А. А. Рубчинский

ДИСКРЕТНЫЕ МАТЕМАТИЧЕСКИЕ МОДЕЛИ

НАЧАЛЬНЫЕ ПОНЯТИЯ И СТАНДАРТНЫЕ ЗАДАЧИ

УЧЕБНОЕ ПОСОБИЕ

DirectMEDIA

Москва

2014

УДК 519.85(075)
ББК 22.176я73
Р82

Рубчинский, А. А.

Р82 Дискретные математические модели. Начальные понятия и стандартные задачи : учебное пособие / А. А. Рубчинский. — М.: Директ-Медиа, 2014. — 269 стр.

ISBN 978-5-4458-3802-9

Предлагаемое пособие посвящено дискретным математическим моделям – в первую очередь, решению разнообразных стандартных задач, в которых надо что-то посчитать, найти, построить и т.д., но не доказать. Особое внимание уделяется технологии ручной реализации алгоритмов.

Аудитория предполагаемого пособия вполне определена. Она состоит из студентов бакалавриата и магистратуры, обучающихся по специальностям, попадающим между точными, естественными и инженерными науками, с одной стороны, и гуманитарными науками, с другой.

УДК 519.85(075)
ББК 22.176я73

ISBN 978-5-4458-3802-9

ПРЕДИСЛОВИЕ

«Труден первый шаг и скучен первый путь.
Преодолел я ранние невзгоды. Ремесло
Поставил я подножием искусству;
Я сделался ремесленник: перстам
Придал послушную, сухую беглость
И верность уху»,

говорит Сальери у Пушкина. Следует подчеркнуть, что без такой, на первый взгляд, рутинной, работы невозможно не только какое-либо творчество, но даже и сколько-нибудь заметное продвижение в любой разумной профессиональной деятельности.

Конечно, математика не является исключением из этого общего правила. Обучаться любому искусству можно на примерах и подражании. И в той степени, в какой математика является искусством (а она им является, на мой взгляд, в значительной степени), именно так – на примерах и подражании – должно быть основано математическое образование, особенно на начальном и среднем уровне. Применительно к более конкретным вещам, мне представляется, что необходимо уделять значительно бóльшее внимание обучению техническим аспектам математики. Ещё конкретнее это значит, что следует уделять внимание методам решения разнообразных <u>стандартных</u> задач, чтобы студент видел, что математика позволяет упрощать ситуации, а не усложнять их (если речь не идёт о наиболее «крутых» математических олимпиадах).

Несмотря на множество книг по дискретной математике и дискретным моделям (часть которых активно использовалась при подготовке предлагаемого материала), книги, в которой подробные определения основных понятий сопровождаются столь большим числом <u>стандартных</u> заданий и поясняющих всех их примеров, просто нет. Мой опыт в преподавании большого числа курсов по этой тематике – от основ дискретной математики до математических моделей и методов принятия решений – говорит о том, что преодолеть естественный страх и скованность большинства студентов можно только одним способом. Надо давать такие задания, которые «получаются», о которых не говорят «вообще не понятно, что и как здесь делать». Только на базе освоения таких простых и подробно описанных заданий можно – для части студентов – переходить к более серьёзным вещам.

Мне представляется, что приступая к написанию любого разумного материала, крайне желательно если не ответить, то хотя бы задать себе три взаимосвязанных вопроса. Они таковы:

1. Для кого предназначен этот материал?
2. Чему могут (или должны) научиться предполагаемые читатели?
3. Что является содержанием предлагаемого материала?

Постараюсь дать мои ответы на эти вопросы.

1. Конечно, хотелось бы сказать, что пособие может быть полезным для студентов, магистров и аспирантов, обучающихся по математическим, программистским, экономическим, управленческим и многим гуманитарным специальностям. Но это не так. Выдающийся математик Питер Халмош в своей известной статье «Как писать математические книги» призывал авторов чётко определять предполагаемую аудиторию. Очень выразителен один из его советов по этому поводу – «книги, нужные всем, не нужны никому».

Аудитория предполагаемого пособия вполне определена. Она состоит из студентов бакалавриата и магистратуры, обучающихся по специальностям, попадающим между точными, естественными и инженерными науками, с одной стороны, и гуманитарными науками, с другой. В этих специальностях – экономике, менеджменте, социологии, политологии, государственном и муниципальном управлении, бизнес-информатике и др., – всё в бóльшей степени используется не традиционная математика с точными количественными зависимостями, а формально более простая математика, оперирующая более наглядными и качественными дискретными моделями. В различных ВУЗах для студентов этих или близких к ним специальностей читаются курсы с названиями «Моделирование и управление», «Методы оптимальных решений», «Математические методы моделирования социальных процессов», «Анализ политической и политологической информации», «Методы анализа и обработки данных для принятия управленческих решений», «Математика конфликтов», «Интеллектуальные системы поддержки управленческих решений» и пр. Здесь перечислены только некоторые курсы, которые я читал, начиная с 2006 г. в Академии внешней торговли, университете «Дубна» и Высшей школе экономики. Учебных пособий, учитывающих реальные нужды этой аудитории и хоть как-то покрывающих указанные курсы, практически нет. Частичным исключением является книга Ф.Т. Алескерова, Э.Л. Хабиной и Д.А. Шварца «Бинарные отношения, графы и коллективные решения», которая вышла уже двумя изданиями. Однако её структура, излагаемый материал и цели отличны от структуры, материала и целей предлагаемого пособия. Именно это обстоятельство – отсутствие необходимых для работы по реальным курсам материалов – и инициировало подготовку настоящего пособия.

2. Как уже говорилось, пособие не рассчитано на математиков и программистов. Тем не менее, представляется, что часть выпускников по упомянутым выше «промежуточным» – между математиками и гуманитариями – специальностям будет встречаться в своей профессиональной деятельности – и не только в научной – с самыми различными дискретными моделями. Поэтому студенты должны получить не только формальное представление о подобных моделях. Они должны привыкнуть к ним («понять – это значит привыкнуть»), не бояться их и уметь работать с ними. Конкретнее, это означает, что они могут (или должны) освоить базовые алгоритмы, связанные с дискретными моделями. Именно поэтому центральной и наиболее важной частью пособия являются многочисленные таблицы и рисунки для «ручной» реализации алгоритмов в небольших размерностях. Только проделав необходимые операции собственными руками, пропустив их перед этим через собственную голову, можно действительно освоить описанный в пособии простой, но разнообразный и непривычный для большинства материал. И уже зная, что, как и зачем делают эти алгоритмы, можно – в дальнейшем – значительно более осознанно и эффективно использовать разнообразное программное обеспечение для решения тех же самых или даже других дискретных задач в реальных размерностях. Таким образом, <u>освоение изложенных в пособии дискретных моделей и алгоритмов работы с ними и представляет собой ответ на второй из поставленных вопросов</u>. Ещё раз стоит повторить, что обучение на примерах и подражании – основной путь начального освоения любой разумной области деятельности.

3. Ответ на третий вопрос, по сути дела, определяется ответами на первые два – кого и чему предполагается научить. Дадим просто несколько более подробные объяснения по поводу того, что в пособии есть, и того, чего в нём нет.

Предлагаемое пособие посвящено дискретным математическим моделям – в первую очередь, решению разнообразных стандартных задач, в которых надо что-то посчитать, найти, построить и т.д., но <u>не доказать</u>. Я совсем не против доказательств как таковых. Но дело в том, что никакого представления о дискретных объектах: кортежах, графах, таблицах, булевых функциях, кодировании, бинарных отношениях, функциях выбора, индексах влияния, стратегиях в играх и др. – даже у хорошего и трудолюбивого школьника, поступившего в ВУЗ, совсем нет (в отличие от интуитивного представления об «обычных» функциях вроде синуса и логарифма). И тем более нет никакого представления о том, как «работают» упомянутые (да и многие другие) дискретные понятия. Поэтому (по моему мнению) надеяться на то, что большинство студентов достаточно быстро научится решать задачи на «доказательство» (даже простые), не приходится. Сначала нужно привыкнуть к рассматриваемым понятиям.

Более того, сам характер дискретных моделей и связанных с ними рассуждений ориентирован больше на алгоритмы, чем на что-либо другое. Основной вопрос при изучении дискретных моделей (и не только на рассматриваемом низком уровне) состоит не в том «как это доказать», а в том, «как это сделать». Да и сами доказательства в дискретной математике в большинстве случаев представляют собой, по сути дела, алгоритмы. Поэтому материал пособия, кроме введения и объяснения многих новых и непривычных понятий, содержит большое число алгоритмов – от совсем очевидных, но всё же специально выделенных именно как алгоритмы, правил выполнения теоретико-множественных операций, до достаточно сложных алгоритмов Форда-Фалкерсона, Дейкстры, нахождения равновесий в биматричных играх, построения устойчивых паросочетаний, различных правил пропорционального представительства, агрегации индивидуальных предпочтений и сравнения векторных оценок методом Подиновского. Не только упомянутые, но и все другие рассматриваемые алгоритмы не просто излагаются и поясняются, но подробно расписываются «в числах» в обозримых и доведённых до конца примерах.

Особое внимание уделяется технологии ручной реализации алгоритмов. Наиболее важные из них (алгоритмы Форда-Фалкерсона, Дейкстры, Флойда-Уоршолла, нахождение критического пути в сетевом графике, задача о замены машины) представлены в виде последовательного заполнения однотипных таблиц, из которых в первую заносятся исходные данные (инициализация), а в последнюю вписывается решение исходной задачи. Приводятся полные примеры последовательного заполнения таких таблиц с подробными объяснениями. Всё это позволяет студентам просто скопировать из соответствующего docx-файла указанные примеры, подставлять вместо содержащихся там исходных данных свои индивидуальные варианты заданий и, следуя имеющимся там же объяснениям, провести требуемые простые однотипные операции со свои-ми данными. Можно сказать, что предлагается своего рода интегрированная система – правда, пока ещё без диагностики ошибок (хотя в принципе понятно, как это сделать). Я убеждён, что чтение готовых примеров не заменяет – с точки зрения освоения – такое выполнение алгоритмов до самого конца по «своим» данным. Но и в тех случаях, где таблиц в явном виде нет (как в изложении алгоритма построения объясняющих цепочек в методе Подиновского), имеется подробное изложение всех шагов на нескольких конкретных примерах, что также позволяет выполнить аналогичную работу самостоятельно. Естественно, что упомянутые файлы прилагаются к пособию.

Материал пособия сгруппирован в четыре части, соответствующие основным типам дискретных моделей и связанных с ними задач. Проблематика каждой части аннонсирована в коротких введениях к ним и здесь не обсуждается. Можно заметить отсутствие части, посвящённой стохастическим моделям. Это связано с особенностями таких моделей, которые, по моему мнению, обосновывают их выделение в отдельную область, естественно включающую в себя материалы как дискретного, так и непрерывного характера. Далее, в соответствие с целями издания, в него включались только такие материалы, которые можно сопроводить простыми и технически легко выполнимыми стандартными заданиями. По этой причине, например, не был включён раздел «Метод ветвей и границ», а также материалы по линейному и целочисленному программированию, многие разделы теории игр и т.д. Разумеется, есть много дискретных моделей, по которым всё же есть задания желаемого типа, но которые также не включены в пособие. Здесь можно ответить словами Козьмы Пруткова «Никто не объемлет необъятного», а также сказать о личных пристрастиях.

Как уже говорилось, пособие инициировано отсутствием изданий того типа и уровня, которые, на мой взгляд, требуются для реального преподавания различных курсов, в значительной степени основанных на дискретных математических моделях или тесно связанных с ними. Речь не идёт об отсутствии книг и учебных пособий по этой тематике в целом и особенно по её отдельным направлениям. Их много, и некоторые из них действительно написаны на высоком научно-методическом уровне. Однако мне представляется, что данное пособие всё же заполняет некоторую нишу, характеризующуюся как указанным кругом специальностей, так и изложением, ориентированным в первую очередь на технические аспекты дискретных моделей. Издание не предполагается использовать как базовый учебник по какому-либо конкретному курсу, но как одно из основных пособий по нескольким курсам для различных специальностей. Назову – в качестве примеров – читаемые в НИУ ВШЭ курсы «Дискретные математические модели», «Теория систем и системный анализ», «Теория игр и конфликтные ситуации», «Методы принятия решений», «Моделирование и управление», «Методы оптимальных решений», «Математика конфликтов». Использование различных частей в различных курсах потребовало максимально возможной независимости материала из разных частей и глав пособия. Например, в первых четырёх главах в рамках изложения «языка математики» даётся достаточно строгое определение понятия функции, наряду с понятиями формы, переменной и т.д.. Однако во всех последующих главах достаточно интуитивного представления о функциях.

Стоит остановиться на некоторых технических вопросах. Пособие состоит из глав с независимой нумерацией формул, примеров, рисунков и т.д. внутри каждой главы. При ссылках на материал из любой другой главы к номеру добавляется отделённый дефисом (-) номер главы. Главы сгруппированы в 4 части, но номера частей при ссылках не указываются. Списки заданий и предметные указатели завершают каждую главу, а небольшие списки литературы – каждую часть. Концы примеров, утверждений, заданий и пр. отмечены знаком ■

Почти все разделы пособия прошли «обкатку» в течение нескольких лет при преподавании упомянутых выше курсов. При выполнении заданий многие вопросы, казавшиеся почти очевидными, вызвали трудности у большинства студентов. Были и обратные примеры, когда некоторые задания, казавшиеся мне более сложными, проблем не вызывали. Полученный при этом очень важный опыт учтён – насколько это возможно – в предлагаемом материале.

Следует сказать, что настоящее пособие предполагает некоторое продолжение. Предполагается подготовка более «продвинутого» пособия по дискретным моделям, в котором будет и более сложный материал, и задачи разной степени трудности «на доказательства», и более структурированное изложение, определяемое одной основной задачей системного анализа – задачей достижения цели. Предполагается также и ещё одно, важное для меня (надеюсь, интересное и для других) пособие, посвящённое реальным прикладным задачам и игровым лабораторным работам, связанным с некоторыми из приложений. В нём предполагается отразить опыт моей (и, конечно, не только моей) работы с реальными прикладными проектами. Наконец, предполагается разработка электронной интерактивной версии настоящего пособия. Но всё это, как писал в своих дневниках Л.Н. Толстой, задумывая новые произведения – ебж (если буду жив).

И последнее. При огромном уважении и любви ко многим замечательным учёным – живым и ушедшим от нас, – с которыми мне «и довелось, и посчастливилось» встречаться, я посвящаю этот скромный труд моим студентам – бывшим, настоящим и будущим. В конце концов, я работаю с ними и для них.

Часть 1. ЭЛЕМЕНТЫ МАТЕМАТИЧЕСКОГО ЯЗЫКА

Начнём первую часть предлагаемого пособия с нескольких цитат из предисловия В.А. Успенского к книге Ю.А. Шихановича «Введение в современную математику» (см. список литературы в конце части).

«… Сейчас, как никогда, становится ясным, что математика – это не только совокупность фактов, изложенных в виде теорем, но прежде всего – арсенал методов, и даже ещё прежде того – я з ы к для описания фактов и методов самых разных областей науки и практической деятельности. Именно этим обстоятельством и обусловливается универсальный характер применимости математики …».

«Математические методы исследования неизбежно начинаются с – явного или неявного – уточнения языка. Причём главное в этом языке … – фундамент, язык н а ч а л ь н ы х понятий. К их числу и относятся, прежде всего … понятия “множество”, “кортеж”, “соответствие”, “функция”, “отношение”». (В обеих цитатах разрядка В.А. Успенского).

К этим понятиям следует добавить – и даже поставить на первое место – понятие “высказывание”, как это и делается в вышеупомянутой книге Ю.А. Шихановича, В ней, как и во многих других изданиях, включая классическую книгу Н. Бурбаки «Теория множеств» (см. список литературы в конце части), понятия высказывания, множества и кортежа считаются исходными и формально неопределяемыми. В то же время понятия графика, соответствия, функции, как и многие другие, будут формально определяться через эти исходные понятия. Именно иерархия основных понятий, связанная с определениями одних через другие, и задаёт порядок изложения. Например, фраза «элемент x принадлежит множеству X» является высказыванием, и уже поэтому множества должны рассматриваться после высказываний, кортежи – после множеств, и т.д. Поэтому материал части 1 «Элементы математического языка» излагается в следующем порядке:

Понятие высказывания вводится и разъясняется (но формально не определяется просто потому, что это невозможно) в главе 1, понятие множества – в главе 2, понятие кортежа – в главе 3. Конечно, вместе с этими понятиями в главах 1 – 3 вводятся и другие «сопутствующие» понятия – простого и составного высказывания, элемента, принадлежности и подмножества, компоненты кортежа и прямого (декартового) произведения множеств, и т.п. Формально определяемые (с помощью уже введённых понятий) понятия графика, соответствия и функции также рассматриваются в главе 3. В главе 4 вводятся понятия формы, переменной и кванторов. Завершает эту часть глава 5, посвящённая булевым (логическим) функциям и некоторымих приложениям. Булевы функции используются в самых различных разделах чистой и прикладной математики, иногда даже за рамками дискретных моделей, и поэтому их также можно отнести к базовым элементам математического языка.

Многие из формальных понятий, вводимых в этой части – кортежа, графика, соответствия, композиции соответствий, формы, – являются обобщениями хорошо известных из обычной школьной математики понятий – вектора, графика, функции, суперпозиции функций, выражения. Эта связь подчёркивается при определении и обсуждении указанных понятий. Более того, во всех главах, кроме первых четырёх, достаточно интуитивного представления о функции и переменных, вместо слова «кортеж» можно использовать более привычные слова «набор», «пара», и т.д. Это возможно, поскольку, опираясь на формализацию указанных понятий из первых четырёх глав, удаётся избежать потерь в точности и общности изложения.

Как и в других частях пособия, изложение сопровождается значительным числом примеров и стандартных заданий, опирающихся на эти примеры.

Глава 1. Высказывания

1. Понятие высказывания

Относительно некоторых предложений естественного языка, при помощи которого мы общаемся, например, «$2 \times 2 = 5$» или «Волга впадает в Каспийское море», имеет смысл задать вопрос: «Истинны они или ложны?» Ясно, что ответ на указанный вопрос относительно первого предложения будет отрицательным: «Нет, это неверно», а относительно второго – положительным: «Да, это так». Легко видеть, что этим свойством обладают не все предложения естественного языка, а только некоторые из них. Так, вряд ли разумно считать, что предложения: «Передай мне, пожалуйста, эту книгу» или «Который час?» обладают свойством быть истинными или ложными. Те из предложений языка, для которых это так, будем называть ***высказываниями***, или ***суждениями***. Заметим, что все математические утверждения, как бы они не назывались (теоремы, леммы, предложения и пр.) являются высказываниями. Заметим, однако, что математические определения не являются высказываниями в указанном смысле. Действительно, неясно, какое значение истинности естественно приписать фразе «Четырёхугольник, противоположные стороны которого попарно параллельны, называется параллелограммом». А вот фразу «В параллелограмме диагонали в точке пересечения делятся пополам» можно рассматривать как высказывание.

Следует сразу сказать, что понятие высказывания, будучи одним из самых важных в математике, не является строгим математическим понятием. Как и важнейшие понятия множества и кортежа, оно не может быть выражено через другие формальные (т.е. ранее уже определённые в математике) объекты и поэтому рассматривается как исходное или базовое. Высказывание можно описать (не определить!) слегка детальнее, как повествовательное предложение, которому естественно сопоставить значение «истина» или «ложь». Только для того, чтобы убедиться в неочевидности этого, на первый взгляд, очевидного, понятия, попробуйте приписать разумное значение истинности предложению «Это высказывание ложно». Разумеется, истинность того или иного конкретного высказывания зачастую трудно, а иногда просто невозможно определить. Но в такой общей постановке это и не входит в задачи математики.

Для удобства дальнейшего изложения высказывания будем обозначать прописными буквами латинского алфавита. Более того, вместо словосочетания «высказывание, обозначенное символом A» будем говорить и писать просто «высказывание A».

Пример 1. Следующие фразы являются высказываниями:

D– «$2 \times 2 = 4$»

E– «Москва – столица России»

F– «Существуют синие яблоки»

G– «Все люди моложе 15-и лет» ■

Поскольку каждое высказывание (согласно данному описанию) может быть истинным или ложным (но не одновременно!), высказыванию естественно сопоставить его ***истинностное***, или ***логическое значение***, т.е. одно из слов – ***истина*** или ***ложь***. Для краткости эти слова обычно сводятся к символам **T** и **F**, и даже к цифрам **1** и **0** (понимаемым именно как символы, а не как числа). Математическая логика (раздел математики, занимающийся операциями с высказываниями) не занимается определением истинностных значений тех или иных конкретных высказываний. В её рамках рассматриваются другие вопросы: как «правильно» формально построить одни высказывания, исходя из других заданных высказываний (синтаксис) и как найти истинностные значения построенных высказываний, исходя из заданных истинностных значений исходных высказываний (семантика).

2. Простые и составные высказывания

Если у нас имеется несколько исходных высказываний, то из них при помощи ***логических союзов*** или ***частиц*** мы можем образовывать новые высказывания, истинностное значение которых зависит только от истинностных значений исходных высказываний и от конкретных союзов и частиц, которые участвуют в построении нового высказывания. Слова и выражения «и», «или», «не», «если ... , то», «поэтому», «тогда и только тогда» являются примерами таких союзов. Исходные высказывания называются ***простыми***, а построенные из них с помощью тех или иных логических союзов новые высказывания –***составными***. Разумеется, слово «простые» никак не связано с сутью или структурой исходных высказываний, которые сами могут быть весьма сложными. В данном контексте слово «простой» является синонимом слова «исходный». Важно то, что значения истинности простых высказываний предполагаются известными или заданными; в любом случае они никак не обсуждаются.

Хотя высказывание типа «Сегодня не четверг» не составлено из двух различных простых высказываний, для единообразия конструкции оно также рассматривается как составное, поскольку его истинностное значение определяется истинностным значением другого высказывания «Сегодня четверг»

Пример 2. Следующие высказывания рассматриваются как составные:

Я читаю «Московский комсомолец» и я читаю «Коммерсант».

Если он сказал это, значит, это верно.

Солнце не является звездой.

Если будет солнечно и температура превысит 25^0, я приеду поездом или автомобилем ■

Простые высказывания, входящие в составные, сами по себе могут быть совершенно произвольными. В частности, они сами могут быть составными. Описываемые ниже базисные типы составных высказываний определяются независимо от образующих их простых высказываний.

2.1. Базисные типы составных высказываний. Рассмотрим основные способы построения составных высказываний из простых, или ***основные операции над высказываниями***. Они перечислены в таблице 1. Разумеется, указанные в левом столбце таблицы слова и выражения не являются исчерпывающим списком логических союзов, а представляют собой только их достаточно распространённые примеры.

Таблица 1. Основные операции над высказываниями

Логические союзы	Знак операции	Имя операции
не; нет	$\neg$	Отрицание
и	$\wedge$	Конъюнкция
или	$\vee$	Дизъюнкция
если ..., то; поэтому; значит; следовательно	$\rightarrow$	Импликация
тогда и только тогда, когда	$\rightleftarrows$	Эквивалентность
или ... или	$\oplus$	Разделительная дизъюнкция

Упомянутая выше связь между истинностными значениями исходных простых высказываний и истинностным значением построенных из них составных высказываний задаётся в так называемых ***таблицах истинности***, в которых истинностное значение составного высказывания даётся при всех комбинациях истинностных значений исходных простых высказываний. Эти таблицы рассматриваются отдельно для шести базисных операций из таблицы 1. В них, как и всюду в дальнейшем, истинностные значения высказываний (истина **T** и ложь **F**) обозначены прямыми буквами, чтобы отличать их от самих высказываний, обозначаемых курсивом.

Нижеприводимые таблицы 2 – 7 дают формальное определение соответствующих операций над истинностными значениями высказываний. В отличие от законов физики, они не являются «Законами Природы», а просто постулируются. Тем не менее, они отражают многие важные черты реальной жизни, повседневного общения и человеческих рассуждений. Именно поэтому понятия и методы исчисления высказываний оказались не только полезными, но и необходимыми во многих современных приложениях – достаточно назвать хотя бы основанные на них компьютеры. Для истинностных значений высказываний, в отличие от самих высказываний, используются строчные латинские буквы, обычно p, q, r, … .

2.1.1. Отрицание. Таблица 2 описывает первый тип составного высказывания – ***отрицание***. Если A – высказывание, то высказывание, состоящее в отрицании A (например, образованное из A путем приписывания перед ним частицы «не» или значка «¬», заменяющего эту частицу), называется отрицанием A и обозначается через $\neg A$. Оно истинно, когда A ложно, и ложно в противном случае (когда A истинно). Из высказываний примера 1 истинными будут высказывания $\neg F$ и $\neg G$, а ложными – высказывания $\neg D$ и $\neg E$.

Таблица 2. Таблица истинности для отрицания

p	$\neg p$
T	**F**
F	**T**

2.1.2. Конъюнкция. Таблица 3 описывает второй тип составного высказывания – ***конъюнкцию***. Конъюнкцией высказываний A, B называется составное высказывание, построенное из высказываний A и B при помощи союза «и», обозначаемого специальным знаком «∧». Конъюнкция считается истинной, когда одновременно истинны оба высказывания A, B. Во всех остальных случаях она считается ложной. Для высказываний, введённых в примере 1, высказывание $D\wedge E$ – истинное, а высказывания $D\wedge F$, $E\wedge F$и $F\wedge G$ – ложны.

Таблица 3. Таблица истинности для конъюнкции

p	q	$p \wedge q$
T	**T**	**T**
T	**F**	**F**
F	**T**	**F**
F	**F**	**F**

2.1.3. Дизъюнкция. Таблица 4 описывает третий тип составного высказывания – ***дизъюнкцию***. Дизъюнкцией высказываний A, B называется составное высказывание, построенное из высказываний A и B при помощи союза «или», обозначаемого специальным знаком «∨». Дизъюнкция считается истинной, когда истинно хотя бы одно из входящих в неё высказываний. Для высказываний, введённых в примере 1, высказывания $D\vee E$, $D\vee F$, $E\vee F$– истинны, а высказывание $F\vee G$ – ложно.

Таблица 4. Таблица истинности для дизъюнкции

p	q	$p \vee q$
T	**T**	**T**
T	**F**	**T**
F	**T**	**T**
F	**F**	**F**

2.1.4. Импликация. Таблица 5 описывает четвёртый тип составного высказывания – ***импликацию***. Импликацией высказываний A, B (в указанном порядке) называется составное высказывание, построенное из высказываний A и B при помощи союза «если ..., то ...» (или других аналогичных выражений), обозначаемого специальным знаком «→». Если $A\to B$ – импликация, то высказывание A называется ***посылкой***, или ***антецедентом*** импликации, а B – её ***заключением***, или ***консеквентом***. Импликация считается ложной в том единственном случае, когда ее посылка истинна, а заключение – ложно.

На первый взгляд неясно, почему импликация считается истинной в случаях, когда её посылка ложна, а заключение – истинно, или когда оба они ложны. Конечно, можно просто сказать, что это – вопрос соглашения, и что именно так принято в математике. Однако возможна и более содержательная аргументация в защиту такого соглашения. Пусть перед выборами кандидат X обещает: «Если меня выберут, то через год пенсии будут увеличены на 50%». Его обещание (и по форме, и по сути) является импликацией с посылкой «меня выберут» и заключением «через год пенсии будут увеличены на 50%». Естественно считать, что X обманул избирателей только в одном случае – если <u>его выбрали, а пенсии не были увеличены на 50%</u>. В остальных трёх случаях

– если его не выбрали, и пенсии не были увеличены на 50%;

– если его не выбрали, и пенсии были увеличены на 50%;

– если его выбрали, и пенсии были увеличены на 50%

обмана со стороны X нет.

Про импликацию с ложной посылкой, которая – по принятому определению – истинна при любом следствии, говорят, что она истинна «в силу ложности посылки». Такое соглашение действительно оказывается удобным во многих случаях. Рассмотрим такой пример. Утверждение о том, что любой четырёхугольник, в котором диагонали в точке пересечения делятся пополам, есть параллелограмм, является импликацией (как и большинство других математических утверждений). Его можно записать в виде импликации с посылкой «диагонали в точке пересечения делятся пополам» и заключением «четырёхугольник является параллелограммом». Но что если диагонали четырёхугольника совсем не пересекаются (это возможно: приведите пример!). Однако и в этом случае с учётом нашего соглашения об импликации эта импликация останется истинной «в силу ложности посылки»: ведь если диагонали не пересекаются, то посылка заведомо является ложной. Без такого соглашения пришлось бы рассматривать отдельно (или хотя бы явно упоминать) этот случай. Многократное использование этого соглашения в данном пособии демонстрирует его полезность.

Для высказываний, введённых в примере 1, высказывания (импликации) $D \to D$, $D \to E$, $F \to D$, $G \to F$ истинны, а высказывания (импликации) $D \to F$, $E \to G$ – ложны.

Таблица 5. Таблица истинности для импликации

p	*q*	*p→q*
T	**T**	**T**
T	**F**	**F**
F	**T**	**T**
F	**F**	**T**

2.1.5. Эквивалентность. Таблица 6 описывает пятый тип составного высказывания – ***эквивалентность.*** Эквивалентностью высказываний *A* и *B* называется составное высказывание, построенное из высказываний *A* и *B* при помощи логического союза «,... тогда и только тогда, когда...» (или других аналогичных выражений), обозначаемого специальным знаком «⇄». Эквивалентность считается истинной, если оба входящих в неё высказывания имеют одинаковое логическое значение (то есть либо оба истинны, либо оба ложны), и ложна в остальных случаях. Для высказываний, введённых в примере 1, высказывания (эквивалентности) $D \rightleftarrows E$, $G \rightleftarrows F$ истинны, а высказывания (эквивалентности) $F \rightleftarrows D$, $G \rightleftarrows E$ – ложны.

Таблица 6. Таблица истинности для эквивалентности

p	*q*	*p⇄q*
T	**T**	**T**
T	**F**	**F**
F	**T**	**F**
F	**F**	**T**

2.1.6. Разделительная дизъюнкция. Таблица 7 описывает шестой тип составного высказывания – ***разделительную дизъюнкцию.*** Разделительной дизъюнкцией высказываний *A*, *B* называется составное высказывание, построенное из высказываний *A* и *B* при помощи логического союза «или ... или» («или разделительное»), обозначаемого специальным знаком «⊕». Оно считается истинным, когда высказывания *A* и *B* принимают разные истинностные значения.

Для высказываний, введённых в примере 1, высказывания (разделительные дизъюнкции) $F \oplus D$, $G \oplus E$ истинны, а высказывания (разделительные дизъюнкции) $D \oplus E$, $G \oplus F$ –ложны.

Таблица 7. Таблица истинности для разделительной дизъюнкции

p	*q*	*p⊕q*
T	**T**	**F**
T	**F**	**T**
F	**T**	**T**
F	**F**	**F**

Если вместо символа **T** используется знак 1, а вместо символа **F** – знак 0, то таблица 7 превращается в таблицу 8a; при записи её строк в другом порядке получаем таблицу 8b. Таблица 8b описывает арифметическую операцию сложения в двоичной системе счисления, которая обычно называется «сложением по модулю 2». Она является одной из основных операций, реализуемых в любом компьютере.

Таблица 8a

p	q	$p\oplus q$
1	**1**	**0**
1	**0**	**1**
0	**1**	**1**
0	**0**	**0**

Таблица 8b

p	q	$p\oplus q$
0	**0**	**0**
0	**1**	**1**
1	**0**	**1**
1	**1**	**0**

Таковы основные операции над высказываниями, представленные в таблицах истинности как операции над истинностными значениями высказываний.

3. Формальные представления составных высказываний

В общем случае произвольное составное высказывание может быть образовано последовательным соединением некоторых простых высказываний некоторыми из имеющихся шести основных операций. Напомним, что простыми называются высказывания, которые в данном контексте рассматриваются как нерасчленимые, с фиксированными истинностными значениями.

Пример 3. Высказывание из примера 2 «***Если будет солнечно и температура превысит 25^{0}, я приеду поездом или автомобилем***» является импликацией двух высказываний «***Будет солнечно и температура превысит 25^{0}***» и «***Я приеду поездом или автомобилем***» (в этом порядке). В свою очередь, посылка этой импликации является конъюнкцией двух высказываний: «***Будет солнечно***» и «***Температура превысит 25^{0}***», а её заключение является дизъюнкцией двух высказываний: «***Я приеду поездом***» и «***Я приеду автомобилем***».

Обозначим высказывания «***Будет солнечно***», «***Температура превысит 25^{0}***», «***Я приеду поездом***» и «***Я приеду автомобилем***» через P, Q, R и S. Тогда рассматриваемое высказывание можно представить формулой

$(P\wedge Q)\to(R\vee S)$. (1) ■

Выражения типа (1) будут рассматриваться как формальные представления составных высказываний. В эти выражения входят прописные латинские буквы, соответствующие простым (т.е. исходным) высказываниям, знаки введённых базисных операций над высказываниями, и скобки, используемые для выделения различных комбинаций букв и знаков (как и в других математических формулах). Конечно, никакого точного определения таких выражений здесь не даётся. Ещё важнее, что построение формальных представлений составных высказываний является содержательной, неформальной операцией. Никакого точного алгоритма здесь нет и не может быть, хотя бы потому, что нет и не может быть формального определения моделируемых объектов – самих составных высказываний. Однако есть многочисленные примеры, поясняющие – в простых и не очень простых случаях – как выглядят такие формальные представления. Обычно этого оказывается достаточным.

Вспомните, как вы сами учились шить, рисовать, играть на пианино, водить машину (список можно легко продолжить). Ни в одном из этих случаев каких бы то ни было точных, однозначно понимаемых инструкций не было. Тем не менее, многим из нас удалось хоть чему-то научиться.

Приведём более сложный пример составного высказывания и его формального представления.

Пример 4. Рассмотрим составное высказывание «***Если я – Ваша женщина и Вы – мой мужчина, то я никогда не перестану любить Вас. Я перестала любить Вас.***

Прежде всего, выделим простые высказывания, из которых и составлено рассматриваемое составное высказывание. Естественно считать, что эти простые высказывания в данном случае таковы:

P= «***Я – Ваша женщина***»,

Q= «***Вы – мой мужчина***»,

R= «***Я никогда не перестану любить Вас***».

В этих обозначениях для высказывания «***Если я – Ваша женщина и Вы – мой мужчина, то я никогда не перестану любить Вас***» формальным представлением является импликация $(P\wedge Q)\to R$. Далее, для высказывания «***Я перестала любить Вас***», которое по сути является отрицанием простого высказывания «***Я никогда не перестану любить Вас***», формальным представлением является отрицание $\neg R$. Высказывание «***Если я – Ваша женщина и Вы – мой мужчина, то я никогда не перестану любить Вас. Я перестала любить Вас***» состоит из двух фраз (до и после запятой), в которых излагаются две последовательные части одной и той же «истории». Ни одна из не является ни следствием, ни причиной другой. В подобных случаях естественно считать, что данное высказывание является конъюнкцией двух высказываний, а его формальное представления также является конъюнкцией двух выражений, уже сопоставленных этим частям. В рассматриваемом случае имеем

$((P\wedge Q)\to R)\wedge\neg R$■

Пример 5. Рассмотрим составное высказывание «***Если я – Ваша женщина и Вы – мой мужчина, то я никогда не перестану любить Вас. Я перестала любить Вас. Значит, я – не Ваша женщина или Вы – не мой мужчина***»

Вся история является импликацией. Её посылкой является текст до слова «значит», а заключением – текст после слова «значит». Посылка уже проанализирована, и её формальное представление найдено в примере 4: $((P\wedge Q)\to R)\wedge\neg R$. Формальным представлением заключения «***Я – не Ваша женщина или Вы – не мой мужчина***» является выражение $\neg P\vee\neg Q$. Таким образом, формальным представлением всего рассматриваемого высказывания является импликация

$(((P\wedge Q)\to R)\wedge\neg R)\to(\neg P\vee\neg Q)$. (2) ■

Примеры 3, 4 и 5 ясно демонстрируют, что выделение исходных простых высказываний и построение сопровождающих формул требует содержательных рассуждений и не сводится к чисто формальным операциям.

Обратим внимание на то, что один и тот же символ может встречаться в одном и том же выражении несколько раз, как в формуле (2) из примера 5.

Непосредственно перед примером 1 говорилось об использовании выражения «высказывание A» вместо более длинного и неудобного выражения «высказывание, обозначенное символом A». При описании базовых операций в разделе 2.1 в том же смысле использовались выражения $D\wedge E$, $G\rightleftarrows E$ и т.д. Аналогично, вместо общего выражения «формальное представление Φ составного высказывания W», будет писать «выражение Φ», «высказывание, представленное выражением Φ», и т.д. Однако надо иметь в виду, что речь идёт об объектах разной природы – неформальных высказываниях и их формальных представлениях. Упомянутая «вольность языка» допустима в тех – впрочем, широко распространённых – случаях, когда такая замена не приводит к неоднозначности понимания и ошибкам. Забегая вперёд, скажем, что примерно то же самое относится к выражениям «множество A», «кортеж α» и т.д. Соответствующие выражения будут использоваться без дополнительных разъяснений.

Пример 6. Пусть P означает высказывание «***Крис собирает видеофильмы***» и Q означает высказывание «***Джек играет на трубе***». Тогда по этим двум простым высказываниям можно определить следующее составное высказывание: «Крис собирает видеофильмы и Джек не играет на трубе». Используя введённые обозначения P и Q, представим его в виде $P\wedge\neg Q$. Здесь и далее используется соглашение, по которому знак отрицания имеет самый высокий приоритет. Это соглашение позволяет в данной формуле не использовать скобок. Точно также предполагается, что конъюнкция «важней» дизъюнкции (в том же смысле, что и в выражении $ab+c$, где сначала делается умножение, а потом – сложение). В остальных случаях очерёдность операций определяется расстановкой скобок ■

Пример 7. Рассмотрим высказывание «***Женат я или не женат, я счастлив***». Обозначим высказывание «я женат» символом P, высказывание «я счастлив» символом Q. Тогда данное составное высказывание имеет формальный вид

$(P\vee\neg P)\to Q$■ (3)

Конечно, такое соответствие между содержательным высказыванием и его представлением вида (1) – (3) далеко не всегда столь очевидно, а сама процедура построения такого соответствия вряд ли может быть полностью формализована. Хотя в рассматриваемых далее примерах всё это делается сравнительно просто, но тренировка здесь, как и во всякой не до конца формализованной деятельности, совершенно необходима. Образцами для такой формализации служат как примеры 3 – 6, так и примеры, приводимые ниже.

Пример 8. Представим высказывание «***Если пол грязный, то я должен вымыть его***» в формальном виде. Для этого прежде всего определим входящие в него простые высказывания. Положим P = «***Пол грязный***», Q= «***Я должен вымыть его***». Тогда рассматриваемое высказывание представляется в виде$P\rightarrow Q$, т.е. оно является импликацией ■

Пример 9. Представим высказывание «***Если пол грязный, то я должен вымыть его. Пол грязный***» в формальном виде. В данном случае у нас есть два последовательных высказывания, разделённых точкой: «***Если пол грязный, то я должен вымыть его***» и «***Пол грязный***». Поскольку оба высказывания относятся к одной «истории», то естественно рассматривать всё высказывание как конъюнкцию этих двух составляющих его простых высказываний. Конечно, это не теорема, и этот факт не может быть формально доказан. Это – вопрос соглашения, но такое соглашение, как минимум, не противоречит здравому смыслу. В большинстве тех случаев, когда высказывания разделены не точкой, а точкой с запятой, просто запятой, или союзом «и», составное высказывание также естественно представлять как конъюнкцию. Точно так же это делалось и в примере 4. Но надо быть внимательным и никогда не забывать о здравом смысле.

Первое высказывание (см. пример 8) представлено как импликация $P\rightarrow Q$, где P = «***Пол грязный***», Q = «***Я должен вымыть его***». Поэтому 1-ая часть рассматриваемого составного высказывания уже представлена, как $P\rightarrow Q$. С учётом нашего соглашения о разделяющей точке получаем следующее формальное представление:$(P\rightarrow Q)\wedge P$, где P и Q – те же, что в примере 8 ■

Пример 10. Представим высказывание «***Если пол грязный, то я должен вымыть его. Пол грязный. Поэтому я должен вымыть его***» в формальном виде.

В данном случае 1-ая часть высказывания: «***Если пол грязный, то я должен вымыть его. Пол грязный***» уже представлена в формальном виде в примере 9: $(P\rightarrow Q)\wedge P$. 2-ая часть исходного составного высказывания после слова «поэтому» такова: «***Я должен вымыть его***», что совпадает с высказыванием Q из примеров 8 и 9. Таким образом, мы имеем составное высказывание $((P\rightarrow Q)\wedge P)$?Q. Осталось «раскрыть» знак вопроса. В разделе 2.1 указывалось, что логические союзы «Если …, то …», «Поэтому», «Потому что», «Поскольку» и т.д. соответствуют импликации, в которой слева стоит посылка («причина»), а справа – заключение («следствие из данной причины»). Таким образом, исходное составное высказывание представляется в виде

$$((P\rightarrow Q)\wedge P)\rightarrow Q. \tag{4}$$

Заметим, что в исходном высказывании перед словом «Поэтому» стоит точка. Однако здесь пишется импликация, а не конъюнкция, поскольку здесь явно сказано, что всё, стоящее перед точкой, является причиной, а всё, стоящее после связки «Поэтому», является заключением. Содержание же всей этой простой истории таково. Мыть грязный пол – это моя обязанность, и если уж он грязный, то я должен его помыть. Ещё раз подчеркнём, что все рассуждения не являются формальными, но зато они являются правдоподобными. Заметим также, что при замене 2-ой точки на точку с запятой или запятую смысл рассказа не изменится и, значит, его формальное представление останется тем же самым ■

Пример 11. Представим высказывание «***Если я видел дальше других, то потому, что я стоял на плечах гигантов***» в формальном виде. (***If I have seen farther than others, it is because I stood on the shoulders of giants*** (quote from Sir Isaac Newton)).

Положим

P = «***Я стоял на плечах гигантов***»,

Q = «***Я видел дальше других***».

Тогда рассматриваемое высказывание представляется импликацией $P\rightarrow Q$.

Заметим, что в исходной фразе заключение («***Я видел дальше других***») написано впереди посылки («***Я стоял на плечах гигантов***»). Действительно, что здесь заключение, а что посылка, непосредственно определяется содержанием фразы, суть которой в том, что если стоишь выше, то и видишь дальше (а не наоборот!). В то же время в формально определённой импликации посылка всегда записывается слева от стрелки, а заключение – справа, т.е. посылка предшествует следствию. Это ещё раз демонстрирует, что при формализации высказывания надо исходить в первую очередь из его содержания. Одна и та же простая мысль может быть записана на русском (и на любом другом) языке многими способами, но её формальное представление от этого не меняется ■

В следующих примерах 12 – 16 рассматривается несколько более сложная ситуация.

Пример 12. Представим высказывание «***Все мужчины созданы равными***» в формальном виде.

Положим

P = «***Быть мужчиной***»,

R = «***Быть созданным равным***».

Естественно переформулировать исходное высказывание «***Все мужчины созданы равными***» следующим образом: «***Если Вы – мужчина, то Вы созданы равным***». При этом смысл высказывания не изменился. Но это модифицированное высказывание можно записать как импликацию $P \to R$ при выше определённых P и R■

Пример 13. Представим высказывание «***Все люди, созданные равными, являются женщинами***» в формальном виде.

Положим

Q = «***Быть женщиной***»,

R = «***Быть созданным равным***».

Естественно переформулировать исходное высказывание «***Все люди, созданные равными, являются женщинами***» следующим образом: «***Если Вы созданы равным, то Вы – женщина***». При этом смысл высказывания не изменился. Но это модифицированное высказывание можно записать как импликацию $R \to Q$ при выше определённых Q и R■

Пример 14. Представим высказывание «***Все мужчины созданы равными. Все люди, созданные равными, являются женщинами***» в формальном виде.

Данное высказывание является конъюнкцией высказываний из примеров 12 и 13. Поэтому оно может быть представлено в виде

$(P \to R) \wedge (R \to Q)$ ■

Пример 15. Представим высказывание «***Все мужчины являются женщинами***» в формальном виде.

Положим, как в примерах 12 и 13

P = «***Быть мужчиной***»,

Q= «***Быть женщиной***».

Естественно переформулировать исходное высказывание «***Все мужчины являются женщинами***» следующим образом: «***Если Вы – мужчина, то Вы – женщина***». При этом смысл высказывания не изменился. Но это модифицированное высказывание можно записать как импликацию $P \to Q$ при выше определённых P и Q■

Пример 16. Представим высказывание «***Все мужчины созданы равными. Все люди, созданные равными, являются женщинами. Поэтому все мужчины являются женщинами***» в формальном виде.

Это высказывание по структуре напоминает высказывания из примеров 5 и 10. Действительно, данное высказывание, как и высказывания из примеров 5 и 10, является импликацией. Её посылка такова: «***Все мужчины созданы равными. Все люди, созданные равными, являются женщинами***»; её заключение таково: «***Все мужчины являются женщинами***».

Посылка представлена в примере 14 формулой $(P \to R) \wedge (R \to Q)$. Заключение представлено в примере 15 формулой $P \to Q$. Таким образом, исходное высказывание можно записать в виде

$$(P \to R) \wedge (R \to Q)) \to (P \to Q). \qquad (5)$$

Ясно, что заключение «***Все мужчины являются женщинами***» является ложным высказыванием. Но здесь следует ещё раз подчеркнуть, что операции над высказываниями и их формальные представления осуществляются <u>независимо от истинности или ложности</u> участвующих в этих операциях простых высказываний ■

Более подробно методы формального анализа высказываний на основе сопоставляемых их формальных выражений, рассмотренных в примерах 3 – 16 вида, демонстрируются в разделе 5-3, сразу после введения в рассмотрение булевых функций и связанных с ними понятий.

4. Задания

Задание 1. Указать, какие из следующих предложений являются высказываниями. Для высказываний установить значение истинности.

01. 7-го декабря 1941 года было воскресенье.

02. Слушайте, мои дети, и вы узнаете о ночной скачке Пола Ривера.

03. 5+8=13 и 4–3=1

04. Некоторые числа отрицательны.

05. Куда Вы идёте сегодня вечером?

06. Эндрю Джексон был президентом Соединённых Штатов в 1867 году.

07. Один галлон молока весит больше 4 фунтов.

08. У меня есть пистолет «Смит и Вессон» ■

Задание 2. Пусть P и Q – высказывания из примера 6. Представить следующие составные высказывания в формальном виде.

01. Крис собирает видеофильмы или Джек играет на трубе.

02. Крис не собирает видеофильмы или Джек не играет на трубе.

03. Крис не собирает видеофильмы и Джек играет на трубе.

04. Ни Крис не собирает видеофильмы, ни Джек не играет на трубе.

05. Либо Крис собирает видеофильмы, либо Джек играет на трубе.

06. Если Джек не играет на трубе, то Крис собирает видеофильмы.

07. Крис собирает видеофильмы, если и только если Джек не играет на трубе.

08. Крис собирает видеофильмы, потому что Джек не играет на трубе.

09. Если Джек не играет на трубе, то Крис собирает видеофильмы, и Джек играет на трубе.

10. Если Джек не играет на трубе, то Крис собирает видеофильмы, или, если Крис собирает видеофильмы, то Джек играет на трубе ■

Задание 3. Представить заданные высказывания в формальном виде. См. примеры 3 – 16 для образца.

01а. Если Эдди едет в город, то Мейбл останется дома.

01б. Если Мейбл не останется дома, то Рита будет готовить.

02. «Мужчина состоит из мужа и чина» (Чехов).

03а. Я покупаю машину или я уезжаю на каникулы.

03б. Я не покупаю машину.

03в. Я уезжаю на каникулы.

04а. Если бы человек мог быть в двух местах одновременно, я был бы с Вами.

04б. Я не был с Вами.

04в. Человек не может быть в двух местах одновременно.

05а. «Если мы будем следовать по стопам Ньютона, то это не будет прогрессом» (Алдос Хаксли).

05б. Мы не следуем по стопам Ньютона.

06а. Джефф любит играть в гольф.

06б. Если Джоан любит шить, то Джефф не любит играть в гольф.

6в. Если Джоан не любит шить, то Брэд поёт в хоре.

07а. Если дерево заражено сосновым короедом, то оно умрёт.

07б. Люди сажают деревья в День Дерева, и оно не умрёт.

07в. Если люди сажают деревья в День Дерева, то оно не заражено сосновым короедом.

08а. Если я напишу чек, он не будет принят.

08б. Если банк гарантирует его, он будет принят.

09а. Кристина Алигера поёт или Рики Мартин – не подростковый идол.

09б. Если Рики Мартин – не подростковый идол, то Бритни Спирс не выиграет Американский Музыкальный Приз.

09в. Бритни Спирс выиграла Американский Музыкальный Приз.

10а. Если я чувствую Вас всей кожей, то Вы глубоко в моём сердце.

10б. Если Вы глубоко в моём сердце, то Вы – не часть меня.

10в. Вы глубоко в моём сердце или Вы часть меня.
11а. Если Отис – диск-жокей, то он живёт в Лексингтоне.
11б. Он живёт в Лексингтоне и он сдвинут на истории.
12. Если Канада – союзник Соединённых Штатов, то Соединённые Штаты относятся к Канаде дружески.
13а. Если Вы спите во время утренних занятий по математике, то Вы хорошо отдохнёте.
13б. Если Вы хорошо отдохнёте, то Вы хорошо сдадите тест по математике.
14а. Мы сбалансируем бюджет или уменьшим налоги,
14б. Если мы сбалансируем бюджет или уменьшим налоги, то будет больше денег для борьбы с загрязнением.
14в. Если мы не сбалансируем бюджет, то мы не уменьшим налоги.
15а. Если у товара низкая цена, то у него нет качества.
15б. Если у товара нет низкой цены или нет качества, то он ненадёжен.
16а. Если тест отрицателен, то Вы не нуждаетесь в лечении.
16б. Если тест положителен, то Вы нуждаетесь в лекарствах.
17а. Если Дэйв одинок сегодня вечером, то он не пойдёт в гости.
17б. Если Дэйв не одинок сегодня вечером или он не пойдёт в гости, то он будет писать свою курсовую работу ■

5. Предметный указатель

Глава 2. Множества

1. Множества и подмножества

Понятие множества (как и уже рассмотренное понятие высказывания) является для нас исходным, неопределяемым. Можно говорить о множестве студентов в данной аудитории, множестве книг в библиотеке и т.д. Множество составлено из элементов, способных обладать некоторыми свойствами и находиться между собой и с элементами других множеств в неких отношениях. Важно, что элементы множества являются различными и (хотя бы в принципе) различимыми. Мы будем обозначать сами множества прописными, а их элементы – строчными буквами латинского алфавита. Высказывание «x есть элемент множества X» символически записывают так: $x \in X$. Эта связь между элементом и множеством называется ***принадлежностью***. Отрицание этого высказывания записывают так: $x \notin X$. Множества X и Y называют ***равными***, если они состоят из одних и тех же элементов. Высказывание о равенстве множеств X и Y записывают в виде: $X = Y$, а отрицание этого высказывания (т.е. высказывание, состоящее в том, что множества X и Y не равны) – в виде $X \neq Y$.

Для удобства вводят так называемое ***пустое множество***, т.е. множество, не содержащее элементов. Его изображают символом $\varnothing$. Таким образом, если x – некоторый элемент (объект), то принадлежность $x \in \varnothing$ – всегда ложное высказывание. Наряду с пустым множеством полезно ввести в рассмотрение и так называемое ***универсальное множество***. Это множество состоит из всех элементов, имеющих отношение к определённой рассматриваемой ситуации. Например, при анализе успеваемости (или проведении социологического опроса) студентов в качестве универсального выступает множество всех студентов данного института или города; в экономическом исследовании в качестве универсального часто выступает множество всех фирм данной страны или группы стран, и т.д. Универсальное множество обычно обозначается буквой U. Заметим, что при переходе от одной ситуации к другой универсальное множество (в отличие от пустого множества) может измениться.

Может случиться так, что все элементы множества X являются одновременно элементами множества Y, т.е.

$$x \in X \rightarrow x \in Y. \qquad (1)$$

В этом случае говорят, что X есть ***подмножество***, или часть Y, что записывается так:

$$X \subseteq Y. \qquad (2)$$

Из (1) и (2) получаем, что $X \subseteq Y$ тогда и только тогда, когда истинна импликация $x \in X \rightarrow x \in Y$. Из определения операции эквивалентности следует, что $(X = Y) \rightleftarrows (X \subseteq Y) \wedge (Y \subseteq X)$. Отрицание высказывания (2) записывается как $X \not\subseteq Y$. $X \subset Y$ означает, что $(X \subseteq Y) \wedge (X \neq Y)$.

Множество может быть задано перечислением своих элементов, получением элементов множества из уже полученных элементов либо из других объектов (порождающей процедурой) или описанием свойств, которыми должны обладать его элементы (разрешающей процедурой). В настоящей главе будут рассматриваться только конечные множества. Конечное множество можно задать (хотя бы в принципе) списком, в котором и перечисляются все его элементы. Список заключается в фигурные скобки, а сами элементы отделяются друг от друга запятыми. Задание множества разрешающей процедурой рассматривается далее, в разделе 4-2.1, порождающей процедурой – в разделе 5.2.

Пример 1. Множество имён летних месяцев: {июнь, июль, август} ■

Пример 2. В описаниях «больших» конечных множеств используются многоточия. Множество неотрицательных целых чисел, не превосходящих 50, записывают как $\{0, 1, \ldots, 50\}$. Заметим, что почёркнутые в предыдущей фразе слова как раз описывают свойство всех элементов данного множества, <u>выделяющее</u> их из всех остальных объектов. Такие свойства называются ***характеристическими***■

Пример 3. Многоточия можно использовать только в тех случаях, когда не возникает двусмысленности. Если $A = \{3, 5, 7, \ldots, 19\}$, то неясно, является ли Aмножеством нечётных чисел, лежащих в интервале от 3 до 19, или это множество простых чисел из того же интервала, т.е. возможны разные «расшифровки» неопределённости, скрывающейся за многоточием ■

Обратим внимание на следующее. Элемент a надо отличать от одноэлементного множества $\{a\}$: это объекты разной природы. Нельзя написать, что$a = \{a\}$ или $a \subseteq \{a\}$. Но можно написать истинное высказывание $a \in \{a\}$. Элементами множеств могут быть другие множества. Множество, элементами которого служат все подмножества множества X, обозначается символом $B(X)$. Его называют ***булеаном*** множества X.

Пример 4. Вставить знак $\subseteq$, $\not\subseteq$, $\in$или$\notin$вместо пробела, чтобы получить истинное высказывание:

(a) $\{3, 4, 5, 6\}$ _______ $\{3, 4, 5, 6, 8\}$.

Поскольку каждый элемент множества $\{3, 4, 5, 6\}$ принадлежит и множеству $\{3, 4, 5, 6, 8\}$, то 1-ое множество является подмножеством 2-го, так что надо вставить $\subseteq$.

(b) 1 _______ $\{2, 4, 6, 8\}$.

Элемент 1 не принадлежит множеству $\{2, 4, 6, 8\}$. Поэтому между ними надо вставить $\notin$■

2. Диаграммы Венна и операции над множествами

В большинстве разделов математики понимание существенно облегчается при помощи разнообразных рисунков и диаграмм. В рассуждениях о множествах часто используются ***диаграммы Венна***, предложенные английским логиком Джоном Венном (1834-1923). В этих диаграммах универсальное множество представлено прямоугольником, а другие рассматриваемые множества – обычно овалами или кружками (размер и форма фигур не имеют значения). Типичный пример диаграммы Венна показан на рис.1. На нём вся область, ограниченная прямоугольником, представляет универсальное множество U, а два овала представляют два множества A и B. Диаграммы Венна дают наглядное представление об операциях над множествами, рассматриваемыми в разделе 2.1.

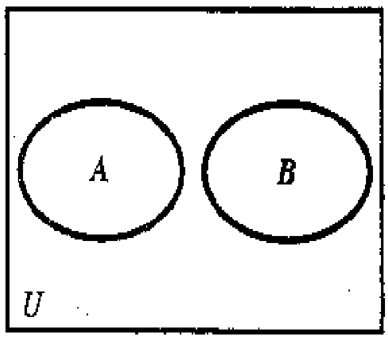


Рис.1. Диаграмма Венна

2.1. Операции над множествами. Если мы располагаем некоторым запасом множеств, то из них мы можем строить новые множества при помощи так называемых ***теоретико-множественных*** операций. Рассмотрим подробно основные из этих операции.

А. Дополнение. Пусть U – универсальное множество, A– некоторое множество (напомним ещё раз, что в рамках любой конкретной ситуации каждое множество содержится в универсальном для данной ситуации множестве U, т.е. $A \subseteq U$). Множество, состоящее из всех элементов U, <u>не содержащихся</u> в A, называют ***дополнением***Aи обозначают через A'. Более формально: $x \in A' \rightleftarrows x \notin A$. Диаграмма Венна на рис.2 иллюстрирует операцию дополнения.

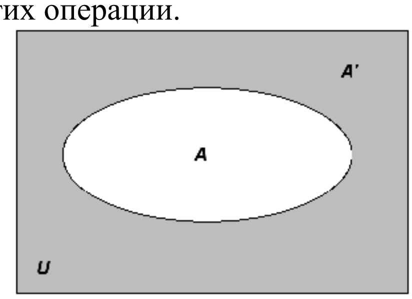


Рис.2. Дополнение

В. Объединение. Множество, элементами которого являются <u>элементы множеств A и B и только они</u>, называют их ***объединением*** и обозначают знакосочетанием $A \cup B$. Очевидно:$(x \in A \cup B) \rightleftarrows (x \in A) \vee (x \in B)$.***Объединением семейства*** множеств $\{A_i\}_{i \in I}$ называют множество, обозначаемое знакосочетанием $\cup_{i \in I} A_i$ и состоящее из тех и только тех элементов, каждый из которых принадлежит хотя бы одному множеству этого семейства. Диаграмма Венна на рис.3 иллюстрирует операцию объединения.

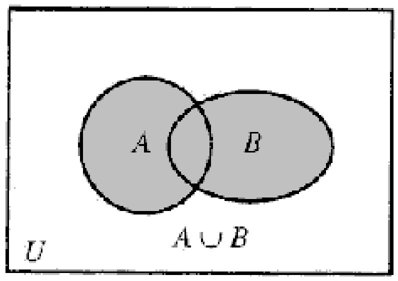


Рис.3. Объединение

С. Пересечение. Множество, элементами которого являются элементы, принадлежащие как множеству A, так и множеству B, называют их ***пересечением*** и обозначают знакосочетанием $A \cap B$. Очевидно: $(x \in A \cap B) \rightleftarrows (x \in A) \wedge (x \in B)$. ***Пересечением семейства*** множеств $\{A_i\}_{i \in I}$ называют множество, обозначаемое знакосочетанием $\bigcap_{i \in I} A_i$ и состоящее из тех и только тех элементов, каждый из которых принадлежит всем множествам этого семейства. Диаграмма Венна на рис.4 иллюстрирует операцию пересечения.

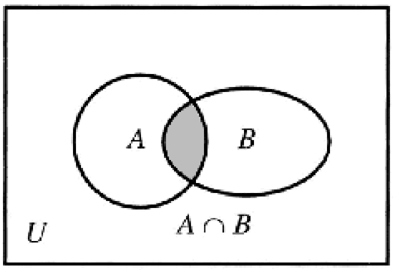


Рис.4. Пересечение

D. Разность. Множество, элементами которого являются элементы множества A, не принадлежащие множеству B, называют их ***разностью*** и обозначают знакосочетанием $A \setminus B$ (саму операцию называют ***вычитанием множеств***). Очевидно: $(x \in A \setminus B) \rightleftarrows (x \in A) \wedge (x \notin B)$. Если, в частности, $B \subseteq A$, то разность $A \setminus B$ называют ***дополнением множества*** B ***до множества*** A и обозначают через $(B')_A$, или просто B', если ясно, о чём идет речь. Диаграмма Венна на рис.4 иллюстрирует операцию вычитания и её результат.

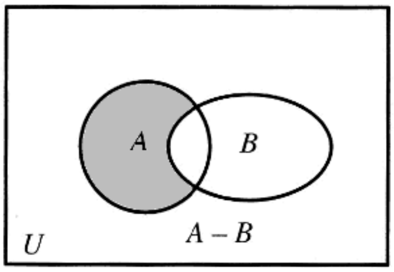


Рис.5. Разность

E. Симметрическая разность. Множество, элементами которого служат элементы множества A, не принадлежащие множеству B, и элементы множества B, не принадлежащие множеству A, называют их ***симметрической разностью*** и обозначают знакосочетанием $A \Delta B$. Очевидно: $(x \in A \setminus B) \rightleftarrows ((x \in A) \wedge (x \notin B)) \vee \big((x \in B) \wedge (x \notin A)\big)$. По определению, $A \Delta B = (A \setminus B) \cup (B \setminus A)$. Диаграмма Венна на рис.6 иллюстрирует симметрическую разность.

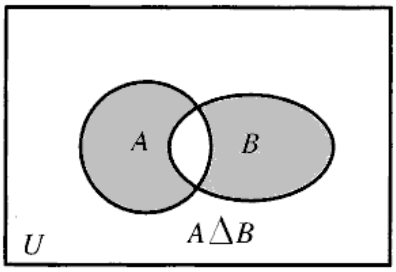


Рис.6. Симметрическая разность

3. Алгоритмы выполнения теоретико-множественных операций

В этом разделе подробно рассмотрена «технология» (точнее, алгоритмы) выполнения введённых в разделе 2 базовых теоретико-множественных операций для конечных множеств.

Пример 5. Дополнение. Пусть $U = \{1, 2, 3, 4, 5, 6, 9\}$, $A = \{2, 5, 4\}$. Сделаем следующее.

1. Выпишем подряд все элементы U: 1, 2, 3, 4, 5, 6, 9.
2. Подчеркнём в этом списке все элементы из A; получим 1, $\underline{2}$, 3, $\underline{4}$, $\underline{5}$, 6, 9.
3. Удалим из списка все подчёркнутые элементы; останутся 1, 3, 6, 9.
4. Положим $A' = \{1, 3, 6, 9\}$ ■

Пример 6. Объединение. Пусть $A = \{a, f, g, r\}$, $B = \{b, d, f, r, t\}$. Сделаем следующее.

1. Выпишем подряд все элементы B: b, d, f, r, t.
2. Подчеркнём в этом списке все элементы, содержащиеся в A; получим $b, d, \underline{f}, \underline{r}, t$.
3. Удалим из списка все подчёркнутые элементы; останутся b, d, t.
4. Добавим оставшиеся элементы b, d, t к списку A; получим a, f, g, r, b, d, t.
5. Положим $A \cup B = \{a, f, g, r, b, d, t\}$ ■

Пример 7. Пересечение. Пусть $A = \{a, f, g, r\}$, $B = \{b, d, f, r, t\}$. Сделаем следующее.

1. Выпишем подряд все элементы A: a, f, g, r.
2. Подчеркнём в этом списке все элементы, не содержащиеся в B; получим $\underline{a}, f, \underline{g}, r$.
3. Удалим из списка все подчёркнутые элементы; останутся f, r.
4. Положим $A \cap B = \{f, r\}$ ■

Пример 8. Разность. Пусть $A = \{a, f, g, r\}$, $B = \{b, d, f, r, t\}$. Сделаем следующее.

1. Выпишем подряд все элементы A: a, f, g, r.
2. Подчеркнём в этом списке все элементы, содержащиеся в B; получим $a, \underline{f}, g, \underline{r}$.
3. Удалим из списка все подчёркнутые элементы; останутся a, g.
4. Положим $A \setminus B = \{a, g\}$ ■

Пример 9. Симметрическая разность. Пусть $A = \{a, f, g, r\}$, $B = \{b, d, f, r, t\}$. Сделаем следующее.

1. Найдём алгоритмом, описанным в примере 8, разность $A \setminus B$. Получим $A \setminus B = \{a, g\}$.
2. Найдём алгоритмом, описанным в примере 8, разность $B \setminus A$. Получим $B \setminus A = \{b, d, t\}$.

3. Найдём алгоритмом, описанным в примере 6, объединение $(A\backslash B)\cup(B\backslash A)$. Получим $\{a, g, b, d, t\}$

4. Положим $A\Delta B = \{a, g, b, d, t\}$ ■

Комбинации нескольких теоретико-множественных операций (например, $A\cap(B'\cup C)$) выполняются аналогичным образом.

Пример 10. Пусть $U = \{1, 2, 3, 4, 5, 6, 9\}$, $A = \{1, 2, 3, 4\}$, $B = \{2, 4, 6\}$, $C = \{1, 3, 6, 9\}$. Найдём следующие множества.

a) $A'\cap B$.

1. Алгоритмом из примера 5 находим $A' = \{5, 6, 9\}$.
2. Алгоритмом из примера 7 находим $A'\cap B = \{5, 6, 9\}\cap\{2, 4, 6\}= \{6\}$.

b) $B'\cup C'$.

1. Алгоритмом из примера 5 находим $B' = \{1, 3, 5, 9\}$ и $C' = \{2, 4, 5\}$.
2. Алгоритмом из примера 6 находим $B'\cup C' = \{1, 3, 5, 9\}\cup\{2, 4, 5\}= \{1, 3, 5, 9, 2, 4\}$.

c) $A\cap(B\cup C')$.

1. $C' = \{2, 4, 5\}$ (см. b)).
2. Алгоритмом из примера 6 находим $B\cup C' =\{2, 4, 6\}\cup\{2, 4, 5\} = \{2, 4, 6, 5\}$.
3. Алгоритмом из примера 7 находим $A\cap(B\cup C') = \{1, 2, 3, 4\}\cap\{2, 4, 6, 5\} = \{2, 4\}$.

d) $(A'\cup C')\cap B'$.

1. $A' = \{5, 6, 9\}$ (см. a)).
2. $B' = \{1, 3, 5, 9\}$, $C' = \{2, 4, 5\}$ (см. b)).
3. Алгоритмом из примера 6 находим $A'\cup C' = \{5, 6, 9\}\cup\{2, 4, 5\}=\{5, 6, 9, 2, 4\}$.
4. Алгоритмом из примера 7 находим $(A'\cup C')\cap B' = \{5, 6, 9, 2, 4\}\cap\{1, 3, 5, 9\}= \{5, 9\}$ ■

Пример 11. Пусть $U = \{a, b, c, d, e, f, g\}$, $X = \{a, c, e\}$, $Y = \{a, b, d, e, f\}$, $Z = \{a, c, d, g\}$. Выполнить операции $X\cap(Y\backslash Z)'$.

1. Алгоритмом из примера 8 находим $Y\backslash Z = \{a, b, d, e, f\}\backslash\{a, c, d, g\} = \{b, e, f\}$
2. Алгоритмом из примера 5 находим $(Y\backslash Z)' = \{b, e, f\}' = \{a, c, d, g\}$.
3. Алгоритмом из примера 7 находим $X\cap(Y\backslash Z)' = \{a, c, e\}\cap\{a, c, d, g\}=\{a, c\}$ ■

4. Проверка равенства двух множеств

Во многих случаях возникает необходимость сравнения двух множеств, каждое из которых получено из одних и тех же исходных подмножеств в результате различных комбинаций нескольких теоретико-множественных операций. Для этого рассмотрим диаграммы Венна для двух и трёх множеств. Они показаны на рис.7 и 8. Цифры на этих рисунках – это просто имена соответствующих подмножеств.

Принципиальными являются следующие почти очевидные факты, которые приводятся без доказательств.

Утверждение 1. α) Результатом выполнения любых теоретико-множественных операций над множествами A и B является объединением некоторых множеств из числа множеств, обозначенных на рис.7 цифрами 1, 2, 3, 4.

β). Результатом выполнения любых теоретико-множественных операций над множествами A, B и C является объединением некоторых множеств из числа множеств, обозначенных на рис.8 цифрами 1, 2, 3, 4, 5, 6, 7, 8.

Таким образом, для определения результатов теоретико-множественных операций можно осуществить операции с множествами, состоящими из символов 1, 2, 3, 4 (для двух множеств) или из символов 1, 2, 3, 4, 5, 6, 7, 8 (для трёх множеств). Такие операции подробно рассмотрены в примерах 10 и 11. Рассмотрим ещё некоторые примеры.

Пример 12. Рассмотрим множество $(A\cap B)'$, где A и B показаны на рис.7. В данном случае $U = \{1, 2, 3, 4\}$; $A =\{2, 4\}$; $B = \{3, 4\}$. В соответствии с описанными ранее алгоритмами выполнения теоретико-множественных операций имеем $A\cap B = \{4\}$, $(A\cap B)' = \{4\}' = \{1, 2, 3\}$. Таким образом, просто подсчитано, что

$$(A\cap B)' = \{1, 2, 3\}. \tag{3a}$$

Рассмотрим теперь множество $A'\cup B'$. Имеем (см. рис.7): $A' = \{1, 3\}$, $B' = \{1, 2\}$, $A'\cup B' = \{1, 3\}\cup\{1, 2\}= \{1, 2, 3\}$, т.е. подсчитано, что

$$A'\cup B' = \{1, 2, 3\} \text{ ■} \tag{3b}$$

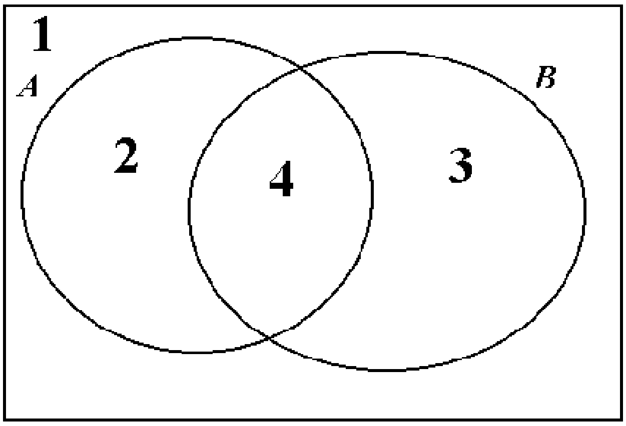


Рис.7

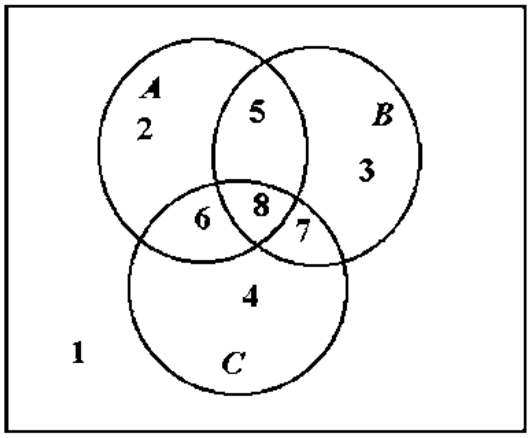


Рис.8

Формулы (3a) и (3b) означают, что $(A \cap B)' = A' \cup B'$ (дополнение к пересечению равно объединению дополнений). Это соотношение называется **1-*м законом де Моргана***. Аналогично выводится и **2-*й закон де Моргана***: $(A \cup B)' = A' \cap B'$ (дополнение к объединению равно пересечению дополнений). Законы де Моргана будут упоминаться в разделе 5-2. Но там они относятся к булевым функциям, а здесь – к двум произвольным подмножествам произвольного универсального множества U.

Пример 13. Рассмотрим множество $A \cup (B \cap C)$, где A, B и C показаны на рис.8. В данном случае $U = \{1, 2, 3, 4, 5, 6, 7, 8\}$; $A = \{2, 5, 6, 8\}$; $B = \{3, 5, 7, 8\}$; $C = \{4, 6, 7, 8\}$. В соответствии с описанными ранее алгоритмами выполнения теоретико-множественных операций имеем $B \cap C = \{3, 5, 7, 8\} \cap \{4, 6, 7, 8\} = \{7, 8\}$; $A \cup (B \cap C) = \{2, 5, 6, 8\} \cup \{7, 8\} = \{2, 5, 6, 7, 8\}$. Таким образом,
$A \cup (B \cap C) = \{2, 5, 6, 7, 8\}$. (4a)

Рассмотрим теперь множество $(A \cup B) \cap (A \cup C)$ с теми же самыми A, B и C. В соответствии с описанными ранее алгоритмами выполнения теоретико-множественных операций имеем $A \cup B = \{2, 5, 6, 8\} \cup \{3, 5, 7, 8\} = \{2, 3, 5, 6, 7, 8\}$; $A \cup C = \{2, 5, 6, 8\} \cup \{4, 6, 7, 8\} = \{2, 4, 5, 6, 7, 8\}$; $(A \cup B) \cap (A \cup C) = \{2, 3, 5, 6, 7, 8\} \cap \{2, 4, 5, 6, 7, 8\} = \{2, 5, 6, 7, 8\}$ Таким образом,
$(A \cup B) \cap (A \cup C) = \{2, 5, 6, 7, 8\}$ ■ (4b)

Формулы (4a) и (4b) означают, что $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$. Эти соотношения выражают ***дистрибутивность объединения относительно пересечения***.

5. Задания

Задание 1. Определить истинность или ложность следующих высказываний:

01.$6 \in \{-2, 5, 8, 9\}$

02.$9 \notin \{6, 3, 4, 8\}$

03.$\{k, c, r, a\} = \{k, c, a, r\}$

04. Множество натуральных чисел, меньших трёх $\neq \{1, 2\}$

05.$8 \subseteq \{3, -2, 5, 7, 8\}$

06.$\{5, 8, 9\} \in \{5, 8, 9, 0\}$

07.$6 \neq \{-2, 5, 8, 9\}$

08.$\varnothing = \{\varnothing\}$

Пусть $A = \{2, 4, 6, 8, 10, 12\}$, $B = \{2, 4, 8, 10, 12\}$, $C = \{4, 10, 12\}$.

09.$4 \in A$

10.$8 \in B$

11.$4 \notin C$

12. Каждый элемент C принадлежит A

13. Каждый элемент C принадлежит B

Пусть $U = \{a, b, c, d, e, f, g\}$, $A = \{a, e\}$, $B = \{a, b, e, f, g\}$, $C = \{b, f, g\}$, $D = \{d, e\}$.

14. $C \subset U$

15. $D \subseteq B$

16. $A \subset B$

17. $B \subseteq C$

18. $\varnothing \subset A$
19. $\varnothing \subseteq \varnothing$
20. $D \subset B$
21. $D \not\subseteq B$
22. $A \not\subseteq B$■

Задание 2. Вставить знак $\in$, $\notin$, $=$ или $\neq$ вместо пробела, чтобы получить истинное высказывание:
01. 5____{2, 4, {5}, 7}
02. {4}____{4, 7, 8, 12}
03. 0____{1, –2, 0, {5}, 9}
04. {3}____{2, 3, 4, 6}
05. 8____{3, –2, 5, 7, 8}
06. –12____{3, 8, 12, 18}
07. 3____{2, 5, 6, 8}
08. b____$\{h, c, d, a, b\}$
09. 9____{6, 3, 4, 8}
10. $\{k, c, r, a\}$____$\{k, c, a, r\}$
11. {5, 8, 9}____{5, 8, 9, 0}
12. 6____{–2, 5, 8, 9}
13. m____$\{l, m, n, o, p\}$
14. $\{e, h, a, n\}$____$\{a, \{h\}, e, n\}$
15. {3, 7, 12, 14}____{3, 7, 12, 14, 0} ■

Задание 3. Вставить знак $\subseteq$ или $\not\subseteq$ вместо пробела, чтобы получить истинное высказывание:
01. {4}____{4, 7, 8, 12}
02. {3}____{2, 3, 4, 6}
03. $\{k, c, r, a\}$____$\{k, c, a, r\}$
04. {5, 8, 9}____{5, 8, 9, 0}
05. $\{e, h, a, n\}$____$\{a, \{h\}, e, n\}$
06. {3, 7, 12, 14}____{3, 7, 12, 14, 0}
07. {–2, 0, 2}_____{–2, –1, l, 2}
08. {Понедельник, Среда, Пятница}___{Воскресенье, Понедельник, Вторник, Среда, Четверг}
09. $\{a, n, d\}$_____$\{r, a, n, d, y\}$
10. $\varnothing$_____$\{a, b, c, d, e\}$
11. $\varnothing$_____$\varnothing$
12. {–7, 4, 9}_____множество всех нечётных целых чисел
13. $\{B, C, D\}$_____$\{B, C, D, F\}$
14. {красный, зелёный, синий, жёлтый}_____{ зелёный, жёлтый, синий, красный}
15. $\varnothing$_____{0}
16. {–1, 0, 1, 2, 3) _____{0, 1, 2, 3, 4} ■

Задание 4. Написать булеан (множество всех подмножеств) для следующих множеств:
01. $\{B, C, D\}$
02. $\{B, C\}$
03. $\{B, \{C\}\}$
04. $\{\{B\},\{C\}\}$
05. $\{B, \{B\}\}$
06. $\{B, \{B\}, C\}$
07. $\{\{\varnothing\}, B, \{C\}\}$
08. $\{a, 1, 0\}$
09. $\{a, 1, \{0\}\}$
10. $\{a, \{a\}, 1, \{0\}\}$

11. $\{-3, 3, \{-3\}\}$
12. {Понедельник, Среда, Пятница}
13. {Понедельник, Среда, August}
14. $\{e, é, è\}$ ■

Задание 5. Найти пересечение множеств (см. пример 7):
01. $\{3, 4, 5, 6, 7\}\cap\{4, 6, 8, 10\}$
02. $\{9, 14, 25, 30\}\cap\{10, 17, 19, 38, 52\}$
03. $\{5, 9, 11\}\cap\varnothing$
04. $\{a, m, n, s, w\}\cap\{c, d, m, o, s\}$
05. $\{P, Q, R\}\cap\{F, H, Q, X, R\}$ ■

Задание 6. Найти объединение множеств (см. пример 6):
01. $\{1, 5, 7\}\cup\{3, 7, 10\}$
02. $\{a, m, n, s, w\}\cup\{c, d, m, o, s\}$
03. $\{P, Q, R\}\cup\{F, H, Q, X, R\}$ ■

Задание 7. Найти разность множеств (см. пример 8):
01. $\{1, 5, 7\}\setminus\{3, 7, 10\}$
02. $\{a, m, n, s, w\}\setminus\{c, d, m, o, s\}$
03. $\{P, Q, R\}\setminus\{F, H, Q, X, R\}$
04. $\{3, 5, 7\}\setminus\{1, 2, 3, 4, 5, 7\}$
05. $\{f, g, h, i, j, k\}\setminus\{g, i, k\}$ ■

Задание 8. Пусть $U = \{a, b, c, d, e, f, g\}$, $X = \{a, c, e, g\}$, $Y = \{a, b, c\}$, $Z = \{b, c, d, e, f\}$. Выполнить указанные операции:
01. $X\cup(Y\cap Z)$
02. $Y\cap(X\cup Z)$
03. $(Y\cap Z')\cup X$
04. $(X'\cup Y')\cap Z$
05. $(Z\cup X')'\cap Y$
06. $(Y\cap X')'\cup Z'$
07. $X\setminus Y$
08. $Y\setminus X$
09. $X'\setminus Y$
10. $Y'\setminus X$
11. $X\cap(X\setminus Y)$
12. $Y\cup(Y\setminus X)$ ■

Задание 9. Пусть $U = \{a, b, c, d, e, f, g\}$, $A = \{a, c, e\}$, $B = \{a, b, d, e, f\}$, $C = \{a, c, d, g\}$. Выполнить указанные операции:
01. $(A'\setminus B)\cup C$
02. $(A\setminus B')\cup C$
03. $(A\setminus B)\cup C'$
04. $(A\setminus B)'\cup C$
05. $A'\setminus(B\cup C)$
06. $A\setminus(B'\cup C)$
07. $A\setminus(B\cup C')$
08. $A\setminus(B\cup C)'$
09. $(A'\setminus B)\cap C$
10. $(A\setminus B')\cap C$
11. $(A\setminus B)\cap C'$
12. $(A\setminus B)'\cap C$
13. $A'\setminus(B\cap C)$
14. $A\setminus(B'\cap C)$
15. $A\setminus(B\cap C')$
16. $A\setminus(B\cap C)'$

17. $(A'\cup B)\setminus C$
18. $(A\cup B')\setminus C$
19. $(A\cup B)\setminus C'$
20. $(A\cup B)'\setminus C$
21. $A'\cup(B\setminus C)$
22. $A\cup(B'\setminus C)$
23. $A\cup(B\setminus C')$
24. $A\cup(B\setminus C)'$
25. $(A'\cap B)\setminus C$
26. $(A\cap B')\setminus C$
27. $(A\cap B)\setminus C'$
28. $(A\cap B)'\setminus C$
29. $A'\cap(B\setminus C)$
30. $A\cap(B'\setminus C)$
31. $A\cap(B\setminus C')$
32. $A\cap(B\setminus C)'$ ■

Задание 10. Путём выполнения заданных теоретико-множественных операций проверить равенство двух множеств (см. примеры 12 и 13):
01. $A\cap(B\cap C) = (A\cap B)\cap C$
02. $A\cup(B\cup C) = (A\cup B)\cup C$
03. $A\cap(B\cup C) = (A\cap B)\cup(A\cap C)$
04. $A\cup(B\cap C) = (A\cup B)\cap(A\cup C)$
05. $(A\cup B)\cap A = (A\cap B)\cup A = A$
06. $(A\cap B)' = A'\cup B'$
07. $(A\cup B)' = A'\cap B'$
08. $A\setminus(B\cup C) = (A\setminus B)\cap(A\setminus C)$
09. $A\setminus(B\cap C) = (A\setminus B)\cup(A\setminus C)$

10. $A\setminus(A\setminus B) = A\cap B$
11. $A\setminus B = A\setminus(A\cap B)$
12. $A\cap(B\setminus C) = (A\cap B)\setminus(A\cap C) = (A\cap B)\setminus C$
13. $(A\setminus B)\setminus C = (A\setminus C)\setminus(B\setminus C)$
14. $A\cup B = A\cup(B\setminus A)$
15. $(A')' = A$
16. $A\cup A' = U$, где U – универсальное множество
17. $A\cap A' = \varnothing$
18. $(A\cap B)\cup(A\cap B') = (A\cup B)\cap(A\cup B') = A$
19. $(A'\cup B)\cap A = A\cap B$
20. $A\cap(B\setminus A) = \varnothing$
21. $(A\cup B)\setminus C = (A\setminus C)\cup(B\setminus C)$

5. Предметный указатель

Глава 3. Кортежи

1. Понятие кортежа

Как понятия высказывания и множества, понятие ***кортежа*** будет исходным, неопределяемым. Можно говорить о кортеже книг, стоящих на полке, о кортеже спортсменов, построенных в один ряд по росту, векторе, состоящем из n координат, и т.д. Синонимами слова «кортеж» служат слова «***вектор***», «***набор***», «***последовательность***» и пр.

Наряду с термином «кортеж» вводится (также в качестве исходного) термин «***компонента***», или «***координата***» кортежа. В самом общем виде компонента кортежа (точнее, его i-ая компонента) – это то, что находится в данном кортеже на i-ом месте. Это может быть i-ая (считая слева) книга на полке, i-ый по алфавиту школьник в классе, i-ый по росту школьник в том же классе, i-ая координата вектора, и т.д. Число компонент кортежа называется его ***длиной***. Кортеж длины s, первая компонента которого есть a_1, вторая $-a_2$, ..., s-я, последняя, $-a_s$, обозначается знакосочетанием $\langle a_1, a_2, ..., a_s\rangle$ (компоненты перечислены по порядку, разделены запятыми и взяты в угловые скобки). Обратим внимание на принципиальное различие между кортежем $\langle a_1, a_2, ..., a_s\rangle$ и множеством $\{a_1, a_2, ..., a_s\}$ всех его компонент. Элементы множества, хотя и выписаны в некотором порядке, на самом деле совершенно «равноправны», и множества $\{a_1, a_2, ..., a_s\}$ и $\{a_s, a_{s-1}, ..., a_1\}$ совпадают. В то же время кортежи $\langle a_1, a_2, ..., a_s\rangle$ и $\langle a_s, a_{s-1}, ..., a_1\rangle$ являются, вообще говоря, разными. При задании множеств не только перечислением (как это делалось до сих пор), а и другими способами (см. далее главу 4) различие между множествами и кортежами станет более явным. Заметим также, что все элементы множества предполагаются различными, а разные компоненты кортежа могут совпадать.

Кортежи длины 2 называются ***парами***, длины 3 – ***тройками*** и т.д. Для удобства вводят в рассмотрение кортеж длины 0, не содержащий вообще компонент, т.е. имеющий вид $\langle\rangle$. Его обозначают буквой Λ (лямбда).

Кортежи будем обозначать строчными буквами из начала греческого алфавита. Будем считать, что кортеж α равен кортежу β, если, во-первых, они одинаковой длины, и, во-вторых, каждая компонента α равна компоненте β с тем же номером. Равенство кортежей α и β выражается как $\alpha = \beta$, а неравенство – как $\alpha \neq \beta$. Кортеж α называется ***кортежем над множеством*** X, если каждая компонента α есть элемент из X. Компонентами кортежа могут быть объекты любой природы, в частности, множества и другие кортежи.

Пример 1. Рассмотрим следующий кортеж длины 3: $\alpha = \langle a, \{a\}, \langle a\rangle\rangle$. Его 1-ая компонента – это некоторый элемент a, 2-ая компонента – одноэлементное множество $\{a\}$, 3-ья компонента – кортеж $\langle a\rangle$ длины 1. Все три компоненты являются <u>разными объектами</u> ■

2. Прямое произведение множеств

Используя понятие кортежа, можно определить ещё одну, очень важную для приложений теоретико-множественную операцию – операцию прямого произведения. ***Прямым произведением множеств*** X и Y называется множество, состоящее из всех тех и только тех <u>пар</u>, первая компонента каждой из которых принадлежит X, а вторая – Y. Прямое произведение множеств X и Y, рассматриваемых в <u>указанном порядке</u>, обозначается знакосочетанием $X\times Y$. Обратим внимание, что пока определено только прямое произведение <u>двух</u> множеств. Заметим также, что прямое произведение двух множеств является множеством кортежей длины 2, т.е. множеством пар.

Пример 2. Пусть $A = \{m, n\}$, $B = \{p, q\}$, $C = \{s, t\}$. Тогда $A\times B= \{\langle m, p\rangle, \langle n, p\rangle, \langle m, q\rangle, \langle n, q\rangle\}$; $B\times C= \{\langle p, s\rangle, \langle p, t\rangle, \langle q, s\rangle, \langle q, t\rangle\}$. Далее, $(A\times B)\times C = \{\langle\langle m, p\rangle, s\rangle, \langle\langle n, p\rangle, s\rangle, \langle\langle m, q\rangle, s\rangle, \langle\langle n, q\rangle, s\rangle, \langle\langle m,$

$p\rangle, t\rangle$, $\langle\langle n, p\rangle, t\rangle$, $\langle\langle m, q\rangle, t\rangle$, $\langle\langle n, q\rangle, t\rangle\}$; $A\times(B\times C) = \{\langle m, \langle p, s\rangle\rangle, \langle m, \langle p, t\rangle\rangle, \langle m, \langle q, s\rangle\rangle, \langle m, \langle q, t\rangle\rangle, \langle n, \langle p, s\rangle\rangle, \langle n, \langle p, t\rangle\rangle, \langle n, \langle q, s\rangle\rangle, \langle n, \langle q, t\rangle\rangle\}$. Заметим, что $(A\times B)\times C \neq A\times(B\times C)$, поскольку эти множества состоят из разных объектов; в частности, их первые компоненты $\langle\langle m, p\rangle, s\rangle$ и $\langle m, \langle p, s\rangle\rangle$ не равны друг другу■

Тем не менее понятие прямого произведения легко распространяется на любое конечное число множеств. Пусть $\{X_i\}$ $(1 \le i \le n)$ – конечное семейство множеств. ***Прямым произведением семейства множеств*** называется множество, состоящее из всех тех и только тех кортежей длины n, первая компонента каждого из которых принадлежит X_1, вторая – X_2, ..., n-ая –X_n. Прямое произведение указанного семейства обозначается знакосочетанием $X_1\times X_2\times \dots \times X_n$, или, короче, $\prod_{i=1}^{n} X_i$. Если в семействе $\{X_i\}$ $(1 \le i \le n)$ все множества одинаковы и равны, например, множеству M, то прямое произведение этого семейства называется ***n-й степенью множества*** M и обозначается через M^n. По определению полагают $M^1 = M$, $M^0 = \Lambda$.

Пример 3. Пусть, как и в примере 2, $A = \{m, n\}$, $B = \{p, q\}$, $C = \{s, t\}$. Тогда $A\times B\times C = \{\langle m, p, s\rangle, \langle n, p, s\rangle, \langle m, q, s\rangle, \langle n, q, s\rangle, \langle m, p, t\rangle, \langle n, p, t\rangle, \langle m, q, t\rangle, \langle n, q, t\rangle\}$, т.е. это множество кортежей длины 3. Естественно, что $A\times B\times C \neq (A\times B)\times C$ и $A\times B\times C \neq A\times(B\times C)$ ■

Пример 4. Пусть $A = \{a, b, c\}$, $B = \{p, q\}$, $C = \{a, q\}$. Найти $(A\times B)\times C$, $A\times(B\times C)$ и $A\times B\times C$.
По определению прямого произведения $A\times B = \{\langle a, p\rangle, \langle a, q\rangle, \langle b, p\rangle, \langle b, q\rangle, \langle c, p\rangle, \langle c, q\rangle\}$, $B\times C = \{\langle p, a\rangle, \langle p, q\rangle, \langle q, a\rangle, \langle q, q\rangle\}$. Далее,

$(A\times B)\times C = \{\langle\langle a, p\rangle, a\rangle, \langle\langle a, p\rangle, q\rangle, \langle\langle a, q\rangle, a\rangle, \langle\langle a, q\rangle, q\rangle, \langle\langle b, p\rangle, a\rangle, \langle\langle b, p\rangle, q\rangle, \langle\langle b, q\rangle, a\rangle, \langle\langle b, q\rangle, q\rangle, \langle\langle c, p\rangle, a\rangle, \langle\langle c, p\rangle, q\rangle, \langle\langle c, q\rangle, a\rangle, \langle\langle c, q\rangle, q\rangle\}$;

$A\times(B\times C) = \{\langle a, \langle p, a\rangle\rangle, \langle a, \langle p, q\rangle\rangle, \langle a, \langle q, a\rangle\rangle, \langle a, \langle q, q\rangle\rangle, \langle b, \langle p, a\rangle\rangle, \langle b, \langle p, q\rangle\rangle, \langle b, \langle q, a\rangle\rangle, \langle b, \langle q, q\rangle\rangle, \langle c, \langle p, a\rangle\rangle, \langle c, \langle p, q\rangle\rangle, \langle c, \langle q, a\rangle\rangle, \langle c, \langle q, q\rangle\rangle\}$;

$A\times B\times C\rangle = \{\langle a, p, a\rangle, \langle a, p, q\rangle, \langle a, q, a\rangle, \langle a, q, q\rangle, \langle b, p, a\rangle, \langle b, p, q\rangle, \langle b, q, a\rangle, \langle b, q, q\rangle, \langle c, p, a\rangle, \langle c, p, q\rangle, \langle c, q, a\rangle, \langle c, q, q\rangle\}$.

Следует обратить внимание на то, что 1-ое и 2-ое множества являются множествами кортежей длины 2, в то время как 3-ье множество является множеством кортежей длины 3 ■

3. Операция проектирования

Используя введенные понятия, определим еще одну теоретико-множественную операцию – ***проектирование***, применяемую только к множеству кортежей одинаковой длины. Поскольку проекция множества кортежей определяется через проекцию кортежа, начнем с её определения.

Пусть $\alpha = \langle a_1, a_2, \dots, a_s\rangle$ – кортеж длины $s > 0$.

1) ***Проекцией кортежа α на i-ю ось*** называется и через $\text{ПР}_i\alpha$ обозначается i-я компонента кортежа α, т.е. a_i. Таким образом, $\text{ПР}_i\alpha = a_i$ $(i = 1, \dots, s)$.

2) Пусть $2 \le q \le s$ и $1 \le i_1 < i_2 < \dots < i_{q-1} < i_q \le s$. ***Проекцией кортежа α на оси с номерами*** $i_1, i_2, \dots, i_q$ называется и через $\text{ПР}_{i_1 \dots i_q}\alpha$ обозначается кортеж $\langle a_{i_1}, \dots, a_{i_q}\rangle$. Таким образом, $\text{ПР}_{i_1 \dots i_q}\alpha = \langle a_{i_1}, \dots, a_{i_q}\rangle$.

3) ***Проекцией кортежа α на пустое множество осей*** называется и через $\text{ПР}_\varnothing\alpha$ обозначается пустой кортеж Λ. Таким образом, $\text{ПР}_\varnothing\alpha = \Lambda$.

4) ***Проекцией пустого кортежа*** Λ ***на пустое множество осей*** называется и через $\text{ПР}_\varnothing\Lambda$ обозначается пустой кортеж Λ. Таким образом, $\text{ПР}_\varnothing\Lambda = \Lambda$.

Пример 5. Если $\alpha = \langle x, y\rangle$, то $\text{ПР}_1\alpha = x$, $\text{ПР}_2\alpha = y$. Если $\alpha = \langle\{x\}, \langle y\rangle\rangle$, то $\text{ПР}_1\alpha = \{x\}$, $\text{ПР}_2\alpha = \langle y\rangle$. Если $\alpha = \langle\langle x, \langle y\rangle\rangle, x\rangle$, то 1-ой компонентой кортежа α является кортеж $\langle x, \langle y\rangle\rangle$ длины 2, а его 2-ой компонентой – элемент x, т.е. $\text{ПР}_1\alpha = \langle x, \langle y\rangle\rangle$, $\text{ПР}_2\alpha = x$■

Определим теперь понятие «проекция множества». Как уже было указано выше, это понятие будет определено только для того случая, когда проектируемое множество состоит из кортежей, причем все эти кортежи имеют одинаковую длину.

Пусть M – множество кортежей длины $s > 0$. Поскольку пустое множество $\varnothing$ не является множеством кортежей длины $s > 0$, то множество M предполагается непустым.

1) ***Проекцией множества M на i-ю ось*** называется и через ПР_iM обозначается множество проекций кортежей из M на i-ю ось $(i = 1, \dots, s)$.

2) Пусть $2 \le q \le s$ и $1 \le i_1 < i_2 < \dots < i_{q-1} < i_q \le s$. ***Проекцией множества M на оси с номерами*** $i_1, i_2,$

..., i_q называется и через $\text{ПР}_{i_1 \dots i_q} M$ обозначается множество проекций кортежей из M на оси с номерамиi_1, i_2, ..., i_{q-1}, i_q.

3) ***Проекцией множества М на пустое множество осей*** называется и через $\text{ПР}_{\varnothing} M$ обозначается пустой кортеж Λ. Таким образом, $\text{ПР}_{\varnothing} M = \Lambda$.

Пример 6. Пусть $M = \{\langle a, p, q\rangle, \langle a, q, q\rangle, \langle b, p, q\rangle, \langle b, q, a\rangle, \langle b, q, q\rangle, \langle c, p, a\rangle, \langle c, p, q\rangle, \langle c, q, q\rangle\}$. Найдём проекцию ПР_{23} заданного множества M кортежей на оси с номерами 2 и 3. По определению проекции множества, $\text{ПР}_{23} M$ состоит из всех проекций всех кортежей из M на указанные оси. Из определения проекции кортежей, проекция отдельного кортежа α – это просто кортеж, состоящий из компонент αс соответствующими номерами. Так, для 1-го по порядку кортежа из M – $\langle a, p, q\rangle$ – его проекцией $\text{ПР}_{23} \langle a, p, q\rangle$ на оси 2, 3 будет кортеж (длины 2) $\langle p, q\rangle$. Далее, $\text{ПР}_{23}\langle a, q, q\rangle = \langle q, q\rangle$, и т.д. В результате получаем: $\text{ПР}_{23}\text{M} = \{\langle p, q\rangle, \langle q, q\rangle, \langle q, a\rangle, \langle p, a\rangle\}$. Здесь в проектируемом множестве M 8 кортежей, а в проекции – 4. Это происходит потому, что проекции различных кортежей из исходного множества совпадают. Например, $\text{ПР}_{23} \langle a, q, q\rangle = \text{ПР}_{23} \langle c, q, q\rangle = \langle q, q\rangle$■

Пример 7. Пусть A и B – два произвольных множества, $M = A \times B$. По определению операций прямого произведения и проектирования имеем $\text{ПР}_1 M = A$, $\text{ПР}_2 M = B$. Поэтому можно сказать, что операции прямого произведения и проектирования являются взаимно-обратными (не уточняя этого понятия) ■

Пример 8. Пусть M – множество точек $\langle x, y\rangle$на плоскости, удовлетворяющих условию $x^2 + y^2 = 1$ (т.е. M – это окружность единичного радиуса с центром в начале координат). Нетрудно видеть, что по определению проекции $\text{ПР}_1 M = \text{ПР}_2 M = [-1, 1]$ (так обозначен отрезок с концами -1 и 1). Таким образом, для геометрических фигур на плоскости (подмножеств двумерного евклидова пространства E^2), состоящих из двумерных точек – кортежей длины 2, введённая здесь операция проектирования совпадает с хорошо известной операцией проектирования вдоль координатных осей в геометрии ■

Далее будут рассмотрены понятия, которые, в отличие от рассмотренных выше исходных понятий высказывания, множества и кортежа, будут формально определены – через эти, ранее введённые неопределяемые понятия.

4. Графики

Одним из важнейших понятий дискретной – и не только дискретной – математики является понятие графика. ***График – это множество пар***, т.е. множество, элементами которого служат пары. Вспоминая (см. раздел 1), что пара – это кортеж длины 2, можно сказать, что графиком называется <u>любое множество кортежей длины 2</u>. Как и рассмотренное в разделе 3 понятие проектирования, понятия графика также является обобщением хорошо известного «школьного» понятия графика функции.

Пример 9. Вспомним хорошо известный график функции $y = \sin x$. По построению, такой график состоит из всех пар чисел $\langle x, y\rangle$ (точек), таких, что $y = \sin x$. Поэтому, как и <u>любое множество точек на плоскости</u>, график данной функции является графиком и в смысле введённого определения, т.е. он является множеством пар ■

Заметим, что множества точек на плоскости из примеров 8 и 9 являются бесконечными. Они задаются не перечислением, а условиями на принадлежащие им элементы – набором характеристических свойств, т.е. таким набором свойств, которым обладают только элементы рассматриваемого множества. Подробнее такой способ задания множеств будет рассмотрен далее, в разделе 4-2.1.

Областью определения графика G называется множество $\text{ПР}_1 G$, а ***областью значений графика*** G – множество $\text{ПР}_2 G$. Таким образом, нахождение областей определения и значений графика сводится к операции проектирования кортежей длины 2 на одну из двух осей. Эта операция является частным случаем операции проектирования (проектирование на одну ось), рассмотренной в разделе 3.

4.1. Операции над графиками. Рассмотрим две важные операции над графиками: одноместную – инверсию, и двухместную – композицию. Инверсия графика определяется через инверсию пары. Пара $\langle c, d\rangle$ называется ***инверсией пары*** $\langle a, b\rangle$, если $c = b$, $d = a$. Другими словами, инверсией пары $\langle a, b\rangle$ является пара $\langle b, a\rangle$. Инверсия пары α обозначается через α^{-1}. Легко видеть, что $(\alpha^{-1})^{-1} = \alpha$. ***Инверсией графика*** G называется множество инверсий всех пар из G. Инверсия графика Gобозначается через G^{-1}. График называется ***симметрическим***, если $G = G^{-1}$. Для симметрических графиков истинны следующие два высказывания: $\alpha\in G \leftrightarrows \alpha\in G^{-1}$ и $\alpha\in G \leftrightarrows \alpha^{-1}\in G$ (напомним, что знаком $\leftrightarrows$ обозначена определённая в разделе 1-2.1.5 операция «эквивалентность» над истинностными значениями высказываний). Легко видеть также, что истинность любого из этих двух высказываний влечёт равенство $G = G^{-1}$.

Пример 10. Пусть X – произвольное множество. Рассмотрим множество X^D всех пар вида $\langle x, x\rangle$, где $x\in X$. Легко видеть, что X^D – симметрический график. Он называется ***диагональю множества*** X^2.

Введём необходимые понятия. Пусть $\alpha = \langle p, q\rangle$, $\beta = \langle s, t\rangle$ – две пары. ***Композицией*** $\boldsymbol{\alpha\circ\beta}$ ***пар*** $\boldsymbol{\alpha}$ ***и*** $\boldsymbol{\beta}$ (в указанном порядке) называется пара γ, определяемая следующим образом:

$$\gamma = \begin{cases} \langle p, t\rangle, & \text{если } q = s \\ \Lambda, & \text{в противном случае} \end{cases}, \qquad (1)$$

где Λ – пустой кортеж (см. раздел 1.1).

Пример 11. Композицией пар $\alpha = \langle p, q\rangle$ и $\beta = \langle q, r\rangle$ при любых p, q и r в соответствие с формулой (1) является пара $\gamma = \langle p, r\rangle$. Композицией пар $\beta = \langle q, r\rangle$ и $\alpha = \langle p, q\rangle$ при $p = r$ является пара $\langle q, q\rangle$. Композицией пар $\beta = \langle q, r\rangle$ и $\alpha = \langle p, q\rangle$ при $p \neq r$ является пустой кортеж Λ. Композицией пар $\alpha = \langle\langle p, r\rangle,q\rangle$ и $\beta = \langle q, r\rangle$ является пара $\langle\langle p, r\rangle,r\rangle$. Действительно, формула (1) при $q = s$ определяет пару $\langle p, t\rangle$ при произвольных p и t. В данном случае на первом месте (вместо p) стоит пара $\langle p, r\rangle$, а на втором месте (вместо t) стоит элемент r. Ещё раз почеркнём, что компонентами кортежа могут быть любые объекты, включая множества и другие кортежи ■

Исходя из операции композиции двух пар, введём теперь операцию композицию двух графиков, т.е. множеств пар (см. определение графика). ***Композиция*** $R= P\circ Q$ определяется как множество композиций всех пар из P со всеми парами из Q. Формально:

$$P\circ Q = \bigcup_{\alpha\in P, \beta\in Q} \alpha\circ\beta, \qquad (2)$$

где композиция двух пар $\alpha\circ\beta$ определена формулой (1).

Пример 12. Пусть график $P = \{\langle a, b\rangle, \langle a, c\rangle\}$, график $Q= \{\langle b, b\rangle, \langle d, c\rangle\}$. Найдём композицию графиков $P\circ Q$. Имеем в соответствии с формулой (2) $P\circ Q = \{\langle a, b\rangle\circ\langle b, b\rangle, \langle a, b\rangle\circ\langle d, c\rangle, \langle a,c\rangle\circ\langle b, b\rangle, \langle a, c\rangle\circ\langle d, c\rangle\}$. В соответствии с формулой (1) для композиций двух пар имеем

$\langle a, b\rangle\circ\langle b, b\rangle = \langle a, b\rangle$, $\langle a, b\rangle\circ\langle d, c\rangle = \langle a, c\rangle\circ\{\langle b, b\rangle = \langle a, c\rangle\circ\langle d, c\rangle = \Lambda$.

Поэтому $P\circ Q = \{\langle a, b\rangle\}$ (множество пар из $P\circ Q$ состоит из одной пары $\langle a, b\rangle$).

Найдём теперь композицию графиков $Q\circ P$ при тех же самых Q и P. Имеем $Q\circ P = \{\langle b, b\rangle\circ\langle a, b\rangle, \langle b, b\rangle\circ\langle a,c\rangle, \langle d, c\rangle\circ\langle a, b\rangle, \langle d, c\rangle\circ\langle a, c\rangle\} = \Lambda$. В этом порядке композиция оказалась пустой ■

Для бесконечных графиков формула (2) остаётся в силе, однако непосредственное рассмотрение всех пар из PиQ (как это делается в примере 12) невозможно. Однако для нахождения композиции $P\circ Q$ можно воспользоваться следующим простым соображением, справедливым для произвольных графиков.

Утверждение 1. Пара $\langle x, y\rangle\in P\circ Q$ тогда и только тогда, когда существует элемент z, такой, что $\langle x, z\rangle\in P$ и $\langle z, y\rangle\in Q$ ■

Пример 13. Рассмотрим композицию двух графиков PиQ: $y = \sin x$ и $y = \ln x$. В соответствии с вышесказанным, пара чисел $\langle x, y\rangle\in P\circ Q$ тогда и только тогда, когда существует элемент z, такой, что $z = \sin x$ и $y = \ln z$. В данном случае это означает, что $\ln(\sin x)$ определён, что может быть при любом x, для которого $\sin x > 0$. А для последнего необходимо и достаточно, чтобы выполнялось условие $2k\pi<x<(2k+1)\pi$ для какого-нибудь целого числа k. Соответствующее значение y из пары $\langle x, y\rangle\in P\circ Q$ определяется формулой $y = \ln(\sin x)$ ■

Пример 13 показывает, что достаточно сложное – на первый взгляд – понятие композиции двух графиков является обобщением хорошо известного «школьного» понятия суперпозиции двух функций.

Если у нас имеется три графика: P, Q и R, то с помощью операции композиции двух графиков из них можно определить два разных графика: $(P\circ Q)\circ R$ и $(P\circ(Q\circ R)$. Имеет место

Утверждение 2. Графики $(P\circ Q)\circ R$ и $P\circ(Q\circ R)$ совпадают, т.е. состоят из одних и тех же пар ■

Утверждение 2 выражает важное свойство операции композиции – её ассоциативность. Это означает, что в выражениях $(P\circ Q)\circ R$ и $P\circ(Q\circ R)$, как и в более сложных выражениях такого же типа, можно убрать скобки и рассматривать композицию не только двух, но и любого числа графиков: $P\circ Q\circ R$, $P\circ Q\circ R\circ S$, и т.д.

4.2. Свойства графиков. График называется ***функциональным*** (***инъективным***), если в нем нет пар с одинаковыми первыми (соответственно одинаковыми вторыми) компонентами.

Пример 14. График $\{\langle b, b\rangle, \langle a, n\rangle\}$ является функциональным и инъективным, поскольку в обеих входящих в него парах и первые, и вторые компоненты являются разными: $b\neq a$ (первые компоненты) и $b\neq n$ (вторые компоненты). Заметим, что совпадение компонент в паре $\langle b, b\rangle$ никак не влияет на рассматриваемые свойства. График $\{\langle x, b\rangle, \langle x, a\rangle\}$ не является функциональным, но является инъективным (первые компоненты совпадают, а вторые – нет). График $\{\langle n, c\rangle, \langle d, c\rangle\}$ является функциональным, но не является инъективным (вторые компоненты совпадают, а первые – нет). Наконец, график $\{\langle n, b\rangle, \langle n, c\rangle, \langle d, c\rangle\}$, состоящий из трёх пар, не является ни функциональным (поскольку он содержит пары $\langle n, b\rangle$ и $\langle n, c\rangle$ с совпадающими первыми компонентами), ни инъективным (поскольку он содержит пары $\langle n, c\rangle$ и $\langle d, c\rangle$ с совпадающими вторыми компонентами) ■

Пример 15. Рассмотрим график, состоящий из всех точек $\langle x, y\rangle$, удовлетворяющих уравнению окружности $x^2 + y^2 = 1$. Этот график не является ни функциональным (поскольку он содержит пары $\langle 0, 1\rangle$ и $\langle 0, -1\rangle$ с совпадающими первыми компонентами), ни инъективным (поскольку он содержит пары $\langle 1, 0\rangle$ и $\langle -1, 0\rangle$ с совпадающими вторыми компонентами) ■

Пример 16. Рассмотрим график, состоящий из всех точек $\langle x, y\rangle$, удовлетворяющих уравнению $y = \ln(x)$. Этот график является функциональным и инъективным, поскольку он является графиком строго возрастающей функции (т.е. $(x_1 \neq x_2) \leftrightarrows (y_1 \neq y_2)$) ■

5. Соответствия и функции

Соответствием называется тройка, первая компонента которой есть подмножество прямого произведения множеств, являющихся ее второй и третьей компонентами. Обратим внимание на то, что все объекты, участвующие в этом определении – тройка, компонента, подмножество, прямое произведение, множество – ранее уже были введены и объяснены. В то же время приведённое определение не включает в себя понятий зависимой и независимой переменной, закона, правила (по которым находится значение зависимой переменной, соответствующее данному значению независимой), и других нуждающихся в объяснении понятий.

Соответствия будут обозначаться прописными греческими буквами. Таким образом, если $\Gamma = \langle G, X, Y\rangle$ – соответствие, то, в согласии с определением, X, Y– множества, а $G\subseteq X\times Y$. По построению, Gявляется графиком, поскольку G – подмножество прямого произведения двух множеств, которое по определению является множеством пар (см. раздел 1.2). Множество G называется ***графиком соответствия*** Γ. Множества X и Y носят название ***области отправления*** и ***области прибытия*** соответствия Γ. Множество ПР_1G называется ***областью определения*** соответствия Γ, а множество ПР_2G – ***областью значений*** соответствия Γ (определения проекции см. в разделе 1.3).

Если пара $\langle x, y\rangle\in G$, то говорят, что элемент y соответствует элементу x в (или при) соответствии Γ. Если $x\in\text{ПР}_1G$, то говорят, что соответствие Γ определено на элементе x. Элемент y называется также ***образом элемента*** x в (или при) соответствии Γ.

Инверсией соответствия $\Gamma = \langle G, X, Y\rangle$ называется и через Γ^{-1} обозначается соответствие $\langle G^{-1}, Y, X\rangle$, где G^{-1} – инверсия графика G (см. начало раздела 4.1). Ясно, что $(\Gamma^{-1})^{-1} = \Gamma$. Если $\Gamma = \langle G, X, Y\rangle$и $\Delta =\langle H, U, V\rangle$ – соответствия, то соответствие $\Sigma = \langle G\circ H,\ X, V\rangle$ называется их ***композицией*** и обозначается через $\Gamma\circ\Delta$. Из ассоциативности композиции графиков следует ***ассоциативность*** композиции соответствий.

Сужением соответствия $\Gamma = \langle G, X, Y\rangle$ на множество A называется и через Γ_A обозначается соответствие $\langle G\cap(A\times Y), X, Y\rangle$. Обратим внимание, что области отправления и прибытия соответствия не меняются. Соответствие $\Delta =\langle H, Z, U\rangle$ называется ***продолжением соответствия*** $\Gamma = \langle G, X, Y\rangle$, если $G\subseteq H$, $X\subseteq Z$, $Y\subseteq U$.

Введём ещё одно понятие, связанное с графиками и соответствиями. Пусть G – произвольный график. Введём в рассмотрение ***соответствие графика G***: $\Gamma_G = \langle G, \text{ПР}_1 G, \text{ПР}_2 G\rangle$ (напомним, что через $\text{ПР}_1 G$ и $\text{ПР}_2 G$ обозначены проекции графика G). У соответствия Γ_G область отправления совпадает с областью определения, а область прибытия – с областью значений. Более того, имеет место простое

Утверждение 3. Любое соответствие с графиком G является продолжением соответствия Γ_G.

Соответствие называется ***функциональным***, или ***функцией***, если его график функционален; ***инъективным***, если его график инъективен; ***всюду определенным***, если его область определения совпадает с областью отправления, и ***сюръективным***, если его область прибытия совпадает с областью значений.

Соответствие, обладающее четырьмя перечисленными свойствами, называется ***взаимнооднозначным***, или ***биективным***, или ***биекцией***.

Функция Γ с областью отправления X и областью прибытия Y называется ***функцией типа*** $X\rightarrow Y$. Напомним, что образом элемента x называется единственный (в силу фунциональности Γ) элемент y, такой, что пара $\langle x, y\rangle\in G$. Образ yэлемента x при функции Γ обозначается через $\Gamma(x)$. Это уже близко к привычному обозначению $y = f(x)$. Для «возвращения» к школьным понятиям необходимо дать аккуратное описание термина «переменная». С этого и будет начинаться следующая глава.

6. Задания

Задание 1. См. примеры 2, 3, 4 для образца.

01. Пусть $A = \{a, b, c\}$, $B = \{\langle p\rangle, q\}$, $C = \{a, q\}$. Найти $(A\times B)\times C$ и $A\times B\times C$.
02. Пусть $A = \{a, b, c\}$, $B = \{p, q\}$, $C = \{a, q\}$. Найти $A\times(B\times C)$ и $A\times B\times C$.
03. Пусть $A = \{\langle p, q\rangle, m\}$, $B = \{m, a\}$, $C = \{a, q\}$. Найти $(A\times B)\times C$ и $A\times B\times C$.
04. Пусть $A = \{\langle p, q\rangle, m\}$, $B = \{m, a\}$, $C = \{q, a\}$. Найти $A\times(B\times C)$ и $A\times B\times C$.
05. Пусть $A = \{p, \langle q, m\rangle\}$, $B = \{m, a\}$, $C = \{a, q\}$. Найти $(A\times B)\times C$ и $A\times B\times C$.
06. Пусть $A = \{a, c\}$, $B = \{b,p, q\}$, $C = \{a, q\}$. Найти $A\times(B\times C)$ и $A\times B\times C$.
07. Пусть $A = \{a, c\}$, $B = \{b,p, q\}$, $C = \{a, q\}$. Найти $(A\times B)\times C$ и $A\times B\times C$.
08. Пусть $A = \{a, c\}$, $B = \{b,p, q\}$, $C = \{a, q\}$. Найти $(A\times B)\times C$ и $A\times(B\times C)$.
09. Пусть $A = \{a, b, c\}$, $B = \{p, q\}$, $C = \{a, p, q\}$. Найти $A\times(B\times C)$ и $A\times B\times C$.
10. Пусть $A = \{a, b, c\}$, $B = \{p, q\}$, $C = \{a, p, q\}$. Найти $A\times B)\times C$ и $A\times B\times C$.
11. Пусть $A = \{a, b, c\}$, $B = \{p, q\}$, $C = \{a, p, q\}$. Найти $(A\times B)\times C$ и $A\times(B\times C)$.
12. Пусть $A = \{a, c\}$, $B = \{b,p, q\}$, $C = \{q\}$. Найти $A\times(B\times C)$ и $A\times B\times C$.
13. Пусть $A = \{a\}$, $B = \{b,p, q\}$, $C = \{a, q\}$. Найти $(A\times B)\times C$ и $A\times B\times C$.
14. Пусть $A = \{a, c\}$, $B = \{b,p\}$, $C = \{a, q\}$. Найти $(A\times B)\times C$ и $A\times(B\times C)$.
15. Пусть $A = \{b, c\}$, $B = \{b, q\}$, $C = \{a, p, q\}$. Найти $A\times(B\times C)$ и $A\times B\times C$.
16. Пусть $A = \{a, b\}$, $B = \{p, q\}$, $C = \{a, p, q\}$. Найти $A\times B)\times C$ и $A\times B\times C$.
17. Пусть $A = \{a, b, c\}$, $B = \{b\}$, $C = \{b, q\}$. Найти $(A\times B)\times C$ и $A\times(B\times C)$.
18. Пусть $A = \{a, b, m\}$, $B = \{m, q\}$, $C = \{p, q\}$. Найти $A\times B)\times C$ и $A\times B\times C$.
19. Пусть $A = \{a, m\}$, $B = \{m, q\}$, $C = \{a, m, q\}$. Найти $(A\times B)\times C$ и $A\times(B\times C)$.
20. Пусть $A = \{a, c\}$, $B = \{b,p, q\}$, $C = \{r, m\}$. Найти $A\times(B\times C)$ и $A\times B\times C$.
21. Пусть $A = \{\langle p, q\rangle, m\}$, $B = \{m, a\}$, $C = \{a, q\}$. Найти $(A\times B)\times C$ и $A\times B\times C$.
22. Пусть $A = \{\langle p, q\rangle, m\}$, $B = \{m, a\}$, $C = \{q, a\}$. Найти $A\times(B\times C)$ и $A\times B\times C$■

Задание 2. См. примеры 5, 6 для образца.

01. Пусть $M = \{\langle a, p, q\rangle, \langle a, q, q\rangle, \langle b, q, p\rangle, \langle b, q, a\rangle, \langle b, q, q\rangle, \langle c, p, a\rangle, \langle c, p, q\rangle, \langle c, q, q\rangle\}$. Найти проекцию ПР_{13} заданного множества M кортежей на оси с номерами 1 и 3.
02. Пусть $M = \{\langle a, p, q\rangle, \langle a, q, q\rangle, \langle b, p, q\rangle, \langle b, q, a\rangle, \langle b, q, q\rangle, \langle c, p, a\rangle, \langle c, p, q\rangle, \langle c, b, q\rangle\}$. Найти проекцию ПР_{12} заданного множества M кортежей на оси с номерами 1 и 2.
03. Пусть $M = \{\langle a, p, q\rangle, \langle a, q, q\rangle, \langle b, p, q\rangle, \langle b, q, a\rangle, \langle b, q, q\rangle, \langle c, p, a\rangle, \langle c, p, q\rangle, \langle c, q, q\rangle\}$. Найти проекцию ПР_{23} заданного множества M кортежей на оси с номерами 2 и 3.

04. Пусть $M = \{\langle a, p, q\rangle, \langle a, q, q\rangle, \langle b, q, a\rangle, \langle b, q, q\rangle, \langle c, p, a\rangle, \langle c, p, q\rangle, \langle c, q, q\rangle\}$. Найти проекцию ПР_{13} заданного множества M кортежей на оси с номерами 1 и 3.

05. Пусть $M = \{\langle a, p, q\rangle, \langle a, q, q\rangle, \langle b, p, q\rangle, \langle b, q, a\rangle, \langle b, q, q\rangle, \langle c, p, a\rangle, \langle c, q, q\rangle\}$. Найти проекцию ПР_{12} заданного множества M кортежей на оси с номерами 1 и 2.

06. Пусть $M = \{\langle a, p, q\rangle, \langle b, p, q\rangle, \langle b, q, a\rangle, \langle b, q, q\rangle, \langle c, p, a\rangle, \langle c, p, q\rangle, \langle c, q, q\rangle\}$. Найти проекцию ПР_{23} заданного множества M кортежей на оси с номерами 2 и 3.

07. Пусть $M = \{\langle a, p, q\rangle, \langle b, p, q\rangle, \langle b, q, a\rangle, \langle b, q, q\rangle, \langle c, p, a\rangle, \langle c, p, q\rangle, \langle c, q, q\rangle\}$. Найти проекцию ПР_{12} заданного множества M кортежей на оси с номерами 1 и 2.

08. Пусть $M = \{\langle a, p, q\rangle, \langle a, q, q\rangle, \langle b, b, q\rangle, \langle b, q, a\rangle, \langle b, q, q\rangle, \langle c, p, a\rangle, \langle q, p, q\rangle, \langle c, q, q\rangle\}$. Найти проекцию ПР_{23} заданного множества M кортежей на оси с номерами 2 и 3.

09. Пусть $M = \{\langle a, p, p\rangle, \langle a, q, q\rangle, \langle b, q, a\rangle, \langle b, q, q\rangle, \langle c, p, a\rangle, \langle c, a, q\rangle, \langle c, q, q\rangle\}$. Найти проекцию ПР_{12} заданного множества M кортежей на оси с номерами 1 и 2.

10. Пусть $M = \{\langle a, p, q\rangle, \langle a, q, q\rangle, \langle b, p, q\rangle, \langle q, q, a\rangle, \langle b, q, q\rangle, \langle c, p, a\rangle, \langle c, c, q\rangle\}$. Найти проекцию ПР_{13} заданного множества M кортежей на оси с номерами 1 и 3 ■

Задание 3. Найти области определения и значения следующих графиков.

01. $y = \sin x$.
02. $\{\langle p, q\rangle, \langle q, q\rangle, \langle q, a\rangle, \langle p, a\rangle\}$.
03. $y = \arcsin x$.
04. $\{\langle a, c\rangle, \langle b, p\rangle, \langle q, f\rangle, \langle \{a\}, \langle q\rangle\rangle\}$.
05. $y = \mathrm{tg}\, x$.
06. $\{\langle b, x\rangle, \langle a, n\rangle, \langle x, b\rangle, \langle d, a\rangle\}$.
07. $y = \mathrm{arcctg}\, x$.
08. $\{\langle x, x\rangle, \langle l, a\rangle, \langle x, b\rangle\}$.
09. $y = \ln(1-x^2)$.
10. $\{\langle b, c\rangle,\langle b, b\rangle, \langle d, c\rangle\}$■

Задание 4. Найти инверсии следующих графиков.

01. $\{\langle a, b\rangle, \langle n, c\rangle, \langle b, q\rangle\}$.
02. $\{\langle a, d\rangle, \langle b, c\rangle, \langle b, b\rangle\}$.
03. $\{\langle b, b\rangle, \langle l, n\rangle, \langle n, b\rangle\}$.
04. $\{\langle x, z\rangle, \langle a, l\rangle, \langle x, y\rangle, \langle z, x\rangle\}$.
05. $\{\langle b, n\rangle, \langle r, p\rangle, \langle m, b\rangle, \langle p, b\rangle\}$ ■

Задание 5. Найти композицию пар в указанном и обратном порядке. См. пример 11 для образца:

01. $\langle n, c\rangle\circ\langle c, c\rangle$.
02. $\langle a, d\rangle\circ\langle b, b\rangle$.
03. $\langle a, b\rangle\circ\langle b, a\rangle$.
04. $\langle b, a\rangle\circ\langle a, b\rangle$.
05. $\langle l, n\rangle\circ\langle n, b\rangle$.
06. $\langle n, b\rangle\circ\langle l, n\rangle$.
07. $\langle b, x\rangle\circ\langle x, f\rangle$.
08. $\langle b, x\rangle\circ\langle x, \langle f\rangle\rangle$.
09. $\langle \{b\}, x\rangle\circ\langle x, f\rangle$.
10. $\langle b, x\rangle\circ\langle y, f\rangle$■

Задание 6. Найти композицию пар в указанном и обратном порядке. См. пример 12 для образца:

01. $P = \{\langle a, b\rangle, \langle n, c\rangle\}$, $Q= \{\langle b, n\rangle, \langle c, c\rangle\}$.
02. $P = \{\langle a, d\rangle, \langle b, c\rangle\}$,$Q = \{\langle b, b\rangle, \langle d, c\rangle\}$.
03. $P = \{\langle a, x\rangle, \langle x, x\rangle\}$,$Q = \{\langle x, b\rangle, \langle b, a\rangle\}$.
04. $P = \{\langle y, d\rangle, \langle y, c\rangle\}$,$Q = \{\langle c, b\rangle, \langle z, y\rangle\}$.
05. $P = \{\langle a, b\rangle, \langle a, c\rangle\}$, $Q = \{\langle b, b\rangle, \langle d, c\rangle\}$.
06. $P = \{\langle b, n\rangle, \langle l, n\rangle\}$, $Q = \{\langle n, b\rangle, \langle d, c\rangle\}$.
07. $P = \{\langle n, b\rangle, \langle c, n\rangle\}$, $Q = \{\langle n, b\rangle, \langle b, c\rangle\}$.
08. $P = \{\langle a, b\rangle, \langle a, c\rangle\}$, $Q = \{\langle b, b\rangle, \langle d, a\rangle\}$.
09. $P = \{\langle a, b\rangle, \langle n, c\rangle\}$,$Q = \{\langle b, n\rangle, \langle c, c\rangle\}$.
10. $P = \{\langle b, b\rangle, \langle l, n\rangle\}$, $Q = \{\langle n, b\rangle, \langle d, l\rangle\}$.
11. $P = \{\langle a, a\rangle, \langle a, c\rangle\}$, $Q = \{\langle b, a\rangle, \langle c, a\rangle\}$.
12. $P = \{\langle b, x\rangle, \langle a, n\rangle\}$, $Q = \{\langle x, b\rangle, \langle d, a\rangle\}$.
13. $P = \{\langle x, x\rangle, \langle l, a\rangle\}$, $Q = \{\langle x, f\rangle, \langle y, x\rangle\}$.
14. $P = \{\langle b, b\rangle, \langle a, l\rangle\}$, $Q = \{\langle x, b\rangle, \langle l, a\rangle\}$.
15. $P = \{\langle r, n\rangle, \langle r, r\rangle\}$, $Q = \{\langle m, b\rangle, \langle d, r\rangle\}$.
16. $P = \{\langle b, x\rangle, \langle a, n\rangle\}$, $Q = \{\langle b, b\rangle, \langle p, q\rangle\}$.
17. $P = \{\langle x, x\rangle, \langle a, l\rangle\}$, $Q = \{\langle x, y\rangle, \langle y, x\rangle\}$.
18. $P = \{\langle b, b\rangle, \langle a, n\rangle\}$, $Q = \{\langle x, y\rangle, \langle b, a\rangle\}$.
19. $P = \{\langle b, n\rangle, \langle l, n\rangle\}$, $Q = \{\langle x, b\rangle, \langle n, a\rangle\}$.
20. $P = \{\langle b, x\rangle, \langle a, n\rangle\}$, $Q = \{\langle n, b\rangle, \langle d, c\rangle\}$.
21. $P = \{\langle x, x\rangle, \langle l, a\rangle\}$, $Q =\{\langle x, b\rangle, \langle l, b\rangle\}$.
22. $P = \{\langle f, b\rangle, \langle a, l\rangle\}$, $Q = \{\langle x, f\rangle, \langle y, x\rangle\}$.
23. $P = \{\langle b, n\rangle, \langle r, p\rangle\}$, $Q = \{\langle m, b\rangle, \langle p, b\rangle\}$.
24. $P = \{\langle b, x\rangle, \langle q, n\rangle\}$, $Q = \{\langle d, r\rangle, \langle p, q\rangle\}$.
25. $P = \{\langle x, z\rangle, \langle a, l\rangle\}$, $Q = \{\langle x, y\rangle, \langle z, x\rangle\}$.
26. $P = \{\langle b, b\rangle, \langle a, n\rangle\}$, $Q = \{\langle x, b\rangle, \langle a, a\rangle\}$.■

Задание 7. Для всех графиков из заданий 3, 4 и 6 проверить наличие (или отсутствие) свойств функциональности и инъективности (см. примеры 14 – 16) ■

Задание 8. Для всех графиков из задания 6 найти проекции $\text{ПР}_1 G$, $\text{ПР}_2 G$. См. примеры 6 – 8 для образца ■

7. Предметный указатель

Глава 4. Высказывательные формы и кванторы

1. Выражения, переменные и формы

В настоящей главе речь идёт о некоторых общих свойствах математических текстов. В них встречаются различные выражения, содержащие те или иные буквы (например, латинского или греческого алфавита), символы операций (например, рассмотренные в главе 2 знаки теоретико-множественных операций), знаки для интегралов, частных производных, и т.д. Все символы, входящие в выражения, будем – в рамках данного материалы – называть для единообразия буквами (именно, буквами математического алфавита). Будем считать, что во все упоминаемые далее выражения входит конечное число букв. Заметим, что здесь нет требования конечности самого никак формально не определяемого «математического алфавита».

Пример 1. Рассмотрим следующие выражения:

1) $x^2 + y^2 = 1$, 2) $\frac{ax+b}{cx+d}$, 3) $\lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n$, 4) $\int_0^1 x^2dx$, 4) $\neg P\wedge\neg Q\equiv\neg(P\vee Q)$, 5) $x> 3$.

В 1-ое из них входит 6 букв: x, y, 1, 2, +, = (буква 2 входит дважды); во 2-ое – входит 10 букв: l, i, m, n, →, ∞, (,), 1, +; в 3-ье – 6 букв: $\int$, x, d, 0, 1, 2; в 4-ое – 7 букв: P, Q, ¬, ∨, ∧, (,); наконец, в 5-ое выражение входит 3 буквы: x, >, 3 ■

Не пытаясь дать формального определения выражения, заметим лишь, что ***выражение*** − это написанная в каком-нибудь разумном, понятном порядке совокупность букв математического алфавита. Мы не будем никак уточнять смысл этого термина. Это потребует от нас, разумеется, некоторой осторожности в обращении с ним. Мы не сможем, в частности, формулировать (и, конечно, доказывать) никаких теорем о выражениях «вообще».

Далее (как и до сих пор, кроме примера 1) почти до конца данного раздела материал представляет собой почти точную цитату из уже упомянутой в самом начале монографии Ю.А. Шихановича. Поскольку в текст внесены небольшие изменения, кавычки опущены.

Перейдем к центральному понятию этого раздела − понятию ***переменной***. Прежде всего, ещё до объяснения этого термина, заметим, что слово «переменная» у нас будет самостоятельным словом, существительным, а не прилагательным, определением к какому-нибудь другому существительному. Никаких терминов вроде «переменная величина» у нас не будет. Что же такое переменная? Поскольку ответ на этот вопрос будет иметь несколько непривычный, не стандартный характер (он не будет, в частности, иметь вид обычного математического определения), мы будем подходить к нему постепенно. Прежде всего, переменная − это просто буква. Некоторые буквы в выражении называются переменными. Какие же буквы в выражении называются переменными? Те, вместо которых можно что-нибудь (например, числа, многочлены и т. д.) подставлять. Следующий вопрос: что значит «можно»? Предварительный ответ на этот вопрос такой: «можно» − это значит, что после подстановки выражение должно иметь смысл. Не пытаясь определить «смысл» в общем виде, ограничимся простым примером: выражение $1/x$ имеет смысл при x= 3 и не имеет смысла при x= 0. Далее, рассмотрим, например, выражение: $a + x$. Это выражение представляет собой слово из трех букв. Какие же буквы в этом выражении являются переменными? По-видимому, прежде всего можно назвать букву x. Ведь общеизвестно, что x − это переменная! Если вместо x подставлять числа, то выражение, полученное после подстановки, будет иметь смысл. Но если подставлять числа вместо буквы a, то выражение, по-видимому, тоже будет сохранять смысл? Значит, буква a тоже является переменной? А если, наконец, вместо буквы + подставить какой-нибудь из знаков −, ×, :, то выражение, очевидно, тоже будет осмысленным. Значит, + тоже переменная? Какие же все-таки буквы в выражении $a + x$ являются переменными?

После всего этого предварительного разговора перейдем к окончательному ответу на вопрос: что такое переменная? Впрочем, он тоже не уложится в одну фразу. Переменными в некотором данном выражении мы будем называть те буквы, которые... специальным указанием (высказанным в момент задания выражения) будут объявлены таковыми. Таким образом, если это специально не указано, никакого ответа на вопрос «какие буквы являются переменными в выражении $a + x$?» дать нельзя. Может показаться, что при таком понимании термина «переменная» («собственно, ведь нет никакого понимания: какую букву хотим, такую назовём переменной») никакой пользы от этого термина не будет. Это не так. Переменными мы будем обычно называть, объявлять в некотором рассматриваемом выражении такие буквы, вместо которых можно, вместо которых мы собираемся подставлять те или иные числа, выражения и т.п. Объявляя букву переменной, мы тем самым привлекаем к ней внимание, предупреждаем, как мы собираемся использовать эту букву при дальнейшем обращении с рассматриваемым выражением, разрешаем применять к этой букве (и ко всему выражению, содержащему эту букву) весь комплекс терминов и обозначений, который мы для понятия «переменная» введём. Объявляя букву переменной, полагается одновременно задавать ***область значений***(***область определения***) этой переменной. Опять-таки, содержательно под областью значений переменной понимается совокупность, из которой мы собираемся черпать ***значения переменной*** для подстановки их на место переменной. Областью значений данной переменной обычно объявляется такая совокупность, объекты которой естественно подставлять в рассматриваемое выражение вместо этой переменной. Разумеется, слово «естественно» употреблено в совершенно не формальном, интуитивном смысле. Тем не менее, понятно, что символы арифметических действий подставлять в выражение $a + x$ вместо буквы + естественно, а тригонометрические функции неестественно. Однако формально областью значений данной переменной может быть объявлено любое множество. Подчеркнём, что переменная считается полностью заданной (так сказать, конституированной в качестве переменной) лишь тогда, когда ей приписана какая-нибудь область значений. Переменную, в область значений которой входят только числа, мы будем называть ***числовой переменной***. При написании выражений в качестве букв, которые мы будем намереваться использовать как числовые переменные, мы будем обычно использовать наиболее привычные для этой цели последние буквы латинского алфавита. Переменная, значениями которой являются истинностные значения «истина» и «ложь», называется ***высказывательной***.

Введём теперь центральные понятия данного раздела. Выражение, содержащее переменные, мы будем называть ***формой***; выражение, не содержащее переменных − ***константой***. Из вышесказанного вытекает, что одно и то же выражение в одном случае, при одном своём употреблении может быть формой, в другом − константой. Форма называется***s-местной***, если она содержит s переменных. Формы могут быть одноместными, двухместными, трехместными и т.д. Если в выражении $(ax + by)x$ объявить переменными буквы x и y, такая форма будет, разумеется, двухместной, а не трехместной. ***Допустимыми значениями*** данной переменной относительно данной одноместной формы называются те её значения (из области значений переменной), подстановка которых вместо переменной превращает форму в осмысленное выражение. Одноместная форма называется ***всюду определённой***, если любое значение её переменной является допустимым. Одноместная форма называется ***нигде не определённой***, если никакое значение её переменной не является допустимым.

Прежде, чем приводить примеры форм, обратим внимание на следующее принципиальное различие между рассмотренными в первых трёх главах понятиями высказывания, множества и кортежа, и введёнными здесь понятиями переменной и формы. Те понятия были содержательными, т.е. не определялись формально через другие введённые ранее понятия. Именно в этом смысле указанные понятия можно считать базовыми или начальными математическими понятиями. В то же время понятие переменной определяется через введённое ранее понятие множества (буква, которой сопоставлено некоторое множество, называемое областью значений данной переменной). Формой называется выражение, содержащее переменные (которые уже формально определены). Может показаться, что отсутствие точного определения выражения делает эти понятия «не совсем» формальными. Но это не так. Важно лишь то, что выражение содержит конечное число различных букв (символов), т.е. можно формально определить множество его букв и, следовательно, формально объявить часть из них переменными. Если же переменные не объявлены, то выражение названо константой. Дальнейшие действия – придание определённого смысла тому или иному выражению (или указанию на его бессмысленность) –

определяются для конкретных выражений конкретным образом и никак сами по себе не влияют на формально определяемые понятия формы и переменной.

Пример 2. Выражение $\sqrt{-1}$ не имеет смысла, если рассматриваются только действительные числа, и имеет вполне конкретное значение (комплексное число i), если рассматриваются комплексные числа. Таким образом, если имеющиеся в данном выражении три буквы не объявлять переменными, то это выражение – по определению – будет константой, значение которой может быть разным, в зависимости от контекста, в котором рассматривается данное выражение ■

Пример 3. Пусть в выражении $1/x$ переменной будет буква x с областью значений $\{\sqrt{},+, 3, 5, 0\}$. Допустимыми значениями переменной x в этом случае являются 3 и 5, поскольку выражения $1/\sqrt{}$, $1/+$, $1/0$ не имеют смысла. Форма не является ни всюду определенной, ни нигде не определенной, так как в множестве значений переменной есть как элементы, для которых форма не определена, так и элементы, для которых форма определена. Переменная x не является числовой, потому что принимает значения $\sqrt{}$, $+$, не являющиеся числовыми ■

s-местная форма называется ***всюду определенной***, если при любых значениях своих переменных (из соответствующих областей значения) она имеет смысл. s-местная форма называется ***нигде не определенной***, если при любом наборе значений своих переменных она не имеет смысла.

Пример 4. Рассмотрим двухместную форму $\sqrt{1-x^2-y^2}$ с числовыми переменными x и y, для каждой из которых областью значений является отрезок [0, 1]. Легко видеть, что допустимыми являются пары чисел $\langle x, y\rangle$, такие, что $x^2 + y^2 \leq 1$. Ясно, что данная форма не является ни всюду определенной, ни нигде не определенной ■

Пусть $\mathfrak{A}$ – форма, x, y, z – переменные. Запись $\mathfrak{A}(x, y, z)$ означает, что в форме нет никаких переменных, отличных от x, y, z. При этом не требуется, чтобы каждая из переменных x, y, z действительно присутствовала в $\mathfrak{A}$ (хотя бы одна из этих переменных должна, конечно, присутствовать, так как $\mathfrak{A}$ – форма). Таким образом, двухместную форму $x(a + y)$ с переменными x и y можно обозначать как просто $\mathfrak{A}$, так и $\mathfrak{A}(x, y)$, $\mathfrak{A}(x, y, z)$, $\mathfrak{A}(x, y, z, u)$, но не $\mathfrak{A}(x)$. Разумеется, если сказано, что $\mathfrak{A}(x, y, z)$ – трехместная форма, то каждая из переменных x, y, z действительно присутствует в $\mathfrak{A}$. Иногда вместо того, чтобы говорить, что $\mathfrak{A}$ – форма с переменными x, y, z, мы будем говорить, что $\mathfrak{A}$ зависит от x, y, z. Соответственно, «$\mathfrak{A}$ не зависит от x» означает, что буква x не является переменной в форме $\mathfrak{A}$.

Пусть $\mathfrak{A}(x, y, z)$ – форма, объекты a, b и c принадлежат областям значений, соответственно, переменных x, y и z. Результат подстановки в форму $\mathfrak{A}$ объектов a, b и c вместо, соответственно, переменных x, y и z мы будем обозначать через $\mathfrak{A}(a, b, c)$. Разумеется, обозначение $\mathfrak{A}(a, b, c)$ только тогда имеет однозначный смысл, когда сказано явно или ясно из контекста, вместо какой из переменных x, y, z какой из объектов a, b, c подставляется.

Пусть $\mathfrak{A}(x, y, z)$ – форма, объекты a, b, c принадлежат областям значений, соответственно, переменных x, y, z. Выражение $\mathfrak{A}(a, b, c)$ либо осмысленно, либо нет. Если $\mathfrak{A}(a, b, c)$ осмысленно, это мы будем обозначать это так: $!\mathfrak{A}(a, b, c)$ и говорить в этом случае либо «$\mathfrak{A}(a, b, c)$ определено», либо «форма $\mathfrak{A}(x, y, z)$ определена при $x=a$, $y=b$, $z=$ c».

Пусть $\mathfrak{A}$ и $\mathfrak{B}$ – две формы (быть может, от одних и тех же, быть может, от различных – полностью или частично – переменных). Если какая-то буква входит в обе рассматриваемые формы, то мы будем считать, что она обозначает в них одно и то же. В частности, если какая-то буква входит в обе формы и является переменной, то она имеет одинаковую (для обеих форм) область значений. При этом предположении назовем формы $\mathfrak{A}$ и $\mathfrak{B}$ ***равносильными***, если при любом наборе значений всех переменных, входящих в обе эти формы, либо они обе не определены, либо обе определены и обозначают одно и то же (имеют одинаковое значение). Обозначать ***равносильность*** форм $\mathfrak{A}$ и $\mathfrak{B}$ мы будем символом $\simeq$: $\mathfrak{A} \simeq \mathfrak{B}$. Обозначать ***неравносильность*** мы будем символом $\not\simeq$: $\mathfrak{A} \not\simeq \mathfrak{B}$. При тех же предположениях форму $\mathfrak{A}$ назовём ***равносильной константе*** A, если при любом наборе значений переменных формы $\mathfrak{A}$ она на этом наборе означает то же, что и константа A. Короче: $\mathfrak{A} \simeq A$. Назовем, наконец, константы A и B ***равносильными***, если они обозначают одно и то же: $A \simeq B$.

Заметим, что для любых форм $\mathfrak{A}$, $\mathfrak{B}$ и $\mathfrak{C}$ верно следующее

Утверждение 1. а) $\mathfrak{A} \simeq \mathfrak{A}$; б) если $\mathfrak{B} \simeq \mathfrak{A}$, то $\mathfrak{A} \simeq \mathfrak{B}$; в) если $\mathfrak{A} \simeq \mathfrak{B}$ и $\mathfrak{B} \simeq \mathfrak{C}$, то $\mathfrak{A} \simeq \mathfrak{C}$■

Пример 5. Формы $(x + 1)^2$ и $x^2 + 2x + 1$ равносильны. Конечно, формально требуется в каждой из них определить переменные и их области значений. Переменной в обеих формах считаем букву x; областью значений является множество всех действительных чисел, т.е. переменные являются числовыми ■

Далее областью значений числовой переменной <u>будет считаться</u> (если противное не оговорено) множество всех действительных чисел (числовая прямая)

Пример 6. Формы $\neg P \wedge \neg Q$ и $\neg(P \vee Q)$ равносильны. Формально в каждой из них переменными объявлены буквы P и Q; множеством значений во всех случаях является множество истинностных значений $\{\mathbf{T}, \mathbf{F}\}$. Данная равносильность обычно называется законом де Моргана ■

Пример 7. Форма $\cos^2 x + \sin^2 x$с одной переменной x равносильна константе 1 (с учётом соглашения, следующего сразу после примера 5) ■

Пример 8. Форма $x^2 \geq 0$ с одной числовой переменной x с произвольной числовой областью значений при любом xявляется истинным высказыванием. Поэтому данная форма равносильна константе **T**. Аналогично, форма $x^2 < 0$ равносильна константе **F**■

Пример 9. <u>Две</u> формы: $\cos(x + y)$ и $\cos x \cos y - \sin x \sin y$ равносильны как две формы от числовых переменных x иy■

Пример 10. <u>Одна</u> форма: $\cos(x + y) = \cos x \cos y - \sin x \sin y$ от числовых переменных x иy равносильна логической константе **T**■

Пример 11. Определим форму следующим образом. Положим

$$F(p) = \begin{cases} 0, & \text{если } p = \mathrm{F} \\ 1, & \text{если } p = \mathrm{T} \end{cases}$$

Объявим переменной букву p с областью значений $\{\mathbf{T}, \mathbf{F}\}$. При этом буквы 0 и 1 предполагаются числовыми константами ■

Важнейшими классами форм являются класс числовых форм и класс высказывательных форм. Форма называется ***числовой***, если при любом наборе значений своих переменных, на котором она определена, она обозначает число (является числом). Форма называется ***высказывательной***, если при любом наборе значений своих переменных, на котором она определена, она обозначает высказывание (является высказыванием).

Не следует путать понятия числовой переменной и числовой формы, высказывательной переменной и высказывательной формы. В примерах 8 и 10 переменные являются числовыми, форма же является высказывательной. В примере 11, наоборот переменная является высказывательной, а форма является числовой. В примере 6 как переменные, так и форма являются высказывательными. Наконец, в примерах 7 и 9 как переменные, так и форма являются числовыми.

Для обозначения равносильности числовых форм, наряду с общим знаком $\simeq$ используется также обычный знак равенства =. Для обозначения равносильности высказывательных форм используется также знак $\equiv$(см. далее формулы (5-2) – (5-7), где этот же знак используется для обозначения равносильности булевых функций – частного случая высказывательных форм).

1.1. Формы и функции. В «обычной» математике понятие переменной было тесно связано с понятием функции. Символом $f(x)$ обозначалась функция f от одной переменной x. Столь же общепринятым является обозначение $f(x_1, \ldots, x_n)$ для функции от нескольких переменных.

В аннотации к части 1 подчёркивалось, что введённые формальные понятия – кортежа, графика и др. – являются обобщениями хорошо известных понятий – вектора, графика и др. из обычной школьной математики. В этом же ряду понятие формы является более точным описанием расплывчатого понятия «выражение». Естественно, что традиционное представление о функции, её значениях и переменных являются частными случаями введённых понятий. Само понятие функции введено в разделе 3.5 как частный случай соответствия. Термины «выражение», «форма» и «переменная» при этом не использовались. Введённое в настоящем разделе понятие переменной относится к формам, а не к функциям.

В настоящем разделе мы установим точное соответствие между некоторыми видами форм и ранее формально определёнными функциями. Именно такое соответствие и позволит «совместить» эти понятия и прийти к формальному определению функции с переменными, являющемуся обобщением соответствующего интуитивного понятия.

Пусть $\mathfrak{A}$– n-местная форма с переменными $x_1, \ldots, x_n$. Напомним, что вместе с переменными задаются множества $A_1, \ldots, A_n$, которым принадлежат все значения соответствующих переменных. Допустимыми были названы такие наборы значений переменных $x_1, \ldots, x_n$, при которых данная форма имеет смысл. Не будем пытаться как-то определить «смысл» в общем виде. Заметим только, что в примерах 1 – 11 под «смыслом» формы всегда можно было понимать либо число, либо одно из двух значений истинности, т.е. все рассмотренные в этих примерах формы были числовыми или высказывательными (см. определение сразу после примера 11). Сделаем ещё один шаг в сторону обобщения. Пусть смыслом формы $\mathfrak{A}(x_1, \ldots, x_n)$ при любых допустимых наборах её переменных являются элементы из некоторого множества B, однозначно определяемые по наборам значений $x_1, \ldots, x_n$. Такие формы будем называть ***функциональными***, а множество B будем называть ***множеством прибытия*** этой функциональной формы.

Смысл этих названий оправдывается следующими рассуждениями. Пусть $\mathfrak{A}(x_1, \ldots, x_n)$ – n-местная функциональная форма с переменными $x_1, \ldots, x_n$ из множеств $A_1, \ldots, A_n$ и множеством прибытия B. Очевидным образом такая форма определяет функциональное соответствие в смысле определения из раздела 3-5. Действительно, положим $A = \prod_{i=1}^{n} A_i$, определим $G \subseteq A \times B$ как множество пар вида $\langle\langle x_1, x_2, ..., x_n\rangle, b\rangle$, где при значениях переменных $x_1, x_2, ..., x_n$форма $\mathfrak{A}(x_1, \ldots, x_n)$ имеет смысл b. Тем самым определено соответствие $\Gamma = \langle G, A, B\rangle$, функциональность которого очевидна, поскольку «смысл» в виде элемента b определяется по условию однозначно. Используя более привычную запись $\Gamma(x_1, x_2, ..., x_n) = b$ вместо записи $\langle\langle x_1, x_2, ..., x_n\rangle, b\rangle \in G$, приходим к следующему выводу. Каждая n-местная функциональная форма$\mathfrak{A}(x_1, \ldots, x_n)$ естественно определяет функцию

$y = f(x_1, \ldots, x_n)$, (1)

переменные которой принадлежат соответственно множествам $A_1, \ldots, A_n$, а значение y– множеству B. Важно то, что для таким образом определённых функций можно одновременно пользоваться как комплексом всех понятий, относящихся к соответствиям, так и комплексом всех понятий, относящихся к формам.

Если для любых двух значений a_i'и a_i''из множества A_i и для любых значений остальных переменных $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$ выполняется равенство

$$f(x_1, \ldots, a_i', \ldots, x_n) = f(x_1, \ldots, a_i'', \ldots, x_n), \qquad (2)$$

то функция $f(x_1, \ldots, x_n)$ на самом деле не зависит от переменной x_i. Такие переменные называются ***несущественными*** или ***фиктивными***.

Основными рассматриваемыми далее случаями являются:

1) булевы функции от n переменных, в которых $A_i = \{0, 1\}$ ($i = 1, \ldots, n$), область прибытия Bтакже равна $\{0, 1\}$;

2) обычные числовые функции от n переменных, в которых $A_i = R$($i = 1, \ldots, n$), область прибытия B также равна R (через R здесь обозначено множество вещественных чисел).

В дальнейших разделах пособия оказывается достаточным интуитивного понятия функции, представляемой стандартной записью (1). Проведённые рассуждения лишь демонстрируют возможность аккуратного определения понятия функции, основывающегося только на базисных понятиях высказывания, множества и кортежа, без привлечения таких трудно формулируемых и нечётких понятиях, как независимая и зависимая переменная величина, изменение, закон, правило, и т.д.

2. Высказывательные формы

Высказывательные формы были определены в предыдущем разделе как формы, которые обозначают высказывания, т.е. являются высказываниями. Например, выражение

$$A \wedge (x > 3)$$

естественно считать двухместной высказывательной формой с высказывательной переменной A и числовой переменной x. Для случаев (которые нельзя априори исключать), когда на некоторых наборах значений переменных высказывательная форма неопределенна, доопределим её логическим значением «Ложь» (т.е. значением **F** из двух элементного множества {**T**, **F**}). Поэтому далее рассматриваются только всюду определённые высказывательные формы. Введённые в разделе 1-2.1 операции над высказываниями очевидным образом можно рассматривать и как операции над высказывательными формами. Для любых высказывательных форм $\mathfrak{A}$ и $\mathfrak{B}$ и без специальных пояснений ясен смысл форм: $\neg\mathfrak{A}$, $\mathfrak{A} \vee \mathfrak{B}$, $\mathfrak{A} \wedge \mathfrak{B}$, $\mathfrak{A} \rightarrow \mathfrak{B}$, $\mathfrak{A} \rightleftarrows \mathfrak{B}$, $\mathfrak{A} \oplus \mathfrak{B}$.

Пусть $\mathfrak{A}(x_1, \ldots, x_s)$ – s-местная ($s \geq 1$) высказывательная форма, где $x_1, \ldots, x_s$ – переменные формы $\mathfrak{A}$,выписанные в каком-нибудь порядке. Пусть областью определения переменной x_i будет множество $M_i(i = 1, \ldots, s)$. Тогда ***областью истинности*** высказывательной формы $\mathfrak{A}$ называется и как $\overline{\mathfrak{A}}$ или $\overline{\mathfrak{A}(x_1, \ldots, x_s)}$ обозначается множество тех кортежей $\langle a_1, a_2, \ldots, a_s\rangle$, для которых $\mathfrak{A}(a_1, \ldots, a_s)$ истинно.

Понятие области истинности позволяет связать операции над множествами с операциями над высказывательными формами. Так, очевидно, при $s \geq 1$

$$\overline{\mathfrak{A}(x_1, \ldots, x_s) \wedge \mathfrak{B}(x_1, \ldots, x_s)} = \overline{\mathfrak{A}(x_1, \ldots, x_s)} \cap \overline{\mathfrak{B}(x_1, \ldots, x_s)},$$

$$\overline{\mathfrak{A}(x_1, \ldots, x_s) \vee \mathfrak{B}(x_1, \ldots, x_s)} = \overline{\mathfrak{A}(x_1, \ldots, x_s)} \cup \overline{\mathfrak{B}(x_1, \ldots, x_s)}.$$

Если M_i– облаcги значений переменных $x_i(i = 1, \ldots, s)$, то

$$\overline{\neg\mathfrak{A}(x_1, \ldots, x_s)} = \prod_{i=1}^{s}(M_i \setminus \overline{\mathfrak{A}(x_1, \ldots, x_s)}).$$

2.1. Высказывательные формы и разрешающие процедуры. Вернёмся к вопросу о разрешающей процедуре задания множеств, упомянутому в разделе 2-1. Суть дела состоит в описании характеристических свойств его элементов и проверке этих свойств. Такая проверка и определяет «разрешение» для данного элемента принадлежать рассматриваемому множеству. В примерах 3-8, 3-9, 3-15 и 3-16 именно так задавались бесконечные графики. Теперь можно дать более детальное описание этого способа задания множеств. Именно, во многих случаях описываемое множество естественно совпадают с областью истинности некоторой высказывательной формы. В примере 3-8 рассматривалось множество M точек $\langle x, y\rangle$на плоскости, удовлетворяющих условию $x^2+y^2= 1$ (т.е. M – это окружность единичного радиуса с центром в начале координат). Определим высказывательную форму $\mathfrak{A}(x,y)$: $x^2 + y^2 = 1$. Понятно, что рассматриваемое множество M совпадает с областью истинности данной высказывательной формы. В пункте 5) примера 1 высказывательная форма $\mathfrak{A}(x)$: $x> 3$ определяет бесконечное множество чисел, бóльших 3.

Вообще, в тех случаях, когда интересующее нас множество M совпадает с областью истинности некоторой высказывательной формы $\mathfrak{A}$, это множество M именно так и может быть задано. Для этого используется стандартное обозначение

$$M = \{x|\mathfrak{A}(x)\}, \qquad (3)$$

которое читается так: M – это множество элементов x, для которых высказывательная форма $\mathfrak{A}$ истинна. В формуле (3) $x = \langle x_1, x_2, \ldots, x_s\rangle$ – это набор переменных данной высказывательной формы. Более сложные вопросы проверки истинности произвольной высказывательной формы, приводящие, в частности, к формальному определению одного важного класса высказывательных форм – предикатов – здесь не рассматриваются.

3. Кванторы

Рассмогрим две операции, применяемые только к высказывательным формам. Эти операции, гак называемые ***навешивания кванторов***, будучи применёнными к высказывательной форме, приводят либо снова к высказывательной форме, либо к высказыванию.

Начнем с простейшего случая. Пусть $\mathfrak{A}(x)$ – одноместная высказывательная форма; область значений переменной x обозначим буквой M. Обозначим через

$$(\forall x \in M)\ \mathfrak{A}(x), \qquad (4)$$

или, если M ясно из контекста, через

$$(\forall x)\ \mathfrak{A}(x), \qquad (5)$$

следующее высказывание: «для любого значения переменной x высказывание, полученное подстановкой этого значения в форму $\mathfrak{A}(x)$ вместо x, истинно». Знак $\forall$ называется ***квантором общности***, переход от формы $\mathfrak{A}(x)$ к высказываниям (4), (5) – навешиванием на форму $\mathfrak{A}(x)$ квантора ***общности по переменной*** *x*.

Обозначим через

$$(\exists x \in M)\ \mathfrak{A}(x), \qquad (6)$$

или, если M ясно из контекста, через

$(\exists x)\,\mathfrak{A}(x)$, (7)

следующее высказывание: «существует такое значение переменной x, что высказывание, полученное подстановкой этого значения в форму $\mathfrak{A}(x)$ вместо x, истинно». Знак $\exists$ называется ***квантором существования***, переход от формы $\mathfrak{A}(x)$ к высказываниям (6), (7) – навешиванием на форму квантора ***существования по переменной*** x.

Пример 12. Рассмотрим следующую высказывательную форму $\mathfrak{A}(x)$: «Девочку в группе зовут Маша» с одной переменной «девочка». Областью значений переменной является множество всех девочек из данной группы. Используя кванторы, получаем высказывания (не высказывательные формы!):

«Всех девочек в группе зовут Маша»

«Некоторую девочку в группе зовут Маша»

«Ни одну девочку не зовут Маша», и т.п ■

Рассмотрим общий случай. Пусть $\mathfrak{A}(x_1, \ldots, x_s)$ – s-местная высказывательная форма ($s \geq 2$); область значений переменной x_i – множество $M_i(i = 1, \ldots, s)$. В силу введённых выше обозначений для любого $i = 1, \ldots, s$ выражения

$(\forall x_i \in M)\,\mathfrak{A}(x_1, \ldots, x_s)$, (8)

$(\exists x_i \in M)\,\mathfrak{A}(x_1, \ldots, x_s)$ (9)

являются (s–1)-местными высказывательными формами, зависящими от переменных $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_s$. Буква x_i в выражениях (8), (9) называется ***связанной переменной***, а все остальные переменные, на которые не навешаны кванторы, называются в таких случаях ***свободными переменными***. Сразу скажем, что свободные переменные – это просто переменные в смысле определения из раздела 1. А связанные переменные <u>переменными в этом смысле (т.е. буквами, вместо которых можно подставлять элементы из множества значений) вообще не являются</u>. Всё, что можно с ними сделать: заменить букву, написанную сразу после квантора, и все вхождения этой буквы в рассматриваемую форму, на одну и ту же букву, отличную от имён всех остальных переменных. Заметим, что близкий смысл имеет индекс суммирования i в суммах вида $\sum_i^N f(i)$, переменная интегрирования tв интегралах $\int_a^b f(t)dt$, переменная, по которой берётся максимум (или минимум): $\max_x f(x)$, и т.д. Общими во всех этих случаях являются два факта: 1) результат (сумма, интеграл, максимум и пр.) <u>не зависит</u> от этой связанной переменной, и 2) эту букву можно везде заменить на любую другую, не совпадающую с другими в данном выражении. Таким образом, можно сказать, что <u>навешивание квантора по переменной x связывает эту переменную</u>.

По определению квантора общности, высказывание

$(\forall x_i \in M)\,\mathfrak{A}(a_1, \ldots, a_{i-1}, x_i, a_{i+1}, \ldots, a_s)$, (10)

истинно тогда и только тогда, когда для любого значения $a_i \in M_i$ переменной x_i высказывание

$\mathfrak{A}(a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_s)$ (11)

истинно.

По определению квантора существования высказывание

$(\exists x_i \in M)\mathfrak{A}(a_1, \ldots, a_{i-1}, x_i, a_{i+1}, \ldots, a_s)$, (12)

истинно тогда и только тогда, когда существует такое значение $a_i \in M_i$ переменной x_i, что высказывание

$\mathfrak{A}(a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_s)$ (13)

истинно.

С помощью кванторов можно ввести некоторые понятия, относящиеся к произвольным соответствиям (см. раздел 3-5). Пусть $\Gamma = \langle G, X, Y\rangle$ – соответствие, A – множество. ***Образом A относительно Г*** называется и через $\Gamma(A)$ обозначается множество всех тех элементов из Y, каждый из которых соответствует в соответствии Γ какому-нибудь элементу множества A. Таким образом,

$\Gamma(A) = \{y \in Y \mid (\exists x \in A)\ \langle x, y\rangle \in G\}$. (14a)

В формуле (14a) выражение после знака | имеет такой же смысл, как в формуле (3). Оно является одноместной высказывательной формой, зависящей от переменной y с областью значений Y. В множество $\Gamma(A)$ входят все те элементы Y, для которых высказывательная форма $(\exists x \in A)\langle x, y\rangle \in G$истинна.

Если $\Gamma = \langle G, X, Y\rangle$ – соответствие, A – множество, то полным прообразом ***A* относительно *Г*** называется и через $\Gamma^{-1}(A)$ обозначается множество всех тех элементов из X, каждому из которых соответствует в соответствии Γ элемент множества A. Таким образом,

$$\Gamma^{-1}(A) = \{x\in X \mid (\exists y\in A)\ \langle x, y\rangle\in G\}. \tag{14b}$$

3.1. Отрицание высказываний с кванторами. Как и для любых высказываний, отрицание высказывательных форм, содержащих кванторы, является высказывательной формой или высказыванием с противоположным значением истинности. Если в высказывательной форме имеется ровно одна переменная, то после навешивания на неё кванторов (общности или существования) эта форма становится высказыванием – именно, высказыванием, содержащим квантор. Именно такие высказывания рассматриваются в данном разделе.

Пример 13. Вернёмся к 1-ому из высказываний из примера 12: «Всех девочек в группе зовут Маша». Отрицанием этого высказывания является высказывание «Некоторых девочек в группе не зовут Маша». Как обычно, то же самое утверждение может быть выражено различными способами:

«Не всех девочек в группе зовут Маша»

«Неверно, что всех девочек в группе зовут Маша»

«По крайней мере одну девочку в группе не зовут Маша», и т.д ■

Пример 14. Рассмотрим высказывание «У некоторых кошек есть блохи». Поскольку «некоторые» в данном контексте значит «по меньшей мере, одна», высказывание «У некоторых кошек есть блохи» по смыслу совпадает с высказыванием «По меньшей мере, у одной кошки есть блохи». Отрицанием этого высказывания является «Ни у одной кошки нет блох», которое может быть записано также в виде «У всех кошек нет блох» ■

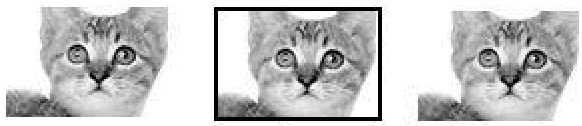
Обобщая рассуждения из примеров 13 и 14, можно прийти к следующей таблице, содержащей схемы для построения отрицаний высказываний с кванторами.

Таблица 1. Отрицания высказываний с кванторами

Высказывание	**Отрицание**
Все делают (имеют и пр.)	Некоторые не делают (Эквивалентно: не все делают)
Некоторые делают (имеют и пр.)	Никто не делает (Эквивалентно: все не делают)

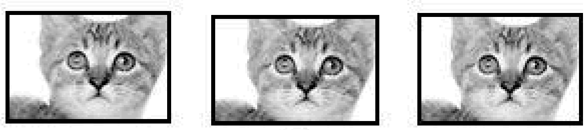
Отрицанием отрицания является само исходное высказывание. Как и все остальные высказывания, они могут быть записаны в разном виде (см. таблицу 1).

Пример 15. Рассмотрим следующие изображения, разделённые на группы A, B, и C.



A

B



C

Отметим одной (или несколькими) буквами группу или группы картинок, удовлетворяющих условию «По меньшей мере хотя бы одна картинка из группы не имеет рамки». Группы A и B удовлетворяют этому условию, а группа C – нет ■

4. Задания

Задание 1. См. пример 3 для образца. Для данной одноместной формы и данной области значений переменной x:

а) найти допустимые значения;

б) установить, является ли форма всюду определённой;

в) установить, является ли форма нигде не определённой;

г) установить, является ли форма числовой.

01. Форма $1/x$, область значений {3, 5, 0}
02. Форма $1/x$, область значений {3, 5}
03. Форма $1/x$, область значений {0}
04. Форма $1/x$, область значений {√ , +}
05. Форма $\frac{x-1}{\sqrt{(x-3)(-4-x)}}$, область значений {−5,–4,–3, −2,–1, 0, 1, 2, 3, 4}
06. Форма $\frac{x-1}{\sqrt{(x-3)(-4-x)}}$, область значений {−5,4}
07. Форма $\frac{x-1}{\sqrt{(x-3)(-4-x)}}$, область значений {3,–4}
08. Форма $\frac{x-1}{\sqrt{(x-3)(-4-x)}}$, область значений {−8, 7, 100}
09. Форма $\frac{x-1}{\sqrt{(x-3)(-4-x)}}$, область значений {3,–4, !}
10. Форма $\frac{x-1}{\sqrt{(x-3)(-4-x)}}$, область значений {3, 1+i}■

Задание 2. Для данных высказывательных форм найти области истинности.

1. $\mathfrak{A}(x,y)$: $x^2 + y^2 \geq 2xy$
2. $\mathfrak{B}(x,y)$: $x + y \geq 2xy$
3. $\mathfrak{C}(x,n)$: $x^n > 1 + n(x - 1)$
4. $\mathfrak{A}(x, y) \to \mathfrak{B}(x, y)$
5. $\mathfrak{D}(x,n)$: $x<n$
6. $\mathfrak{C}(x,n) \wedge \mathfrak{D}(x,n)$ ■

Задание 3. Нарисовать на координатной плоскости область истинности высказывательной формы $\mathfrak{A}(x, y)$ с числовыми переменными x, y, если

1. $\mathfrak{A}(x, y)$: $y = x^2$.
2. $\mathfrak{A}(x, y)$: $2x + 3y - 1 > 0$.
3. $\mathfrak{A}(x, y)$: $\sin(x + y) = 0$.
4. $\mathfrak{A}(x, y)$: $y = x^2 + 1/x$.
5. $\mathfrak{A}(x, y)$: $y<x^2 + 1/x$.
6. $\mathfrak{A}(x, y)$: $y = \frac{1}{1+x^2}$.
7. $\mathfrak{A}(x, y)$: $y = \frac{2x}{1+x^2}$■

Задание 4. Проверить следующие равносильности

1. $(\forall x)[\mathfrak{A}(x) \wedge \mathfrak{B}(x)] \equiv (\forall x)\mathfrak{A}(x) \wedge (\forall x)\mathfrak{B}(x)$
2. $(\forall x)[\mathfrak{A}(x) \vee \mathfrak{B}(x)] \equiv (\forall x)\mathfrak{A}(x) \vee (\forall x)\mathfrak{B}(x)$
3. $(\exists x)[\mathfrak{A}(x) \wedge \mathfrak{B}(x)] \equiv (\exists x)\mathfrak{A}(x) \wedge (\exists x)\mathfrak{B}(x)$
4. $(\exists x)[\mathfrak{A}(x) \vee \mathfrak{B}(x)] \equiv (\exists x)\mathfrak{A}(x) \vee (\exists x)\mathfrak{B}(x)$
5. $(\forall x)[\mathfrak{A}(x) \to \mathfrak{B}(x)] \equiv (\forall x)\mathfrak{A}(x) \to (\forall x)\mathfrak{B}(x)$
6. $(\forall x)[\mathfrak{A}(x) \rightleftarrows \mathfrak{B}(x)] \equiv (\forall x)\mathfrak{A}(x) \rightleftarrows (\forall x)\mathfrak{B}(x)$
7. $(\exists x)[\mathfrak{A}(x) \to \mathfrak{B}(x)] \equiv (\exists x)\mathfrak{A}(x) \to (\exists x)\mathfrak{B}(x)$
8. $(\exists x)[\mathfrak{A}(x) \rightleftarrows \mathfrak{B}(x)] \equiv (\exists x)\mathfrak{A}(x) \rightleftarrows (\exists x)\mathfrak{B}(x)$ ■

Задание 5. Написать отрицания следующих высказываний
1. У каждой собаки есть свой день
2. На равнинах Испании сегодня нет дождей
3. Некоторые книги длиннее, чем эта книга
4. Ни один мастер по ремонту компьютеров не умеет играть в «блэкджек»
5. Некоторым людям всегда везёт
6. Каждый иногда любит кого-то
7. Все любят победителя
8. Все мужчины созданы равными
9. Некоторые ученики из нашего класса ездили на загородную экскурсию ■

Задание 6. Для изображений из примера 15 отметить одной (или несколькими) буквами группу или группы картинок, удовлетворяющих условиям:
1. Ни одна картинка не имеет рамки
2. По меньшей мере одна картинка не имеет рамки
3. Не каждая картинка имеет рамку
4. По меньшей мере одна картинка имеет рамку
5. Нет картинок, которые не имеют рамки
6. Все картинки не имеют рамки
7. Не каждая картинка не имеет рамки ■

5. Предметный указатель

Глава 5. Булева алгебра

1. Понятие булевой функции

Булевы функции названы в честь английского математика XIX века Джорджа Буля, который впервые применил алгебраические методы для решения логических задач. Иногда они называются также ***логическими функциями*** и функциями ***алгебры логики***. Булевы функции находят применение в вычислительной технике, логике, теории управления, принятии решений и многих разделах информатики.

Обозначим через B двухэлементное множество $\{0,1\}$, состоящее из двух символов 0 и 1. Тогда $B^n = B\times B\times\ldots\times B$ (n сомножителей) – это множество всех кортежей длины n, компоненты которых равны 0 или 1. По определению, $B^1 = B = \{0,1\}$. Обратим особое внимание на тот факт, что 0 и 1 – это, вообще говоря, не числа ноль и один, а формальные символы, некоторые операции над которыми действительно напоминают обычные арифметические операции сложения и умножения, однако другие операции над ними совсем не похожи на арифметические.

Определим булевы функции, используя общую конструкцию функции из раздела 4-1.1. Булевой функцией называется функция с областью отправления B^n и областью прибытия B. Число nможет быть произвольным натуральным числом: 1, 2, … . Возвращаясь к привычным обозначениям, можно сказать, что булева функция является функцией от n переменных, принимающих значения из множества $B = \{0,1\}$; значения булевой функции принадлежат тому же самому двухэлементному множеству B. Если не указано противное, булевы функции будем считать определёнными на всех наборах из B^n. Это значит, что область определения булевой функции совпадает с её областью отправления.

Имеется несколько различных способов представления и интерпретации булевых функций. В данной главе рассматривается табличное представление, а также представление с помощью булевых или логических формул.

1.1. Табличное представление. Булевы функции от небольшого числа переменных удобно представлять с помощью таблиц. Таблица для функции $f(x_1, \ldots, x_n)$ имеет $n + 1$ столбец. В первых n столбцах указываются значения переменных $x_1, \ldots, x_n$, а в $(n + 1)$-ом столбце – значение $f(x_1, \ldots, x_n)$ функции f на этом наборе значений.

Таблица .1. Табличное представление функции $f(x_1, \ldots, x_n)$

x_1	……	x_{n-1}	x_n	$f(x_1, \ldots, x_{n-1}, x_n)$
0	……	0	0	$f(0, \ldots, 0, 0)$
0	……	0	1	$f(0, \ldots, 0, 1)$
0	……	1	0	$f(0, \ldots, 1, 0)$
	……			…..
1	……	1	1	$f(1, \ldots, 1, 1)$

Наборы переменных в строках обычно располагаются в ***лексикографическом порядке***:

$(\alpha_1, \ldots, \alpha_n) \prec (\beta_1, \ldots, \beta_n) \leftrightarrow \exists i\in\{1, \ldots, n\}$, такое, что при всех $j<i \alpha_j = \beta_j$и $\alpha_i<\beta_i$. (1)

Если эти наборы рассматривать как записи чисел в двоичной системе счисления, то 1-ая строка является записью числа 0, 2-ая – числа 1, 3-я – числа 2, ... , а последняя – числа 2^n–1. Типичное представление булевой функции от трёх переменных даётся в таблице 2. В этой таблице, как и во всех последующих таблицах, задающих булевы функции, наборы переменных расположены в лексикографическом порядке. Обратим особое внимание на удобство такого расположения. Его не надо запоминать. Просто в последнем (самом правом) столбце значения меняются, начиная с нуля, через одну позицию; в предпоследнем столбце, начиная с двух нулей, через две позиции, и т.д., вплоть до первого столбца, в котором в первой половине позиций стоят нули, а во второй половине – единицы. Другие важные свойства наборов, расположенных именно в таком порядке, будут рассмотрены в разделе 5.

Таблица 2. Функция от трёх переменных

x_1	x_2	x_3	$f(x_1,x_2,x_3)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

При больших n табличное представление становится громоздким. Например, для функции от 10 переменных потребуется таблица с 1024 строками. Но для малых n ($n = 2, 3, 4$) оно достаточно наглядно.

1.2. Булевы функции от одной и двух переменных. Прежде чем двигаться дальше и говорить о других представлениях булевых функций, необходимо детально рассмотреть булевы функции от одной и двух переменных. Функции от одной переменной записаны в таблице 3. Этих функций – четыре:

Таблица 3. Булевы функции от 1-ой переменной

x	φ_0	φ_1	φ_2	φ_3
0	0	0	1	1
1	0	1	0	1

Левый столбец содержит два значения переменной x. Следующие 4 столбца определяют все 4 функции от одной переменной. Функции φ_0 и φ_3 – константы 0 и 1; их значения не зависят от значения переменной x. В конце раздела 4-1.1 такие переменные названы ***несущественными***. Функция φ_1 «повторяет» x: $\varphi_1(x) = x$. Функция $\varphi_2(x)$ называется ***отрицанием*** x. Для отрицания будет использоваться тот же знак $\neg$, который использовался в таблицах 1-1 и 1-2 для отрицания высказываний. Иногда для отрицания используются также обозначения $\overline{\boldsymbol{x}}$, x^0, $\sim x$.

Отличие от введённой в разделе 1-2 операции отрицания состоит в том, что там отрицание относилось только к высказываниям и их истинностным значениям. Здесь отрицание относится к формальным символам 0 и 1, для которых интерпретация их как истинностных значений F и T является хотя и часто используемой, но далеко не единственной. То же самое относится и к булевым функциям от любого числа переменных.

Рассмотрим теперь логические функции от двух переменных. Их число равно 16. Все они записаны в таблице 4.

Таблица 4. Булевы функции от 2-ух переменных

x_1	x_2	ψ_0	ψ_1	ψ_2	ψ_3	ψ_4	ψ_5	ψ_6	ψ_7	ψ_8	ψ_9	ψ_{10}	ψ_{11}	ψ_{12}	ψ_{13}	ψ_{14}	ψ_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Два левых столбца содержат все четыре набора переменных: (0, 0), (0, 1), (1, 0), (1, 1), расположенные в лексикографическом порядке. Следующие 16 столбцов определяют все 16 различных функций от двух переменных: в столбце под символом ψ_i находятся 4 значения функции ψ_i на этих 4-ёх наборах. Эти столбцы также расположены в лексикографическом порядке, но не сверху-вниз, как в таблицах 1 и 2, а слева направо.

Многие из этих 16-и функций часто используются в качестве «элементарных» и имеют собственные названия и обозначения. Часть этих названий уже использовалась в главе 1 для описания операций над высказываниями и их истинностными значениями. При замене F и T на 0 и 1 все 6 рассмотренных в разделе 1-2.1 операций перейдут в 6 функций, представленных в таблице 4. Естественно, что используются те же названия.

1. $\psi_0(x_1, x_2) = 0$ – константа 0;
2. $\psi_{15}(x_1, x_2) = 1$– константа 1;
3. $\psi_3(x_1, x_2) = x_1$– функция, равная 1-му аргументу;
4. $\psi_{12}(x_1, x_2) = \neg x_1$– отрицание x_1;
5. $\psi_5(x_1, x_2) = x_2$– функция, равная 2-му аргументу;
6. $\psi_{10}(x_1, x_2) = \neg x_2$– отрицание x_2;
7. $\psi_1(x_1, x_2) = x_1 \wedge x_2$ – конъюнкция x_1 и x_2 (обозначается также через $x_1 \& x_2$ и x_1x_2);
8. $\psi_7(x_1, x_2) = x_1 \vee x_2$ – дизъюнкция x_1 и x_2;
9. $\psi_{13}(x_1, x_2) = x_1 \rightarrow x_2$ – импликация x_1 и x_2 (читается «x_1 влечёт x_2» или «из x_1 следует x_2»;
10. $\psi_6(x_1, x_2) = x_1 \oplus x_2$ – сложение по модулю 2 или разделительная дизъюнкция (используется также обозначение $x_1 + x_2$);
11. $\psi_9(x_1, x_2) = x_1 \rightleftarrows x_2$ – эквивалентность x_1 и x_2 (читается как «x_1 эквивалентно x_2» или «$x_2 = 1$ тогда и только тогда, когда $x_1 = 1$»);
12. $\psi_{14}(x_1, x_2) = x_1 \mid x_2$ – штрих Шеффера (отрицание конъюнкции – «антиконъюнкция»);
13. $\psi_8(x_1, x_2) = x_1 \downarrow x_2$ – стрелка Пирса (отрицание дизъюнкции – «антидизъюнкция»).

Заметим, что для 2-местных функций из этого списка в правых частях равенств использованы не только собственные имена, но и ***инфиксная запись***, в которой значок операции помещается между 1-ым и 2-ым аргументами (запись вида $\psi(x_1, x_2)$ называется ***префиксной***). Функции $\psi_2(x_1, x_2)$, $\psi_4(x_1, x_2)$, $\psi_{11}(x_1, x_2)$ не имеют специальных названий.

Пример 1. Таблица 4 задаёт все функции от двух переменных. Приведём в качестве примера использования этой таблицы несколько тождеств для булевых функций, в верности которых легко убедиться проверкой значений функций на всех четырёх наборах – 00, 01, 10 и 11 – значений двух переменных. Под тождествами здесь и далее понимаются равенства функций при всех наборах значений их переменных.

$$a \rightarrow b \equiv \neg a \vee b \qquad (2)$$
$$a \oplus b \equiv (\neg a \wedge b) \vee (a \wedge \neg b) \qquad (3)$$
$$a \oplus 1 \equiv \neg a \qquad (4)$$
$$a \rightleftarrows b \equiv (a \wedge b) \vee (\neg a \wedge \neg b) \qquad (5)$$
$$a \mid b \equiv \neg(a \wedge b) \qquad (6)$$
$$a \downarrow b \equiv \neg(a \vee b). \qquad (7)$$

Использование других имён переменных вместо x_1, x_2 и x подчёркивает независимость свойств функций и связей между ними от этих имён, что, впрочем, совершенно естественно для всех разделов математики ■

Отметим, что функции $\psi_3(x_1, x_2)$, $\psi_{12}(x_1, x_2)$, $\psi_5(x_1, x_2)$, $\psi_{10}(x_1, x_2)$ зависят только от одной переменной, так что в $\psi_3(x_1, x_2)$ и $\psi_{12}(x_1, x_2)$ несущественной является переменная x_2, в $\psi_5(x_1, x_2)$ и $\psi_1(x_1, x_2)$ несущественной является переменная x_1. Функции $\psi_0(x_1, x_2)$ и $\psi_{15}(x_1, x_2)$ являются константами 0 и 1, т.е. в них обе переменные несущественны. Конъюнкция называется также ***логическим умножением***, а дизъюнкция – ***логическим сложением***.

Для удобства дальнейшего использования перепишем таблицу 4 ещё раз, заменив в ней нейтральные обозначения ψ_i на принятые значки для соответствующих функций:

Таблица 5. Булевы функции от 2-ух переменных со значками функций

x_1	x_2	0	$x_1 \wedge x_2$	ψ_2	x_1	ψ_4	x_2	$x_1 \oplus x_2$	$x_1 \vee x_2$	$x_1 \downarrow x_2$	$x_1 \rightleftarrows x_2$	$\neg x_2$	ψ_{11}	$\neg x_1$	$x_1 \rightarrow x_2$	$x_1 \mid x_2$	1
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Понятие несущественных переменных позволяет дать несколько непривычное, но очень удобное определение равенства функций. Именно, две булевы функции f_1 и f_2 называются ***равными***, если функцию f_2 можно получить из функции f_1 путём добавления и / или удаления несущественных переменных. В частности, это позволяет считать, что во всяком конечном множестве функций все функции зависят от одного и того же множества переменных.

Разумеется, эти понятия относятся не только к булевым, но и к произвольным функциям. Хорошо известным примером является равенство $\sin^2 x + \cos^2 x = 1$, означающее, что переменная xв функции $f(x) = \sin^2 x + \cos^2 x$ несущественна.

2. Функции и формулы

Табличное представления булевых функций подходят лишь для функций с небольшим числом аргументов. Формулы позволяют удобно представлять многие функции от бóльшего числа аргументов и оперировать различными представлениями одной и той же функции. Пусть $\mathcal{B}$– некоторое (конечное или бесконечное) множество булевых функций. Зафиксируем некоторое счётное множество переменных $V = \{x_1, x_2, \ldots\}$. Определим по индукции множество формул над $\mathcal{B}$ с переменными из V. Одновременно будем определять числовую характеристику $dep(\Phi)$ формулы Φ, называемую её ***глубиной***, и множество её ***подформул***.

а) **Базис индукции.** Каждая переменная $x_i \in V$ и каждая константа $c \in \mathcal{B}$ является формулой глбины 0, т.е. $dep(x_i) = dep(c) = 0$. Множество её подформул состоит из неё самой. Напомним, что константа является функцией, все переменные которой несущественны, и $c \in \{0, 1\}$.

б) **Шаг индукции.** Пусть $f(x_1, \ldots, x_m) \in \mathcal{B}, \Phi_1, \ldots, \Phi_m$ – формулы, и $\max_{1\le i\le m} dep(\Phi_i) = k$. Тогда выражение $\Phi = f(\Phi_1, \ldots, \Phi_m)$ является формулой, её глубина $dep(\Phi)$ равна $k + 1$, а множество подформул Φ включает саму формулу Φ и все подформулы формул $\Phi_1, \ldots, \Phi_m$.

Понятие «доказательства по индукции», может быть, встречалось читателю. Но вот «определения по индукции», чаще называемые индуктивными или рекурсивными определениями, для большинства читателей являются новыми. Остановимся на этом подробнее.

Речь идёт о задании, построении или описании множеств – в данном случае множества формул – некоторым процессом, который обычно называется порождающей процедурой (см. раздел 2-1). Идея такой процедуры состоит в определении некоторого достаточно простого исходного множества (это и есть базис индукции) и затем в последовательном добавлении в рассматриваемое множество новых элементов, полученных из уже имеющихся элементов, либо из каких-либо других объектов. Точная формулировка этих понятий далеко выходит за рамки настоящего пособия. Здесь мы просто проиллюстрируем «работу» порождающей процедуры на примере.

Пример 2. Определение множества формул над множеством функций $\mathcal{B}$, состоящем из четырёх функций: константы 0, константы 1, $\neg$ (отрицания) и $\vee$ (дизъюнкции).

Сначала определяется исходное множество формул: $\{0, 1, x_1, x_2, \ldots\}$. В соответствии с описанием базиса индукции, это множество объявляется множеством всех формул глубины 0. Далее, формулами глубины 1 являются все формулы вида $\neg x_1, \neg x_2, \ldots$, а также формулы вида $x_i \vee x_j$ ($i, j = 1, 2, \ldots$) и вида $c \vee x_i$, $x_i \vee c$($i = 1, 2, \ldots, c = 0, 1$). Далее, формулами глубины 2 являются все формулы вида $\neg x_i \vee x_j$, $x_i \vee \neg x_j$, $\neg x_i \vee \neg x_j$, $\neg(\neg x_i)$, $(x_i \vee x_j) \vee x_k$, и т.д.

Убедимся, например, что формулы $\neg x_i \vee x_j$ и $(x_i \vee x_j) \vee x_k$ имеют глубину 2. По построению, подформулами формулы $\neg x_i \vee x_j$ являются формулы $\Phi_1 = \neg x_i$ и $\Phi_2 = x_j$. Формула Φ_1 имеет глубину 1, формула Φ_2 имеет глубину 0, максимальное из этих чисел равно 1 и по описанию шага индукции глубина формулы $\Phi = \neg x_i \vee x_j$ равна 1 + 1 = 2. В формуле $(x_i \vee x_j) \vee x_k \Phi_1 = x_i \vee x_j$, $\Phi_1 = x_k$, их глубина равна 1 и 0 и, значит, глубина $\Phi = (x_i \vee x_j) \vee x_k$ равна 2. Заметим, что глубина формулы $\neg x_i \vee \neg x_j$ также равна 2 ■

Указанная конструкция определяет только «внешний» вид выражений, являющихся формулами, или их ***синтаксис***. Для сопоставления каждой корректно построенной формуле определяемой ею булевой функции, т.е. для определения ***семантики*** формул, нам понадобится ещё одна индуктивная процедура, в большой степени аналогичная уже рассмотренной.

Базис индукции. Пусть $dep(\Phi) = 0$. Тогда $\Phi = x_i \in V$ или $\Phi = c \in \mathcal{B}$. В 1-ом случае Φ определяет функцию $f_\Phi(x_i) = x_i$, во 2-ом – функцию, тождественно равную константе c.

Шаг индукции. Предположим (по индукции), что всем формулам над $\mathcal{B}$ глубины не более k булевы функции уже сопоставлены. Пусть $\Phi(x_1, \ldots, x_n)$ – произвольная формула глубины $k + 1$. По процедуре построения формул это означает, что $\Phi = f(\Phi_1, \ldots, \Phi_m)$, где $f(x_1, \ldots, x_m) \in \mathcal{B}$ и при этом $\max_{1\le i\le m} dep(\Phi_i) = k$. По предположению индукции, всем формулам глубины неболее k уже сопоставлены булевы функции $g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n)$. Тогда формула Φопределяет функцию $f_\Phi(x_1, \ldots, x_n) = f(g_1, \ldots, g_m)$. Для полной аккуратности конструкции надо включить в индуктивное предположение очевидное условие, что формулы с переменными $x_1, \ldots, x_n$ определяют функции от этих же переменных.

Заметим, что разделение конструкций на синтаксическую и семантическую части достаточно естественно для дискретной математики. Примерно по такой же схеме (но заметно сложнее) выглядит определение предикатов (здесь не рассматривается).

Далее будут рассматриваться формулы над множествами функций от одной и двух переменных, которые являются подмножествами одного множества так называемых элементарных функций $\mathcal{B}_e$ ={0, 1, ¬, ∧, ∨, →, ⊕, ⇄, |, ↓} (см. раздел 1.1). Определение формул в этих базисах и сопоставляемых им функций является частным случаем рассмотренных выше общих конструкций.

Пример 3. Для определения функции, задаваемой (говорят также ***реализуемой***) формулой, удобно использовать таблицу, строки которой соответствуют наборам значений переменных, а в столбцах под знаком каждой операции стоят значения функции, реализуемой соответствующей подформулой. Для формулы $\Phi(x_1, x_2, x_3) = (x_1 \vee x_2) \to (x_3 \rightleftarrows (x_1 \downarrow \neg x_2))$ функция f_Φ определяется в последнем столбце следующей таблицы 6:

Таблица 6. Построение таблицы истинности для функции $f_\Phi = (x_1 \vee x_2) \to (x_3 \rightleftarrows (x_1 \downarrow \neg x_2))$

x_1	x_2	x_3	1) $x_1 \vee x_2$	2) $\neg x_2$	3) $x_1 \downarrow 2$	4) $x_3 \rightleftarrows 3$	5) $1 \to 4$
0	0	0	0	1	0	1	1
0	0	1	1	1	0	0	0
0	1	0	1	0	1	0	0
0	1	1	1	0	1	1	1
1	0	0	0	1	0	1	1
1	0	1	1	1	0	0	0
1	1	0	1	0	0	1	1
1	1	1	1	0	0	0	0

Заголовок $x_1 \vee x_2$ столбца 1) и заголовок $\neg x_2$ столбца 2) понятны – в эти столбцы заносятся результаты указанных действий. Заголовок $x_1 \downarrow 2$ столбца 3) означает стрелку Пирса, применённую к x_1 и функции, уже подсчитанной в столбце 2). Заголовок$x_3 \rightleftarrows 3$ столбца 4) означает эквивалентность, применённую к x_3 и функции, уже подсчитанной в столбце 3). Заголовок $1 \to 4$ столбца 5) означает импликацию, посылка которой уже подсчитана в столбце 1), а заключение – в столбце 4). Таблицы, помогающие аналогичным образом вычислять булевы функции от двух-четырёх переменных по заданным формулам, называются ***таблицами истинности***■

Укажем на различие между таблицами 6 и 2. В таблице 2 в последнем столбце значения функции на всех наборах задавались, а понятие формулы не использовалось. В то же время в таблице 6 функция задаётся формулой, а заголовки столбцов подробно указывает на процесс вычисления булевой функции, реализуемой данной формулой.

2.1. Равносильность булевых формул. Булевы формулы Φ и Ψ называются ***равносильными,*** если соответствующие им функции f_Φ и f_Ψ равны. Равносильность двух формул обозначается как $\Phi \equiv \Psi$. Равносильные формулы называют также ***тождественно равными***, а выражения вида $\Phi \equiv \Psi$ – ***логическими тождествами***. Вместо термина «равносильность» иногда используется термин «эквивалентность».

Таким образом, равносильные формулы реализуют одну и ту же булеву функцию. Ниже приводится ряд пар равносильных формул (тождеств), отражающих существенные свойства логических операций и важные соотношения между различными операциями. Они часто позволяют находить для булевых функций по реализующим их формулам более простые эквивалентные формулы. Большинство из приводимых тождеств имеют собственные имена. Часто их называют ***законами логики.***

Пусть ∘ – это одна из функций ∧, ∨, ⊕. Для этих трёх функций выполнены следующие две равносильности (законы ***ассоциативности*** и ***коммутативности***).

1) Закон ***ассоциативность***: $(a \circ b_2) \circ c \equiv a \circ (b_2 \circ c)$.

Законы ассоциативности показывают, что значения формул, составленных из переменных и одних операций конъюнкции, одних операций дизъюнкции, одних операций сложения по модулю 2 не зависят от расстановки скобок. Поэтому вместо формул $(a\circ x_2)\circ c$ и $a\circ(b\circ c)$ мы будем для упрощения писать выражение $a\circ b\circ c$, которое не является формулой непосредственно по определению, но может быть легко превращено в неё с помощью расстановки скобок.

2) Закон ***коммутативность***: $a\circ b\equiv b\circ a$

3) Законы ***дистрибутивные***:

$(a\vee b)\wedge c\equiv(a\wedge c)\vee(b\wedge c)$

$(a\wedge b)\vee c\equiv(a\vee c)\wedge(b\vee c)$

$(a\oplus b)\wedge c\equiv(a\wedge c)\oplus(b\wedge c)$

4) ***Закон двойного отрицания***: $\neg(\neg a)\equiv a$

5) ***Законы де Моргана*** (внесение отрицания внутрь скобок):

$\neg(a\vee b)\equiv\neg a\wedge\neg b$

$\neg(a\wedge b)\equiv\neg a\vee\neg b$

6) ***Законы упрощения***:

законы идемпотентности $a\equiv a$, $a\vee a\equiv a$

закон противоречия $a\wedge\neg a\equiv 0$; ***закон исключенного третьего*** $a\vee\neg a\equiv 1$

законы 0 и 1 $a\wedge 0\equiv 0$, $a\vee 0\equiv a$, $a\wedge 1\equiv a$, $1\equiv 1$

7) ***Правила поглощения***: $a\vee ab\equiv a$; $a(a\vee b)\equiv a$

8) ***Правило склеивания***: $ab\vee a(\neg b)\equiv a$

9) ***Правило вычёркивания***: $ab\vee\neg a\equiv b\vee\neg a$

Далее будем часто писать ab вместо $a\wedge b$. Эта «вольность записи» общепринята, так как она приводит к заметному сокращению формул, а «оправдывается» тем, что результаты операции конъюнкции $0\wedge 0 = 0\wedge 1 = 1\wedge 0 = 0$, $1\wedge 1 = 1$ полностью совпадает с таблицей умножения чисел 0 и 1. При отсутствии скобок знак отрицания, стоящий перед переменной, имеет самый высокий приоритет, т.е. операция отрицания выполняется прежде всего. Следующий приоритет имеет логическое умножение. Остальные операции выполняются в порядке, указанном скобками.

Обратим также внимание на следующее. Все приведённые законы и правила – это не законы природы. Они доказываются (точнее, проверяются) подстановкой в левую и правую части логических тождеств всех наборов значений переменных (при n = 2 – четырёх наборов, при n = 3 – восьми).

Пример 4. Проверим справедливость 2-го закона дистрибутивности

$(a\wedge b)\vee c\equiv(a_1\vee c)\wedge(b\vee c)$

Для левой и правой частей проверяемого тождества получаем две таблицы истинности (см. для сравнения таблицу 5):

Таблица 7а. Левая часть тождества

a_1	b	c	1)$a_1\wedge b$	2)$1\vee c$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

Таблица 7б. Правая часть тождества

a_1	b	c	1)$a_1\vee c$	2)$b\vee c$	3) $1\wedge 2$
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	0	1	0
0	1	1	1	1	1
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

Поскольку правые столбцы полностью совпадают, то функция, реализуемая формулой из левой части, совпадает с функцией, реализуемой формулой из правой части. Это и означает справедливость 2-го закона дистрибутивности

Если в этом законе удалить знак конъюнкции (см. выше) и заменить дизъюнкцию на сложение, то данное тождество примет вид $ab+c\equiv(a+c)(b+c)$, что, разумеется, для чисел с операциями умножения и сложения неверно. Заметим, что при такой же замене в 1-ом законе дистрибутивности получаем закон, очевидно выполняющийся для чисел (и даже матриц):

$(a+b)c \equiv ac + bc$, что ещё раз говорит об аккуратности при использовании аналогий между булевыми и числовыми формулами ■

Многие из приведённых тождеств (2) – (7) и 1) – 9) можно не проверять, а выводить из других тождеств. Но на этом мы останавливаться не будем, стремясь к максимально возможной стандартизации рассуждений (для чего на самом деле и разрабатывалась булева алгебра ещё в 19-ом веке). Заметим, что все тождества остаются в силе, если вместо переменных a, b, c подставлять произвольные формулы в любом базисе.

Из определения равносильности формул непосредственно следует так называемый ***принцип замены равносильных подформул***: пусть формула α является подформулой формулы Φ, формула α'равносильна α и формула Φ' получена из Φ посредством замены некоторого вхождения α на α'. Тогда Φ' равносильна Φ, т.е. $\Phi' \equiv \Phi$.

Применяя этот принцип и используя основные тождества, можно находить для заданной формулы другие равносильные ей формулы.

Пример 5. Упрощение формулы. Имеет место цепочка тождеств, устанавливающая равносильность всех входящих в неё формул:

$((x_1 \to x_2)x_1 \vee \neg x_1x_2)(x_1 \vee \neg x_2) \equiv ((\neg x_1 \vee x_2)x_1 \vee \neg x_1x_2)(x_1 \vee \neg x_2) \equiv (\neg x_1x_1 \vee x_1x_2 \vee \neg x_1x_2)(x_1 \vee \neg x_2) \equiv$
$(x_1x_2 \vee \neg x_1x_2)(x_1 \vee \neg x_2) \equiv x_2(x_1 \vee \neg x_2) \equiv x_1x_2$.

Дадим пояснения. В исходной формуле заменяем подформулу $x_1 \to x_2$ на равносильную в силу формулы (2) подформулу $\neg x_1 \vee x_2$. Получаем формулу, стоящую справа от 1-го знака $\equiv$. Далее, заменяя в ней по 1-му закону дистрибутивности 3) формулу $(\neg x_1 \vee x_2)x_1$ на $\neg x_1x_1 \vee x_1x_2$, получаем формулу, стоящую справа от 2-го знака $\equiv$. Далее, используя в формуле $\neg x_1x_1$ закон противоречия, получаем $\neg x_1x_1 \equiv 0$, откуда в силу закона нуля $x \vee 0 \equiv x$ выводим, что $\neg x_1x_1 \vee x_1x_2 \vee \neg x_1x_2 \equiv x_1x_2 \vee \neg x_1x_2$, и получаем формулу справа от 3-го знака $\equiv$. Далее, по правилу склеивания, заменяем подформулу $x_1x_2 \vee \neg x_1x_2$ на x_2 и приходим к формуле справа от 4-го знака $\equiv$. Пользуясь ещё раз дистрибутивностью и законом противоречия, заменяем $x_2(x_1 \vee \neg x_2)$ на окончательную простую формулу x_1x_2, равносильную исходной формуле $((x_1 \to x_2)x_1 \vee \neg x_1x_2)(x_1 \vee \neg x_2)$ ■

Как уже говорилось, закон ассоциативности позволяет рассматривать многоместную конъюнкцию $\bigwedge_{i=1}^{n} x_i$ и многоместную дизъюнкцию $\bigvee_{i=1}^{n} x_i$ в качестве формул. В случае $n = 1$ обе формулы означают x_1. Пустая конъюнкция (при $n = 0$) полагается равной 1, пустая дизъюнкция – равной 0.

Законы де Моргана 5) могут быть обобщены для произвольного неотрицательного n:

$$\neg\left(\bigvee_{i=1}^{n} x_i\right) \equiv \bigwedge_{i=1}^{n} \neg x_i \tag{8а}$$

$$\neg\left(\bigwedge_{i=1}^{n} x_i\right) \equiv \bigvee_{i=1}^{n} \neg x_i \tag{8б}$$

При $n > 2$ в этом можно убедиться, представив многоместную функцию расстановкой скобок с помощью двуместных. При $n = 3$, например, цепочка преобразований для (8б) имеет вид:

$\neg(x_1 \wedge x_2 \wedge x_3) = \neg((x_1 \wedge x_2) \wedge x_3) = \neg(x_1 \wedge x_2) \vee \neg x_3) = (\neg x_1 \vee \neg x_2) \vee \neg x_3 = \neg x_1 \vee \neg x_2 \vee \neg x_3$.

В случае $n = 0$ и $n = 1$ соотношения (8) выполняются очевидным образом.

Соотношения (8) допускают дальнейшее обобщение. Пусть функция f реализуется некоторой формулой в базисе $\mathcal{B} = \{0, 1, \neg, \wedge, \vee\}$. Тогда формула, реализующая её отрицание $\neg f$,может быть получена по следующему алгоритму.

Алгоритм отрицания в базисе $\{0, 1, \neg, \wedge, \vee\}$.

1. Все конъюнкции (значки $\wedge$) должны быть заменены на дизъюнкции (значки $\vee$) и наоборот, все дизъюнкции должны быть заменены на конъюнкции.

2. Переменные x_i должны быть заменены на их отрицания $\neg x_i$.

3. Константы 0 и 1 должны быть заменены противоположными константами 1 и 0 ■

Обоснованием алгоритма являются законы де Моргана. Приведём пример, иллюстрирующий работу алгоритма отрицания.

Пример 6. Пусть функция f задается формулой $(x_1 \wedge x_2) \vee (\neg x_1 \wedge \neg x_2) \equiv x_1 \rightleftarrows x_2$ (см. тождество (5)). При отрицании левой части предполагаемого тождества получаем

$\neg((x_1 \wedge x_2) \vee (\neg x_1 \wedge \neg x_2)) \equiv \neg(x_1 \vee x_2) \wedge \neg(\neg x_1 \wedge \neg x_2) \equiv (\neg x_1 \wedge \neg x_2) \wedge (\neg(\neg x_1) \vee \neg(\neg x_2)) \equiv (\neg x_1 \wedge \neg x_2) \wedge (x_1 \vee$
$x_2) \equiv (\neg x_1 \wedge x_1 \vee \neg x_2 \wedge x_1 \vee \neg x_1 \wedge x_2 \vee \neg x_2 \wedge x_2 \equiv \neg x_2 \wedge x_1 \vee \neg x_1 \wedge x_2 \equiv \neg x_1 \wedge x_2 \vee x_1 \wedge \neg x_2 \equiv x_1 \oplus x_2$. (9)

1-ое тождество следует из 2-го закон де Моргана. 2-ое тождество получено применением законов де Моргана к обоим слагаемым дизъюнкции. 3-ье тождество получено применением закона двойного отрицания к выражениям $\neg(\neg x_1)$ и $\neg(\neg x_2)$. Далее последовательно использована дистрибутивность для 4-го тождества, закон исключённого третьего для 5-го тождества (два члена из четырёх пропадают), коммутативность в 6-ом тождестве и тождество (3) в последнем, 7-ом тождестве. Преобразования заканчиваются формулой $x_1 \oplus x_2$. Таблица 5 подтверждает, что функции $x_1 \rightleftarrows x_2$ и $x_1 \oplus x_2$ (т.е. эквивалентность и сложение по модулю 2) действительно являются отрицаниями друг друга, поскольку на любом наборе, где одна из них равна 1, другая равна 0, и наоборот ■

2.2. СДНФ и СКНФ. Всякая булева функция от n переменных, по определению, задаётся своими значениями на 2^n наборах нулей и единиц. Однако уже при сравнительно небольших значениях nзадание функции таблицей громоздко и неудобно. Поэтому желательно иметь некоторый регулярный способ представления булевых функций при помощи формул, которые (в отличие от таблиц) могут быть преобразованы, упрощены и т.д. В настоящем разделе даются наиболее известные представления булевых формул в виде совершенной дизъюнктивной нормальной формы (СДНФ) и совершенной конъюнктивной нормальной формы (СКНФ).

Пусть $\sigma \in \{0, 1\}$. Положим

$$x^{\sigma} = \begin{cases} \boldsymbol{x}, \text{ если } \sigma = 1 \\ \neg \boldsymbol{x}, \text{ если } \sigma = 0 \end{cases}. \tag{10}$$

Сформулируем несколько почти очевидных утверждений, необходимых для определения СДНФ и СКНФ.

Утверждение 1. $x^{\sigma} = 1$ тогда и только тогда, когда $x = \sigma$ ■

Утверждение 2. Конъюнкция $\boldsymbol{x_1^{\sigma_1} x_2^{\sigma_2}} \ldots \boldsymbol{x_k^{\sigma_k}}$ равна 1 на единственном наборе значений переменных $x_1, \ldots, x_k$: $x_1 = \sigma_1, \ldots, x_k = \sigma_k$■

Следующее утверждение позволяет выразить функцию $f(x_1, \ldots, x_n)$ через функции от ме́ньшего числа аргументов.

Утверждение 3. Всякая логическая функция $f(x_1, \ldots, x_n)$ при любом $k = 1, 2, \ldots, n$ представляется в виде

$$f(x_1, \ldots, x_k, x_{k+1}, \ldots, x_n) = \bigvee_{\sigma_1, \sigma_2, \ldots, \sigma_k} \boldsymbol{x_1^{\sigma_1} x_2^{\sigma_2}} \ldots \boldsymbol{x_k^{\sigma_k}} f(\sigma_1, \ldots, \sigma_k, x_{k+1}, \ldots, x_n), \tag{11}$$

где дизъюнкция берётся по всем 2^k наборам $\sigma_1, \ldots, \sigma_k$ значений 0 и 1.

Доказательство следует из того, что для любого набора $\alpha_1, \ldots, \alpha_n$ левая и правая части (11) совпадают ■

Представление (11) произвольной булевой функции называется разложением по переменным $x_1, \ldots, x_k$. Его частный случай при $k = 1$ имеет вид

$$f(x_1, x_2, \ldots, x_n) = x_1 f(1, x_2, \ldots, x_n) \vee \neg x_1 f(0, x_2, \ldots, x_n) \tag{12}$$

и носит название формулы ***разложения по одной переменной***. Разумеется, вместо x_1 можно взять любую другую переменную.

Воспользовавшись утверждением 3 при $k = n$, получаем представление

$$f(x_1, \ldots, x_n) = \bigvee_{\sigma_1, \sigma_2, \ldots, \sigma_n} \boldsymbol{x_1^{\sigma_1} x_2^{\sigma_2}} \ldots \boldsymbol{x_n^{\sigma_n}} f(\sigma_1, \ldots, \sigma_n). \tag{13}$$

При этом в правую часть (13) входят только те «слагаемые», на которых значения $f(\sigma_1, \ldots, \sigma_n)$ равны 1. Опуская значения $f(\sigma_1, \ldots, \sigma_n)$, поскольку логическое умножение» на 1 можно не писать, получаем из (13) равенство

$$f(x_1, \ldots, x_n) = \bigvee_{\sigma_1, \sigma_2, \ldots, \sigma_n | f(\sigma_1, \sigma_2, \ldots, \sigma_n) = 1} \boldsymbol{x_1^{\sigma_1} x_2^{\sigma_2}} \ldots \boldsymbol{x_n^{\sigma_n}}, \tag{14}$$

где дизъюнкция берётся по всем наборам $\sigma_1, \ldots, \sigma_n$, на которых функция $f(\sigma_1, \ldots, \sigma_n) = 1$.

Представление (8) и называется ***совершенной дизъюнктивной нормальной формой*** (сокращённо СДНФ) булевой функции f от n переменных. С учетом соглашения о том, что пустая дизъюнкция равна 0, оно распространяется на функцию $f(x_1, \ldots, x_n)$, тождественно равную 0.

СДНФ обладает следующими свойствами.

1. Она является дизъюнкцией некоторых конъюнкций $K_1 \vee K_2 \vee \ldots \vee K_s$.
2. Каждая из конъюнкций K_iимеет вид $\boldsymbol{x_1^{\sigma_1} x_2^{\sigma_2}} \ldots \boldsymbol{x_n^{\sigma_n}}$, где n – число переменных функции.
3. Все конъюнкции $K_1, K_2, \ldots, K_s$ различны.

Утверждение 4а. Представление булевой функции, обладающее свойствами 1 – 3, единственно с точностью до перестановки конъюнкций ■

Введём в рассмотрение также важное для дальнейшего понятие ***дизъюнктивной нормальной формы* (*ДНФ*)**. От СДНФ ДНФ отличается только одним: входящие в неё конъюнкции не обязаны включать все переменные. Например, формула $xy\vee y\neg z$ является ДНФ, но не является СДНФ. Однако СДНФ является частным случаем ДНФ.

Всякая функция $f(x_1, \ldots, x_n) \not\equiv 1$ может быть выражена также в виде конъюнкции некоторых дизъюнкций $\boldsymbol{x_1^{\tau_1}}\vee\boldsymbol{x_2^{\tau_2}}\vee \ldots\vee\boldsymbol{x_n^{\tau_n}}$. Чтобы получить это представление, выпишем СДНФ функции $\neg f(x_1, \ldots, x_n) \not\equiv 0$:

$$\neg f(x_1, \ldots, x_n) = \bigvee_{\sigma_1,\sigma_2,\ldots,\sigma_n \mid \neg f(\sigma_1,\sigma_2,\ldots,\sigma_n)=1} x_1^{\sigma_1} x_2^{\sigma_2} \ \ldots\ x_n^{\sigma_n}. \tag{15}$$

Воспользовавшись алгоритмом отрицания из раздела 2.1 и тем, что $\neg(\neg f) = f)$, из (9) получаем

$$f(x_1, \ldots, x_n) = \bigwedge_{\sigma_1,\sigma_2,\ldots,\sigma_n \mid f(\sigma_1,\sigma_2,\ldots,\sigma_n)=0} x_1^{\sigma_1}\vee x_2^{\sigma_2}\vee \ldots \vee x_n^{\sigma_n} \tag{16}$$

(условие суммирования $\neg f(x_1, \ldots, x_n) = 1$ заменено на эквивалентное условие $f(x_1, \ldots, x_n) = 0$).

Представление (16) носит название ***совершенной конъюнктивной нормальной формы***(сокращённо СКНФ). С учетом соглашения о том, что пустая конъюнкция равна 1, оно распространяется на функцию $f(x_1, \ldots, x_n)$, тождественно равную 1.

СКНФ обладает следующими свойствами.

1. Она является конъюнкцией некоторых дизъюнкций $D_1\wedge D_2\wedge \ldots \wedge D_t$.
2. Каждая из дизъюнкций D_i имеет вид $\boldsymbol{x_1^{\sigma_1}}\vee\boldsymbol{x_2^{\sigma_2}}\vee \ldots \vee\boldsymbol{x_n^{\sigma_n}}$, где n – число переменных функции.
3. Все дизъюнкции $D_1, D_2, \ldots, D_s$ различны.

Утверждение 4б. Представление булевой функции, обладающее свойствами 1 – 3, единственно с точностью до перестановки дизъюнкций ■

Это утверждение может быть доказана непосредственно либо на основе утверждения для СДНФ, применённой к функции $\neg f$.

Понятие КНФ определяется аналогично понятию ДНФ.

2.2.1. Равносильность формул. На основе утверждения 4а можно разработать алгоритм установления равносильности формул. На «верхнем» уровне алгоритм таков.

Алгоритм установления эквивалентности формул. Формулы Φ_1 и Φ_2 над стандартным базисом $\mathcal{B}_e =\{0, 1, \neg, \wedge, \vee, \to, \oplus, \rightleftarrows, |, \downarrow\}$ и одним и тем же конечным множеством переменных $\{x_1, x_2, \ldots, x_n\}$ предполагаются заданными.

1. Приведём формулу Φ_1 к СДНФ.
2. Приведём формулу Φ_2 к СДНФ.
3. Если СДНФ совпадают (с точностью до порядка конъюнкций), то это и означает равносильность формул Φ_1 и Φ_2.

Суть дела в том, что преобразования перехода от любой формулы к равносильной ей формуле обратимы. Более точно это означает следующее. Пусть в результате замены подформулы формулы Φ', совпадающей с левой частью некоторого тождества из (2) – (7) и 1) – 9), правой частью этого же тождества, получаем формулу Φ''. Но тогда при замене в формуле Φ'' правой части этого же тождества его левой частью, получаем снова формулу Φ'. Поэтому представление двух формул Φ_1 и Φ_2 одной и той же СДНФ не только автоматически влечёт равносильность Φ_1 и Φ_2, но и определяет явный вид перехода от одной формуле к другой. Таким образом, алгоритм сводится к основному шагу – построению СДНФ по заданной формуле Φ. Последний, в свою очередь, сводится к трём шагам:

1. Преобразование формулы в стандартном базисе в формулу над уже упоминавшимся базисом $\mathcal{B} =\{0, 1, \neg, \wedge, \vee\}$.
2. Представление формулы в базисе $\mathcal{B} =\{0, 1, \neg, \wedge, \vee\}$ в виде ДНФ
3. ДНФ преобразуется в СДНФ.

Рассмотрим алгоритмы, реализующие указанные шаги по отдельности

1. Функции $\to, \oplus, \rightleftarrows, |, \downarrow$ из стандартного базиса выражается через дизъюнкцию, конъюнкцию и отрицание с помощью тождеств (2) – (7). Делая указанные замены, в соответствии с принципом замены равносильных подформул, получаем равносильную исходной формулу, в которую входят только символы переменных и функции из множества $\{0, 1, \neg, \wedge, \vee\}$.

2. **Алгоритм построения ДНФ по формуле в базисе $\mathcal{B}$ ={0, 1, ¬, ∧, ∨}.** Наиболее сложный из шагов 1 – 3. Он состоит из следующих более простых шагов.

2.1. Убираем все двойные отрицания.

2.2. Переносим все отрицания «внутрь подформул», пока они не останутся только перед переменными. Примером такого перенесения являются сами законы де Моргана.

2.3. Раскрываем по закону дистрибутивности все скобки. Результирующее выражение будет представлять собой дизъюнкцию конъюнкций.

2.4. Удаляем все конъюнкции, в которые входят одновременно и некоторая переменная, и её отрицание (закон исключённого третьего).

2.5. В оставшихся конъюнкциях оставляем по одному вхождению каждой переменной, если их было несколько (1-ый закон идемпотентности).

2.6. Из нескольких совпадающих конъюнкций (если они есть) оставляем только одну (2-ой закон идемпотентности).

Среди перечисленных шагов наиболее сложным является шаг 2.2. Приведём формальный алгоритм его выполнения. Пусть Φ – произвольная формула в базисе $\mathcal{B}$ ={0, 1, ¬, ∧, ∨}. Пусть в ней есть отрицание, не стоящее непосредственно перед переменной. Тогда отрицание стоит непосредственно перед некоторой подформулой Φ*. Учитывая заданный базис, можно сказать, что имеет место один из двух случаев:

А. Отрицание стоит перед скобками, внутри которых имеется дизъюнкция некоторых подформул.

Б. Отрицание стоит перед скобками, внутри которых имеется конъюнкция некоторых подформул.

В случае А применим 1-ый закон де Моргана, в случае Б – 2-ой закон де Моргана. В обоих случаях знак отрицания становится «ближе» к переменным. Формально это определяется как изменение глубины формулы, перед которой стоял знак отрицания. Легко доказать, пользуясь индуктивным определением формулы, что максимальная глубина формулы, перед которой знак отрицания стоит после перенесения, ровно на 1 меньше, чем глубина формулы, перед которой знак отрицания стоял до перенесения. Поэтому сам алгоритм таков.

1. Если в формуле есть отрицание, не стоящее непосредственно перед переменной, то преобразуем формулу по одному из законов де Моргана.

2. Удаляем двойные отрицания, если они возникли.

3. Выполняем шаги 1 и 2 до тех пор, пока знаков отрицания, не стоящих непосредственно перед переменной, не останется. Это означает, что все знаки отрицания стоят непосредственно перед переменными.

Остальные пункты 2.3 – 2.6 не нуждаются в специальных пояснениях. Остановимся подробнее на шаге 3 построения СДНФ – построении СДНФ по ДНФ. Суть дела здесь также проста. Пусть $K = \boldsymbol{x_{i_1}^{\sigma_1} x_{i_2}^{\sigma_2} \ldots x_{i_k}^{\sigma_k}}$ – одна из входящих в ДНФ конъюнкций, которая не содержит некоторых переменных. Такие конъюнкции называются ***неполными***. Пусть переменная x не входит в K. Определим две конъюнкции $K' = Kx^1$ и $K'' = Kx^0$ (см. формулу (10) по поводу обозначения x^{σ}). Тогда по правилу склеивания $x_1x_2 \vee x_1 (\neg x_2) \equiv x_1$, подставляя вместо $x_1 K$ и вместо $x_2 x$, получаем $K' \vee K'' \equiv K$, причём обе конъюнкции K' и K'' содержат все «старые» переменные, а также переменную x, которая не входит в K. Совершенно аналогично поступаем со всеми конъюнкциями, входящими в рассматриваемую ДНФ. Когда вместо исходной ДНФ получим дизъюнкцию конъюнкций, содержащих все переменные, останется только из каждой группы совпадающих конъюнкций (если таковые окажутся) удалить все конъюнкции, кроме одной ■

В большинстве случаев нетрудно найти гораздо более короткие цепочки преобразований, чем получаемые в результате применения описанного алгоритма. Но у него есть значительное преимущество – он позволяет находить эти цепочки не в результате догадок и сообразительности, а в результате выполнения стандартных операций, которые можно поручить компьютеру. Перефразируя известное высказывание Кронекера, можно сказать, что это преимущество примерно того же типа, как у честного труда над воровством. Изложение различных алгоритмов, позволяющих стандартным образом решать достаточно сложные задачи, а также сопутствующих понятий и обозначений, и является основным содержанием настоящего пособия.

Пример 7а. Рассмотрим формулу $(x_2\to x_1)((x_2\oplus 1)\neg x_3\rightleftarrows\neg x_3)$. В соответствии с алгоритмом построения СДНФ, запишем её в базисе $\mathcal{B}=\{0, 1, \neg, \wedge, \vee\}$. Для этого, как указано выше, выразим входящие в исходную формулу функции $\to$, $\oplus$, $\rightleftarrows$ через функции 0, 1, $\neg$, $\wedge$, $\vee$ из $\mathcal{B}$, используя тождества (2) – (7). Заменим подформулу $x_2\to x_1$ равносильной ей формулой $\neg x_2\vee x_1$ (см. тождество (2) при $a = x_2$, $b = x_1$). Получим равносильную исходной формулу $(\neg x_2\vee x_1)((x_2\oplus 1)\neg x_3\rightleftarrows\neg x_3)$. Заменим в последней подформулу $x_2\oplus 1$ равносильной ей формулой $\neg x_2$.(см. тождество (4) при $a = x_2$). Получим равносильную исходной формулу $(\neg x_2\vee x_1)(\neg x_2\neg x_3\rightleftarrows\neg x_3)$. Заменим в последней подформулу $\neg x_2\neg x_3\rightleftarrows\neg x_3$ равносильной ей формулой $(\neg x_2\neg x_3)\neg x_3\vee\neg(\neg x_2\neg x_3)\neg\ (\neg x_3)$ (см. тождество (5) при $a = \neg x_2\neg x_3$,$b = \neg x_3$). Получим равносильную исходной формулу $(\neg x_2\vee x_1)\ ((\neg x_2\neg x_3)\ \neg x_3\vee\neg(\neg x_2\neg x_3)\neg(\neg x_3))$. Последняя формула уже является формулой в базисе $\mathcal{B}$, так что шаг 1 выполнен ■

Пример 7б. Рассматриваемая формула имеет вид $(\neg x_2\vee x_1)\ ((\neg x_2\neg x_3)\neg x_3\vee\neg(\neg x_2\neg x_3)\neg(\neg x_3))$. В ней имеется единственное двойное отрицание $\neg(\neg x_3)$. Убрав его в соответствии с шагом 2.1, получим равносильную исходной формулу $(\neg x_2\vee x_1)\ ((\neg x_2\neg x_3)\neg x_3\vee\neg(\neg x_2\neg x_3)x_3)$, в которой нет двойных отрицаний.

В последней формуле имеется единственное вхождение отрицания перед подформулой: $\neg(\neg x_2\neg x_3)$. В соответствии с алгоритмом выполнения шага 2.2, используем закон де Моргана: $\neg(ab) = \neg a\vee\neg b$ при $a = \neg x_2$, $b = \neg x_3$ (как и выше, знак $\wedge$ опущен). Получаем $\neg(\neg x_2\neg x_3) = x_2\vee x_3$. Заменяя подформулу $\neg(\neg x_2\neg x_3)$ на формулу $x_2\vee x_3$, получаем равносильную исходной формулу $(\neg x_2\vee x_1)((\neg x_2\neg x_3)\neg x_3\vee(x_2\vee x_3)\wedge x_3)$, в которой уже все отрицания стоят непосредственно перед переменными, и двойных отрицаний нет.

Далее, в соответствии с шагом 2.3, в последней формуле раскрываем все скобки, используя законы дистрибутивности. Начинаем со 2-го сомножителя: $(\neg x_2\vee x_1)((\neg x_2\neg x_3)\neg x_3\vee(x_2\vee x_3)\wedge x_3). \equiv (\neg x_2\vee x_1)(\neg x_2\neg x_3\neg x_3\vee x_2x_3\vee x_3x_3)$. Далее перемножаем скобки. В первой содержатся два члена, соединённых знаком дизъюнкции $\vee$, во второй – три члена, соединённых знаком дизъюнкции $\vee$: Перемножая скобки почленно, получаем 6 слагаемых:

$$\neg x_2\neg x_2\neg x_3\neg x_3\vee x_1\neg x_2\neg x_3\neg x_3\vee\neg x_2x_2x_3\vee x_1x_2x_3\vee\neg x_2x_3x_3\vee x_1x_3x_3 \quad (17)$$

Поскольку в формуле (17) скобки отсутствуют, то шаг 2.3 выполнен.

На следующем шаге 2.4 удаляем единственный член $\neg x_2x_2x_3$, в который входит как переменная x_2, так и её отрицание $\neg x_2$. В результате получим формулу (18):

$$\neg x_2\neg x_2\neg x_3\neg x_3\vee x_1\neg x_2\neg x_3\neg x_3\vee x_1x_2x_3\vee\neg x_2x_3x_3\vee x_1x_3x_3, \quad (18)$$

содержащую 5 слагаемых.

В соответствии с шагом 2.5, в оставшихся конъюнкциях оставляем по одному вхождению каждой переменной, если их было несколько (1-ый закон идемпотентности). Получаем формулу:

$$\neg x_2\neg x_3\vee x_1\neg x_2\neg x_3\vee x_1x_2x_3\vee\neg x_2x_3\vee x_1x_3. \quad (19)$$

Поскольку совпадающих конъюнкций в формуле (19) нет, то она является одной из многих ДНФ, равносильных исходной формуле ■

Пример 7в. Осталось преобразовать построенную ДНФ в СДНФ, в соответствии с приведённым выше описанием шага 3. Получаем, просматривая последовательно все члены, содержащие не все переменные в (19):

$\neg x_2\neg x_3\equiv x_1\neg x_2\neg x_3\vee\neg x_1\neg x_2\neg x_3$;

$\neg x_2x_3\equiv x_1\neg x_2x_3\vee\neg x_1\neg x_2x_3$;

$x_1x_3\equiv x_1x_2x_3\vee x_1\neg x_2x_3$.

Подставляя эти выражения в (19) вместо соответствующих подформул, получаем

$x_1\neg x_2\neg x_3\vee\neg x_1\neg x_2\neg x_3\vee x_1\neg x_2\neg x_3\vee x_1x_2x_3\vee x_1\neg x_2x_3\vee\neg x_1\neg x_2x_3\vee x_1x_2x_3\vee x_1\neg x_2x_3$.

Удаляя из последней формулы одну из двух конъюнкций $x_1x_2x_3$ и одну из двух конъюнкций $x_1\neg x_2x_3$, получаем формулу

$$x_1\neg x_2\neg x_3\vee\neg x_1\neg x_2\neg x_3\vee x_1\neg x_2\neg x_3\vee x_1x_2x_3\vee x_1\neg x_2x_3\vee\neg x_1\neg x_2x_3, \quad (20)$$

являющейся СДНФ функции, реализуемой исходной формулой $(x_2\to x_1)((x_2\oplus 1)\neg x_3\rightleftarrows\neg x_3)$ ■

Во многих случаях требуется не построить СДНФ по заданной формуле, а упростить формулу в том же базисе $\mathcal{B}$, насколько это возможно. Для этого используются те же самые тождества (2) – (7) и 1) – 9).

Пример 8. Рассмотрим ДНФ (19), найденную при преобразованиях в примере 7:$\neg x_2\neg x_3\vee x_1\neg x_2\neg x_3\vee x_1x_2x_3\vee\neg x_2x_3\vee x_1x_3$. Постараемся найти равносильную ей более простую формулу. Сначала выпишем отдельные равносильности:

$\neg x_2\neg x_3\vee x_1\neg x_2\neg x_3\equiv\neg x_2\neg x_3$ (см. правило поглощения $a\vee ab\equiv a$ при $a = \neg x_2\neg x_3$, $b = x_1$);

$x_1x_2x_3\vee x_1x_3\equiv x_1x_3$ (см. правило поглощения $a\vee ab\equiv a$ при $a = x_1x_3$, $b = x_2$).

Подставляя правые части этих тождеств в формулу $\neg x_2\neg x_3\vee x_1\neg x_2\neg x_3\vee x_1x_2x_3\vee\neg x_2x_3\vee x_1x_3$, получаем $\neg x_2\neg x_3\vee x_1x_3\vee\neg x_2x_3$. Далее, так как $\neg x_2\neg x_3\vee\neg x_2x_3\equiv\neg x_2$ (правило склеивания), то $\neg x_2\neg x_3\vee x_1x_3\vee\neg x_2x_3\equiv\neg x_2\vee x_1x_3$. Таким образом, простая формула $\neg x_2\vee x_1x_3$ равносильна как исходной формуле $(x_2\rightarrow x_1)((x_2\oplus 1)\neg x_3\rightleftarrows\neg x_3)$, так и её СДНФ (14), найденной в примере 7 ■

Разнообразные понятия сложности формул, а также ещё более важные понятия сложности уже не формул, а алгоритмов, вычисляющих заданные различными способами булевы функции, выходят за рамки настоящего пособия и далее не рассматриваются.

3. Формальный анализ высказываний

В настоящем разделе мы возвращаемся к формальному анализу составных высказываний, начатому в главе 1. Введённые в этой главе булевы функции, таблицы истинности и связанные с ними понятия оказываются очень удобным инструментом для такого анализа.

В разделе 1-3 было продемонстрировано, как формально описать составное высказывание с помощью более короткого выражения. В это выражение входят символы простых высказываний, из которых образовано данное составное высказывание, а также соединяющие их знаки операций над высказываниями. Входят, если необходимо, и скобки, которые указывают порядок выполнения этих операций, подобно тому, как скобки указывают порядок выполнения алгебраических и арифметических операций в «школьной» математике. Все шесть возможных операций над высказываниями и их истинностными значениями ($\neg$, $\wedge$, $\vee$, $\rightarrow$, $\oplus$, $\rightleftarrows$) подробно определены и объяснены в разделе 1-2.1.

Исходя из формальной записи составного высказывания, можно получить его новое представление – сопровождающую высказывание булеву формулу и реализуемую ей булеву функцию. Опишем конструкцию перехода от высказывания к булевой формуле. Напомним, что в формальное описание составного высказывания входят символы простых высказываний (обычно это прописные буквы латинского алфавита), знаки операций $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\oplus$, $\rightleftarrows$ и скобки. Заменим все прописные буквы латинского алфавита на те же самые, но строчные буквы того же алфавита. Будем рассматривать эти буквы как знаки булевых переменных. Само же получившееся (после указанной замены и интерпретации букв как булевых переменных) выражение будем теперь рассматривать, как формулу в базисе $\{\neg, \wedge, \vee, \rightarrow, \oplus, \rightleftarrows\}$ (в смысле точного рекурсивного определения из раздела 2 настоящей главы). Далее именно эти формулу и реализуемую ей булеву функцию будем называть ***формулой и функцией, сопровождающей данное составное высказывание***. Вместо символов F и T будем использовать символы 0 и 1.

Внимательный читатель может заметить, что формальное выражение составного высказывания не является, несмотря на свой формальный характер, строго определённым объектом. В то же время булева формула (над заданным базисом) и реализуемая ей булева функция строго определены рекурсивными процедурами из раздела 2. Поэтому нет гарантии, что для любого составного высказывания его формальное выражение после объявления символов простых высказываний булевыми переменными действительно становится корректно определённой формулой. Такую гарантию даёт рекурсивная процедура точного определения составных высказываний и соответствующих им выражений, в которые входят только символы простых высказываний, знаки операций и скобки. Такая процедура будет практически совпадать с описанными в разделе 2 процедурами. Мы этого не делаем, не желая загромождать изложение не очень существенными, но долго описываемыми техническими деталями. Достаточно сказать, что во всех рассматриваемых и упоминаемых ситуациях никаких проблем с однозначным пониманием получающихся выражений именно как булевых формул не возникает.

Ещё раз подчеркнём, что формализация составных высказываний является содержательной операцией (см. раздел 1-3), в то время как сами формулы и функции являются, конечно, формальными, точно определёнными объектами, к которым приложим достаточно сложный и разнообразный математический аппарат. Именно указанное сопоставление формул и функций составным высказываниям и позволяет использовать алгебру логики для их формального анализа. В настоящем пособии затрагивается лишь небольшая часть этого мощного аппарата.

Пример 9. Вернёмся к высказываниям из примеров 1-3,1-4, 1-5, 1-7, 1-9, 1-10, 1-16 и перепишем оттуда полученные там формальные выражения для рассмотренных составных высказываний:

$(P\wedge Q)\to(R\vee S),$ (21а)

$((P\wedge Q)\to R)\wedge\neg R,$ (22а)

$(((P\wedge Q)\to R)\wedge\neg R)\to(\neg P\vee\neg Q),$ (23а)

$(P\vee\neg P)\to Q,$ (24а)

$(P\to Q)\wedge P,$ (25а)

$((P\to Q)\wedge P)\to Q,$ (26а)

$(P\to R)\wedge(R\to Q))\to(P\to Q).$ (27а)

После замены прописных букв строчными и интерпретации их как булевых переменных получаем булевы формулы, сопровождающие исходные составные высказывания:

$(p\wedge q)\to(r\vee s),$ (21б)

$((p\wedge q)\to r)\wedge\neg r,$ (22б)

$(((p\wedge q)\to r)\wedge\neg r)\to(\neg p\vee\neg q),$ (23б)

$(p\vee\neg p)\to q,$ (24б)

$(p\to q)\wedge p,$ (25б)

$((p\to q)\wedge p)\to q,$ (26б)

$(p\to r)\wedge(r\to q))\to(p\to q).$ (27б)

Далее для иллюстрации свойств высказываний мы будем обращаться к этим формулам ■

Пример 10. Рассмотрим ещё раз высказывание из примера 1-4: ***«Если я – Ваша женщина и Вы – мой мужчина, то я никогда не перестану любить Вас. Я перестала любить Вас»***. Её сопровождающая формула (22б) равна $((p\wedge q)\to r)\wedge\neg r$. Найдём значения этой функции при всех возможных значениях переменных. Для этого построим её таблицу истинности по аналогии с таблицами истинности 6 и 7. В правом столбце таблицы записаны значения этой функции на соответствующих наборах. Из таблицы 8 можно сделать следующий вывод. Если все три исходных простых высказывания «Я ***– Ваша женщина»***, «***Вы – мой мужчина»*** и «Я ***никогда не перестану любить Вас»*** ложны, то рассматриваемое составное высказывание истинно. А в случае истинности этих трёх высказываний рассматриваемое составное высказывание ложно.

Таблица 8. Построение таблицы истинности для функции $f = ((p\wedge q)\to r)\wedge\neg r$.

p	q	r	1) $p\wedge q$	2) $1\to r$	3) $\neg r$	4) $2\wedge 3$
0	0	0	0	1	1	1
0	0	1	0	1	0	0
0	1	0	0	1	1	1
0	1	1	0	1	0	0
1	0	0	1	0	1	0
1	0	1	1	1	0	0
1	1	0	1	0	1	0
1	1	1	1	1	0	0

■

Пример 11. Рассмотрим теперь всю «историю» из примера 1-5: «***Если я – Ваша женщина и Вы – мой мужчина, то я никогда не перестану любить Вас. Я перестала любить Вас. Значит, я – не Ваша женщина или Вы – не мой мужчина***». Сопровождающая формула (23б) равна $(((p\wedge q)\to r)\wedge\neg r)\to(\neg p\vee\neg q)$. Последовательное заполнение соответствующей таблицы истинности 9 показано ниже. Заметим, что часть таблицы 9 скопирована из таблицы 8, поскольку формула (22б) является подформулой формулы (23б).

Таблица 9. Построение таблицы истинности для функции $f =(((p\wedge q)\to r)\wedge\neg r)\to(\neg p\vee\neg q)$

p	q	r	1) $p\wedge q$	2) $1\to r$	3) $\neg r$	4) $2\wedge 3$	5) $\neg p$	6) $\neg q)$	7) $5\vee 6$	8) $4\to 7$
0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	1	0	0	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1

0	1	1	0	1	0	0	1	0	1	1
1	0	0	1	0	1	0	0	1	1	1
1	0	1	1	1	0	0	0	1	1	1
1	1	0	1	0	1	0	0	0	0	1
1	1	1	1	1	0	0	0	0	0	1

Результат может показаться неожиданным – ведь при всех значениях переменных, т.е. при любых сочетаниях истинности и ложности исходных простых высказываний, данное составное высказывание оказывается истинным! Более того, если построить таблицы истинности для всех сопровождающих формул (21б) – (27б), кроме уже рассмотренной формулы (22) и формулы (25), результат окажется таким же – соответствующие функции тождественно равны 1. Подробнее эти вопросы рассматриваются в разделе 3.2 ■

Пример 12. Построим таблицу истинности для формулы (27б), называемой ***правилом силлогизма***. Последовательные шаги показаны в таблице 10.

Таблица 10. Построение таблицы истинности для функции $f=((p\to r)\wedge(r\to q))\to(p\to q)$

p	q	r	1) $p\to r$	2) $r\to q$	3) $1\wedge 2$	4) $p\to q$	5) $3\to 4$
1	1	1	1	1	1	1	1
1	1	0	0	1	0	1	1
1	0	1	1	0	0	0	1
1	0	0	0	1	0	0	1
0	1	1	1	1	1	1	1
0	1	0	1	1	1	1	1
0	0	1	1	0	0	1	1
0	0	0	1	1	1	1	1

■

3.1. Равносильность составных высказываний. Пусть A = A(P, Q, ..., Z) и B= B(P, Q, ..., Z) – составные высказывания, построенные из высказываний P, Q, ..., Z. Высказывания A и B называются ***равносильными***, что обозначается $A\equiv B$, если сопровождающие их булевы формулы равносильны (см. определение равносильности формул в разделе 2.1) Таким образом, установление равносильности высказываний сводится к проверке равносильности формул в базисе $\{\neg, \wedge, \vee, \to, \oplus, \rightleftarrows\}$. Последняя задача подробно рассмотрена в разделе 2.2.1.

Обратим особое внимание на разницу между понятиями эквивалентности и равносильности высказываний. Эквивалентность – это операция над двумя высказываниями; её результатом является новое высказывание, истинностное значение которого определяется таблицей 1-6. Равносильность – это логическое значение (истина или ложь), сопоставляемое двум составным высказываниям, зависящим от одних и тех же простых высказываний. Никакая комбинация этих двух высказываний, в отличие от эквивалентности, при этом не определяется.

Составное высказывание называется ***тождественно истинным*** (***тождественно ложным***), если сопровождающая его булева функция есть константа 1 (0). Тождественно истинные высказывания называются также ***тавтологиями***. Высказывания с таблицами истинности 9 и 10 являются тавтологиями.

3.2. Логические рассуждения и их значимость. Среди всевозможных составных высказываний особую роль играют составные высказывания специального вида, называемые ***логическими рассуждениями***. Он определяется по сопровождающей булевой формуле данного высказывания. Эта формула должна представлять собой импликацию, посылка которой (то, что стоит слева от стрелки) является конъюнкцией отдельных подформул произвольного вида (которым в исходном высказывании соответствуют предположения, правила, законы, наблюдения и т.д.). Эти отдельные подформулы называются ***предпосылками*** логического рассуждения. Подформула произвольного вида, являющаяся заключением импликации (то, что стоит справа от стрелки) называется ***выводом*** логического рассуждения. Таким образом, логическое рассуждение можно условно изобразить в следующем виде:

$$(\;P_1\wedge P_2\wedge\ldots\ldots\wedge P_k\;)\to C \qquad (28)$$

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

(*предпосылка* 1 и *предпосылка* 2 и…….и *предпосылка k*) влекут *вывод*

Как предпосылки, так и вывод могут быть достаточно сложными формулами, построенными по составными высказываниями ■

Пример 13. Формулы (21б), (23б), (24б), (26б), (27б) сопровождают высказывания, которые в силу определения являются логическими рассуждениями. Приведём все эти высказывания, для бóльшей наглядности, вместе:

1. «*Если будет солнечно и температура превысит 25^0, я приеду поездом или автомобилем*»

2. «***Если я – Ваша женщина и Вы – мой мужчина, то я никогда не перестану любить Вас. Я перестала любить Вас. Значит, я – не Ваша женщина или Вы – не мой мужчина***»

3. «*Женат я или не женат, я счастлив*»

4. «*Если пол грязный, то я должен вымыть его. Пол грязный. Поэтому я должен вымыть его*»

5. «*Все мужчины созданы равными. Все люди, созданные равными, являются женщинами. Поэтому все мужчины являются женщинами*» ■

Пример 14. Рассмотрим высказывание: «***Если Вы хотите улучшить свою сердечно-сосудистую систему, то стоит кататься на беговых (не горных) лыжах. Вы катаетесь на беговых лыжах. Значит, Вы хотите улучшить свою сердечно-сосудистую систему***». В этом составном высказывании выделяются два простых высказывания

P = «***Вы хотите улучшить свою сердечно-сосудистую систему***»,

Q = «***Вы катаетесь на беговых лыжах***».

Сопровождающая функция равна $((p\to q)\wedge q)\to p$. Эта функция является импликацией, а её посылка является конъюнкцией двух предпосылок: $p\to q$ и q. Таким образом, исходное высказывание является логическим рассуждением ■

Пример 15. Вернёмся к высказыванию из примера 10: ***«Если я – Ваша женщина и Вы – мой мужчина, то я никогда не перестану любить Вас. Я перестала любить Вас»***. Её сопровождающая формула (22б) равна $((p\wedge q)\to r)\wedge\neg r$. Эта формула не представлена в виде (28), и, следовательно, это высказывание не является логическим рассуждением ■

Важным понятием является понятие ***значимости*** логического рассуждения. Рассуждение называется значимым, если из истинности всех предпосылок обязательно следует истинность вывода. Проверка значимости осуществляется следующим образом.

Алгоритм проверки значимости логического рассуждения

1. Представить рассуждение, обозначая символами отдельные входящие в предпосылки и в вывод простые (т.е. исходные) высказывания, в формальном виде, как это делалось в примерах 1-3 – 1-16.

2. Построить по формальному представлению высказывания булеву функцию, как это делалось в конструкции, описанной в начале настоящего раздела 3. Если полученная функция не представлена в виде (28), то исходное высказывание просто не является логическим рассуждением и далее на предмет значимости не рассматривается.

3. Составить таблицу истинности для данной булевой функции, как это делалось в примерах 3, 4, 10 – 12. Если функция окажется тождественно равной 1, то исходное рассуждение является значимым. В противном случае оно не значимо ■

Указанный алгоритм даёт несколько больше, чем просто проверка значимости. Логика рассуждения нарушается как раз на тех наборах переменных, на которых функция равна 0. Поскольку импликация равна 0 тогда и только тогда, когда посылка истинна, а заключение ложно, то это и значит, что для этих наборов заключение не следует из посылки. Это и есть нарушение логики.

Приведём примеры значимых и не значимых логических рассуждений.

Пример 16. Для всех логических рассуждений из примера 15 сопровождающая функция равна 1 на всех наборах, т.е. все эти логические рассуждения значимы ■

Пример 17. Рассмотрим следующее высказывание: «***Если я видел дальше, чем другие, то потому, что я стоял на плечах гигантов (Ньютон). Я не видел дальше, чем другие. Следовательно, я не стоял на плечах гигантов***». Его первая часть (до 1-ой точки) рассмотрена в примере 1-11. Эта часть представлена формально в виде $P\to Q$, где

P = «***Я стоял на плечах гигантов***»,

Q = «***Я видел дальше других***».

Рассматриваемое высказывание представляется в виде $((P\to Q)\wedge\neg Q)\to\neg P$ при тех же самых P и Q. Сопровождающая его формула $((p\to q)\wedge\neg q)\to\neg p$ записывается в виде (28):

↑↑↑↑↑ $((p\to q)$ $\quad\wedge\neg q)$ $\qquad\to\neg p$

предпосылка 1 и предпосылка 2 влекут вывод

т.е. рассматриваемое высказывание является логическим рассуждением.

Построим таблицу истинности для данной булевой функции. Все необходимые шаги указаны в таблице 11.

Таблица 11. Построение таблицы истинности для функции $((p\to q)\wedge\neg q)\to\neg p$

p	q	1) $p\to q$	2) $\neg q$	3) $1\wedge 2$	4) $\neg p$	5) $3\to 4$
0	0	1	1	1	1	1
0	1	1	0	0	1	1
1	0	0	1	0	0	1
1	1	1	0	0	0	1

Так как последний столбец содержит только единицы, то рассматриваемое логическое рассуждение значимо ■

Пример 18. Рассмотрим логическое рассуждение из примера 14: «***Если Вы хотите улучшить свою сердечно-сосудистую систему, то стоит кататься на беговых (не горных) лыжах. Вы катаетесь на беговых лыжах. Значит, Вы хотите улучшить свою сердечно-сосудистую систему***». Его сопровождающая функция равна $((p\to q)\wedge q)\to p$. В таблице истинности для данной функции (см. таблицу 12) последний (самый правый) столбец содержит не только единицы, но и нули. Это по определению и означает, что данное рассуждение не значимо. Более того, по той строчке, в которой в последнем столбце стоит 0, можно понять, где именно «ломается» логика. В данном случае это происходит при $p = 0$ и $q = 1$. Другими словами, Вы не хотите улучшать Вашу сердечно-сосудистую систему (да просто Вас здоровье, к счастью, не беспокоит), а на лыжах катаетесь для удовольствия!

Таблица 12. Построение таблицы истинности для функции $((p\to q)\wedge q)\to p$

p	q	1) $p\to q$	2) $1\wedge q$	3) $2\to p$
0	0	1	0	1
0	1	1	1	0
1	0	0	0	1
1	1	1	1	1

■

Заметим, что значимость и истинность рассуждения – понятия разные. Значимое рассуждение обязано быть истинным высказыванием при любых истинностных значениях входящих в него простых высказываний, а истинное высказывание вообще не обязано иметь специальный вид логического рассуждения. И в любом случае истинность любого составного высказывания определяется истинностью его простых составляющих, которая вообще не обсуждается. В частности, значимым логическим рассуждением является высказывание 5 из примера 13, вывод которого утверждает, что все мужчины являются женщинами. Но оно действительно следует из имеющихся предпосылок (все мужчины рождаются равными и др,), смысл которых не обсуждается.

4. Булевы функции для описания систем голосования

В этом разделе будет описано использование булевых функций для формализации различных правил голосования в небольших группах (совет директоров, правление банка и пр.). Приведём пример рассматриваемой ситуации и возникающей в ней формальной задачи.

Пример 19. Комитет состоит из четырёх членов и председателя. Решения принимаются большинством голосов, однако, если председатель голосует «против», то решение не принимается. Требуется построить булеву функцию, зависящую от 5 переменных: x_1, x_2, x_3, x_4, y ($x_i = 1$ тогда и только тогда, когда i-ый член комитета голосует «за» ($i = 1, 2, 3, 4$), $y = 1$ тогда и только тогда, когда председатель голосует «за»). Предполагается, что значение этой булевой функции на некотором наборе x_1, x_2, x_3, x_4, y равно 1 тогда и только тогда, когда в результате голосования, соответствующего этому набору, решение принимается. Можно считать, что такая функция (если она существует) описывает данную схему голосования. Её можно назвать ***функцией голосования*** ■

Из самой постановки вопроса ясно, что искомая функция является ***неубывающей*** (говорят также, ***монотонно неубывающей***). Действительно, пусть при некотором наборе голосов x_1, x_2, x_3, x_4, yрешение принимается, т.е. $f(x_1, x_2, x_3, x_4, y) = 1$. Если бы кто-то из тех, кто голосовал «против», проголосовал «за», решение тем более должно быть принято. Формально неубывающая функция $f(z_1, \ldots, z_n)$ характеризуется условием:

$$\boldsymbol{z_i'} \le \boldsymbol{z_i''}\ (i = 1, 2, \ldots, n) \rightarrow f(\boldsymbol{z_1'}, \ldots, \boldsymbol{z_n'}) \le f(\boldsymbol{z_1''}, \ldots, \boldsymbol{z_n''}). \qquad (29)$$

Будем искать функцию голосования в виде реализующей её ДНФ, в которую все переменные входят без отрицания. Существование такой ДНФ для произвольной неубывающей булевой функции является хорошо известным фактом, который здесь не доказывается.

Алгоритм построения функции голосования.

1. Составить таблицу типа таблицы 1 с заданным числом переменных (т.е. участников голосования). В самом левом столбце для удобства записываются числа, двоичным разложением которых служит набор нулей и единиц в данной строке. Для каждого набора значений переменных в правый столбец записать 1, если по исходному словесному описанию схемы голосования, при голосовании участников, соответствующем данному набору, решение принимается. В противном случае записать туда 0. Разумеется, выполнение этого шага требует содержательного анализа рассматриваемой схемы голосования, т.е. шаг не является чисто формальным.

2. Удалить из таблицы все строчки, в правой части которых записан ноль, а из оставшейся таблицы удалить правый столбец, содержащий (после удаления строк) одни единицы. В результате выполнения этого шага останутся строчки, соответствующие голосованиям, при которых решения принимаются.

3. Самый сложный шаг. Просматривая последовательно все строки таблицы, удаляем i-ую строку, если для неё существует другая строка (среди ещё не удалённых), которая содержит единицы только на тех местах, на которых есть единицы и в i-ой строке. В результате остаются все минимальные голосования, при которых решения принимаются. Минимальность означает, что при изменении позиции <u>любого одного участника</u> с «за» на «против» решение не будет принято.

4. Каждой оставшейся строке соответствует конъюнкция, содержащая те и только те переменные (без отрицаний), номера которых совпадают с номерами позиций, на которых в данной строке стоят единицы. Например, строке 01101 соответствует конъюнкция x_2x_3y (если последняя переменная y сопоставлена председателю).

5. Образуем дизъюнкцию всех полученных на предыдущем шаге конъюнкций. Полученная формула реализует искомую функцию голосования ■

Пример 20. Применим алгоритм для схемы голосования, описанной в примере 19. Составим таблицу 13 в соответствии с шагом 1 алгоритма. Сначала часть таблицы, состоящая из столбцов с заголовками x_1, x_2, x_3, x_4, y заполняется в лексикографическом порядке (см. формулу (1)). Затем делается самое главное – заполняется столбец со значениями функции голосования f. По условию, если председатель против, то решение не принимается. Поэтому во всех строках, где $y = 0$, в столбец f записываем 0, независимо от значений всех других переменных. Поэтому 0 будет в столбце fво всех строках с чётными номерами. В остальных строках, в которых $y = 1$, используется правило большинства, т.е. 1 записывается в последнюю позицию во всех строках, в которых число единиц больше числа нулей, т.е. рано 3, 4 или 5 (а не 2, 1 или 0).

Шаг 2. Для большей ясности сначала подсветим удаляемые строки и столбец (см. таблицу 14). После их удаления оставшиеся строки и столбцы образуют таблицу 15.

Шаг 3. Понятно, что строчки, в которых есть ровно три единицы, не могут быть удалены (в данной схеме с учётом правила большинства не может быть строк, в которых меньше трёх единиц). Строка 15 должна быть удалена, поскольку все единицы строки 13 находятся на тех местах, где есть единицы в строке 15. Строка 23 должна быть удалена, поскольку все единицы строки 21 находятся на тех местах, где есть единицы в строке 23. Строка 27 должна быть удалена по той же причине из-за строки 25. Строка 29 должна быть удалена по той же причине из-за строки 25. Строка 31 содержит 1 на всех позициях и, конечно, также должна быть удалена. Оставшиеся 6 строк образуют таблицу 16.

Шаг 4. В соответствии с алгоритмом по строчкам таблицы 16 образуем следующие 6 конъюнкций: x_3x_4y, x_2x_4y, x_2x_3y, x_1x_4y, x_1x_3y, x_1x_2y.

Шаг 5. В соответствии с алгоритмом образуем дизъюнкцию всех полученных на предыдущем шаге конъюнкций. Функция $f(x_1, x_2, x_3, x_4, y) = x_3x_4y \vee x_2x_4y \vee x_2x_3y \vee x_1x_4y \vee x_1x_3y \vee x_1x_2y$ и является в данном случае искомой функцией голосования ■

Таблица 13. Исходные данные для схемы голосования

N	x_1	x_2	x_3	x_4	y	*0*
00	**0**	**0**	**0**	**0**	**0**	**0**
01	**0**	**0**	**0**	**0**	**1**	**0**
02	**0**	**0**	**0**	**1**	**0**	**0**
03	**0**	**0**	**0**	**1**	**1**	**0**
04	**0**	**0**	**1**	**0**	**0**	**0**
05	**0**	**0**	**1**	**0**	**1**	**0**
06	**0**	**0**	**1**	**1**	**0**	**0**
07	**0**	**0**	**1**	**1**	**1**	**1**
08	**0**	**1**	**0**	**0**	**0**	**0**
09	**0**	**1**	**0**	**0**	**1**	**0**
10	**0**	**1**	**0**	**1**	**0**	**0**
11	**0**	**1**	**0**	**1**	**1**	**1**
12	**0**	**1**	**1**	**0**	**0**	**0**
13	**0**	**1**	**1**	**0**	**1**	**1**
14	**0**	**1**	**1**	**1**	**0**	**0**
15	**0**	**1**	**1**	**1**	**1**	**1**
16	**1**	**0**	**0**	**0**	**0**	**0**
17	**1**	**0**	**0**	**0**	**1**	**0**
18	**1**	**0**	**0**	**1**	**0**	**0**
19	**1**	**0**	**0**	**1**	**1**	**1**
20	**1**	**0**	**1**	**0**	**0**	**0**
21	**1**	**0**	**1**	**0**	**1**	**1**
22	**1**	**0**	**1**	**1**	**0**	**0**
23	**1**	**0**	**1**	**1**	**1**	**1**
24	**1**	**1**	**0**	**0**	**0**	**0**
25	**1**	**1**	**0**	**0**	**1**	**1**
26	**1**	**1**	**0**	**1**	**0**	**0**
27	**1**	**1**	**0**	**1**	**1**	**1**
28	**1**	**1**	**1**	**0**	**0**	**0**
29	**1**	**1**	**1**	**0**	**1**	**1**
30	**1**	**1**	**1**	**1**	**0**	**0**
31	**1**	**1**	**1**	**1**	**1**	**1**

Таблица 14. Удаляемые строки и столбец

N	x_1	x_2	x_3	x_4	y	*0*
00	**0**	**0**	**0**	**0**	**0**	**0**
01	**0**	**0**	**0**	**0**	**1**	**0**
02	**0**	**0**	**0**	**1**	**0**	**0**
03	**0**	**0**	**0**	**1**	**1**	**0**
04	**0**	**0**	**1**	**0**	**0**	**0**
05	**0**	**0**	**1**	**0**	**1**	**0**
06	**0**	**0**	**1**	**1**	**0**	**0**
07	**0**	**0**	**1**	**1**	**1**	**1**
08	**0**	**1**	**0**	**0**	**0**	**0**
09	**0**	**1**	**0**	**0**	**1**	**0**
10	**0**	**1**	**0**	**1**	**0**	**0**
11	**0**	**1**	**0**	**1**	**1**	**1**
12	**0**	**1**	**1**	**0**	**0**	**0**
13	**0**	**1**	**1**	**0**	**1**	**1**
14	**0**	**1**	**1**	**1**	**0**	**0**
15	**0**	**1**	**1**	**1**	**1**	**1**
16	**1**	**0**	**0**	**0**	**0**	**0**
17	**1**	**0**	**0**	**0**	**1**	**0**
18	**1**	**0**	**0**	**1**	**0**	**0**
19	**1**	**0**	**0**	**1**	**1**	**1**
20	**1**	**0**	**1**	**0**	**0**	**0**
21	**1**	**0**	**1**	**0**	**1**	**1**
22	**1**	**0**	**1**	**1**	**0**	**0**
23	**1**	**0**	**1**	**1**	**1**	**1**
24	**1**	**1**	**0**	**0**	**0**	**0**
25	**1**	**1**	**0**	**0**	**1**	**1**
26	**1**	**1**	**0**	**1**	**0**	**0**
27	**1**	**1**	**0**	**1**	**1**	**1**
28	**1**	**1**	**1**	**0**	**0**	**0**
29	**1**	**1**	**1**	**0**	**1**	**1**
30	**1**	**1**	**1**	**1**	**0**	**0**
31	**1**	**1**	**1**	**1**	**1**	**1**

Таблица 15. Все голосования, при которых принимаются решения

N	x_1	x_2	x_3	x_4	y
07	0	0	1	1	1
11	0	1	0	1	1
13	0	1	1	0	1
15	0	1	1	1	1
19	1	0	0	1	1
21	1	0	1	0	1
23	1	0	1	1	1
25	1	1	0	0	1
27	1	1	0	1	1
29	1	1	1	0	1
31	1	1	1	1	1

Таблица 16. Минимальные голосования, при которых принимаются решения

N	x_1	x_2	x_3	x_4	y
07	0	0	1	1	1
11	0	1	0	1	1
13	0	1	1	0	1
19	1	0	0	1	1
21	1	0	1	0	1
25	1	1	0	0	1

Пример 21. В правление банка входят четыре человека: председатель *A*, имеющий два голоса в своём распоряжении, и члены правления *B, C* и *D*, обладающие одним голосом каждый. Для принятия какого-либо решения при голосовании должно быть набрано хотя бы четыре голоса. Найдём функцию голосования для описанной схемы.

Сопоставим членам правления *B, C* и *D* переменные x_1, x_2 иx_3, председателю *A* – переменную y. В соответствии с описанной схемой голосования на шаге 1 заполняем таблицу 17:

Таблица 17. Исходные данные для схемы голосования

N	x_1	x_2	x_3	y	*0*
00	0	0	0	0	0
01	0	0	0	1	0
02	0	0	1	0	0
03	0	0	1	1	0
04	0	1	0	0	0
05	0	1	0	1	0
06	0	1	1	0	0
07	0	1	1	1	1

N	x_1	x_2	x_3	y	*0*
08	1	0	0	0	0
09	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	1

Легко видеть, что без участия председателя остальные участники могут набрать не более 3-ёх голосов, поэтому во всех строчках, где председатель «против», решение не будет принято. В то же время мнения одного председателя не достаточно для принятия решения (нужно ведь 4 голоса «за», а у него только два). Такие же очевидные соображения позволяют легко заполнить эту таблицу.

Далее в соответствии с алгоритмом построения функции голосования выполняем шаг 2, оставляя строчки, соответствующие принятию решения (см. таблицы 18 и 19).

Таблица 18. Удаляемые строки и столбец

N	x_1	x_2	x_3	y	*0*
00	0	0	0	0	0
01	0	0	0	1	0
02	0	0	1	0	0
03	0	0	1	1	0
04	0	1	0	0	0
05	0	1	0	1	0
06	0	1	1	0	0
07	0	1	1	1	1

N	x_1	x_2	x_3	y	*0*
08	1	0	0	0	0
09	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	1

Таблица 19. Все голосования, при которых принимаются решения

N	x_1	x_2	x_3	y
07	0	1	1	1
11	1	0	1	1
13	1	1	0	1
15	1	1	1	1

Таблица 20. Минимальные голосования, при которых принимаются решения

N	x_1	x_2	x_3	y
07	0	1	1	1
11	1	0	1	1
13	1	1	0	1

В результате шага 3 получается таблица 20 минимальных голосований, поскольку лишь набор из строки 15, содержащий только единицы, не является минимальным. Объединяя простые шаги 4 и 5, запишем ответ: функция голосования $f(x_1, x_2, x_3, y) = x_2x_3y \vee x_1x_3y \vee x_1x_2y$■

5. Код Хэмминга

В настоящем разделе рассматривается важное приложение булевых функций – ***помехоустойчивое кодирование***. Его суть такова. Имеется двоичный вектор $\alpha = (\alpha_1, \ldots, \alpha_d)$ некоторой фиксированной длины d, который необходимо передать (например, из узла А в узел В телекоммуникационный сети). Известно, что при передаче двоичного вектора возможно не более одной ошибки (т.е. символ 0 может быть воспринят после передачи как 1 или 1 как 0). Требуется разработать схему кодирования и декодирования, т.е. предложить:

1. Построение по двоичному вектору $\alpha = (\alpha_1, \ldots, \alpha_d)$ его двоичного кода $\beta = (\beta_1, \ldots, \beta_n)$.

2. Построение по полученному при передаче вектора $\beta = (\beta_1, \ldots, \beta_n)$ вектору $\overline{\boldsymbol{\beta}} = (\overline{\boldsymbol{\beta}}_1, \ldots, \overline{\boldsymbol{\beta}}_n)$ нового вектора $\overline{\boldsymbol{\alpha}} = (\overline{\boldsymbol{\alpha}}_1, \ldots, \overline{\boldsymbol{\alpha}}_d)$, такого, что $\overline{\boldsymbol{\alpha}} = \alpha$, независимо от <u>одной</u> ошибки при передаче вектора β.

Данная задача имеет много решений. Например, каждый символ α_i при передаче можно заменить тремя совпадающими символами, а при декодировании принимать решение по правилу большинства: если большинство из трёх переданных символов – 0, то $\alpha_i = 0$, в противном случае $\alpha_i = 1$. Недостатком такого рода мажоритарных схем является сильное увеличение длины кодов β (в упомянутом случае в три раза). Поэтому требуется, чтобы отношение длины кода n к длине исходного слова d было бы как можно меньше. В частности, очень желательно, чтобы с ростом dуказанное отношение стремилось бы к 1. Требуемыми свойствами обладает код Хэмминга, к описанию которого сейчас и переходим.

5.1. Понятия и обозначения. Введём необходимые понятия, обозначения и утверждения, относящиеся к двоичным векторам. Рассмотрим все двоичные векторы длины k. Расположим их в лексикографическом порядке, т.е. в порядке возрастания целых чисел, двоичными разложениями которых являются эти векторы:

000…00
000…01
000…10 (30)
………..
111…10
111…11

(см. также формулу (1)). Вектор, стоящий на i-ом сверху месте (начиная с 0) в множестве векторов (16), является двоичным разложением числа i. Обозначим его через $e^k(i)$ $(0 \le i \le 2^k - 1)$. Например, при $k = 3$ получаем

$e^3(0) = 000$, $e^3(1) = 001$, $e^3(2) = 010$, $e^3(3) = 011$, $e^3(4) = 100$, $e^3(5) = 101$, $e^3(6) = 110$, $e^3(7) = 111$ (31a)

при $k = 4$ получаем

$e^4(0) = 0000$, $e^4(1) = 0001$, $e^4(2) = 0010$, $e^4(3) = 0011$, $e^4(4) = 0100$, $e^4(5) = 0101$, $e^4(6) = 0110$, $e^4(7) = 0111$,
$e^4(8) = 1000$, $e^4(9) = 1001$, $e^4(10) = 1010$, $e^4(11) = 1011$, $e^4(12) = 1100$, $e^4(13) = 1101$, $e^4(14) = 1110$, $e^4(15) = 1111$. (31b)

Обозначим j-ую координату вектора $e^k(i)$ через $\boldsymbol{e_j^k(i)}$$(j = 1, \ldots, k)$. Здесь координаты считаются, как обычно, слева направо. Например,$e^3(3) = 011$, откуда $\boldsymbol{e_1^3(3)} = 0$, $\boldsymbol{e_2^3(3)} = \boldsymbol{e_3^3(3)} = 1$.

Особую роль в построении кода Хэмминга играют векторы $e^k(i)$ для $i = 2^m$ $(m = 0, 1, \ldots, k - 1)$. Их координаты обладают тремя почти очевидными свойствами

1. Для любого $m = 0, 1, \ldots, k - 1$ координата $\boldsymbol{e_{k-m}^k(2^m)} = 1$,
2. Для любого $p < 2^m$ координата $\boldsymbol{e_{k-m}^k(p)} = 0$.
3. Для любого $q > m$ координата $\boldsymbol{e_{k-m}^k(2^q)} = 0$.

Все три свойства непосредственно следуют из того, что двоичным разложением длины k числа 2^m является вектор 0…010…0, содержащий только одну единицу на $(k–m)$-ом месте. Действительно, так как $1 = 2^0$, то $m = 0$, $k–m = k$ и в векторе $e^k(1) = 00…01$ единица действительно стоит на последнем, т.е. k-ом, месте. Далее, так как $2 = 2^1$, то $m = 1$, $k–m = k–1$, и в векторе $e^k(2) = 00…10$ единица действительно стоит на предпоследнем, т.е. $(k–1)$-ом, месте. Аналогично, $e^k(4) = 00…100$, и т.д.

Пусть $\beta = (\beta_1, \ldots, \beta_n)$ – двоичный вектор длины n. Определим вектор $h(\beta)$ формулой

$$h(\beta) = \sum_{i=1}^{n} \boldsymbol{\beta_i e^k(i)}, \tag{32}$$

где число k является минимальным числом, удовлетворяющим неравенству

$$n < 2^k. \tag{33}$$

По построению, вектор $h(\beta)$ имеет kкоординат. Он получается сложением векторов вида $e^k(i)$, умноженных на 0 или 1. В настоящем разделе под сложением понимается сложение по модулю 2: $0\oplus0 = 1\oplus1 = 0$, $0\oplus1 = 1\oplus0 = 1$. При покоординатной записи (32) имеет вид

$$h_j(\beta) = \sum_{i=1}^{n} \boldsymbol{\beta_i e_j^k(i)} \ (j = 1, \ldots, k). \quad (34)$$

Обратим внимание, что в (32) и (34) суммирование начинается с 1, а не с 0, поскольку добавление вектора $e^k(0)$, состоящего из одних нулей, не изменит сумм (32) и (34) при любом β_0. Из закона 3) дистрибутивности для функции $\oplus$ (см. раздел 2) и формулы (32) непосредственно следует, что для любых двух двоичных векторов β^1 и β^2

$$h(\beta^1\oplus\beta^2) = h(\beta^1) \oplus h(\beta^2). \quad (35)$$

5.2. Схема кодирования и декодирования. Прежде всего по заданной длине d исходных слов $\alpha = (\alpha_1, \ldots, \alpha_d)$ определим число k как минимальное число, для которого $d + k< 2^k$. Положим $n = d + k$. Так как $n< 2^k$, то любое число от 1 до $d + k$ имеет двоичное разложение длины k (см. векторы (30) и (31)). В таблицу 21 вписаны значения k и n для нескольких небольших d. Из таблицы ясен характер отношения d/n: с ростом d оно стремится к 1.

5.2.1. Кодирование. Определим кодовый вектор $\beta = (\beta_1, \ldots, \beta_n)$ по исходному слову $\alpha = (\alpha_1, \ldots, \alpha_d)$ следующим образом. Занумеруем в порядке возрастания все координаты вектора β,

Таблица 15. Параметры кодирования

d	***k***	***n=d+k***	**2^k**
3	**3**	**6**	**8**
4	**3**	**7**	**8**
5	**4**	**9**	**16**
6	**4**	**10**	**16**
11	**4**	**15**	**16**
26	**5**	**31**	**32**
57	**6**	**63**	**64**

номера которых не являются степенями двойки: $\beta_3, \beta_5, \beta_6, \beta_7, \beta_9, \ldots$. По конструкции, число таких координат в точности равно числу d. Далее, положим

$$\beta_3 = \alpha_1, \beta_5 = \alpha_2, \beta_6 = \alpha_3, \quad (36)$$

и т.д. В результате все координаты, номера которых не являются степенями двойки, оказываются «заполненными» координатами исходного вектора α. Указанные координаты вектора β называются ***информационными*** (или ***информационными разрядами***), как раз потому, что они содержат подлежащую передаче информацию. Остальные разряды, номера которых равны степеням двойки: 1, 2, 4, …, называются ***контрольными***.

Пример 22. Рассмотрим исходный набор $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ длины 4 (d= 4). Прежде всего определим число k из условия $d + k< 2^k$. Имеем $4 + 2 > 2^2$, $4 + 3 < 2^3$. Поэтому k= 3, $n = d + k = 7$ (см. таблицу 21). В векторе $\beta = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7)$ координаты 3, 5, 6 и 7 являются информационными. После присвоения им значений $\alpha_1, \alpha_2, \alpha_3, \alpha_4$) получаем $\beta = (\beta_1, \beta_2,\alpha_1, \beta_4,\alpha_2,\alpha_3,\alpha_4)$ ■

Определим теперь контрольные разряды $\beta_1, \beta_2, \beta_4, \ldots$ формулой:

$$\beta_{2^m} = \sum_{i=1, i\neq 2^m}^{n} \beta_i e_{k-m}^k(i) \ (m = 1, \ldots, k). \quad (37)$$

В силу указанных выше свойств векторов $e^k(i)$ для $i = 2^m$ в сумму (37) входят только те слагаемые β_i, для которых

1) номер $i>2^m$ (все ме́ньшие координаты равны 0 по свойству 2);

2) номер $i \neq 2^q$ для всех $q>m$ (см. свойство 3).

Учитывая, что информационные разряды β_i – это как раз те, для которых $i \neq 2^q$ для любых q, формула (37) выражает контрольные разряды только через информационные. Другими словами, формулы (36) и (37) вместе полностью определяют кодирование, поскольку выражают кодовый вектор $\beta = (\beta_1, \beta_2,\ldots, \beta_n)$ через исходный вектор $\alpha = (\alpha_1, \ldots, \alpha_d)$. Таким образом, **алгоритм кодирования** состоит всего из двух шагов.

1. По исходному слову α определяются информационные разряды вектора β по формуле (36).
2. По информационным разрядам вектора β определяются его контрольные разряды по формуле (37).

Пример 23. Построим в явном виде кодовый вектор $\beta = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7)$ по заданному вектору $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ длины 4. Поскольку при d= 4 число k= 3 (см. пример 21), то можно записать в явном виде (см. формулы (37)):

$\beta_1 = \beta_3 \oplus \beta_5 \oplus \beta_7$;

$\beta_2 = \beta_3 \oplus \beta_6 \oplus \beta_7$;

$\beta_4 = \beta_5 \oplus \beta_6 \oplus \beta_7$.

Поскольку в данном случае по построению $\beta_3 = \alpha_1$, $\beta_5 = \alpha_2$, $\beta_6 = \alpha_3$, $\beta_7 = \alpha_4$, то вместе с предыдущими равенствами получаем, что все координаты вектора $\beta = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7)$ выражаются (притом достаточно просто) через координаты $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ ■

Нетрудно понять и общую закономерность. При произвольном d и соответствующим ему k (напомним, что k – минимальное число, удовлетворяющее неравенству $d + k < 2^k$) точно также получим

$\beta_1 = \beta_3 \oplus \beta_5 \oplus \beta_7 \oplus \ldots$ и далее через 1 вплоть до $n = d + k$,

$\beta_2 = \beta_3 \oplus \beta_6 \oplus \beta_7 \oplus \beta_{10} \oplus \beta_{11} \oplus \ldots$ и далее через 2 вплоть до $n = d + k$,

$\beta_4 = \beta_5 \oplus \beta_6 \oplus \beta_7 \oplus \beta_{12} \oplus \beta_{13} \oplus \beta_{14} \oplus \beta_{15} \oplus \ldots$ и далее через 4 вплоть до $n = d + k$, (38)

……………………………………………………………………………………..

……………………………………………………………………………………..

Для кода Хемминга имеет место следующее

Утверждение 5. Для кодового вектора $\beta = (\beta_1, \ldots, \beta_n)$, в котором информационные разряды заполнены произвольно, а контрольные выражаются через них по формулам (37), верно векторное равенство

$h(\beta) = \mathbf{0}$, (39)

где $h(\beta)$ определяется формулой (32)

Действительно, если к обеим частям равенства (37) прибавить его левую часть $\boldsymbol{\beta_{2^m}}$, справа получится сумма (32), а слева 0, в силу очевидного тождества А⊕А = 0 при любом А ■

Пример 24. Продолжение примера 23. Положим α = 1011. Определим кодовый вектор β формулами из примера 11:

$\beta_1 = \beta_3 \oplus \beta_5 \oplus \beta_7 = 1 \oplus 0 \oplus 1 = 0$;

$\beta_2 = \beta_3 \oplus \beta_6 \oplus \beta_7 = 1 \oplus 1 \oplus 1 = 1$;

$\beta_4 = \beta_5 \oplus \beta_6 \oplus \beta_7 = 0 \oplus 1 \oplus 1 = 0$.

Таким образом, β = 0110011. Имеем (см. 18) и (17a))

$h(\beta) = 0 \times 001 \oplus 1 \times 010 \oplus 1 \times 011 \oplus 0 \times 100 \oplus 0 \times 101 \oplus 1 \times 110 \oplus 1 \times 111 = 010 \oplus 011 \oplus 110 \oplus 111 = 000 = \mathbf{0}$,

(40)

в соответствии с утверждением 5.

5.2.2. Декодирование. Передаётся кодовый вектор $\beta = (\beta_1, \ldots, \beta_n)$, определяемый по входному слову $\alpha = (\alpha_1, \ldots, \alpha_d)$ формулами (36) для информационных разрядов и (37) для контрольных разрядов. В результате получен вектор $\overline{\boldsymbol{\beta}} = (\overline{\boldsymbol{\beta}}_1, \ldots, \overline{\boldsymbol{\beta}}_n)$, отличающийся от β не более, чем в одном разряде. В основе декодирования, т.е. построения по вектору $\overline{\boldsymbol{\beta}}$ переданного вектора β, лежит следующее

Утверждение 6. Если вектор $h(\overline{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \overline{\boldsymbol{\beta}}_i \boldsymbol{e}^{\boldsymbol{k}}(\boldsymbol{i})$ (см. формулу (32)) не равен $\mathbf{0}$, то он является двоичным разложением числа, равного номеру той единственной координаты вектора $\overline{\boldsymbol{\beta}} = (\overline{\boldsymbol{\beta}}_1, \ldots, \overline{\boldsymbol{\beta}}_n)$, которая была искажена при передаче кодового вектора $\beta = (\beta_1, \ldots, \beta_n)$. Если же $h(\overline{\boldsymbol{\beta}}) = \mathbf{0}$, то ошибки при передаче не произошло, т.е. $\overline{\boldsymbol{\beta}} = \beta$.

Доказательство. Предположим, что при передаче вектора $\beta = (\beta_1, \ldots, \beta_n)$ получен вектор $\overline{\boldsymbol{\beta}} = (\overline{\boldsymbol{\beta}}_1, \ldots, \overline{\boldsymbol{\beta}}_n)$, и при этом произошло искажение в одном разряде: вместо числа β_s было передано ошибочное число $\beta_s \oplus 1$ (0 вместо 1 или 1 вместо 0). Определим вектор δ, все координаты которого, кроме s-ой, равны 0, а s-ая координата $\delta_s = 1$. Тогда $\overline{\boldsymbol{\beta}} = \beta + \delta$. В силу линейности функции h и равенства (39) имеем $h(\overline{\boldsymbol{\beta}}) = h(\beta) \oplus h(\delta) = h(\delta) = e^k(s)$. По построению, вектор $e^k(s)$ представляет собой двоичное разложение числа s длины k, что и доказывает утверждение в случае $h(\overline{\boldsymbol{\beta}}) \neq \mathbf{0}$. В случае $h(\overline{\boldsymbol{\beta}}) = \mathbf{0}$ имеем $\overline{\boldsymbol{\beta}} = \beta$, поскольку они могут отличаться только в одном разряде, что и доказывает утверждение в случае $h(\overline{\boldsymbol{\beta}}) = \mathbf{0}$■

Доказательство утверждения 6 фактически содержит в себе

Алгоритм декодирования кодов Хэмминга.

1. По вектору $\overline{\boldsymbol{\beta}} = (\overline{\boldsymbol{\beta}}_1, \ldots, \overline{\boldsymbol{\beta}}_n)$ и формуле (32) определяем вектор $h(\overline{\boldsymbol{\beta}})$.
2. Если $h(\overline{\boldsymbol{\beta}}) = \mathbf{0}$, то переходим к шагу 4; в противном случае – к шагу 3.
3. По вектору $e^k(s) = h(\overline{\boldsymbol{\beta}}) \neq \mathbf{0}$ определяем число s, двоичным разложением которого является $e^k(s)$, и изменяем s-ую координату вектора $\overline{\boldsymbol{\beta}}$ на противоположное значение.
4. Вектор $\overline{\boldsymbol{\beta}}$ совпадает с передающимся вектором β. Поэтому компоненты β, находящиеся на местах 1, 3, 5, 6, и т.д., не являющимися степенями 2, как раз и образуют исходное слово $\alpha = (\alpha_1, \ldots, \alpha_d)$ ■

Пример 25. Продолжение примера 24. При передаче кода $\beta = 0110011$ произошёл сбой в 3-ьем разряде, т.е. оказался получен вектор $\overline{\boldsymbol{\beta}} = 0100011$ вместо $\beta = 0110011$. В соответствии с алгоритмом декодирования прежде всего подсчитаем вектор $h(\overline{\boldsymbol{\beta}})$. Имеем (см. для сравнения формулу (40)):

$$h(\overline{\boldsymbol{\beta}}) = 0\times001\oplus1\times010\oplus0\times011\oplus0\times100\oplus0\times101\oplus1\times110\oplus1\times111 = 010\oplus110\oplus111 = 011 = 3,$$

что и означает – в силу утверждения 6 – что ошибка действительна произошла в 3-ем разряде. Заменяя 0 в 3-ьем разряде на 1, получаем из ошибочного кода $\overline{\boldsymbol{\beta}} = 0100011$ правильный код$\beta =$ 0110011 ■

Пример 26. Пусть получен вектор $\overline{\boldsymbol{\beta}} = 01001001$. Так как его длина $n = 9$, то из таблицы 15 видно, что $k = 4$ и длина d исходного сообщения равна 5. Воспользуемся алгоритмом декодирования. Получаем (см. (17b) для векторов $\mathrm{e}^4(i)$):

$h(\overline{\boldsymbol{\beta}}) = 0010 \oplus 0101 \oplus 1001 = 1110 = 14$

(в сумму вошли те слагаемые, которые соответствуют координатам $\overline{\boldsymbol{\beta}}$, равным 1). Это означает, что сбой произошёл в 14-ом разряде. Но 14-го разряда в данном случае просто нет – вектор $\overline{\boldsymbol{\beta}}$ содержит только 9 координат. В чём же ошибка?

Никакой ошибки нет. Алгоритм декодирования относится только к двоичным векторам, полученных при кодировании методом Хэмминга, или отличающимся от них только в одном разряде. Это означает, что данный вектор 01001001 таковым не является. Если взять все вектора α длины 4, закодировать их описанным в разделе 4.2.1 методом и затем ещё менять все получившиеся векторы ровно в одном разряде, мы никогда не получим данного вектора ■

6. Задания

Задание 1. Для заданной формулы составить её таблицу истинности. См. таблицы 6 – 12 для образца.

Варианты формул для задания 1:

01. $\neg p\wedge(q\to\neg r)$	11. $(\neg r\to\neg q)\oplus\vee(\neg p\wedge q)$	21. $(p\oplus q)\to r$
02. $(\neg r\rightleftarrows\neg q)\to(\neg p\wedge q)$	12. $(p\wedge r)\rightleftarrows\neg q$	22. $((p\wedge q)\to r)\wedge\neg r$
03. $(p\to r\vee\neg q$	13. $(q\to r)\wedge p$	23. $(p\to q)\rightleftarrows r$
04. $(q\vee\neg r)\oplus p$	14. $(\neg p\wedge q)\oplus\neg r$	24. $p\to(q\oplus r)$
05. $(\neg p\to q)\rightleftarrows\vee\neg r$	15. $\neg(p\to q)\wedge(r\vee\neg q)$	25. $(p\to\neg q)\to r$
06. $\neg(p\wedge q)\wedge(r\to\neg q)$	16. $\neg((\neg p\rightleftarrows q)\to r)$	26. $p\rightleftarrows(\neg q\to r)$
07. $\neg((\neg p\to q)\oplus r)$	17. $\neg(r\to(\neg p\wedge\neg q))$	27. $(p\oplus\neg q)\to\neg r$
08. $\neg(r\vee(\neg p\to\neg q))$	18. $(\neg p\wedge\neg q)\vee(\neg r\oplus\neg p)$	28. $p\neg q\rightleftarrows q\neg r\vee r\neg p$
09. $(\neg p\to\neg q)\vee(\neg r\wedge\rightleftarrows\neg p)$	19. $(p\rightleftarrows\neg p)\to q$	29 $pq\vee q\neg r\vee r\neg p$
10.$\neg p\to(q\vee\neg r)$	20. $((p\to q)\wedge p)\to q$	30 $(p\neg q\oplus q\neg r)\to pq$■

Задание 2. Представить заданную формулу в базисе $\{0, 1, \neg, \wedge, \vee\}$, выражая другие функции через дизъюнкцию, конъюнкцию и отрицание с помощью тождеств (2) – (7). Далее не упрощать! См. пример 7а для образца.

Варианты задания те же, что в задании 1.■

Задание 3. Формулу в базисе $\{0, 1, \neg, \wedge, \vee\}$, полученную при выполнении задания 2, представить в виде ДНФ. См. пример 7б для образца■

Задание 4. ДНФ, полученную при выполнении задания 3, преобразовать в СДНФ. См. пример 7в для образца■

Задание 5. Для данной истории, рассматриваемой как составное высказывание, исходя из её формального представления, полученного при выполнении в задании 1-3, построить сопровождающую формулу. См. пример 9 для образца■

Задание 6. Для сопровождающей формулы, полученной при выполнении задания 5, построить таблицу истинности. См. таблицы 6 – 12 для образца■

Задание 7. Используя таблицу истинности, проверить значимость данного логического рассуждения. Если оно не значимо, указать, в каких ситуациях его вывод не следует из предпосылок. См. примеры 17 и 18 для образца.

Варианты для заданий 5 – 7:

01. Если это дерево заражено сосновым короедом, то оно умрёт. Люди посадили дерево в День Дерева (американский праздник) и оно не умрёт. Поэтому, если люди посадили дерево в День Дерева, то оно не заражено сосновым короедом.

02. Кристина Агилера поёт или Рики Мартин не является подростковым идолом. Если Рики Мартин не является подростковым идолом, то Бритни Спирс не выиграет Американскую музыкальную премию. Бритни Спирс выиграла Американскую музыкальную премию. Следовательно, Кристина Агилера не поёт.

03. Если Отис диск-жокей, то он живёт в Лексингтоне. Он живёт в Лексингтоне и он сдвинут на истории. Следовательно, если Отис не сдвинут на истории, то он не диск-жокей.

04. Я покупаю машину или я уезжаю на каникулы. Я не покупаю машину, следовательно, я уезжаю на каникулы.

05. Если бы человек мог быть в двух местах одновременно, я был бы с Вами. Я не был с Вами. Следовательно, человек не может быть в двух местах одновременно.

06. «Если мы будем следовать по стопам Ньютона, это не будет прогрессом» (Алдос Хаксли). Мы не следуем по стопам Ньютона. Значит, это прогресс».

07. Если Эдди едет в город, то Мейбл останется дома. Если Мейбл не останется дома, то Рита будет готовить. Рита не готовит. Поэтому Эдди не едет в город.

08. Если я чувствую Вас всей кожей, то Вы глубоко в моём сердце. Если Вы глубоко в моём сердце, то Вы – не часть меня. Вы глубоко в моём сердце или Вы часть меня. Поэтому, если я чувствую Вас всей кожей, то Вы – часть меня.

09. Если я подпишу чек, он не будет принят. Если банк гарантирует его, он будет принят. Банк гарантирует его. Поэтому я не подпишу чек.

10. Джефф любит играть в гольф. Если Джоан любит шить, то Джефф не любит играть в гольф. Если Джоан не любит шить, то Брэд поёт в хоре. Поэтому Брэд поёт в хоре.

11. Если Канада – союзник Соединённых Штатов, то Соединённые Штаты относятся к Канаде дружески. Соединённые Штаты относятся к Канаде дружески. Поэтому Канада – союзник Соединённых Штатов.

12. Если Вы спите во время утренних занятий по математике, то Вы хорошо отдохнёте. Если Вы хорошо отдохнёте, то Вы хорошо сдадите тест по математике. Поэтому, если Вы спите во время утренних занятий по математике, то Вы хорошо сдадите тест по математике.

13. Если мы сбалансируем бюджет или уменьшим налоги, то будет больше денег для борьбы с загрязнением. Если мы не сбалансируем бюджет, то мы не уменьшим налоги. Мы не уменьшим налоги. Поэтому будет больше денег для борьбы с загрязнением.

14. Если Дэйв одинок сегодня вечером, то он не пойдёт в гости. Если Дэйв не одинок сегодня вечером или он не пойдёт в гости, то он будет писать вечером свою курсовую работу. Дэйв не будет писать вечером свою курсовую работу. Значит, Дэйв не одинок сегодня вечером.

15. Если тест отрицателен, то Вы не нуждаетесь в лечении. Если тест положителен, то Вы нуждаетесь в лекарствах. Вы не нуждаетесь в лекарствах. Значит, Вы не нуждаетесь в лечении ■

Задание 8. Построить булеву функцию голосования для данной схемы голосования. См. примеры 20 и 21 для образца.

Варианты для задания 8:

01. Комитет состоит из пяти членов. Решения принимаются большинством голосов, однако, если председатель голосует «против», то решение не принимается.

02. В комитете, состоящим из трёх членов *B*, *C*, *D* и председателя *A*, решения принимаются большинством голосов, а в случае равенства голосов голос председателя является решающим.

03. Совет директоров фирмы состоит из четырёх человек. Глава совета *A* обладает при голосовании тремя голосами, члены *B* и *C* – двумя голосами каждый, а член совета *D* – одним голосом. Для принятия решения необходимо набрать не менее 6 голосов.

04. Совет директоров фирмы состоит из четырёх человек. Глава совета A обладает при голосовании тремя голосами, его заместитель B – двумя голосами, члены C и D – одним голосом каждый. Для принятия решения необходимо набрать не менее 6 голосов.
05. Совет директоров фирмы состоит из четырёх человек. Глава совета A обладает при голосовании тремя голосами, его заместитель B – двумя голосами, члены C и D – одним голосом каждый. Для принятия решения необходимо набрать не менее 5 голосов.
06. Совет директоров фирмы состоит из четырёх человек. Глава совета A обладает при голосовании тремя голосами, его заместитель B – двумя голосами, члены C и D – одним голосом каждый. Для принятия решения необходимо набрать не менее 4 голосов.
07. Совет директоров фирмы состоит из четырёх человек. Глава совета A обладает при голосовании тремя голосами, члены B и C – двумя голосами каждый, а член совета D – одним голосом. Для принятия решения необходимо набрать не менее 5 голосов.
08. В правление банка входят четыре человека: председатель A, имеющий два голоса в своём рас поряжении, и члены правления B, C и D, обладающие одним голосом каждый. Для принятия какого-либо решения при голосовании должно быть набрано хотя бы три голоса.
09. В правление банка входят четыре человека: председатель A, имеющий два голоса в своём распоряжении, и члены правления B, C и D, обладающие одним голосом каждый. Для принятия какого-либо решения при голосовании должно быть набрано хотя бы четыре голоса.
10. Совет директоров фирмы состоит из четырёх человек. Глава совета A обладает при голосовании тремя голосами, члены B и C – двумя голосами каждый, а член совета D – одним голосом. Решение принимается большинством голосов, а в случае равенства голос члена D является решающим.
11. Совет директоров фирмы состоит из четырёх человек. Глава совета A обладает при голосовании тремя голосами, члены B и C – двумя голосами каждый, а член совета D – одним голосом. Решение принимается большинством голосов, а в случае равенства голос члена B является решающим.
12. Совет директоров фирмы состоит из четырёх человек. Глава совета A обладает при голосовании тремя голосами, его заместитель B – двумя голосами, члены C и D – одним голосом каждый. Решение принимается большинством голосов, но не принимается, если глава совета против.
13. Совет директоров фирмы состоит из четырёх человек. Глава совета A обладает при голосовании тремя голосами, члены B и C – двумя голосами каждый. члены D – одним голосом. Решение принимается большинством голосов, а в случае равенства голос председателя является решающим ■

Задание 9. По данному входному слову построить его код Хемминга. См. примеры 22 – 24 для образца.

Варианты входных слов для задания 9

01) 0111	11) 010	21) 10101	31) 0001
02) 1100	12) 111	22) 11010	32) 0110
03) 001	13) 10100	23) 0000	33) 1011
04) 110	14) 11001	24) 0101	34) 0101
05) 10011	15) 01110	25) 1010	35) 000
06) 01000	16) 0100	26) 1111	36) 101
07) 01101	17) 1001	27) 100	37) 00010
08) 0011	18) 1110	28) 00001	38) 00111
09) 1000	19) 011	29) 00110	39) 01100
10) 1101	20) 10000	30) 11011	40) 0010■

Задание 10. По данному полученному слову восстановить входное сообщение или убедиться, что полученное слово не является кодовым словом. См. примеры 25, 26 для образца.

Варианты полученных слов для задания 10:

01) 1110011	06) 001001110	11) 011101110	16) 0110001
02) 111001110	07) 011001010	12) 0110111	17) 011000110
03) 111001110	08) 0100011	13) 1110001	18) 0110010
04) 0010011	09) 010001110	14) 011011110	19) 011001010
05) 011001100	10) 0111011	15) 011001111	20) 0010111■

7. Предметный указатель

Литература к части 1

1. Бурбаки Н. Теория множеств. М.: Мир, 1965.
2. Гендлер М.Б., Прокопчук Ю.Ю. Конечная математика. Пособие для практических занятий. // Под ред. дейст. члена АН СССР Емельянова С.В. – М.: ВИНИТИ, 1988. − 112 с.
3. Дехтярь М.И. Лекции по дискретной математике. – М.: Интернет-Университет Информацион-ных Технологий; БИНОМ. Лаборатория знаний, 2009, 259 с. ISBN 978-5-94774-714-0 (БИНОМ ЛЗ).
4. Кузнецов О.П. Дискретная математика для инженера. − 5-е изд., стер. – СПб: Изд-во «Лань», 2007. – 400 с.
5. Шиханович Ю.А. Введение в современную математику. Начальные понятия. – М.: Наука. Гл. ред. физ.-мат. лит., 1965. – 376 с.
6. Шоломов Л.А. Основы теории дискретных логических и вычислительных устройств. – М.: Наука. Гл. ред. физ.-мат. лит., 1980. – 400 с.

Часть 2. ОПТИМИЗАЦИЯ НА ГРАФАХ

Под дискретной оптимизацией понимается оптимизация числовых функций, заданных на конечных множествах. В терминологии раздела 3-5 такие функции называются функциями типа $A \to R$, где A – конечное множество, R – множество всех вещественных чисел. Трудно рассчитывать на какую-либо теорию для столь широкого класса задач. Поэтому обычно рассматриваются специальные подклассы задач, которые, с одной стороны, вызывают практический интерес, и, с другой стороны, поддаются решению при реальных размерностях.

Среди разнообразных задач дискретной оптимизации выделяются задачи оптимизации на графах. Во-первых, графами моделируются многие разнообразные реальные системы, в силу чего задачи оптимизации таких систем представляются именно как задачи оптимизации на графах. Во-вторых, специфические свойства и геометрическая наглядность многих объектов, связанных с графами – путей, циклов, остовных деревьев, разрезов, потоков и пр. – позволяет предложить эффективные методы выбора из них объектов, оптимальных в том или ином смысле. В третьих, многие общие и частные задачи дискретной оптимизации, в которых нет никакого упоминания о графах, тем не менее естественно представляются как задачи оптимизации на графах.

Естественно и начать эту часть с достаточно подробного определения и описания основных рассматриваемых в ней объектов - ориентированных и неориентированных графов. Этому посвящена 6 глава 6. В главе 7 рассмотрена одна из наиболее известных и хорошо исследованных задач оптимизации на графах – классическая задача поиска максимального потока. В главе 8 рассмотрена не менее знаменитая задача о кратчайших путях между вершинами и описаны два существенно различных подхода к её решению – алгоритм Дейкстры и алгоритм Флойда-Уоршалла. В главе 9 рассмотрена почти столь же знаменитая задача о паросочетаниях в двух оптимизационных постановках – о максимальном паросочетании и о назначении. Подходы к их решению основаны на алгоритмах, ранее рассмотренных в этой части: потоковом алгоритме и алгоритме нахождения циклов отрицательной длины. В главе 10 рассматривается одна из центральных задач дискретной оптимизации – многошаговая задача. Эта общая задача эквивалентна задаче нахождения пути максимальной стоимости на графе, что и послужило одним из оснований включения этой задачи в данную часть пособия. Рассмотрены также некоторые важные модификации общей постановки.

Центральное место в этой части пособия занимают классические алгоритмы нахождения максимального потока, кратчайшего пути, эффективного назначения и пр. Практически во всех случаях предложены таблицы и простые правила их последовательного заполнения, позволяющие «вручную» решать упомянутые задачи в небольшой размерности (но всё же в такой, при которой решение прямым перебором является затруднительным). Представляется, что именно такая работа с алгоритмами позволяет действительно понять, как они «работают» не только на чисто формальном, но и на содержательном уровне.

Глава 6. Элементы теории графов

В главе вводятся в рассмотрение графы – объекты, позволяющие просто представлять, точно формулировать и эффективно решать многие теоретические и прикладные задачи. Даются содержательное и формальное определения графа (как в ориентированном, так и неориентированном случаях). Определяются базовые понятия и термины, связанные с графами. Все понятия, связанные с взвешенными графами, т.е. с графами, вершинам и/или рёбрам которых сопоставлены некоторые числа – стоимости, длины и т.д., будут рассмотрены далее.

1. Понятие и определения графа

Во многих случаях жизни привычка толкает нас рисовать на бумаге точки, изображающие населенные пункты, агрегаты, людей, химические вещества и т.д., и соединять эти точки линиями или стрелками, указывающими на связи между соответствующими объектами. Такие схемы встречаются всюду под разными названиями: карты дорог, технологические схемы предприятий, диаграммы организаций, электрические цепи, сети коммуникаций, блок-схемы алгоритмов, генеалогические деревья и пр. Хотя такого рода объекты изучались достаточно давно (начиная с Эйлера), немецкий математик Д. Кёниг был первым, кто предложил называть такие схемы «графами» и систематически изучать их свойства (в 1936 году).

Граф характеризует <u>связи между объектами</u>; условно эти объекты изображаются точками, а связи между ними – линиями, соединяющими соответствующие точки. При этом положение точек, наклон и длина линий совершенно не имеют значения (в отличие, например, от изображений в геометрии); важно лишь то, какие именно пары точек <u>соединены</u>, а какие – <u>нет</u>. Для удобства будем обозначать вершины натуральными числами (вообще говоря, их можно обозначать любыми символами).

Пример 1. На рис.1 приведен пример трёх графов. На первый взгляд эти графы различны. Однако на самом деле во всех трёх случаях изображен один и тот же граф. Действительно, во всех трех изображениях соединены между собой те же самые пары точек: {1, 2}, {1, 4}, {1, 5}, {2, 3}, {2, 4}, {2, 5}, {3, 4}, {4, 5}; никакие другие пары точек не соединены ■

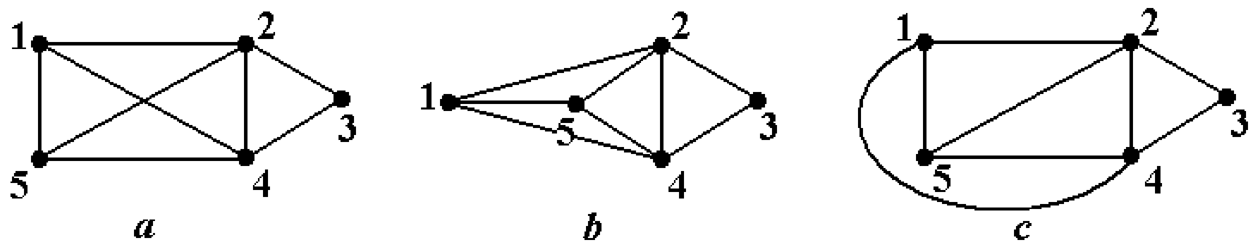


Рис.1

В одних случаях при описании связей между объектами «направление» связи не имеет значения. Соответствующие точки в графе соединяются линией без стрелки (как на рис.1); граф в этом случае называется <u>неориентированным</u>. В других случаях важно не только то, что объекты связаны, но и то, как именно «направлена» эта связь. Соответствующие точки в графе соединяются линией со стрелкой; граф в этом случае называется <u>ориентированным</u>. С помощью неориентированных графов удобно представлять, например, карты дорог; технологические схемы естественно описывать с помощью ориентированных графов. Пример ориентированного графа (так называемая «сеть питания») показан на рис.2.

1.1. Формальное определение графов. Нам понадобятся введённые в разделе 3.5 понятия функции типа $A \to B$, инъективной функции, множеств отправления, прибытия, определения и значения функции, а также введённое в разделе 3.1 понятия кортежа (пары и тройки).

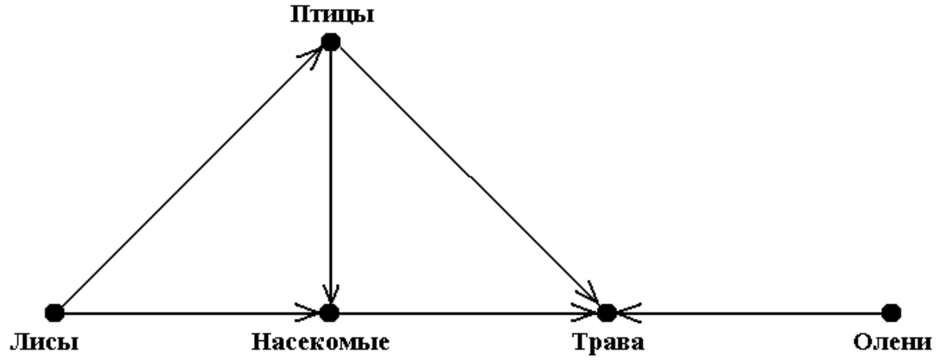


Рис.2

1.1.1. Определение неориентированных графов. Обозначим через $X^{1\text{-}2}$ множество всех одно- и двухэлементных подмножеств множества X. Дадим теперь необходимые формальные определения. ***Неориентированным графом*** G называется тройка $\langle V, E, F\rangle$, где V – множество ***вершин*** графа, E – множество его ***рёбер***, F – всюду определённая функция типа $E \to V^{1\text{-}2}$ (т.е. функция, у которой область определения совпадает с областью отправления). Это определение может показаться сложным и запутанным, особенно в сравнении с интуитивно ясным представлением о графе, данном в примере 1. Остановимся на этом подробнее.

Пример 2. Рассмотрим граф, показанный на рис.3. В этом графе множество рёбер $E = \{a, b, c, d, e, f, g\}$; множество вершин $V = \{1, 2, 3, 4\}$; $V^{1\text{-}2} = \{\{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}$. Рёбра a и b соединяют одни и те же вершины 1 и 3; рёбра c и d соединяют одни и те же вершины 1 и 4; ребро e соединяет вершины 2 и 3; ребро f соединяет вершины 1 и 2; наконец, ребро g соединяет вершины 2 и 4. Именно такого рода соответствия формализуются с помощью функции F. В данном случае

$$F(a) = F(b) = \{1, 3\}, F(c) = F(d) = \{1, 4\}, F(f) = \{1, 2\}, F(e) = \{1, 3\}, F(g) = \{2, 4\}. \qquad (1)$$

Рёбра, соединяющие одну и ту же пару вершин, называются ***кратными***. В рассматриваемом случае кратными являются рёбра a и b, а также c и d. Заметим, что концы определены для всех рёбер, что и означает, что функция F всюду определена. Другими словами, никаких «болтающихся» концов у рёбер нет ■

В общем случае можно сказать, что значение функции F на произвольном ребре p, т.е. одно- или двухэлементное множество вершин, как раз представляет собой множество концов данного ребра p (сравните рис.3 и соотношения (1) и убедитесь в этом!). В примере 2 все эти множества были двухэлементными, т.е. рёбра, у которых оба конца совпадают, в графе на рис.3 отсутствуют. Однако в общем случае такие рёбра могут присутствовать.

Пример 3. Рассмотрим граф, показанный на рис.4. В этом графе $F(a) = \{1\}$, $F(b) = \{1, 2\}$. Рёбра, у которых концы совпадают (с вершиной A), называются ***петлями при вершинах*** (вершине A). Вообще говоря, петель при одной и той же вершине (т.е. кратных петель) может быть несколько, как и рёбер, соединяющих одну и ту же пару разных вершин ■

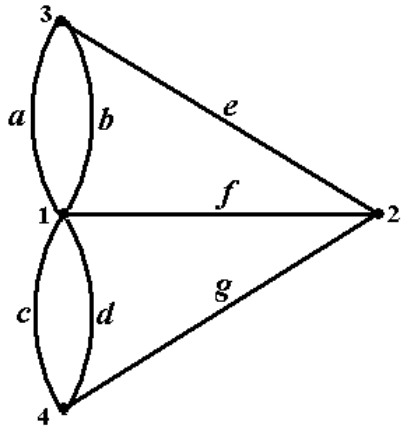


Рис.3

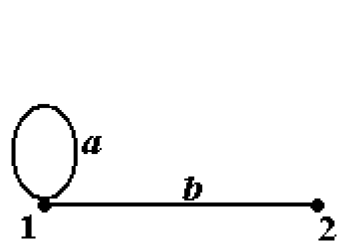


Рис.4

Во многих случаях разным рёбрам графа соответствуют разные пары вершин (если они не являются петлями) и разные вершины (если они являются петлями). В этих случаях функция F является инъекцией (см. раздел 3-5), а соответствующий граф называется ***простым***.

Напомним, что инъективной функцией, или инъекцией, называется функция F, такая, что

$[x \neq y] \rightarrow [F(x) \neq F(y)]$ (2)

(определение импликации $P \rightarrow Q$ см. в разделе 1.2).

Поскольку каждое ребро простого графа однозначно определяется своими концами, то такое ребро можно однозначно задать двухэлементным или одноэлементным (для петли) множеством его концов. Таким образом, можно дать следующее формальное определение. Неориентированным простым графом G называется пара $\langle V, E\rangle$, где V – множество вершин графа, $E \subseteq V^{1\text{-}2}$ – множество его рёбер. Здесь одноэлементное множество из V, т.е. множество вида $\{i\}$, означает ребро – петлю при вершине i, а двухэлементное множество $\{i, j\}$ означает ребро с концами i и j. Заметим также, что иногда в литературе простые графы называются графами, а графы, содержащие кратные рёбра – ***мультиграфами.***

Пример 4. Рассмотрим граф, показанный на рис.4. Этот граф является простым. Поэтому его можно задать парой множеств $\langle V, E\rangle$, где $V = \{1, 2\}$ – множество вершин графа, $E = \{\{1\}, \{1, 2\}\}$ – множество его рёбер ■

Пример 5. Граф, показанный на рис.1, является простым графом без петель. Он задаётся парой множеств $\langle V, E\rangle$, где $V = \{1, 2, 3, 4, 5\}$ – множество его вершин, $E = \{\{1, 2\}, \{1, 4\}, \{1, 5\}, \{2, 3\}, \{2, 4\}, \{2, 5\}, \{3, 4\}, \{4, 5\}\}$ – множество его рёбер ■

Поскольку простой граф – частный случай графа, то введённое выше задание графа в виде тройки $\langle V, E, F\rangle$ является более общим и может быть использовано во всех случаях, в том числе и для простых графов. В то же время задание парой $\langle V, E\rangle$, которое применимо ко всем простым парам, не может использоваться в общем случае: если несколько рёбер имеют общие концы i и j, то их, конечно, нельзя задать одним и тем же двухэлементным множеством $\{i, j\}$. Следует сказать, что задание парой проще, чем задание тройкой. Поэтому во всех случаях, когда это возможно, т.е. для простых графов, будем использовать задание парой $\langle V, E\rangle$.

Резюмируем рассмотренные примеры 1 – 5.

Граф на рис.1 задаётся парой $\langle V, E\rangle$, где $V = \{1, 2, 3, 4, 5\}$, $E = \{\{1, 2\}, \{1, 4\}, \{1, 5\}, \{2, 3\}, \{2, 4\}, \{2, 5\}, \{3, 4\}, \{4, 5\}\}$.

Граф на рис.3 задаётся тройкой $\langle V, E, F\rangle$, где $V = \{1, 2, 3, 4\}$, $E =\{a, b, c, d, e, f, g\}$, $F(a) = F(b) = \{1, 3\}$, $F(c) = F(d) = \{1, 4\}$, $F(f) = \{1, 2\}$, $F(e) = \{1, 3\}$, $F(g) = \{2, 4\}$.

Граф на рис.4 может быть задан как парой $\langle V, E\rangle$, где $V = \{1, 2\}$, $E = \{\{1\}, \{1, 2\}\}$, так и тройкой $\langle V, E, F\rangle$, где $V = \{1, 2\}$, $E =\{a, b\}$, $F(a) = \{1\}$, $F(b) = \{1, 2\}$.

В связи с введёнными способами задания графов (напомним, что пока рассматриваются только неориентированные графы) предлагаются следующие виды стандартных заданий:

1) по заданному формальному описанию (в виде тройки или пары) нарисовать граф;

2) по заданному изображению дать формальное описание графа в виде пары $\langle V, E\rangle$ (если граф простой) или тройки $\langle V, E, F\rangle$ (в противном случае). Если номера вершин и/или имена рёбер на рисунке не указаны, дать их самостоятельно.

Конечно, существует много других способов формального задания графов, один из которых описан далее, в разделе 4.1. В основном выбор формального задания графа (структуры данных) определяется эффективностью алгоритма, при программной реализации которого этот вид задания используется. Подробнее эти вопросы рассматриваются в других курсах (например, в курсе «Структуры и алгоритмы обработки данных»).

1.1.2. Определение ориентированных графов. Оно во многом схоже с определением неориентированных графов. Ориентированным графом G называется тройка $\langle V, A, F\rangle$, где V – множество ***вершин*** графа, A – множество его ***дуг***, F – всюду определённая функция типа $A \rightarrow V^2$ (напомним, что через V^2 обозначается прямое произведение (см. раздел 1-3.2) множества вершин V на себя или $V \times V$, т.е. множество всех упорядоченных пар $\langle x, y\rangle$, где $x, y \in V$). В отличие от двухэлементных множеств $\{x, y\}$, в которых по самому смыслу слова «двухэлементный» x и y не совпадают, в паре $\langle x, y\rangle$ компоненты могут совпадать. Это определение близко по смыслу к определению неориентированных графов. Именно, $F(a) = \langle i, j\rangle$ означает, что дуга a выходит из вершины i и входит в вершину j; если же $i = j$, то это означает, что данная дуга является петлёй при вершине i. Как и в неориентированном случае, если функция F инъективна, то соответствующий граф называется простым; при этом для каждой пары вершин $\langle i, j\rangle$ существует не более одной дуги с началом i и концом j. Поэтому здесь также можно задавать любую дугу упорядоченной парой $\langle i, j\rangle$, состоящей из её начала i и конца j, и представлять граф G в виде пары $\langle V, A\rangle$, где множество дуг $A\subseteq V^2$. Обратим внимание на то, что дуги $\langle i, j\rangle$ и $\langle j, i\rangle$ являются разными.

Для сокращения записи иногда (если это не вызовет недоразумений) неориентированные графы будем называть просто графами, а ориентированные – орграфами.

Пример 6. Граф на рис.2 является простым орграфом без петель. Занумеруем его вершины так: Лисы – 1; Насекомые – 2; Трава – 3; Олени – 4; Птицы – 5. Тогда $G = \langle V, A\rangle$, где $V = \{1, 2, 3, 4, 5\}$, $A = \{\langle 1, 2\rangle, \langle 1, 5\rangle, \langle 2, 3\rangle, \langle 4, 3\rangle, \langle 5, 2\rangle, \langle 5, 3\rangle\}$ ■

2. Внутренне- и внешне устойчивые множества вершин

Дадим определения сразу для неориентированного и ориентированного случаев, учитывая, что эти определения очень похожи (отличия будем указывать в скобках). Две различные вершины, которые соединены ребром (дугой), т.е. являющиеся концами одного и того же ребра (дуги), называются ***смежными***. В противном случае две вершины не смежны. Множество X вершин графа G, любые две из которых не смежны, называется ***внутренне устойчивым***, а максимально возможное число элементов внутренне устойчивого множества − ***числом внутренней устойчивости*** графа G (обозначается α (G)).

Сразу поясним это не очень простое понятие. Прежде всего, любое множество вершин, состоящее из одной вершины, является по определению внутренне устойчивым как в неориентированном, так и в ориентированном случае. Действительно, определение внутренней устойчивости множества вершин X, как и почти все математические определения, является импликацией. Его можно переформулировать так: «Если любые две разные вершины множества X не смежны, то множество X называется внутренне устойчивым» или, пользуясь определёнными в главе 1-4 кванторами, короче

$$(\forall a\in X)(\forall b\in X)[(a\neq b)\to(a \text{ не смежна с } b)]. \qquad (3)$$

Но если множество X состоит из одной вершины, то посылка $a \neq b$ импликации (3) всегда ложна, поскольку a и b – две разные вершины из множества X. Напомним, что по определению импликации (см. раздел 1-2.1.4), импликация с ложной посылкой всегда истинна «в силу ложности посылки». Но это и означает истинность импликации (3), что и означает внутреннюю устойчивость любого одноэлементного множества.

Утверждение 1. Любое подмножество внутренне устойчивого множества внутренне устойчиво.

Действительно, по определению, во внутренне устойчивом множестве никакие две вершины не соединены ребром. Значит, этим же свойством обладает и любое его подмножество ■

Внутренне устойчивое множество вершин называется ***максимальным по включению***, если оно не содержится ни в каком другом внутренне устойчивом множестве. Утверждение 1 позволяет при поиске всех внутренне устойчивых множеств ограничиться поиском только максимальных по включению внутренне устойчивых множеств. Действительно, всякое внутренне устойчивое множество либо само является максимальным по включению, либо содержится в некотором максимальном по включению внутренне устойчивом множестве. (Заметим, что таких множеств может быть несколько). Поэтому семейство всех максимальных по включению внутренне устойчивых множеств определяет все внутренне устойчивые множества: множество является внутренне устойчивым, если оно содержится хотя бы в одном максимальном по включению внутренне устойчивом множестве.

Пример 7. Проиллюстрируем понятия внутренней устойчивости множества вершин в неориентированном графе. Для этого рассмотрим граф, показанный на рис.5а. На рис.5б – 5е показаны все максимальные по включению внутренне устойчивые множества в данном графе. На каждом из этих рисунков видно, что все выделенные большими кружками вершины не соединены непосредственно друг с другом рёбрами. Ясно также, что при добавлении любой вершины все эти множества перестают быть внутренне устойчивыми – появляются рёбра. Обратим внимание на то, что максимальные по включению внутренне устойчивые множества могут содержать различное число вершин – три на рис.5б – 5г и два на рис.5д и 5е.

Множество, показанное на рис.5ж, является внутренне устойчивым, но не максимальным по включению. Оно содержится в множестве, показанном на рис.5б. Наконец, множество, показанное на рис.5з, не является внутренне устойчивым. Входящие в него вершины 5 и 6 соединены ребром.

Поскольку максимальное число вершин во внутренне устойчивом множестве равно 3, то число внутренней устойчивости $\alpha(G) = 3$.

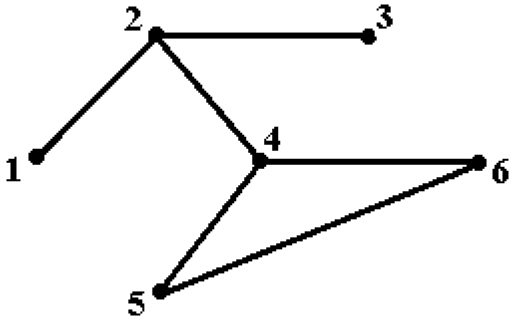


Рис.5а. Исходный граф

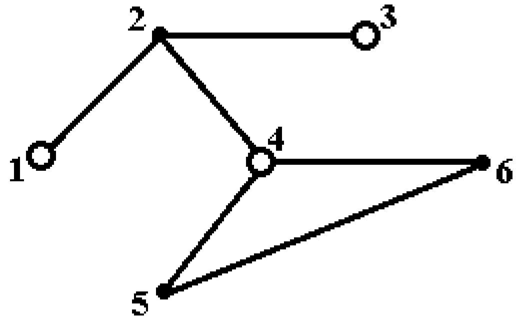


Рис.5б. Максимальное по включению внутренне устойчивое множество {1,3,4}

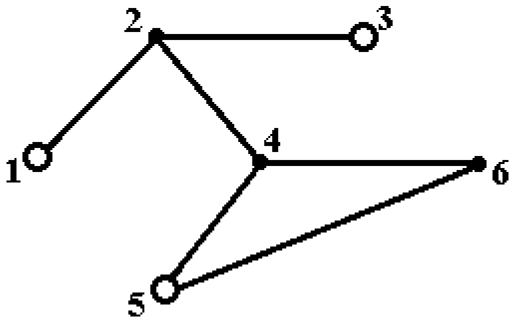


Рис.5в. Максимальное по включению внутренне устойчивое множество {1,3,5}

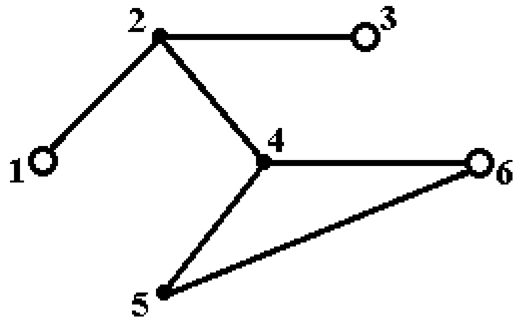


Рис.5г. Максимальное по включению внутренне устойчивое множество {1,3,6}

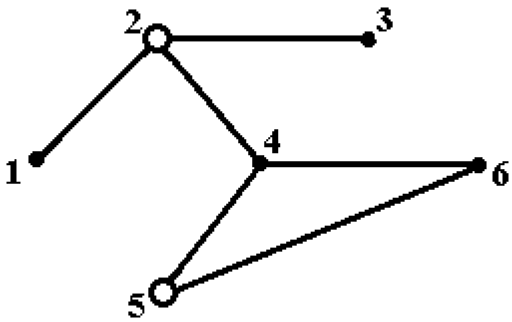


Рис.5д. Максимальное по включению внутренне устойчивое множество {2,5}

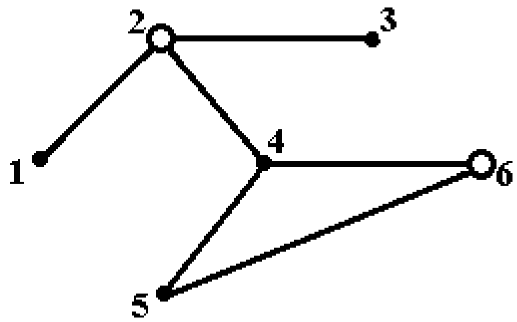


Рис.5е. Максимальное по включению внутренне устойчивое множество {2,6}

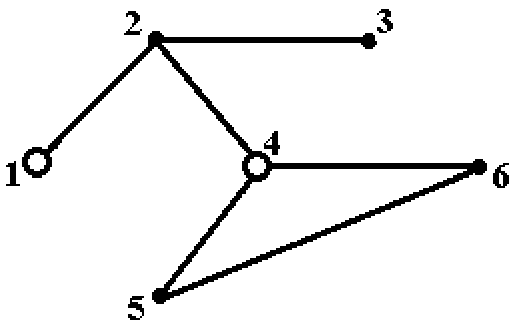


Рис.5ж. Внутренне устойчивое множество {1,4}, не максимальное по включению

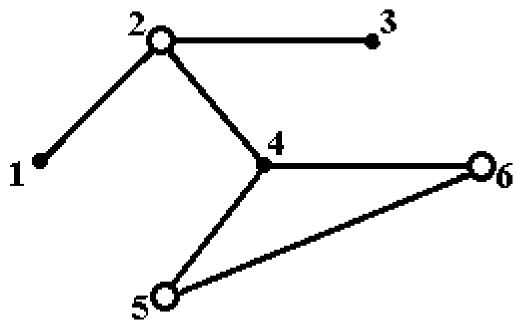


Рис.5з. Множество вершин {2,5,6}, не являющееся внутренне устойчивым

Можно проверить, что любое множество, содержащее 4 вершины, не будет внутренне устойчивым. Действительно, любое внутренне устойчивое множество не может содержать никаких двух вершин из подмножества {4,5,6}, так как они все попарно смежны (соединены рёбрами). Значит, в него может входить не более одной вершины из множества {4,5,6}. Но так как в нём должно быть 4 вершины, а всего есть 6 вершин, то в него должны входить все вершины из множества {1,2,3}. А так как вершины 1 и 2 смежны, то такое множество не может быть внутренне устойчивым. В силу того же утверждения 1 бóльшие множества вершин также не могут быть внутренне устойчивыми■

В произвольном графе с небольшим числом вершин максимальные по включению внутренне устойчивые множества могут быть найдены простым перебором всех различных подмножеств, начиная с множества всех вершин. Однако самый простой и эффективный подход – внимательно посмотреть на граф собственными глазами (и подумать собственной головой). Конечно, в графах, содержащих порядка 15 и более вершин, такой просмотр практически нереален. Формальные алгоритмы решения данной задачи в силу своей сложности здесь не рассматриваются.

Введём дальнейшие понятия. Множество X вершин графа G называется ***внешне устойчивым***, если любая вершина из $V\setminus X$, т.е. вершина, не принадлежащая X, смежна хотя бы с одной вершиной из X(является концом некоторой дуги, начало которой принадлежит множеству вершин X, в ориентированном случае). Минимально возможное число элементов внешне устойчивого множества называется ***числом внешней устойчивости*** графа G (обозначается $\beta(G)$). Обратим внимание на то, что связность вершин внутри множества X не оказывает никакого влияния на данное свойство. Это значит, что добавление или удаление любого ребра (дуги) между вершинами из X не повлияет ни на наличие свойства внешней устойчивости у множества X, ни на его отсутствие. Обратим внимание, что в определении не требуется, чтобы какая-нибудь вершина из X была бы смежной со всеми вершинами из $V\setminus X$. Достаточно, если любая вершина из $V\setminus X$ смежна хотя бы с какой-нибудь вершиной из X. Для разных «внешних» вершин (т.е. из $V\setminus X$) смежные с ними «внутренние» вершины (т.е. из X) могут (но не обязаны!) быть различными.

Заметим, что само множество всех вершин V является внешне устойчивым в силу ложности посылки. Действительно, при $X = V$ имеем $V\setminus X =\varnothing$, т.е. вершин, не принадлежащих X, просто не существует. Как и определение внутренне устойчивого множества, определение внешне устойчивого множества является импликацией:

$(\forall a\in V\setminus X)\rightarrow(\exists b\in X)(a$ смежна с $b)$, (4)

которая всегда верна при ложной посылке $(\forall a\in V\setminus X)$.

Имеет место простое

Утверждение 2. Любое множество, содержащее внешне устойчивое множества вершин, также внешне устойчиво.

Действительно, пусть внешне устойчивое множество $X\subseteq Y$. Для всякой вершины $z\in V\setminus Y$ верно $z\in V\setminus X$. Так как X – внешне устойчивое множество, то, по определению, z смежна с некоторой вершиной $w\in X$ (является концом дуги с началом w). А в силу включения $X\subseteq Y$ имеем $w\in Y$. Таким образом, начав с произвольной вершины $z\in V\setminus Y$, установили, что она смежна с некоторой вершиной $w\in Y$ (является концом дуги с началом w), что, по определению, означает внешнюю устойчивость Y ■

Внешне устойчивое множество называется ***минимальным по включению***, если оно не содержит ни одного другого внешне устойчивого множества. В силу утверждения 2, для нахождения всех внешне устойчивых множеств достаточно найти только минимальные внешне устойчивые множества. Все остальные являются произвольными множествами, содержащими хотя бы одно минимальное по включению внешне устойчивое множество.

Проиллюстрируем теперь это понятие на том же графе рис.5а.

Пример 8. На рис.6а – 6е показаны все минимальные по включению внешне устойчивые множества в данном графе. На каждом из этих рисунков видно, что любая вершина, показанная чёрным кружком (как в исходном графе) соединена хотя бы с одной вершиной из внешне устойчивого множества, выделенных квадратами. Ясно также, что при удалении любой вершины из внешне устойчивого множества все эти множества перестают быть внешне устойчивыми – появляются вершины, не соединённые ни с одной из оставшихся вершин. Обратим внимание на то, что минимальные по включению внешне устойчивые множества могут содержать различное число вершин – две на рис.6а – 6в и три на рис.6г – 6е.

Множество, показанное на рис.6ж, является внешне устойчивым, но не минимальным по включению. Оно содержит множества, показанные на рис.6б и 6в. Наконец, множество, показанное на рис.6з, не является внешне устойчивым. Вершина 3 не соединена ребром ни с вершиной 1, ни с вершиной 4, что противоречит определению внешне устойчивого множества.

Поскольку минимальное число вершин во внешне устойчивом множестве равно 2, то число внешней устойчивости $\beta(G) = 2$.

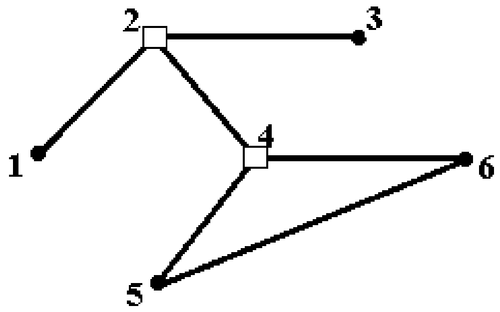


Рис.6а. Минимальное по включению внутренне устойчивое множество {2,4}

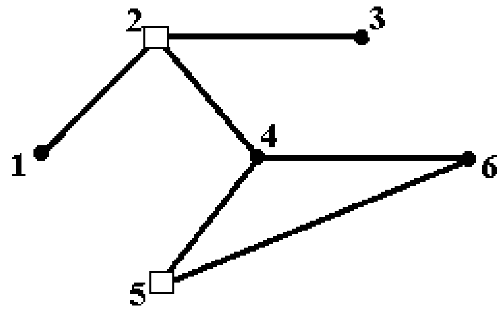


Рис.6б. Минимальное по включению внутренне устойчивое множество {2,5}

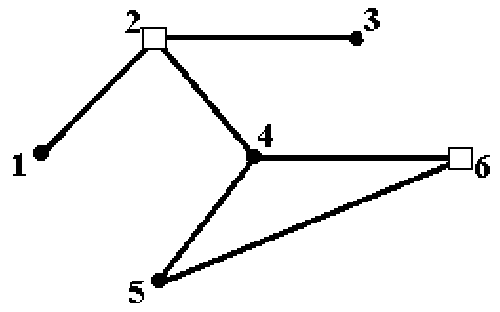


Рис.6в. Минимальное по включению внутренне устойчивое множество {2,6}

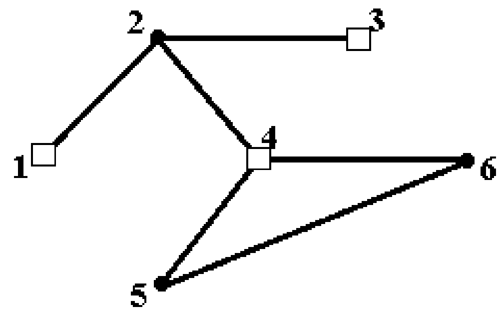


Рис.6г. Минимальное по включению внутренне устойчивое множество {1,3,4}

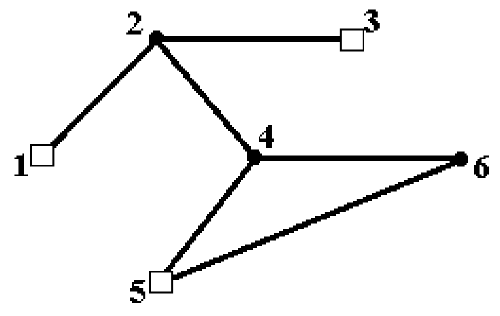


Рис.6д. Минимальное по включению внутренне устойчивое множество {1,3,5}

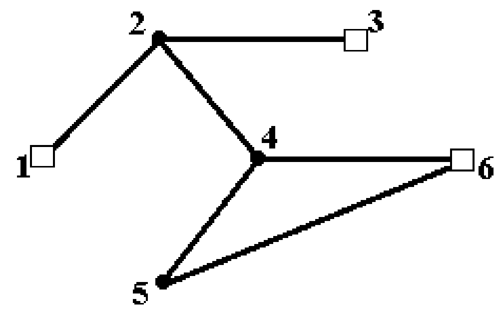


Рис.6е. Минимальное по включению внутренне устойчивое множество {1,3,6}

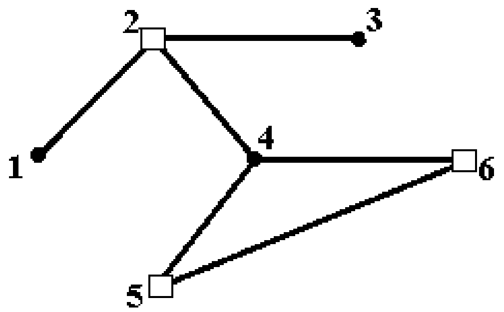


Рис.6ж. Внешне устойчивое множество {2,5,6}, не минимальное по включению

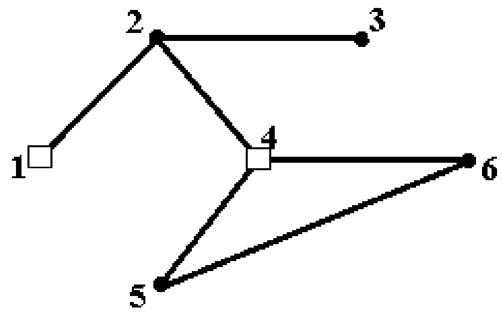


Рис.6з. Множество вершин {1,4}, не являющееся внешне устойчивым

Видно непосредственно на рисунке, что любое множество, содержащее одну вершину, не будет внешне устойчивым – просто потому, что в исходном графе нет ни одной вершины, соединённой со всеми остальными ■

Как и для внутренне устойчивых множеств, формальные алгоритмы решения данной задачи в силу своей сложности здесь не рассматриваются.

Множество X вершин графа G, одновременно внутренне и внешне устойчивое, называется ***ядром*** графа G.

Напомним, что определения внутренне устойчивого множества вершин в неориентированном и ориентированном случаях совпадают. Совпадают и определения ядра. Отличаются только определения внешне устойчивого множества: в ориентированном случае требуется, чтобы дуга, соединяющая множества X и $V\setminus X$, имела направление от X к $V\setminus X$.

Из определений внутренней и внешней устойчивости непосредственно следует, что:

- любая изолированная вершина графа может быть добавлена к любому не содержащему её внутренне устойчивому множеству и это расширенное множество также будет внутренне устойчивым;

- любое внешне устойчивое множество обязано содержать любую изолированную вершину.

Из утверждений 1 и 2 непосредственно следует

Утверждение 3. Множество вершин Z является ядром графа G тогда и только тогда, когда существуют минимальное по включению внешне устойчивое множество X и максимальное по включению внутренне устойчивое множество Y, такие, что

$$X\subseteq Z\subseteq Y. \qquad (5)$$ ■

Если уже известны все минимальные по включению внешне устойчивые множества X_1, …, X_s и все максимальные по включению внутренне устойчивые множества Y_1, …, Yt, то утверждение 3 позволяет предложить следующий

Алгоритм нахождения всех ядер графа.

1. Находим все такие пары индексов $\langle i, j\rangle$, такие, что

$$Xi\subseteq Y_j. \qquad (6)$$

2. Для каждой такой пары индексов находим все множества Z, удовлетворяющие двойному включению

$$Xi\subseteq Z\subseteq Y_j. \qquad (7)$$

3. Из всех множеств Z, найденных на шагах 1 и 2, выделяем все различные множества. Они и являются всеми ядрами графа ■

Алгоритм применим как в неориентированном, так и в ориентированном случаях. Если включения (6) ни при каких i, j не выполняются, то это означает, что в данном случае ядра не существуют. Заметим также, что сами списки X_1, …, X_s и Y_1, …, Yt никогда не являются пустыми, поскольку в любом графе и орграфе любое одноэлементное множество является внутренне устойчивым, а множество всех вершин – внешне устойчивым, в силу ложности посылки.

Пример 9. Рассмотрим снова граф на рис.5a. Максимальными по включению внутренне устойчивыми множествами являются множества $Y_1=\{1,3,4\}$, $Y_2=\{1,3,5\}$, $Y_3=\{1,3,6\}$, $Y_4=\{2,5\}$, $Y_5=\{2,6\}$ (см. рис.5). Минимальными по включению внешне устойчивыми множествами являются множества $X_1=\{2,4\}$,$X_2=\{2,5\}$, $X_3=\{2,6\}$, $X_4=\{1,3,4\}$, $X_5=\{1,3,5\}$, $X_6=\{1,3,6\}$ (см. рис.6). По алгоритму нахождения ядер рассмотрим все пары множеств $\langle Xi, Y_j\rangle$, удовлетворяющие включению (6). Таковыми являются следующие пары:$\langle X_2, Y_4\rangle$, $\langle X_3, Y_5\rangle$, $\langle X_4, Y_1\rangle$, $\langle X_5, Y_2\rangle$, $\langle X_6, Y_3\rangle$. Для всех этих пяти пар включения являются равенствами, т.е. $Xi= Y_j$. Поэтому расположенные «между ними» (см. (7)) ядра Z совпадают с $Xi=Y_j$. В данном случае получаем список ядер {2,5}, {2,6}, {1,3,4}, {1,3,5}, {1,3,6} (шаг 3 не потребовался, поскольку все эти множества различны). Таким образом, в данном случае множество ядер совпадает с множеством максимальных по включению внутренне устойчивых множеств. Такое совпадение не является общим правилом, однако важная связь между этими множествами в случае неориентированных графом действительно присутствует. Эти вопросы будут рассмотрены немного далее ■

Пример 10. На рис.7 показан орграф из примера 6 (сеть питания) с пронумерованными вершинами 1 – 5. В этом орграфе внутренне устойчивыми множествами являются все 1-элементные множества {1}, {2}, {3}, {4}, {5} и 2-элементные множества {1,3}, {1,4}, {2,4}, {4,5}.

Других внутренне устойчивых множеств в данном орграфе нет. Максимальными по включению являются только эти 2-элементные множества.

Одноэлементных внешне устойчивых множеств в данном случае нет (так как нет ни одной вершины, откуда дуги ведут во все остальные). Единственным 2-элементным внешне устойчивым множеством вершин является $\{1, 4\}$ (так как дуги $\langle 1, 5\rangle$, $\langle 1, 2\rangle$ и $\langle 4, 3\rangle$ ведут из $\{1, 4\}$ в 5, 2 и 3, т.е. во все остальные вершины, в соответствии с определением). Поэтому множество $\{1, 4\}$ является минимальным по включению внешне устойчивым множеством. Далее, пусть X–любое внешне устойчивое множество в данном орграфе. Предположим, что вершина 1 не входит в X. Тогда, по определению для ориентированных графов, должна быть вершина в X, из которой выходит дуга с концом в вершине 1. Но поскольку в вершину 1 ни одна дуга не входит (см. рис.7), то сделанное предположение неверно. Поэтому вершина 1 входит в любое внешне устойчивое множество X. Принадлежность вершины 4 любому внешне устойчивому множеству вершин устанавливается точно также. Таким образом, множество вершин $\{1, 4\}$ входит в любое внешне устойчивое множество. Следовательно, $\{1, 4\}$ является единственным минимальным по включению внешне устойчивым множеством вершин.

Сравнивая по алгоритму список максимальных по включению внутренне устойчивых множеств $\{1,3\}$, $\{1,4\}$, $\{2,4\}$, $\{4,5\}$ и список минимальных по включению внешне устойчивых множеств вершин, состоящий из единственного множества $\{1,4\}$, сразу получаем, что единственным ядром является множество вершин $\{1,4\}$■

Пример 11. Рассмотрим простой орграф без петель, показанный на рис.8. Единственными внутренне устойчивыми множествами вершин являются три одноэлементных множества $\{1\}$, $\{2\}$, $\{3\}$. Они же, естественно, максимальны по включению (так как бóльших внутренне устойчивых множеств нет).

Найдём все минимальные по включению внешне устойчивые множества вершин. Как уже упоминалось выше, множество всех вершин является внешне устойчивым. Далее, 2-элементное множество $\{1,2\}$ является внешне устойчивым, поскольку есть дуга $\langle 2,3\rangle$ в единственную не входящую в $\{1,2\}$ вершину 3. Точно также, внешне устойчивыми множествами являются $\{2,3\}$ и $\{1,3\}$. Далее, 1-элементное множество $\{1\}$ не является внешне устойчивым, потому что в графе нет дуги $\langle 1,3\rangle$, что противоречит определению. Аналогично, 1-элементные множества $\{2\}$ и $\{3\}$ также не являются внешне устойчивыми.

Таким образом, имеется список минимальных по включению внешне устойчивых множеств $\{1,2\}$, $\{2,3\}$, $\{1,3\}$ и список максимальных по включению внутренне устойчивых множеств $\{1\}$, $\{2\}$, $\{3\}$. Условия (6) $Xi \subseteq Y_j$ не выполняются уже потому, что все Xi состоят из двух элементов, а все Y_j – из одного элемента. В силу утверждения 3 в данном случае ядра отсутсутствуют, так как включения (5) не могут иметь места ■

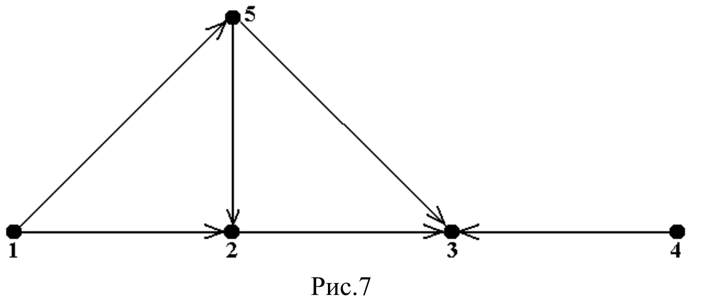


Рис.7

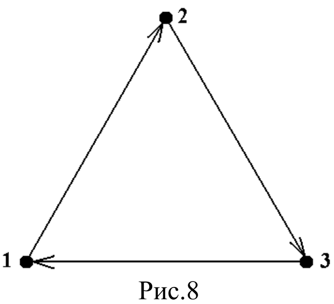


Рис.8

Примеры 10 и 11 показывают, что в ориентированных графах ядра могут быть, а могут и не быть. Возможно ли отсутствие ядер внеориентированных графах? Полный ответ на этот вопрос даёт

Утверждение 4. В неориентированном графе любое максимальное по включению внутренне устойчивое множество является и внешне устойчивым, т.е. является ядром.

Действительно, предположим, что максимальное по включению внутренне устойчивое множество вершин A не является внешне устойчивым. По определению, это означает, что существует вершина $y \in V \setminus A$)., не смежная ни с одной вершиной $x \in A$. Но тогда множество $A \cup \{y\}$ не содержит ни одного ребра, поскольку без вершины y, т.е. в самом множестве A, рёбер нет по определению внутренней устойчивости, а добавление y также не добавило рёбер по построению. Это означает, что множество $A \cup \{y\}$ внутренне устойчиво. Но тогда множество A, будучи внутренне устойчивым, не будем максимальным по включению, поскольку содержится в бóльшем внутренне устойчивом множестве $A\cup\{y\}$, что противоречит предположению о максимальности A■

Утверждение 4 даёт гарантию существования ядер в любых неориентированных графах. Значительно более сложный вопрос об условиях существования и единственности ядер в ориентированных графах здесь не рассматривается.

3. Пути в графах

Как и в разделе 2, будем давать определения сразу для неориентированного и ориентированного случая, указывая отличия в скобках. Наиболее общим понятием является понятие пути (орпути). ***Путём*** (***орпутём***) в графе (орграфе) называется чередующаяся последовательность μ вершин и рёбер (дуг), начинающаяся и кончающаяся в вершинах:

$\mu = \langle v_0, b_1, v_1, b_2, \ldots, v_{k-1}, b_k, v_k\rangle$, (8)

такая, что вершины v_i и v_{i+1} соединены ребром (дугой) b_k.

Вершина v_0 называется началом пути (орпути) μ, вершина v_k – концом пути (орпути) μ, а число k, равное числу рёбер (дуг), входящих в путь (орпуть) – его длиной. Говорят также, что путь μ соединяет вершины v_0 и v_k (орпуть μ ведёт из v_0 в v_k). ***Обратным путём*** для пути μ называется путь $\mu^{-1} = \langle v_k, b_k, v_{k-1}, b_{k-1}, \ldots, v_1, b_1, v_0\rangle$. Другими словами, путь μ^{-1} состоит из тех же самых вершин и рёбер, но проходимых в обратном порядке. Обратим внимание, что обратный путь определён только в неориентированном случае.

Цепью (***орцепью***) называется путь (орпуть), в котором все рёбра (дуги) различны. ***Простой цепью*** (***орцепью***) называется цепь (орцепь), в которой все вершины различны. Путь (орпуть), в котором начало и конец совпадают, называется ***циклическим путём*** (***орпутём***). Циклический путь, являющийся цепью (орцепью), называется ***циклом*** (***орциклом***). Цикл (орцикл), в котором все вершины, кроме начала и конца, различны и не совпадают с началом, называется ***простым циклом*** (***орциклом***).

Говорят, что не циклический путь ν содержится в не циклическом пути μ, если последовательность ν или последовательность ν^{-1} является подпоследовательностью последовательности μ. В ориентированном случае остаётся только часть этого определения: не циклический орпуть ν содержится в не циклическом орпути μ, если последовательность ν является подпоследовательностью последовательности μ.

Для циклических путей последнее определение требуется модифицировать, учитывая то, что такой путь можно «обходить», начиная с любой вершины, отчего он, конечно же, не изменится. Поэтому можно дать такие четыре определения:

1) не циклический путь ν содержится в циклическом пути μ, если последовательность μ можно циклически перенумеровать так, что последовательность ν или последовательность ν^{-1} станет подпоследовательностью этой перенумерованной последовательности μ;

2) не циклический орпуть ν содержится в циклическом орпути μ, если последовательность μ можно циклически перенумеровать так, что последовательность ν станет подпоследовательностью этой перенумерованной последовательности μ;

3) циклический путь ν содержится в пути μ, если последовательность ν или последовательность ν^{-1} можно циклически перенумеровать так, что эта перенумерованная последовательность ν станет подпоследовательностью последовательности μ;

4) циклический орпуть ν содержится в орпути μ, если последовательность ν можноциклически перенумеровать так, что эта перенумерованная последовательность ν станет подпоследовательностью последовательности μ.

Эти на первый взгляд сложные понятия разъясняются далее (см. примеры 17 – 21).

Все вышеприведённые в данном разделе понятия относятся к произвольным графам и орграфам (см. раздел 1.1). В случае простых графов и орграфов понятие пути (орпути) упрощается: под путём (орпутём) понимается последовательность вершин

$\mu = \langle v_0, v_1, \ldots, v_{k-1}, v_k\rangle$, (9)

такая, что множество вершин $\{v_i, v_{i+1}\}$ определяет ребро графа (петлю), если v_i и v_{i+1} совпадают (пара вершин $\langle v_i, v_{i+1}\rangle$ определяет дугу орграфа (петлю, если v_i и v_{i+1} совпадают)). Напомним, что в простых графах (орграфах) пара вершин однозначно определяет ребро (дугу), в то время как в графах общего вида таких рёбер (дуг) может быть несколько (см. рис.3). Соответственно упрощаются все остальные приведённые выше определения – достаточно указывать только последовательности вершин.

Приведём примеры, иллюстрирующие и разъясняющие введённые понятия.

Пример 12. В графе, показанном на рис.9, обратным к пути $\mu = \langle 1, f, 4, d, 2, c, 4, g, 3\rangle$ является путь $\mu^{-1} = \langle 3, g, 4, c, 2, d, 4, f, 1\rangle$. Обратным к циклическому пути $\mu = \langle 1, f, 4, d, 2, e, 3, b, 1\rangle$ является циклический путь $\mu^{-1} = \langle 1, b, 3, e, 2, d, 4, f, 1\rangle$ (проверьте!) ■

Пример 13. В графе, показанном на рис.10, обратным к простому пути $\mu = \langle 2, 1, 4, 3\rangle$ является простой путь $\mu^{-1} = \langle 3, 4, 1, 2\rangle$. Обратным к циклическому пути $\mu^{-1} = \langle 2, 4, 1, 3, 4, 2\rangle$ является циклический путь $\mu^{-1} = \langle 2, 4, 3, 1, 4, 2\rangle$■

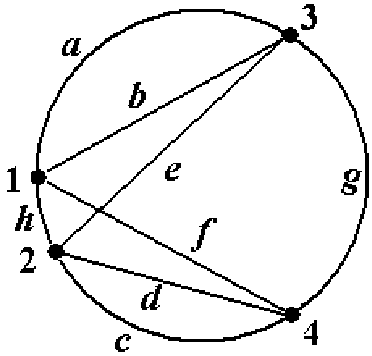


Рис.9

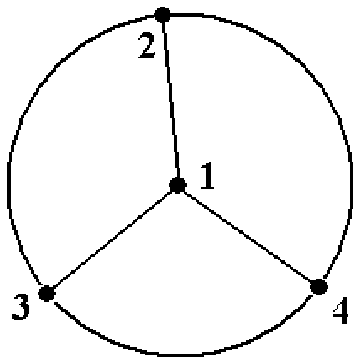


Рис.10

Пример 14. В графе, показанном на рис.11 последовательность $\mu = \langle 1, d, 3, e, 2, b, 1, a, 2, b, 1, c, 3\rangle$ является путём с началом в вершине 1 и концом в вершине 3. Его длина, т.е. число входящих в него рёбер, равно 6. Этот путь не является цепью, так как не все его рёбра различны (ребро b встречается два раза). В этом же графе последовательность $v = \langle 1, d, 3, e, 2, b, 1, c, 3\rangle$ является цепью с началом в вершине 1 и концом в вершине 3. Эта цепь не является простой, так как в ней не все вершины различны (вершины 1 и 3 встречаются по два раза). Последовательность $\lambda = \langle 1, d, 3\rangle$ является простой цепью. Последовательность $\lambda^* = \langle 1, c, 3\rangle$ также является простой цепью, отличной от простой цепи $\lambda = \langle 1, d, 3\rangle$ (см. рис.11). Заметим, что все четыре рассмотренных в примере пути $\mu, v, \lambda, \lambda^*$ соединяют одну и ту же пару вершин 1 и 3 ■

Пример 15. Граф, показанный на рис.12 является простым графом, поэтому для задания путей можно пользоваться упрощёнными последовательностями, состоящими только из вершин. Последовательность $\mu = \langle D, C, G, E, C, B, G, E, F\rangle$ является путём с началом в вершине D и концом в вершине F; его длина равна 8. Путь μ не является цепью, так как ребро $\{G, E\}$ входит в него дважды. Последовательность $v = \langle D, C, G, E, C, B, F\rangle$ является цепью, так как все её рёбра не повторяются (проверьте!), но она не является простой цепью, так как вершина C встречается в ней два раза. Последовательность $\lambda = \langle D, C, B, F\rangle$ является простой цепью, поскольку все её вершины различны. Заметим, что все три рассмотренные пути – μ, v, λ – соединяют одну и ту же пару вершин D и F ■

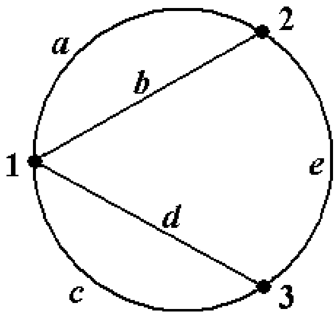


Рис.11

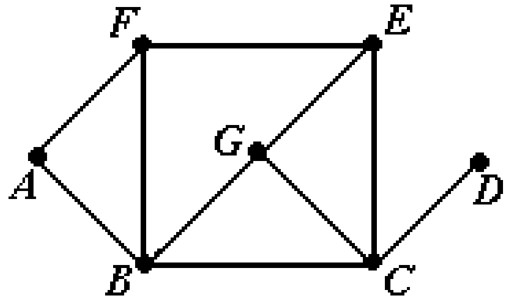


Рис.12

Замечания в конце примеров 14 и 15 приводят к достаточно простому умозаключению, сформулированному в следующем утверждении.

Утверждение 5. Для всякого пути (орпути), соединяющего две разные вершины, существует содержащаяся в нём цепь (орцепь), соединяющая те же две вершины. Для всякой цепи (орцепи), соединяющей две вершины, существует содержащаяся в ней простая цепь (орцепь), соединяющая те же две вершины ■

Заметим, что существующая в силу утверждения 5 цепь может совпадать с содержащим её путём (если сам путь уже является цепью); существующая в силу утверждения 5 простая цепь может совпадать с содержащей её цепью (если сама эта цепь уже является простой цепью).

Пример 16. В графе, показанном на рис.13, последовательность $\mu = \langle 1, a, 2, b, 1, a, 2\rangle$ является путём. Этот путь не является цепью, поскольку он проходит два раза по ребру a. Двумя содержащимися в этом пути цепями являются только цепи $\nu = \langle 1, a, 2\rangle$ и $\nu^* = \langle 1, b, 2\rangle$. Обе эти цепи уже являются простыми, и поэтому любая содержащаяся в одной из них простая цепь совпадает с содержащей цепью ■

Заметим также, что во всех рассмотренных в примерах 14 − 16 путях начальная и конечная вершины всегда различны, т.е. ни один из них не является циклическим путём.

Пример 17. Рассмотрим простой граф, показанный на рис.14. Не циклический путь $\mu = \langle 6, 2, 5\rangle$ содержится в циклическом пути $\nu = \langle 2, 5, 3, 6, 2\rangle$, хотя последовательность μ не является подпоследовательностью последовательности ν. Действительно, если <u>тот же самый</u> циклический путь ν начать с вершины 6, т.е. записать в виде $\nu = \langle 6, 2, 5, 3, 6\rangle$, то после такой перенумерации последовательность μ становится подпоследовательностью последовательности ν. Рассмотрим теперь не циклический путь $\mu = \langle 4, 1, 5\rangle$ и циклический путь $\nu = \langle 1, 4, 2, 5, 1\rangle$. Последовательность $\langle 4, 1, 5\rangle$ не является подпоследовательностью ни самой последовательности ν,ни любой её циклической перестановки: $\langle 4, 2, 5, 1, 4\rangle$, $\langle 2, 5, 1, 4, 2\rangle$, $\langle 5, 1, 4, 2, 5\rangle$. Однако обратная последовательность $\mu = \langle 5, 1, 4\rangle$ содержится в циклически переставленной $\nu = \langle 4, 2, 5, 1, 4\rangle$. Значит, в силу введённых определений, путь $\mu = \langle 4, 1, 5\rangle$ содержится в циклическом пути ν ■

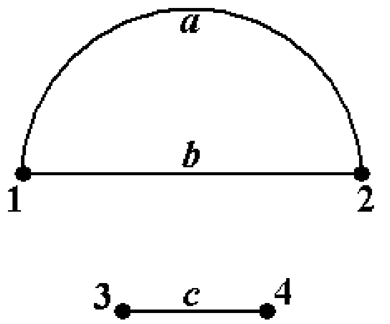


Рис.13

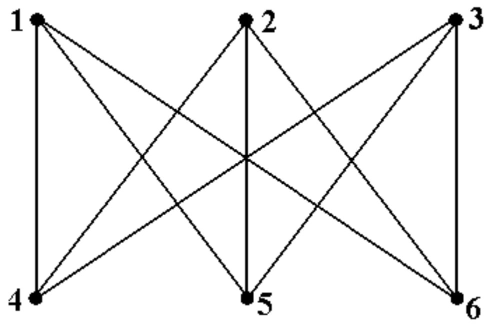


Рис.14

Пример 18. В графе, показанном на рис.11, последовательность $\mu = \langle 1, d, 3, e, 2, e, 3, c, 1, b, 2, a, 1\rangle$ является циклическим путём, но не является циклом, поскольку ребро e входит в него дважды. Содержащийся в пути μ путь $\nu = \langle 1, d, 3, c, 1, b, 2, a, 1\rangle$ является циклом, но не является простым циклом, так начальная и конечная вершина 1 входит в него 3 раза, вместо двух, требуемых определением. Наконец, содержащийся в пути ν путь $\lambda = \langle 1, d, 3, c, 1\rangle$ является простым циклом ■

Пример 19. В графе, показанном на рис.12, последовательность $\mu = \langle E, C, G, B, A, F, B, G, E\rangle$ является циклическим путём, но не является циклом, поскольку ребро $\{G,B\}$ входит в него дважды. Содержащийся в пути μ путь $\nu = \langle E, C, B, A, F, B, G, E\rangle$ является циклом, но не является простым циклом, так вершина B входит в него 2 раза. Наконец, содержащийся в пути ν путь $\lambda = \langle E, C, B, G, E\rangle$ является простым циклом ■

Имеет место утверждение, аналогичное утверждению 5 для не циклических путей.

Утверждение 6. Для всякого циклического пути (орпути) есть содержащийся в нём цикл (орцикл). Для всякого цикла (орцикла) есть содержащийся в нём простой цикл (орцикл) ■

Пример 20. В орграфе, показанном на рис.15, последовательность $\mu = \langle 1, b, 2, a, 1, d, 3, c, 1\rangle$ является циклическим орпутём, который является орциклом. Этот орцикл содержит два простых цикла: $\nu_1 = \langle 1, b, 2, a, 1\rangle$ и $\nu_2 = \langle 1, d, 3, c, 1\rangle$. Кроме них, в данном орграфе есть простой орцикл $\langle 1, d, 3, e, 2, a, 1\rangle$■

Пример 21. В орграфе, показанном на рис.16, имеется два орцикла: $\langle 1, 2, 3, 4, 5, 6, 1\rangle$ и $\langle 7, 7\rangle$. Они оба являются простыми орциклами ■

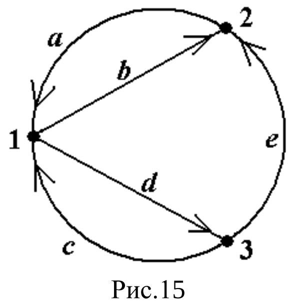


Рис.15

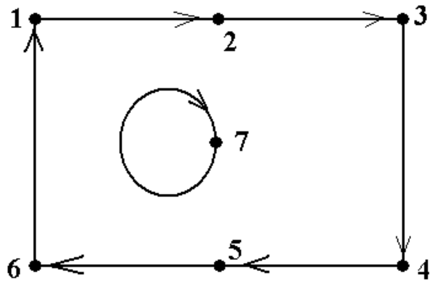


Рис.16

Введём ещё одно важное для дальнейшего понятие. Граф (орграф) называется ***ациклическим***, если он не содержит циклов (орциклов). Рассмотрим подробнее ациклические орграфы. В ациклическом орграфе по самому определению отсутствуют орциклы и, следовательно, все орпути являются простыми. Однако для ациклических орграфов верен и более специальный факт, в каком-то смысле описывающий их структуру. Имеет место

Утверждение 7. Пусть $G(V, A)$ – простой ациклический орграф с множеством вершин V и множеством дуг A. Тогда множество вершин V разбивается на непересекающиеся подмножества $V_1, \ldots, V_m$, такие, что для любой дуги (p, q) графа G начало p содержится в множестве V_iс бóльшим номером, чем конец q.

Доказательство этого утверждения, по сути, является алгоритмом построения указанных подмножеств. Поскольку граф ациклический, то в нём найдётся хотя бы одна вершина, из которой не выходят дуги. Обозначим множество всех таких вершин через V_1. Удалим из графа G все вершины из V_1 и все ведущие в них дуги. Оставшийся граф также будет ациклическим. Множество его вершин, из которых не выходят дуги, обозначим через V_2. Продолжая этот процесс, получим разбиение с требуемыми свойствами (последний оставшийся граф G_m не может содержать дуг, иначе на нём процесс не мог бы окончиться) ■

Это утверждение будет использоваться несколько раз в различных главах пособия.

Перед завершением этого раздела определим ещё одно важное «путевое» понятие. ***Маршрутом*** в графе (орграфе) называется чередующаяся последовательность μ вершин и рёбер (дуг), начинающаяся и кончающаяся в вершинах v_0 и v_k: $\mu = \langle v_0, b_1, v_1, b_2, ..., v_{k-1}, b_k, v_k\rangle$, такая, что вершины v_i и v_{i+1} соединены ребром (вершина v_i соединена дугой с вершиной v_{i+1} или вершина v_{i+1} соединена дугой с вершиной v_i). Вершина v_0 называется началом маршрута μ, вершина v_k – концоммаршрута μ, а число k, равное числу рёбер (дуг), входящих в маршрут – его длиной. Говорят также, что маршрут μ соединяет вершины v_0 и v_k. ***Обратным маршрутом*** для маршрута μ называется маршрут $\mu^{-1} = \langle v_k, b_k, v_{k-1}, b_{k-1}, ..., v_1, b_1, v_0\rangle$. Другими словами, маршрут μ^{-1} состоит из тех же самых вершин и рёбер, но проходимых в обратном порядке. Обратим внимание, что обратный маршрут, в отличие от обратного пути, определён как для графов, так и для орграфов. Дуга (v_i, v_{i+1}) называется ***прямой дугой*** маршрута μ, дуга (v_{i+1}, v_i) – ***обратной дугой*** маршрута μ.

Из приведённого определения следует, что для неориентированных графов понятия пути и маршрута совпадают. Но для орграфов это понятия является новым. Можно сказать, что маршрут – это путь, который не принимает во внимание ориентации дуг. При произвольном изменении ориентаций в дугах орграфа каждый маршрут останется тем же самым маршрутом.

4. Связность и компоненты связности

Вершина j графа (орграфа) называется ***достижимой из вершины*** i, если существует путь (орпуть) с началом в i и концом в j. Множество K вершин графа (орграфа) называется ***компонентой связности*** (***сильной связности***), если любые две вершины из K взаимно достижимы, и при этом множество K является максимальным по включению множеством, обладающим данным свойством (это означает, что при добавлении к K хотя бы одной новой вершины свойство взаимной достижимости теряется). Заметим, что множество, состоящее из одной вершины, является связным (в орграфе – сильно связным) в «силу ложности посылки», но далеко не всегда является компонентой связности (сильной связности), поскольку не обязано быть максимальным по включению. Заметим также, что множество всех вершин графа (орграфа) также не обязано быть компонентой связности (сильной связности). Если же граф (орграф) состоит из единственной компоненты, то он называется связным (сильно связным).

Пример 22. Рассмотрим граф, показанный на рис.17. У него есть три компоненты связности: $\{A, B, C, D\}$, $\{E, F, G, H, I, J\}$, $\{R\}$■

Пример 23. Рассмотрим орграф, показанный на рис.18. В нём вершины 1 и 3 взаимно достижимы (с помощью дуг a и b), поэтому они по определению входят в одну и ту же компоненту сильной связности. Каждая из вершин 2 и 4 образует отдельную компоненту сильной связности, поскольку из 3 нельзя перейти в 4, а из 4 нельзя перейти в 2. Таким образом, у данного графа имеется три компоненты сильной связности: $\{1,3\}$, $\{2\}$ и $\{4\}$■

Наличие нескольких кратных рёбер и любого числа петель, естественно, никак не сказывается на компонентах связности ни в неориентированном, ни в ориентированном случае. Поэтому в обоих случаях для нахождения компонент связности можно преобразовать граф к простому графу без петель, оставив по одному ребру (дуге) среди нескольких кратных и удалив все петли.

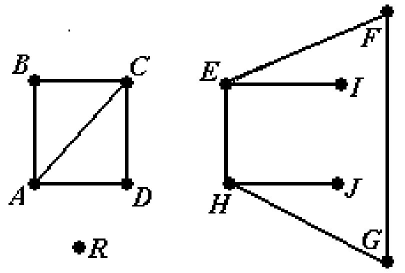


Рис.17

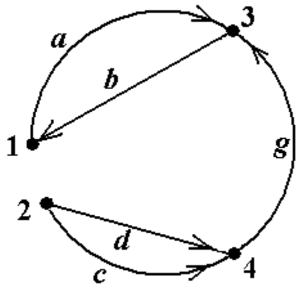


Рис.18

Пример 24. Граф, показанный на рис.19a, не является простым и содержит петлю при вершине. После удаления кратной дуги и петли (а также введения нумерации вершин) получается простой граф, показанный на рис.19b, в котором, как и в исходном графе, имеется одна компонента связности.

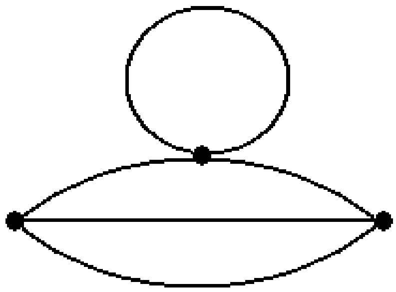


Рис.19a

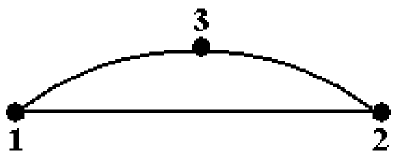


Рис.19b

Пример 25. Орграф, показанный на рис.18, не является простым, поскольку дуги c и d кратны. После удаления одной из кратных дуг получается простой орграф, показанный на рис. 20. В нём, как и в исходном орграфе, содержатся те же компоненты сильной связности: $\{1,3\}$, $\{2\}$ и $\{4\}$. Заметим, что дуги a и b в графе рис.18 не являются кратными, в отличие от дуг c и d■

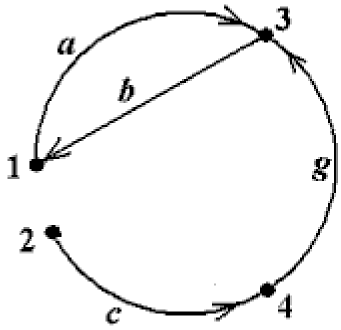


Рис.20

Компоненты связности (сильной связности) в небольших графах, показанных на всех предыдущих рисунках, находятся непосредственно (как говорят, «глазами»). Надо сказать, что «глазами» искались внутренне и внешне устойчивые множества вершин (см. примеры 7 и 8). Однако ведущую роль в решении прикладных задач дискретного характера играют алгоритмы. В этом разделе излагается алгоритм нахождения компонент связности неориентированных графов. Учитывая суть задания, можно обойтись простыми графами без петель. Аналогичный алгоритм для орграфов здесь не излагается.

Прежде, чем перейти к формальным конструкциям, рассмотрим следующий

Пример 26. На рисунке 21 показан граф, содержащий 300 вершин и около 1200 рёбер. Число компонент связности в нём равно двум. Однако даже в данном, сравнительно простом, случае проверить это «глазами», мягко говоря, затруднительно ■

4.1. Алгоритм нахождения компонент связности простых неориентированных графов. Когда речь заходит о каких бы то ни было алгоритмах, первым – и часто определяющим – вопросом является вопрос о формальном представлении рассматриваемых объектов или, как часто говорят, о выборе соответствующей структуры данных. В разделе 1.1 было дано формальное представление простых неориентированных графов в виде пары $\langle V, E\rangle$, где $V = \{1, 2, \ldots, n\}$ – множество вершин графа, $E \subseteq V^{1\text{-}2}$, где $V^{1\text{-}2}$ – множество всех 1- и 2-элементных подмножеств множества вершин V. Двухэлементное множество $\{x, y\}$, принадлежащее E, интерпретируется как ребро с концами x и y.

Однако в данном случае, как и во многих других, более адекватной структурой данных является следующая. Граф задаётся в виде массива, i-ый элемент которого соответствует вершине i. Сам же i-ый элемент является массивом, состоящим из номеров всех вершин, смежных с вершиной i. В рамках такой структуры удобно просматривать все вершины, смежные с данной, чем и оправдано их применение. Такое представление графов назовём <u>массивом смежных вершин</u>.

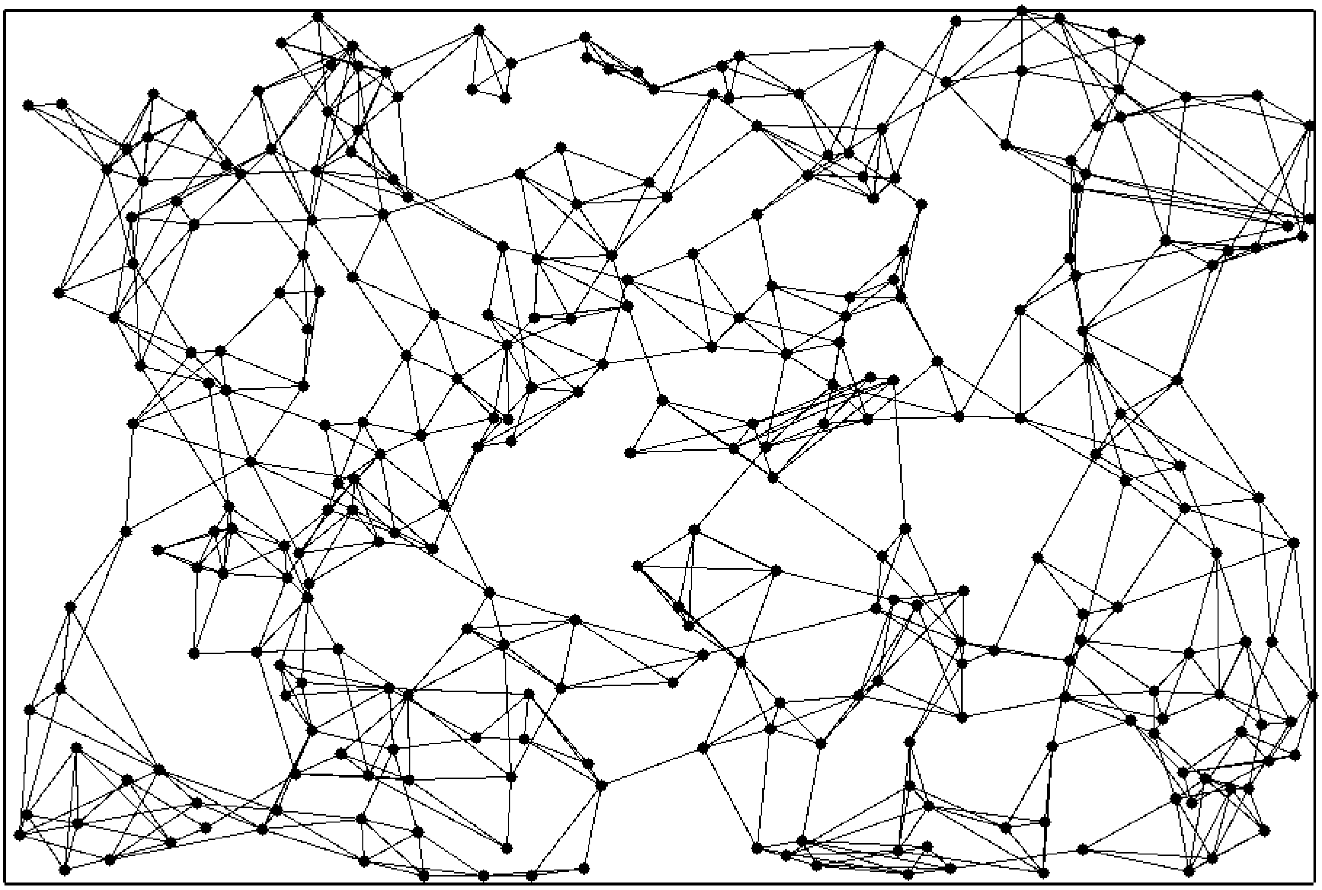


Рис.21

Для орграфов аналогичное представление определяется как массив, i-ый элемент которого является массивом номеров всех вершин, <u>в которые</u> ведёт дуга из вершины i.

Пример 27. Для графа, показанного на рис.5а, массив смежных вершин имеет следующий вид:

$\langle\langle 2\rangle, \langle 1,3,4\rangle, \langle 2\rangle, \langle 2,5,6\rangle, \langle 4,6\rangle, \langle 4,5\rangle\rangle$■

Пример 28. Для графа, показанного на рис.22 (полученным из графа на рис.17 перенумерацией вершин), указанное представление имеет следующий вид:$\langle\langle 2,3,4\rangle, \langle 1,3\rangle, \langle 1,2,4\rangle, \langle 1,3\rangle, \langle 6, 8,9\rangle, \langle 5,7,10\rangle, \langle 6,8\rangle, \langle 5,7\rangle, \langle 5\rangle, \langle 6\rangle, \Lambda\rangle$. Обратите внимание на последний (11-ый) массив: поскольку вершина 11 изолирована, кортеж смежных с ней вершин пуст, а пустые кортежи обозначаются знакомΛ(см. раздел 3-1) ■

Пример 29.Для орграфа, показанного на рис.23, указанное представление имеет следующий вид: $\langle\langle 3,4\rangle, \langle 1,4\rangle, \langle 2,4\rangle, \Lambda\rangle$. На 4-ой позиции стоит пустой кортеж, так как из вершины 4 не выходит ни одна дуга■

Далее рассматривается основной материал данного раздела. Предлагается алгоритм, по массиву смежных вершин определяющий компоненты связности графа. Суть его такова. Определим массив M длины n (через n обозначено число вершин графа). Присвоение i-ой вершине метки a означает формальную операцию присвоения $M[i] = a$. Начинаем с любой вершины (для определённости, с первой). Присваиваем ей метку –1.

Пусть некоторое множество вершин имеет метки –1 и +1. Возьмём любую вершину x с меткой –1 (если таковая найдётся) и у всех вершин, смежных с ней и ещё не имеющих метку,

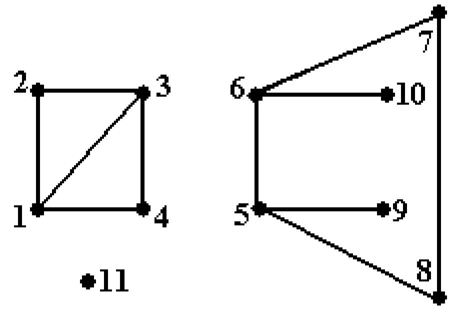


Рис.22

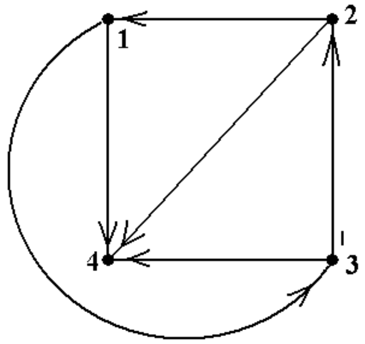


Рис.23

поставим метку –1. После этого, <u>независимо от того, была ли помечена хотя бы одна вершина</u>, у вершины x заменим метку - –1 на +1. Если же ни одной вершины с меткой - –1 не найдётся, то все вершины с меткой +1 образуют 1-ую компоненту связности. Если больше непомеченных вершин нет, то граф состоит из одной компоненты связности. Если есть хоть одна непомеченная вершина y, то пометим её меткой –2 и повторим весь процесс с заменой 1 на 2. Далее, если ещё остались непомеченные вершины, пометим любую из них меткой –3, и т.д, вплоть до того, когда все вершины становятся помеченными. При этом все вершины с одной и той же меткой i образуют i-ую компоненту связности.

Пример 30. Применим описанный алгоритм для графа из примера 28 (рис.22). Для него массив смежных вершин таков: $\langle\langle 2,3,4\rangle, \langle 1,3\rangle, \langle 1,2,4\rangle, \langle 1,3\rangle, \langle 6,8,9\rangle, \langle 5,7,10\rangle, \langle 6,8\rangle, \langle 5,7\rangle, \langle 5\rangle, \langle 6\rangle, \Lambda\rangle$.

1. Инициализация. В данном случае число вершин равно 11. Поэтому определим массив M длины 11 и положим $M[1] = –1$.
2. В соответствии с алгоритмом положим $M[2] = –1$, $M[3] = –1$, $M[4] = –1$.
3. Поскольку помечены все вершины из списка смежных с вершиной 1 вершин 2, 3, 4, то положим $M[1] = 1$.
4. Возьмём следующую вершину 2 с меткой –1. Смежные с ней вершины 1, 3 уже помечены. В соответствии с алгоритмом положим $M[2] = 1$.
5. Возьмём следующую вершину 3 с меткой –1. Смежные с ней вершины 1, 2, 4 уже помечены. В соответствии с алгоритмом положим $M[3] = 1$.
6. Возьмём следующую вершину 4 с меткой –1. Смежные с ней вершины 1, 3 уже помечены. В соответствии с алгоритмом положим $M[4] = 1$.
7. Поскольку более вершин с меткой –1 нет ($M[1] = M[2] = M[3] = M[4] = 1$), то выбираем следующую непомеченную вершину 5 и присваиваем ей метку $M[5] = –2$.
8. В соответствии с алгоритмом для смежных с вершиной 5 вершин 6, 8, 9 положим $M[6] = –2$, $M[8] = –2$, $M[9] = –2$.
9. Поскольку помечены все вершины из списка смежных с вершиной 5 вершин 6, 8, 9, то положим $M[5] = 2$.
10. Возьмём следующую вершину 6 с меткой –2. Смежными с ней вершинами является 5 (уже помеченная), а также 7 и 10 (ещё непомеченные). В соответствии с алгоритмом положим $M[7] = –2$, $M[10] = –2$.
11. В соответствии с алгоритмом положим $M[6] = 2$.

12. Возьмём следующую вершину 7 с меткой –2. Смежными с ней вершинами является 6 и 8 (уже помеченные). В соответствии с алгоритмом положим $M[7] = 2$.

13. Возьмём следующую вершину 8 с меткой –2. Смежными с ней вершинами является 5 и 7 (уже помеченные). В соответствии с алгоритмом положим $M[8] = 2$.

14. Возьмём следующую вершину 9 с меткой –2. Смежной с ней вершинами является только 5 (уже помеченная). В соответствии с алгоритмом положим $M[9] = 2$.

15. Возьмём следующую вершину 10 с меткой –2. Смежной с ней вершинами является только 6 (уже помеченная). В соответствии с алгоритмом положим $M[10] = 2$.

16. Поскольку вершин с меткой –2 нет ($M[5] = M[6] = M[7] = M[8] = M[9] = M[10] = 2$), то выбираем следующую непомеченную вершину 11 и присваиваем ей метку $M[11] = –3$.

17. В соответствии с алгоритмом просматриваем все вершины, смежные с 11. Поскольку вершин, смежных с 11, не существует, то меняем метку $M[11]$: $M[11] = 3$.

18. Поскольку более вершин с меткой –3 не существует, то в соответствии с алгоритмом ищем непомеченную вершину. Так как таковых не имеется, то алгоритм останавливается.

После остановки имеем следующий массив M: $M = \langle 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3\rangle$. Это и означает, что 1-ую компоненту связности образуют вершины 1, 2, 3, 4; 2-ую компоненту связности – вершины 5, 6, 7, 8, 9, 10; 3-ью компоненту связности – изолированная вершина 11■

Число вершин, смежных вершине неориентированного графа, называется ***степенью*** данной вершины.

5. Эйлеровы циклы

5.1. Кёнигсбергские мосты. Не каждому городу выпадает честь быть отмеченным в такой точной науке, как классическая математика. Кёнигсберг же благодаря своим мостам и великому учёному-математику XVIII-го века Леонарду Эйлеру вошёл в число математических знаменитостей.

Рис.24. Великий математик Леонард Эйлер (1707 – 1783)

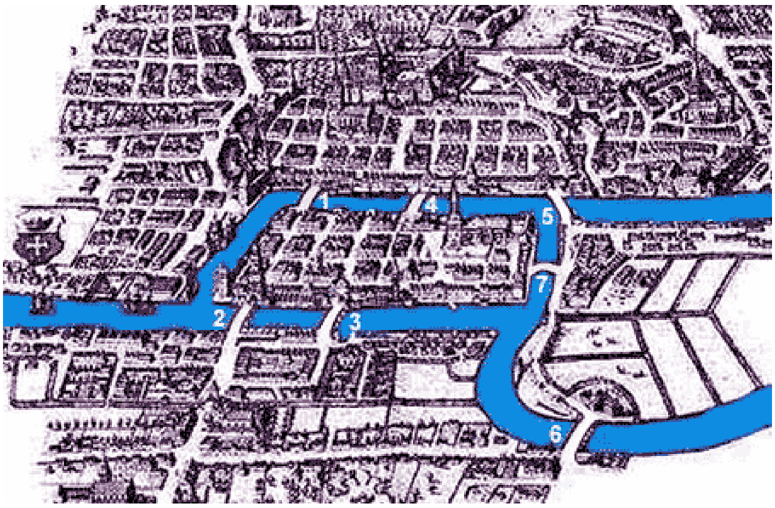


Рис.25. Кенигсберг начала XVIII века с мостами

Леонард Эйлер (Leonhard Euler) родился 15 апреля 1707 г. в маленькой тихой Швейцарии, куда изо всей Европы приезжали мастера и учёные, не желавшие тратить дорогое рабочее время на частые в других странах гражданские смуты и религиозные распри. Так переселилась в Базель из Голландии и семья Бернулли, давшая миру уникальное созвездие научных талантов во главе с братьями Якобом и Иоганном. Семья небогатого протестантского пастора Пауля Эйлера жила в доме Бернулли. Иоганн Бернулли быстро заметил и оценил талант Леонарда, стал его наставником, а сыновья Иоганна Николай и Даниил – его друзьями.

Более всего поражает способность великого учёного говорить о сложных вещах простым языком. Для решения серьёзных математических задач Эйлер использовал наглядные головоломки. Одна из них положила начало совершенно новой области исследований, выросшей впоследствии в самостоятельный раздел математики – теорию графов. Особенность этой теории – в геометрическом подходе к изучению объектов. Живя в Кёнигсберге, прогуливаясь по его набережным, Эйлер обратил внимание на оригинальное расположение семи мостов города. Причиной этому было причудливое течение рукавов Прегеля, соединенных протокой, охватывающих с севера и юга остров Кнайпхоф, а затем сливающихся вместе (рис.25). Эти мосты назывались так:

1. Kramer-Brücke (Лавочный мост)
2. Grüne Brücke (Зелёный мост)
3. Köttel-Brücke (Потроховый мост)
4. Schmiede-Brücke (Кузнечный мост)
5. Holz-Brücke (Деревянный мост)
6. Hohe-Brücke (Высокий мост)
7. Honig-Brücke (Медовый мост)

В 1736 Эйлер решил задачу, известную с тех пор как задача о кенигсбергских мостах: можно ли пройти по всем этим мостам и при этом вернуться в исходную точку так, чтобы по каждому мосту пройти только один раз. На рис.26 изображена условная схема семи мостов Кёнигсберга (заметим, что сейчас осталось только два из них).

Упростим схему, показанную на рис.26. Для этого представим каждый из четырёх участ-ков суши одной точкой, а каждый из мостов – линией, соединяющей соответствующие точки. В результате получаем граф, показанный на рис.3. Обратим внимание на то, что данный граф не является простым (см. пример 2). Нетрудно понять, что исходная задача стала значительно более обозримой. В уже введённых терминах (см. раздел 4) вопрос ставится так: существует ли в графе на рис.3 цикл, проходящий по всем рёбрам? Напомним, что однократность прохождения каждого ребра включена в определение цикла в разделе 4. Эйлер дал общий ответ на поставленный вопрос в следующем виде.

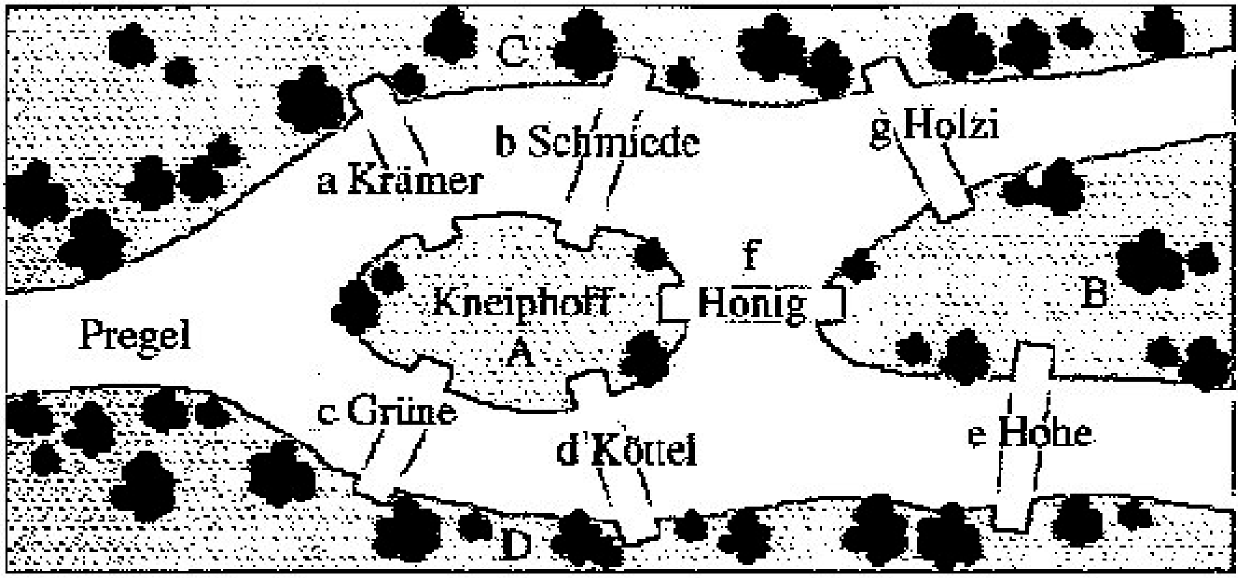


Рис.26

Теорема. Для того чтобы данный связный граф имел цикл, проходящий по всем рёбрам, необходимо и достаточно, чтобы степени всех вершин графа были чётными (степень вершины определена в самом конце предыдущего раздела 4). Для того чтобы данный связный граф имел цепь между вершинами A и B, проходящую по всем рёбрам, необходимо и достаточно, чтобы степени всех вершин, кроме A и B, были чётными, а степени этих вершин были нечётными■

Поскольку в графе на рис.3 степени некоторых вершин нечётны, то, следовательно, требуемого цикла не существует. В честь Эйлера циклы в неориентированных графах, проходящие по всем рёбрам, были названы ***эйлеровыми циклами***, как и соответствующие цепи.

5.2. Алгоритм Флёри. Теорема Эйлера сама по себе является теоремой существования: в ней не говорится о том, как построить эйлеров цикл. Эффективный алгоритм был предложен французским математиком Флёри в 1873 году – почти через 150 лет после открытия Эйлера. Для его изложения нам понадобится важное понятие перешейка в графе. ***Перешеек*** – это ребро, удаление которого разделяет связную компоненту графа на две несвязные части.

Пример 31. Найдём все перешейки в графе, показанном на рис.27*a*. Граф состоит только из одной компоненты связности, так что надо найти каждое ребро, удаление которого делает граф несвязным. Ребро {3,5} является перешейком; его удаление делает граф несвязным, как показано на рис. 27*b*. Ребро {7,8} также является перешейком; его удаление делает граф несвязным, как показано на рис.27*c*; при этом вершина 8 образует отдельную компоненту связности. Удаление ребра {6,7} не приводит к разделению графа, как показано на рис.27*d*; то же самое относится к удалению любого ребра (кроме {3,5}и {7,8}).

Алгоритмтм Флёри.

1. Начинаем с любой вершины. Проходим вдоль любого ребра до другой вершины. Запоминаем это ребро как начало цикла и удаляем его из графа.
2. Пусть мы достигли некоторой вершины. Выбираем любое выходящее из неё ребро, с одним условием: не выбираем перешеек, если есть другая возможность. Проходим вдоль выбранного ребра до другой вершины. Добавляем это ребро в цикл и удаляем его из графа.
3. Выполняем шаг 2 до тех пор, пока не удалим все рёбра из графа ■

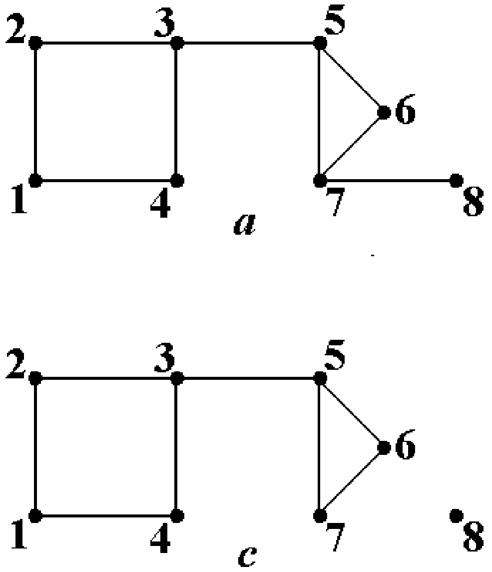


Рис.27

Пример 32. Применим алгоритм Флёри к графу, показанному на рис.28*a*. Прежде всего проверяем, что граф связен и степени всех вершин чётны. По теореме Эйлера из этого следует, что эйлеров цикл в таком графе должен быть.

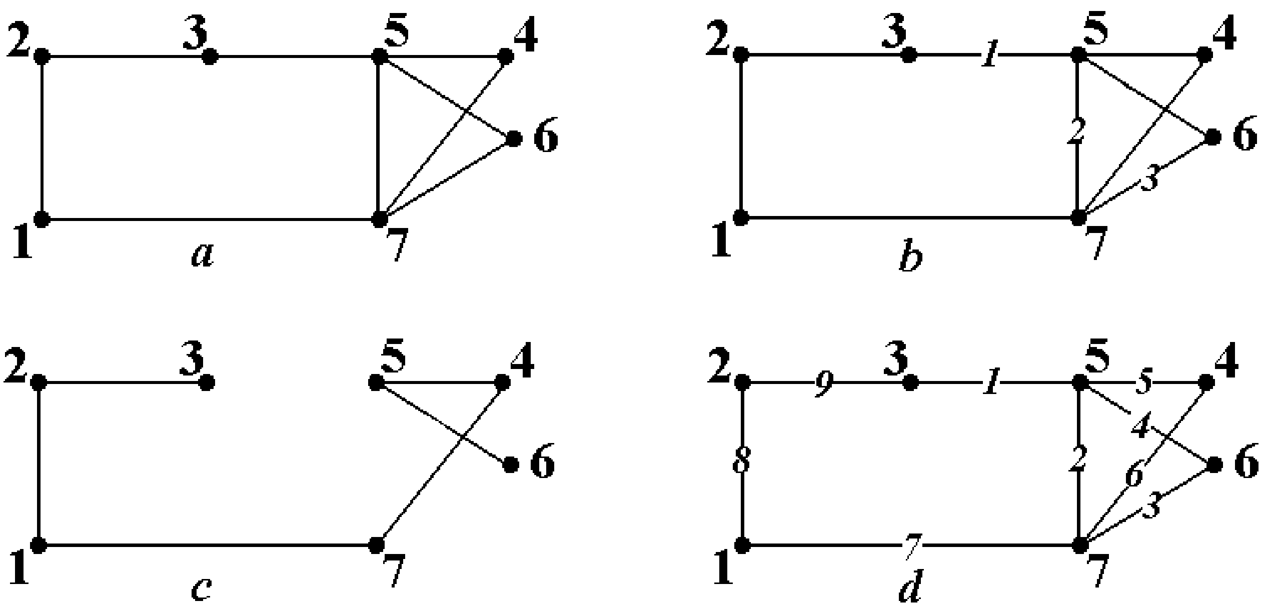


Рис.28

Начинаем процесс с вершины 3. Из 3 можно двигаться либо в 2, либо в 5. Пойдём по направлению к 5, удалив ребро {3,5} (на рисунке 28*b* указаны номера удаляемых рёбер в порядке их удаления). Таким образом, началом искомого цикла является ребро {3,5}. Далее можно двигаться в вершины 4, 6 или 7. Поскольку ни одно из рёбер {5,4}, {5,6} и {5,7} в графе после удаления ребра {3,5} не является перешейком, то можно выбрать в качестве 2-го ребра цикла любое из этих трёх рёбер. Выбираем ребро {5,7}. На рис.28*b* это ребро имеет номер 2, а начало искомого цикла выглядит так: 3→5→7.

Далее, из вершины 7можно перейти в вершины 1, 4 или 6. Однако ребро {7,1} является перешейком в графе, полученном удалением рёбер {3,5} и {5,7} (см. рис.28*b*). Так как есть другие возможности ({7,4} и {7,6}), в соответствии с алгоритмом Флёри выберем одну из них. Выбрав {7,6}, продлеваем строящийся цикл этим самым ребром {7,6} и получаем 3→5→7→6; удаляемые рёбра и модифицированный граф показаны на рис.28*b*. Заметим, что выбор перешейка {7,1} в графе, показанном на рис.28*b*, и его последующее удаление привели бы к невозможности когда-либо пройти по рёбрам {7,4} и {7,6}. Это поясняет условие на шаге 2 алгоритма Флёри.

Граф, оставшийся после удаления первых трёх рёбер искомого цикла – {3,5}, {5,7} и {7,6} – показан на рис.28*c*. Из очередной вершины 6 в графе рис.28*c* выходит только одно ребро {6,5}, которое <u>является</u> перешейком в этом графе. В соответствии с алгоритмом Флёри, мы должны двигаться далее по этому ребру {6,5}, несмотря на то, что {6,5} является перешейком, так как <u>других возможностей нет</u>. Далее все последующие рёбра определяются также совершенно однозначно (см. рис.28*c*). В результате получаем эйлеров цикл 3→5→7→6→5→4→7→1→2→3. В него по одному разу входят все 9 рёбер рассматриваемого графа. Полная последовательность удаляемых по порядку рёбер представлена на рис.28*d*■

6. Двудольные графы

В заключение данной главы рассмотрим один специальный класс простых неориентированных графов – так называемые ***двудольные графы***. Они широко используются в приложениях. Граф называется двудольным, если множество его вершин можно разбить на два непустых подмножества (доли), так что любое его ребро соединяет две вершины из разных долей (эквивалентно: концы любого ребра принадлежат разным долям). Из этого определения следует, что двудольный граф не содержит петель. Множество рёбер двудольного графа, которые не имеют общих вершин, называется ***паросочетанием***. Паросочетание, содержащее максимальное число рёбер, называется ***максимальным паросочетанием***. Алгоритмы поиска максимальных (а также некоторых других) паросочетаний рассмотрены в главе 9 и разделе 13-1.

Пример 33. Двудольный граф показан на рис.29. Множество X = {{1,7}, {3,11}, {5,10}}, состоящее из трёх рёбер является паросочетанием, поскольку входящие в него рёбра не имеют общих концов. Множество Y = {{4,7}, {1,8}, {3,9}, {5,10}, {6,11}}, состоящее из пяти рёбер, также является паросочетанием. Его максимальность следует из того, что «нижняя» доля содержит 5 вершин, в силу чего паросочетания, состоящего более чем из 5 рёбер, просто не существует – если бы оно было, то по крайней мере два его ребра были бы инцидентны одной вершине, что противоречит определению паросочетания ■

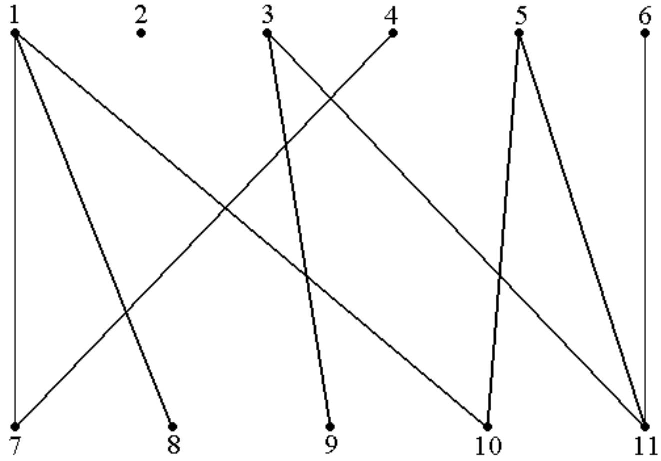


Рис.29. Двудольный граф

Задания

Задание 1. По заданному формальному описанию нарисовать граф (см. примеры 1 – 5).

1. $G = \langle V, E, F\rangle$, где $V = \{1, 2, 3, 4\}$, $E =\{a, b, c, d, e, f, g, h\}$, $F(a) = F(b) = \{1, 3\}$, $F(c) = F(d) = \{2, 4\}$, $F(e) = \{2, 3\}$, $F(f) = \{1, 4\}$, $F(g) = \{3, 4\}$, $F(h) = \{1, 2\}$.

2. $G = \langle V, E\rangle$, где $V = \{1, 2, 3, 4\}$, $E = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{1, 4\}, \{2, 4\}\}$.

3. $G = \langle V, E, F\rangle$, где $V = \{1, 2, 3\}$, $E =\{a, b, c, d, e, f\}$, $F(a) = F(b) = \{1, 2\}$,$F(c) = F(d) = \{1, 3\}$, $F(e) = F(f) = \{2, 3\}$.

4.$G = \langle V, E, F\rangle$, где $V = \{1, 2, 3\}$, $E =\{a, b, c, d, e\}$, $F(a) = F(b) = \{1, 2\}$,$F(c) = F(d) = \{1, 3\}$, $F(e) = \{2, 3\}$.

5. $G = \langle V, E\rangle$, где $V = \{1, 2, 3, 4\}$, $E = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}$.

6. $G = \langle V, E\rangle$, где $V = \{1, 2, 3, 4, 5, 6\}$, $E = \{\{2\}, \{6\}, \{1, 4\}, \{1, 5\}, \{1, 6\}, \{2, 5\}, \{3, 6\}\}$.

7. $G = \langle V, E, F\rangle$, где $V = \{1, 2, 3, 4\}$, $E = \{a, b, c\}$, $F(a) = F(b) = \{1, 2\}$, $F(c) = \{3, 4\}$.

8. $G = \langle V, E\rangle$, где $V = \{1, 2, 3, 4, 5, 6, 7\}$, $E = \{\{7\}, \{1, 2\}, \{2, 3\}, \{3, 4\}, \{5, 6\}, \{1, 6\}\}$.

9. $G = \langle V, E\rangle$, где $V= \{1, 2, 3, 4, 5\}$, $E = \{\{5\}, \{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{1, 5\}\}$.

10. $G = \langle V, E, F\rangle$, где $V = \{1, 2, 3\}$, $E = \{a, b, c, d, e\}$, $F(a) = F(b) = \{1, 2\}$, $F(c) = \{1, 3\}$, $F(d) = \{2, 3\}$, $F(e) = \{3\}$.

11. $G = \langle V, E, F\rangle$, где $V = \{1, 2\}$, $E = \{a, b, c\}$, $F(a) = F(b) = F(c) = \{1, 2\}$.

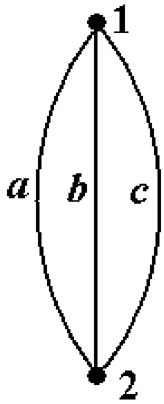
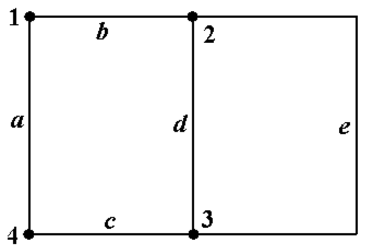
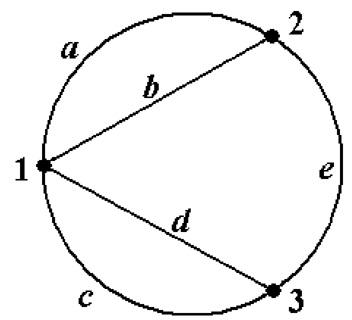
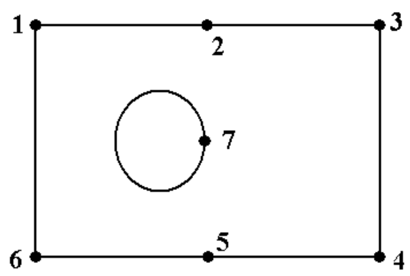
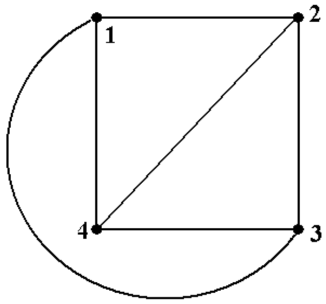
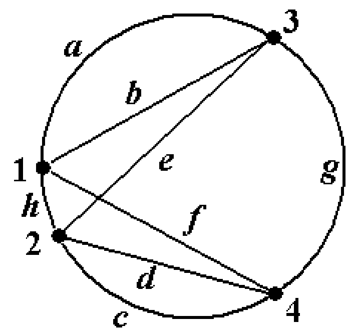
12. $G = \langle V, E\rangle$, где $V = \{1, 2, 3, 4\}$, $E = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{1, 4\}, \{2, 4\}, \{1, 3\}\}$.

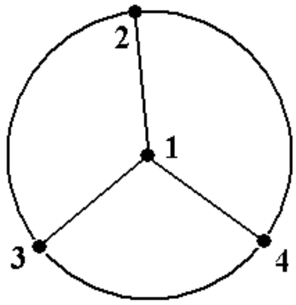
13. $G = \langle V, E\rangle$, где $V = \{1, 2, 3, 4, 5, 6\}$, $E = \{\{1, 4\}, \{1, 5\}, \{1, 6\}, \{2, 4\}, \{2, 5\}, \{2, 6\}, \{3, 4\}, \{3, 5\}, \{3, 6\}\}$.

14. $G = \langle V, E, F\rangle$, где $V = \{1, 2, 3, 4\}$, $E = \{a, b, c, d\}$, $F(a) = F(b) = \{1, 3\}$, $F(c) = \{1, 4\}$,$F(d) = \{2, 3\}$.

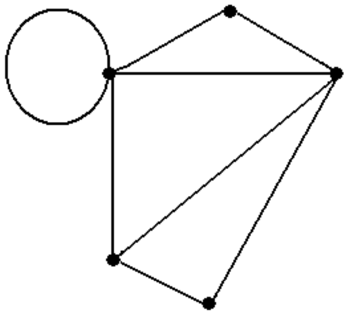
15. $G = \langle V, E, F\rangle$, где $V = \{1, 2, 3, 4\}$, $E = \{a, b, c, d, e\}$, $F(a) = \{1, 4\}$,$F(b) = \{1, 2\}$, $F(c) = \{3, 4\}$, $F(d) = F(e) = \{2, 3\}$ ■

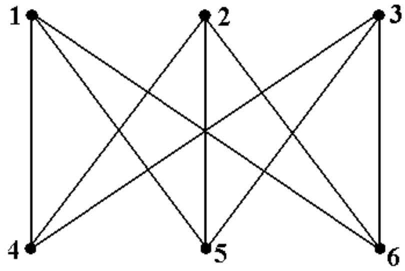
Варианты неориентированных графов для заданий 2, 4, 6, 8, 10, 12

<table>
<tr>
<td>
01</td>
<td>
02</td>
<td>
03</td>
</tr>
<tr>
<td>
04</td>
<td>
05</td>
<td>
06</td>
</tr>
</table>

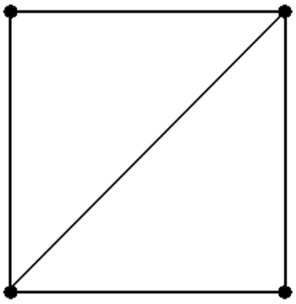


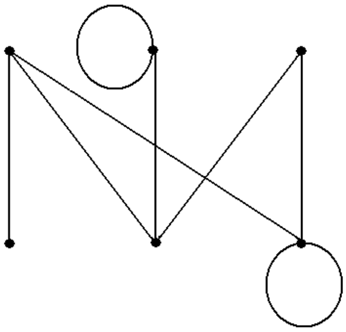
07

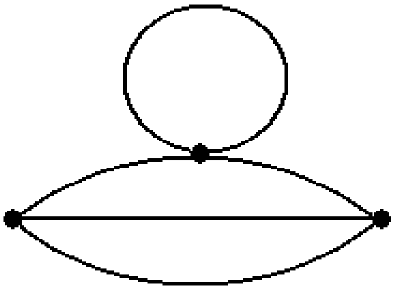

08

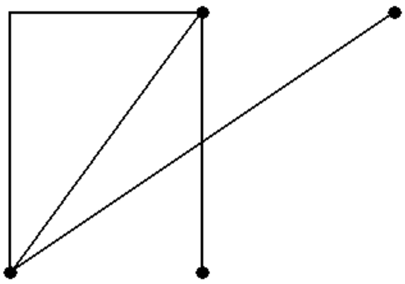


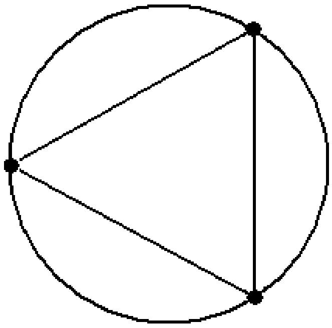
09

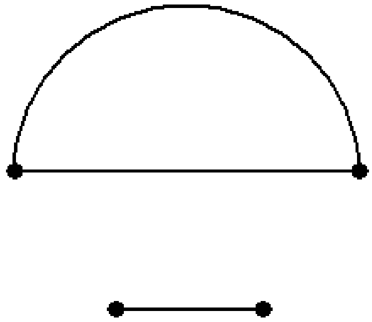

10

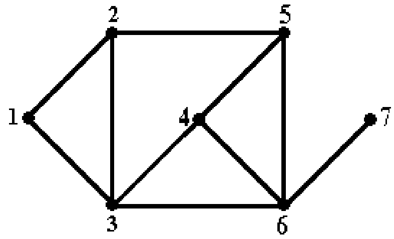

11


12

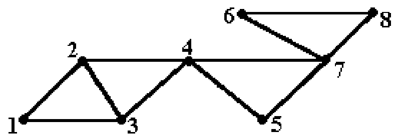

13


14

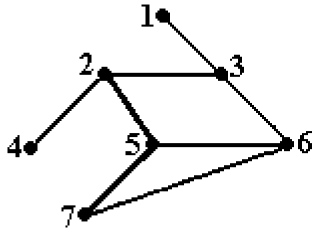

15



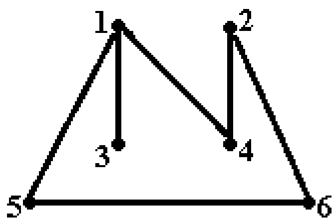
16



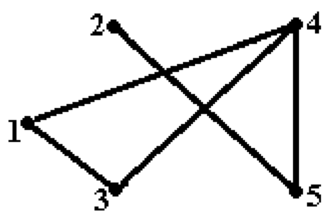
17



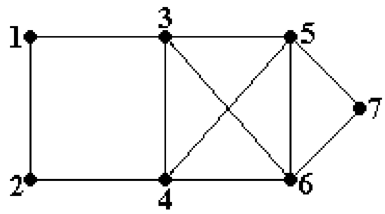
18



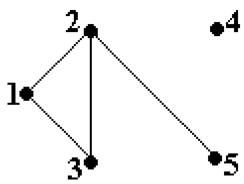
19



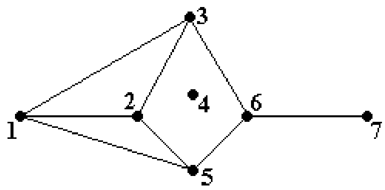
20



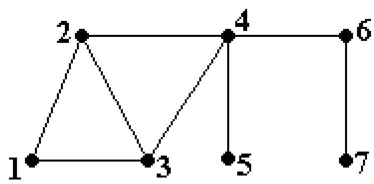
21



22

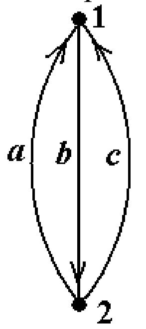


23

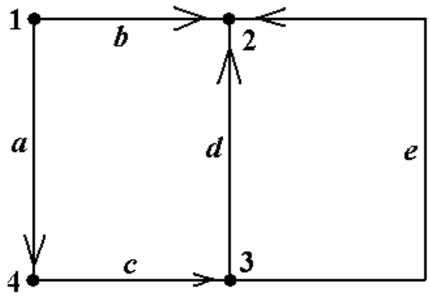


24

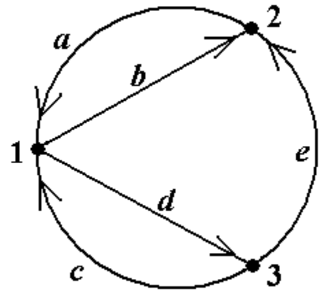
Варианты ориентированных графов для заданий 3, 5, 7, 9, 11, 13



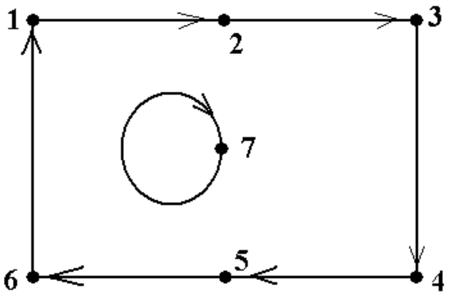
01



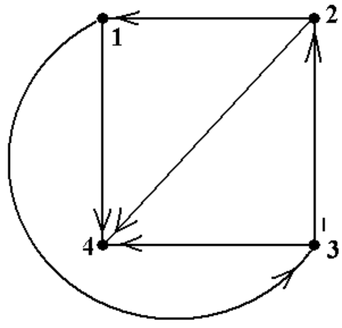
02



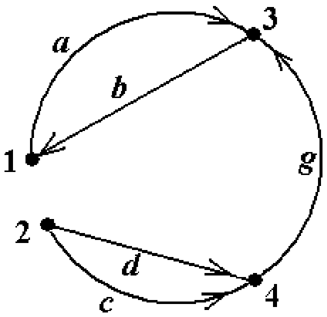
03



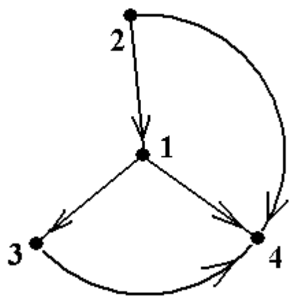
04



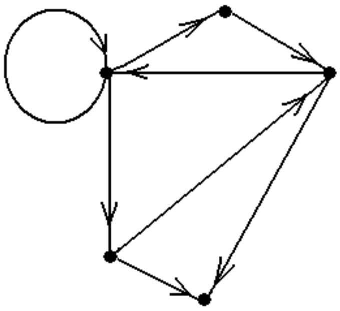
05

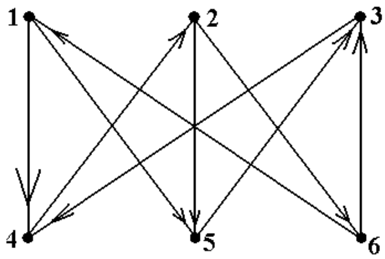


06

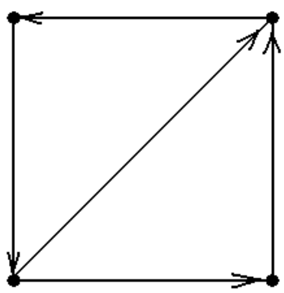


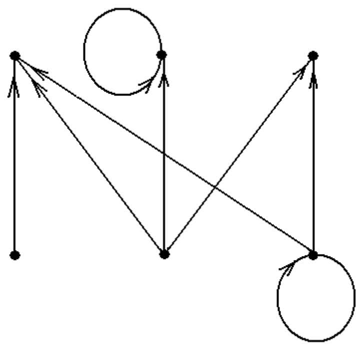
07

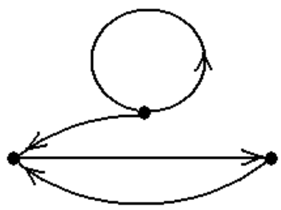

08

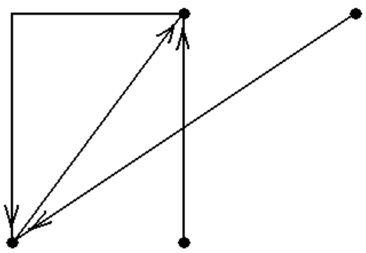


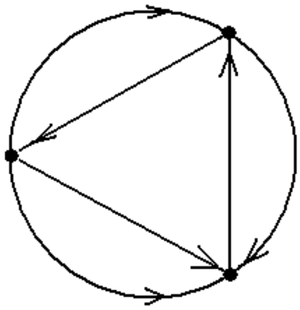
09

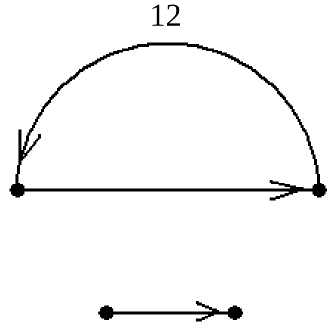

10


11


12


13


14


15

Задание 2. Представить заданный граф парами $\langle V, E\rangle$ или тройками $\langle V, E, F\rangle$ (см. примеры 2 – 5). При необходимости дать имена рёбрам и вершинам ■

Задание 3. Представить заданный орграф парами $\langle V, A\rangle$ или тройками $\langle V, A, F\rangle$ (см. пример 6). При необходимости дать имена дугам и вершинам ■

Задание 4. В заданном графе найти все максимальные по включению внутренне устойчивые множества вершин, все минимальные по включению внешне устойчивые множества вершин и все ядра (см. примеры 7 – 9) ■

Задание 5. В заданном орграфе найти все максимальные по включению внутренне устойчивые множества вершин, все минимальные по включению внешне устойчивые множества вершин и все ядра (если они есть). Если ядер нет, то указать на это и объяснить, в чём дело (см. примеры 10 и 11) ■

Задание 6. В данном графе найти:

1) путь, не являющийся цепью;
2) содержащуюся в найденном в 1) пути цепь, не являющуюся простой, соединяющую те же самые вершины;
3) содержащуюся в найденной в 2) цепи простую цепь, соединяющую те же самые вершины.

Если некоторые искомые объекты не существуют (как в примере 16), то объяснить это (см. примеры 14 – 17)■

Задание 7. В данном орграфе найти:

1) орпуть, не являющийся орцепью;
2) не содержащуюся в найденном в 1) орпути орцепь, не являющуюся простой;
3) не содержащуюся в найденной в 2) орцепи простую орцепь.

Если некоторые искомые объекты не существуют, то объяснить это (см. примеры 20 и 21)■

Задание 8. В данном графе найти:

1) циклический путь, не являющийся циклом;
2) содержащийся в найденном в 1) пути цикл, не являющийся простым;
3) содержащийся в найденном в 2) цикле простой цикл. (см. примеры 18 – 19) ■

Если некоторые искомые объекты не существуют, то объяснить это ■

Задание 9. В данном орграфе найти:

1) циклический орпуть, не являющийся орциклом;
2) содержащийся в найденном в 1) орпути орцикл, не являющийся простым;
3) содержащийся в найденном в 2) орцикле простой орцикл.

Если некоторые искомые объекты не существуют, то объяснить это (см. примеры 20 и 21) ■

Задание 10. В данном графе «найти глазами» компоненты связности (см. примеры 22 и 24) ■

Задание 11. В данном орграфе «найти глазами» компоненты сильной связности (см. примеры 23 и 25) ■

Задание 12. Привести данный граф, если он содержат кратные рёбра и/или петли, к простому графу без петель. В качестве ответа нарисовать пару «исходный – изменённый граф». Если исходный граф не содержал кратных рёбер и петель, то эту часть задания не выполнять. Описать изменённый (или исходный) граф парой $\langle V, E\rangle$ как это делалось в примерах 4 и 5. ■

Задание 13. Привести данный орграф, если он содержат кратные дуги и/или петли, к простому орграфу без петель. В качестве ответа нарисовать пару «исходный – изменённый орграф». Если исходный орграф не содержал кратных рёбер и петель, то эту часть задания не выполнять. Описать изменённый (или исходный) орграф парой $\langle V, A\rangle$ как это делалось в примере 6■

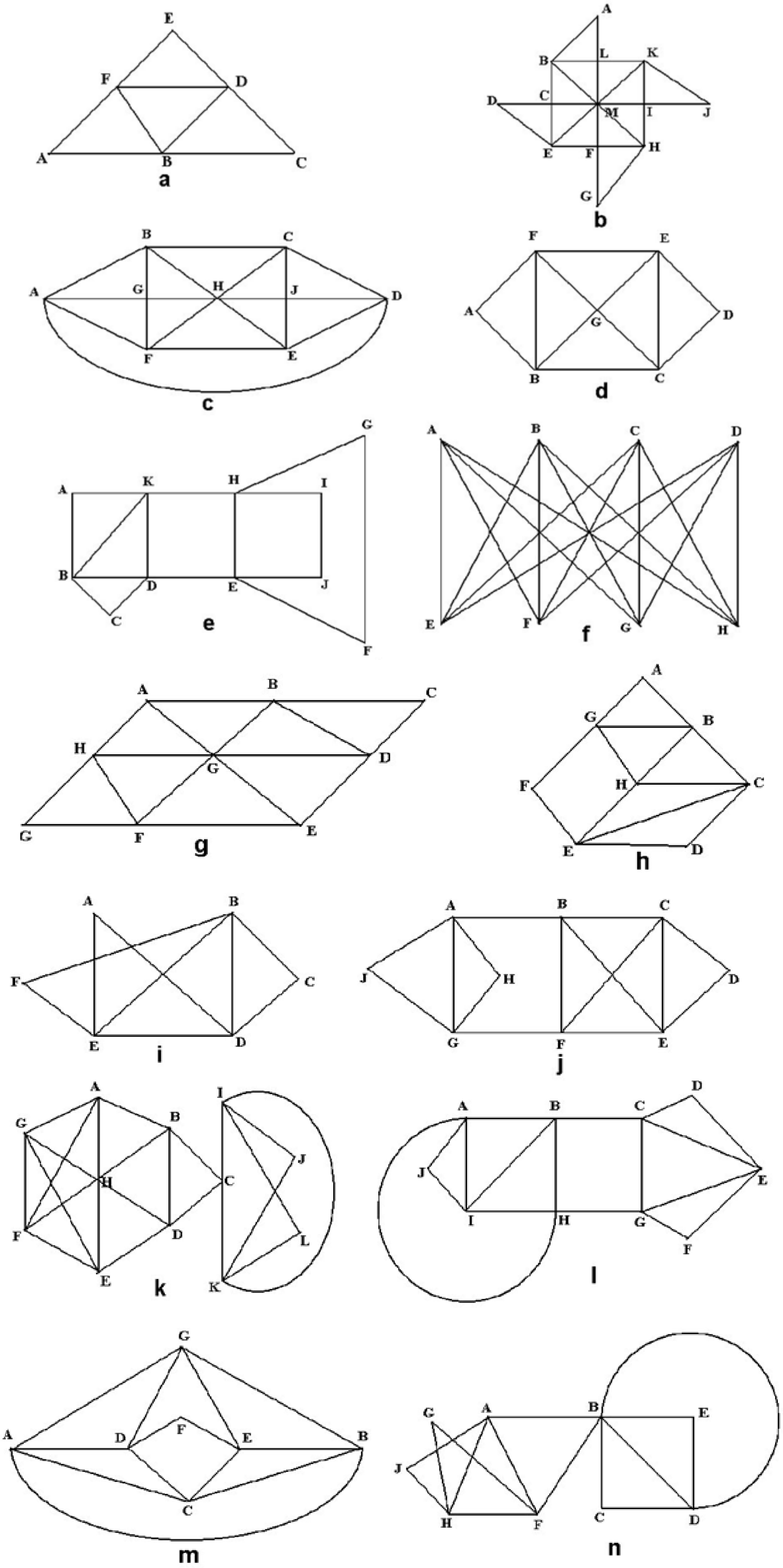
Задание 14. Граф, полученный в результате выполнения задания 12, представить в виде массива смежных вершин (см. примеры 27 – 29) ■

Задание 15. Орграф, полученный в результате выполнения задания 13, представить в виде массива смежных вершин (см. примеры 27 – 29) ■

Задание 16. Для графа, рассмотренного в задании 14, найти указанным в примере 30 алгоритмом компоненты связности ■

Задание 16. В данном неориентированном графе, найти эйлеров цикл, пользуясь алгоритмом Флёри. Решение должно быть представлено, как в примере 32. Это означает, что должны быть подробно показаны все шаги и должны быть отмечены все встречаемые перешейки, вплоть до ситуации, когда дальнейшие рёбра определяются совершенно однозначно ■

Варианты графов для задания 16 показаны на следующем рисунке.



7. Предметный указатель

Глава 7. Потоки в сетях

Во многих практически важных случаях функционирование системы, моделируемой ориентированным графом, определяется передачей между её отдельными частями (которым сопоставлены вершины графа) некоторых потоков (материальных, энергетических, информационных). Теория, в рамках которой изучаются распределения потоков, называется теорией потоков в сетях. Распределение потоков во многих случаях оценивается некоторым числом, характеризующим эффективность данного распределения. Поэтому задача оптимизации распределения потоков является одной из основных задач теории потоков в сетях. Формальной постановке и решению базовой задачи этой теории – задаче о максимальном потоке – и посвящена настоящая глава.

1. Определение потоковой сети.

Потоковая сеть представляет собой ориентированный граф, удовлетворяющий некоторым специальным условиям и обладающий некоторыми дополнительными (по отношению к произвольным графам) параметрами. Необходимые понятия и определения теории графов даны в предыдущей главе 6; будем пользоваться введёнными там понятиями без дополнительных разъяснений, останавливаясь лишь на новых (ранее не определённых) понятиях и терминах.

При определении сети в базовом простейшем случае рассматривают ориентированные связные графы, обладающие следующими специальными свойствами:
1) существует ровно одна вершина, в которую не входит ни одна дуга (эта вершина называется ***источником***);
2) существует ровно одна вершина, из которой не выходит ни одна дуга (эта вершина называется ***стоком***);
3) ни одна дуга не является петлёй.

Потоковой сетью называется граф указанного типа, у которого каждой дуге $v_j (j = 1, 2, ..., m)$ сопоставлено положительное число $c_j (j = 1, 2, ..., m)$, называемое ***пропускной способностью*** дуги v_j. Пример потоковой сети приведен на рис.1. Пропускные способности проставлены в кружках около дуг.

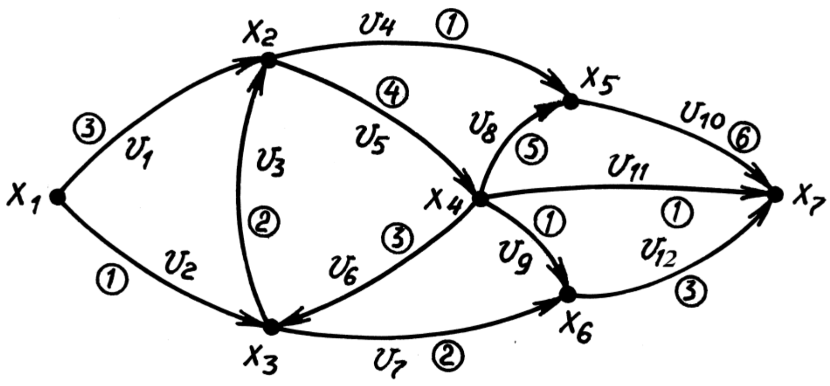


Рис.1

Таким образом, потоковая сеть – это ориентированный граф, в котором заданы пропускные способности дуг. В дальнейшем будем удобно занумеровать вершины графа так, чтобы вершина с минимальным номером (x_1) являлась источником, а вершина с максимальным номером (x_n) являлась стоком сети. Потоковую сеть будем обозначать через S.

Введем обозначения:

A_i^+ – множество номеров всех дуг, входящих в вершину x_i;

A_i^- – множество номеров всех дуг, выходящих из вершины x_i.

Пример 1. В сети, показанной на рис.1:

$n = 7$, $m = 12$,

$A_1^- = \{1,2\}$,

$A_2^+ = \{1,3\}$, $A_2^- = \{4,5\}$, $A_3^+ = \{2,6\}$, $A_3^- = \{3,7\}$, $A_4^+ = \{5\}$, $A_4^- = \{6,8,9,11\}$,

$A_5^+ = \{4,8\}$, $A_5^- = \{10\}$, $A_6^+ = \{7,9\}$, $A_6^- = \{12\}$,

$A_7^+ = \{10,11,12\}$,

$A_1^+ = A_7^- = \varnothing$ (по определению сети, так как у источника нет входящих, а у стока - выходящих дуг);

$c_1=3$, $c_2=1$, $c_3=2$, $c_4=1$, $c_5=4$, $c_6=3$, $c_7=2$, $c_8=5$, $c_9=1$, $c_{10}=6$, $c_{11}=1$, $c_{12}=3$ ■

Потоком в сети называется вектор $u = (u_1, u_2, ..., u_m)$ удовлетворяющий условиям

$0 \le u_j \le c_j (j = 1, 2, ..., m)$, (1)

$\sum_{j\in A_i^+} u_j = \sum_{j\in A_i^-} u_j$ $(i = 2, 3, ..., n-1)$. (2)

Компонента u_j вектора $u = (u_1, u_2, ..., u_m)$ называется ***потоком по дуге*** v_j $(j = 1, 2, ..., m)$. Таким образом, условие (1) означает, что поток по любой дуге – неотрицательное число, не превосходящее пропускной способности этой дуги. Условие (2) означает, что сумма потоков во всех дугах, входящих в вершину, равна сумме потоков во всех дугах, выходящих из этой же вершины (кроме источника и стока).

Содержательно поток описывает распределение по дугам сети некоторого продукта, доставляемого из источника в сток. При этом поток u_j по дуге v_j – это количество продукта, передаваемого по данной дуге от её начала к её концу. Условие (1) выражает ограничение на поток по каждой дуге: нельзя передать больше, чем эта дуга может пропустить. Условие (2) выражает закон сохранения: сколько продукта поступило в промежуточную вершину, столько же оттуда отправлено далее.

Величиной $P(u)$ потока u называется сумма потоков во всех дугах, выходящих из источника, т.е. общее количество передаваемого по сети продукта.

Пример 2. Для иллюстрации введённых понятий рассмотрим сеть, показанную на рис. 2. Векторы

$u^1 = (2, 0, 1, 1, 1, 0, 0, 2, 0)$;

$u^2 = (0, 1, 0, 0, 0, 1, 1, 1, 0)$;

$u^3 = (1, 0, 1, 0, 0, 1, 0, 0, 1)$;

$u^4 = (2, 1, 1, 1, 1, 1, 0, 2, 1)$

являются различными потоками в этой сети (проверьте это!). Легко видеть, что

$P(u^1) = 2$, $P(u^2) = 1$, $P(u^3) = 2$, $P(u^4) = 3$

Интересно изобразить эти четыре потока графически, сопоставив каждой единице потока по дуге пунктирную линию вдоль этой же дуги. Такое представление дается на рис.3.

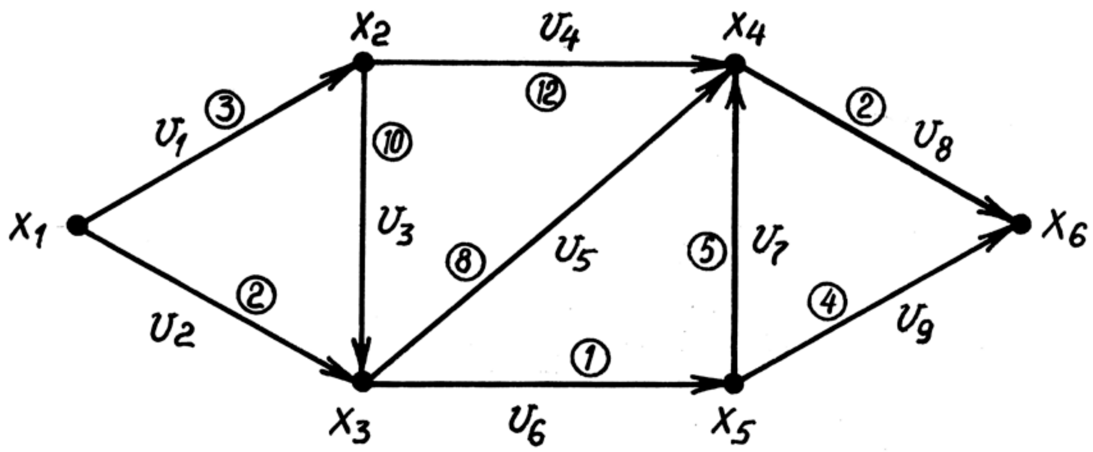


Рис.2 ■

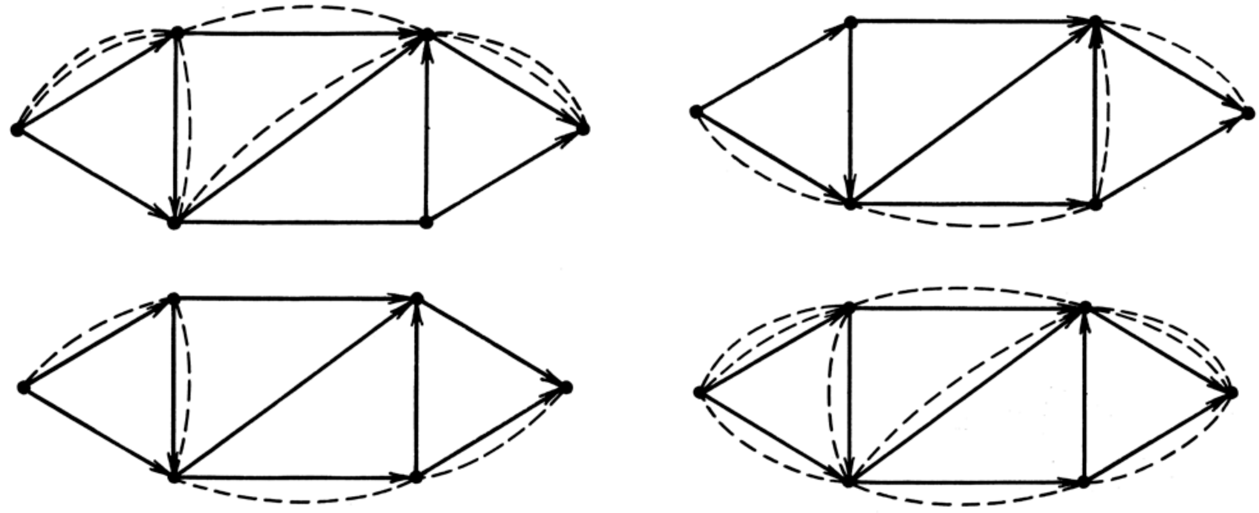


Рис.3

2. Постановка задачи о максимальном потоке

Одной из основных задач, связанных с потоковыми сетями, является задача нахождения так называемого ***максимального потока*** в сети. Максимальным потоком в сети S называется такой потокu, величина которого $P(u)$ не меньше величины $P(u')$ любого другого потока u' в этой же сети S. Содержательно это значит, что суммарное количество продукта, протекающего по сети от источника до стока, максимально. Формально задача нахождения максимального потока записывается следующим образом:

$$\sum_{j \in A_1^-} u_j \to \max \qquad (3)$$

при условиях (1) и (2).

Данная задача (1) – (3) является задачей линейного программирования, поскольку её целевая функция и ограничения линейны.

Пример 3. Для сети, показанной на рис. 2, задача о максимальном потоке запишется следующим образом:

$u_1 + u_2 \to \max$

при условиях

$0 \le u_1 \le 3,$

$0 \le u_2 \le 2,$

$0 \le u_3 \le 10,$

$0 \le u_4 \le 12,$

$0 \le u_5 \le 8,$

$0 \le u_6 \le 1,$

$0 \le u_7 \le 5,$

$0 \le u_8 \le 2,$

$0 \le u_9 \le 4;$

$u_1 = u_3 + u_4,$

$u_2 + u_3 = u_5 + u_6,$

$u_4 + u_5 + u_7 = u_8,$

$u_6 = u_7 + u_9$■

Для решения задачи о максимальном потоке в принципе можно применить любой метод решения задачи линейного программирования. Однако специфика данной задачи позволяет предложить более эффективные методы решения; один из них будет рассмотрен в разделе 4.

3. Модификация основной постановки

Во многих практически важных случаях содержательное описание ситуации отличается от рассмотренного ранее. Рассмотрим основные модификации данной модели.

3.1. Число источников больше единицы. Прежде всего, дадим точные модифицированные формулировки введённых понятий. Предполагается, что у сети имеется несколько источников. Это означает, что есть несколько вершин, в каждой из которых нет входящих дуг (есть только выходящие). Потоком в такой сети по-прежнему является вектор, число компонент которого равно числу дуг; неравенства (1) – ограничения на потоки в дугах – также не меняются. Закон сохранения (2) выполняются во всех вершинах, кроме всех источников и стока. Определение величины потока заменяется на аналогичное: величиной потока является сумма потоков во всех дугах, выходящих из всех источников. Как и в базовом случае, максимальным потоком называется любой поток с максимальной величиной $P(u)$.

Убедимся, что задача о максимальном потоке в данной модификации сводится к той же задаче в исходной постановке раздела 2. Пусть вершины графа сети $x_1, \ldots, x_k$ ($k > 1$) являются источниками. Добавим к графу сети S новую вершину x_0, и соединим её дугами со всеми «старыми» источниками $x_1, \ldots, x_k$, как это показано на рис.4. Пропускные способности в новых дугах положим равными ∞, оставив их без изменений во всех дугах старой сети S. Таким образом, определена новая потоковая сеть T.

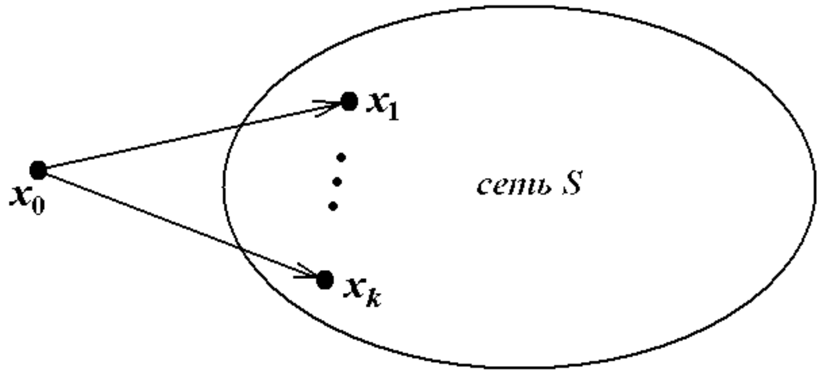


Рис.4

Пусть $u = (u_1, u_2, ..., u_m)$ – произвольный поток в исходный сети Sс источниками $x_1, \ldots, x_k$. Определим поток $v = (w_1, \ldots, w_k. u_1, u_2, ..., u_m)$ в сети T, положив поток w_i в новых дуге (x_0, x_i) равным сумме потоков во всех старых дугах, выходящих из стока x_i ($i = 1, \ldots, k$). По построению, старые источники стали внутренними вершинами в новой сети; при этом выполнение закона сохранения (2) сразу следует из определения потоков w_i. Ясно, что величины потоков u и v совпадают. Поэтому любой максимальный поток в исходной сети S получается из максимального потока в новой сети T просто отбрасыванием первых k компонент. Таким образом, нахождение максимального потока в сети с несколькими источниками сводится к нахождению максимального потока в сети с одним источником.

2. Число стоков больше единицы. Конструкция практически не отличается от предыдущей. Отличие только в том, что теперь старые стоки $y_1, \ldots, y_l$ соединяются с новым стокомy_0.

3. Пропускные способности вершин. В рассматриваемой модели до сих пор считалось, что заданы пропускные способности дуг; это соответствует реальным ограничениям, существующим в транспортных сетях. Однако при описании с помощью графов систем производственного типа отдельным агрегатам (машинам, станкам и др.) естественно сопоставляются именно вершины графа. Вершины x и y соединяются дугой, если продукция, произведенная на агрегате, которому сопоставлена вершина x, может поступать на агрегат, которому сопоставлена вершина y (для последующей обработки). При этом в первую очередь приходится учитывать не пропускные способности дуг, а производительность агрегатов, которая отображается (в модифицированной модели) в виде пропускной способности вершин.

Формально в данном случае сетью называется граф с теми же свойствами 1) – 3), что и ранее (см. раздел 1), у которого заданы ***пропускные способности вершин*** $b_1, b_2, ..., b_n$. При этом потоком в такой сети (назовём их сетями 2-го рода в отличие от ранее введённых сетей 1-го рода) называется любой вектор $u = (u_1, u_2, ..., u_m)$, удовлетворяющий, наряду с условием сохранения потоков (2) и условием неотрицательности

$$u_j \geq 0 \quad (j = 1, 2, ..., m), \tag{4}$$

ограничениям в вершинах

$$\sum_{j \in A_i^+} u_j \leq b_i (i = 2, ..., n), \tag{5}$$

$$\sum_{j \in A_1^-} u_j \leq b_1 \tag{6}$$

(условие (6) понадобилось выделить, так как в источник x_1 не входит ни одна дуга).

Задача о максимальном потоке для сетей 2-го рода содержательно состоит в определении максимально возможной производительности данного участка (моделируемого сетью 2-го рода), если заданы производительности $b_1, b_2, ..., b_n$ всех отдельных агрегатов этого участка. Формально эта задача записывается следующим образом:

$\sum_{j\in A_1^-} u_j \to \max$

при условиях (2), (4), (5) и (6).

Заметим, что целевые функции в задаче о максимальном потоке для сетей 1-го и 2-го рода совпадают.

Поскольку ограничения в задаче для сетей 1-го и 2-го рода различны, на первый взгляд, эта задача отличается от предыдущей задачи о максимальном потоке в сети 1-го рода. Покажем, что на самом деле задача для сети 2-го рода сводится к задаче для сети 1-го рода. Пусть задана сеть 2-го рода, x_i – любая вершина в ней, b_i – её пропускная способность. Заменим в графе сети вершину x_i на пару вершин x_i', x_i'' и соединяющую их дугу $\tilde{v}_i$ с той же пропускной способностью b_i. При этом все дуги, входившие в x_i, будут входить в x_i', а все дуги, выходившие из x_i, будут выходить из вершины x_i''. Указанный переход от фрагмента старой сети к фрагменту новой сети показан на рис.5.

Аналогичным образом заменим все вершины исходной сети 2-го рода на пары вершин. Пропускные способности оставшихся дуг исходной сети (ранее они не задавались) положим равными достаточно большому положительному числу (достаточно взять число $m\times\sum_{i=1}^{n} b_i$). Таким образом, по исходной сети S 2-го рода построена сеть S* 1-го рода.

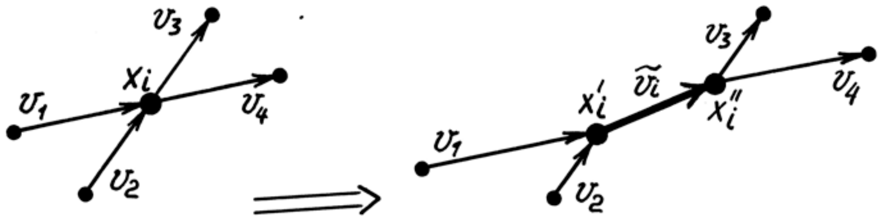


Рис.5

Утверждение 1. Максимальный поток в исходной сети S 2-го рода равен максимальному потоку в построенной по ней указанным выше способом сети S* 1-го рода ■

В силу утверждения 1 достаточно рассмотреть методы решения задачи о максимальном потоке только в сети 1-го рода. Решение задачи этой основной задачи (1) – (3) рассмотрено в следующем разделе 4.

4. Поиск максимального потока.

Пусть задана сеть S. Введём важное понятие разреза сети. ***Разрезом сети*** называется такое множество C дуг сети S, после удаления которых сеть распадается на несколько изолированных друг от друга частей, так что источник x_1 и сток x_n оказываются в разных частях, причём ни одно подмножество C этим свойством не обладает.

Пример 4. В сети, показанной на рис.2, множество дуг $\{v_4, v_5, v_6\}$ образует разрез; разрез образуют также множества дуг $\{v_6, v_7, v_8\}$ и $\{v_4, v_5, v_7, v_9\}$ (все эти разрезы показаны на рис.6). Множество дуг $\{v_1, v_5, v_6, v_8\}$ разреза не образует (см. рис.7). Дуги, образующие указанные множества, на рис.6 и 7 отмечены штриховкой ■

Разрезы удобно изображать также линиями (не обязательно прямыми), «разрезающими» соответствующие дуги. На рис.8 показан разрез $\{v_6, v_7, v_8\}$. Дуги относительно разреза «ведут себя» по-разному. Обозначим через A_s множество вершин, каждая из которых может быть соединена маршрутом (определение маршрута см. в конце раздела 6-3) с источником, через A_t– множество остальных вершин графа, которые соединяются неориентированным путём со стоком.

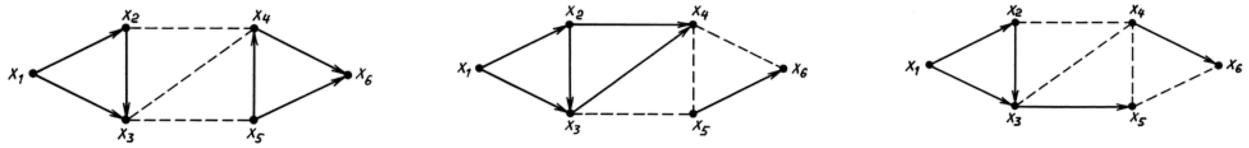


Рис.6

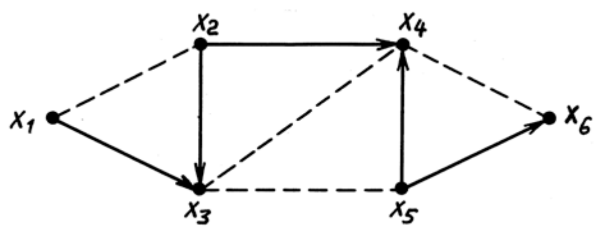


Рис.7

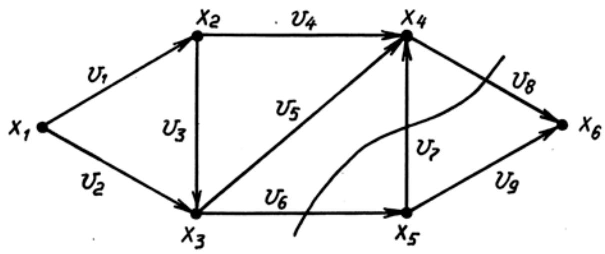


Рис.8

Пример 4. Для разреза $\{v_6, v_7, v_8\}$, показанного на рис.6 b и 8, $A_s = \{x_1, x_2, x_3, x_4\}$, $A_t = \{x_5, x_6\}$; для разреза $\{v_4, v_5, v_6\}$ (рис.6a) $A_s = \{x_1, x_2, x_3\}$, $A_t = \{x_4, x_5, x_6\}$ ■

Легко понять, что каждая дуга разреза соединяет между собой вершины, лежащие в разных множествах: одна в A_s, а другая в A_t. Дуга v называется ***прямой дугой*** разреза, если она выходит из A_s и входит в A_t; ***обратной***, если она выходит из A_t и входит в A_s. В разрезе $\{v_6, v_7, v_8\}$ дуги v_6 и v_8 – прямые, а дуга v_7 – обратная. Далее разрезы будем обозначать латинскими буквами A, B, ...

Пусть u – поток в сети, $P(u)$ – величина этого потока. ***Потоком u(A) через разрез А*** называется число, равное сумме потоков u_j во всех прямых дугах разреза A минус сумма потоков u_j во всех обратных дугах разреза A.

Утверждение 2. Для любого разреза A

$$u(A) = P(u). \tag{7}$$

Содержательно это утверждение понятно: все, что «вытекает» из источника, проходит по сети и «втекает» в сток. Это же количество продукта протекает через любой разрез. Поэтому надо сложить все, что течёт в нужном направлении (учтя с обратным знаком то, что течёт в противоположном направлении, т.е. по обратным дугам).

Пропускной способностью $c\ (A)$ разреза A называется сумма пропускных способностей всех его прямых дуг. Ясно, что для любого разреза A и любого потока в сети u

$u(A) \le c(A)$.

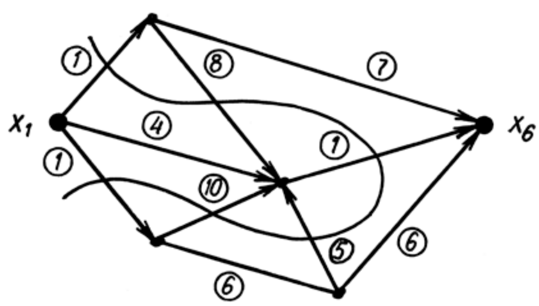
(8)Действительно, сумма потоков в прямых дугах разреза не превосходит $c(A)$; потоки же в обратных дугах входят в $u(A)$ со знаком минус, откуда сразу следует (8).

Разрез A, пропускная способность которого $c(A)$ минимальна (минимум берется по всем разрезам сети S), называется ***минимальным разрезом***. Из определений минимального разреза, максимального потока и формул (7) и (8) следует, что максимальный поток в сети не превосходит пропускной способности минимального разреза. Гораздо более сложным является следующее

Утверждение 3. Максимальный поток равен пропускной способности минимального разреза ■

Алгоритм поиска потока в сети, для которого выполняется указанное в утверждении 3 равенство, излагается далее, в разделе 4.1.

Пример 6. Найдем минимальные разрезы (и тем самым величины максимальных потоков) в нескольких простых сетях. В сети рис.1 минимальный разрез образует дуги v_1 и v_2, выходящие из источника x_1. Его пропускная способность равна 4. В сети рис.2 минимальным является разрез $\{v_6, v_7, v_8\}$, показанный на рис.6 b и 8. Его пропускная способность равна 3. Обратим внимание, что дуга v_7 (с пропускной способностью 5) здесь не учитывается, поскольку она является обратной дугой этого разреза. В сети рис.9 минимальный разрез указан линией. Его пропускная способность равна 3. В этом разрезе всего три дуги являются прямыми, остальные обратными.



■

Рис.9

4.1. Алгоритм Форда-Фалкерсона (АФФ). Перейдем к изложению алгоритма, предложенному американскими математиками Фордом и Фалкерсоном в 1945г. (впервые опубликован в 1951 г.). Он находит поток u с максимальной величиной $P(u)$, т.е. решает задачу (1) – (3). В основе этого алгоритма лежит переход от уже построенного потока u к новому потоку u', такому, что $P(u') > P(u)$. Если такой переход оказывается невозможным, то это значит, что поток u является решением исходной задачи (т.е. поток u уже является максимальным).

Алгоритм перехода от потока u к новому потоку u' состоит из четырёх этапов:

1. Расстановка ***меток*** у вершин сети (исходя из заданного потока u).
2. Построение так называемого ***увеличивающего маршрута*** между источником и стоком (исходя из построенных на этапе 1 меток).
3. Вычисление ***инкремента*** найденного увеличивающего маршрута (исходя из самого маршрута и заданного потока u).
4. Построение нового потока u' (исходя из заданного потока u, увеличивающего маршрута, построенного на этапе 2, и вычисленного на этапе 3 инкремента) ■

Наиболее важным и сложным этапом является этап 1. Рассмотрим его отдельно, учитывая, в частности, то обстоятельство, что расстановка меток у вершин используется в самых разнообразных задачах, связанных с графами (см., например, раздел 6-4.1).

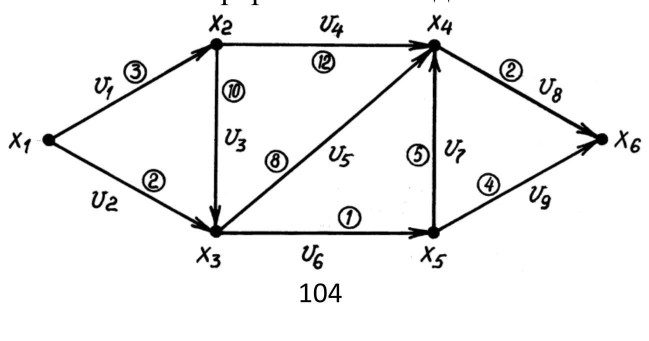
1. **Алгоритм расстановки меток.** В процессе работы этого алгоритма каждая вершина сети находится в одном из следующих трёх состояний:

1. ***Не помечена.***
2. ***Помечена и не просмотрена.***
3. ***Помечена и просмотрена.***

Каждая вершина может перейти из состояния с ме́ньшим номером в состояние с бо́льшим номером (не помеченная вершина может стать помеченной и не просмотренной; помеченная и не просмотренная вершина может стать помеченной и просмотренной). Обратные переходы, как и изменение полученных меток, невозможны. Как и во всех других алгоритмах, при отсутствии явного указания после проверки условия всегда выполняется следующий шаг.

0. Инициализация. Источник 1 получает метку 0; вершина 1 объявляется помеченной и не просмотренной; все остальные вершины объявляются не помеченными.
1. Из множества помеченных и не просмотренных вершин выбираем вершину с минимальным номером. Пусть это будет вершина i.
2. Выбранная на шаге 1 вершина переходит в состояние 3, т.е. она объявляется помеченной и просмотренной.
3. Помечаем индексом i все не помеченные ранее вершины x, обладающие следующим свойством: существует дуга v_j, ведущая из i в x, для которой $u_j < c_j$ (говорят, что дуга v_j ***не насыщена в прямом*** направлении). Вершина x объявляется помеченной и не просмотренной.
4. Если после выполнения шага 3 сток n оказывается помеченным, алгоритм 1 расстановки меток завершает работу и следует переход к описываемому далее алгоритму 2 построения увеличивающего маршрута.
5. Помечаем индексом – i все не помеченные ранее вершины x, обладающие следующим свойством: существует дуга v_j, ведущая из x в i, для которой $u_j > 0$ (говорят, что дуга v_j ***не насыщена в обратном*** направлении). Вершина x объявляется помеченной и не просмотренной.
6. Если текущее множество помеченных и не просмотренных вершин не пусто, то переходим к шагу 1.
7. АФФ завершает работу. Рассматриваемый в качестве исходного для алгоритма 1 расстановки меток поток u является искомым максимальным потоком■

Пример 7. Применим алгоритм 1 расстановки пометок для сети, показанной на рис.2 и для удобства дальнейшего изложения перерисованной здесь. В качестве исходного рассмотрим



поток u = (2, 0, 1, 1, 1, 0, 0, 2, 0) (см. пример 2). При «ручной» реализации алгоритма воспользуемся несколькими последовательно заполняемыми таблицами. При инициализации (шаг 0 алгоритма) заполняется таблица 1.0. В соответствии с алгоритмом расстановки меток, вершина 1 (источник) объявляется помеченной и не просмотренной (состояние 2), а все остальные вершины объявляются непомеченными (состояние 1). Метки при инициализации не определяются. В левую клетку ячейки (i, j), находящейся на пересечении i-ой строки и j-ого столбца запишем поток по дуге (i, j), в правую – ограничение в ней. Таким образом, в таблице 1.0 представлен результат инициализации, и в этой таблице собрана вся необходимая для дальнейшего информация. Обратим внимание, что вся информация, записанная в таблице 1.0, далее не меняется. Все новые данные (о состояниях и метках вершин) будут записываться в добавляемые снизу строки.

Таблица 1.0. Инициализация для расстановки меток по заданному потоку

	1		2		3		4		5		6	
1			2	3	0	2						
2					1	10	1	12				
3							0	8	0	1		
4											2	2
5							0	5			0	4
	2		1		1		1		1		1	

Итерация 1. Результаты итерации будут записываться в нижнюю строку таблицы 1.1, которая сначала отличается от таблицы 1.0 только добавленной нижней строкой, в которой ничего не записано. Из всех помеченных и не просмотренных вершин (т.е. вершин в состоянии 2) выбирается вершина с минимальным номером. В состоянии 2 есть только одна вершина 1, поэтому она и выбирается. В соответствии с шагом 2 алгоритма эта вершина переходит из состояния 2 в состояние 3, и число 3 записывается в нижнюю строку, в левую клетку ячейки 1. Далее, в соответствии с шагом 3 алгоритма, рассматриваются все вершины, в которые ведут дуги из вершины 1. Этим вершинам соответствуют заполненные ячейки 1-ой строки: 2-ая и 3-ья. Поскольку в обеих ячейках число в левой клетке меньше числа в правой клетке, то это и значит, что обе дуги: v_1 = (1,2) и v_2 = (1,3) не насыщены в прямом направлении. Поэтому, в соответствии с шагом 3 алгоритма, вершины 2 и 3 получают метку 1 и становятся помеченными, но не просмотренными, т.е. переходят из состояния 1 в состояние 2, что и записывается в нижнюю строку таблицы 1.1. Поскольку сток не помечен, то, в соответствии с шагом 5 алгоритма, надо проверить вершины, из которых ведут дуги в вершину 1. Но поскольку заполненных ячеек в 1-ом столбце таблицы 1.1 нет, то, значит, нет и таких вершин, и все операции шага 5 просто не выполняются. Для завершения итерации осталось скопировать данные из предпоследней строки таблицы на те же самые пустые позиции в нижней строке, не меняя ничего в уже заполненных позициях в нижней строке.

Таблица 1.1. Итерация 1 расстановки меток по заданному потоку

	1		2		3		4		5		6	
1			2	3	0	2						
2					1	10	1	12				
3							0	8	0	1		
4											2	2
5							0	5			0	4
	2		1		1		1		1		1	
	3		2	1	2	1	1		1		1	

Итерация 2. Результаты итерации будут записываться в нижнюю строку таблицы 1.2, которая сначала отличается от таблицы 1.1 только добавленной нижней строкой, в которой ничего не записано. Рассматриваем предпоследнюю строку. Из всех помеченных и не просмотренных вершин (т.е. вершин в состоянии 2)выбирается вершина с минимальным номером. В состоянии 2 есть две вершины: 2 и 3, поэтому выбирается вершина с ме́ньшим номером 2. В соответствии с шагом 2 алгоритма эта вершина переходит из состояния 2 в состояние 3, и число 3 записывается в нижнюю строку, в левую клетку ячейки 2. Далее, в соответствии с шагом 3 алгоритма,

рассматриваются все вершины, в которые ведут дуги из вершины 2. Этим вершинам соответствуют заполненные ячейки 2-ой строки: 3-ья и 4-ая. Поскольку вершина 3 уже помечена, то она не рассматривается. В 4-ой ячейке число 1 в левой клетке меньше числа 12 в правой клетке, т.е. дуга (2,4) не насыщена в прямом направлении. Поэтому, в соответствии с шагом 3 алгоритма, вершина 4 получает метку 2 и становятся помеченной, но не просмотренной, т.е. переходят из состояния 1 в состояние 2, что и записывается в нижнюю строку таблицы 1.2. Поскольку сток не помечен, то, в соответствии с шагом 5 алгоритма, надо проверить вершины, из которых ведут дуги в вершину 2. Но поскольку во 2-ом столбце заполнена только ячейка (1,2), то в неё ведёт только одна дуга из уже помеченной вершины 1. Поэтому все операции шага 5 просто не выполняются. Для завершения итерации осталось скопировать данные из предпоследней строки таблицы на те же самые пустые позиции в нижней строке, не меняя ничего в уже заполненных позициях в нижней строке.

Таблица 1.2. Итерация 2 расстановки меток по заданному потоку

	1		2		3		4		5		6	
1			2	3	0	2						
2					1	10	1	12				
3							0	8	0	1		
4											2	2
5							0	5			0	4
	2		1		1		1		1		1	
	3		2	1	2	1	1		1		1	
	3		3	1	2	1	2	2	1		1	

Итерация 3. Результаты итерации будут записываться в нижнюю строку таблицы 1.3, которая сначала отличается от таблицы 1.2 только добавленной нижней строкой, в которой ничего не записано. Рассматриваем предпоследнюю строку. Из всех помеченных и не просмотренных вершин (т.е. вершин в состоянии 2)выбирается вершина с минимальным номером. В состоянии 2 есть две вершины: 3 и 4, поэтому выбирается вершина с ме́ньшим номером 3. В соответствии с шагом 2 алгоритма эта вершина переходит из состояния 2 в состояние 3, и число 3 записывается в нижнюю строку, в левую клетку ячейки 3. Далее, в соответствии с шагом 3 алгоритма, рассматриваются все вершины, в которые ведут дуги из вершины 3. Этим вершинам соответствуют заполненные ячейки 3-ей строки: 4-ая и 5-ая. Поскольку вершина 4 уже помечена, то она не рассматривается. В 5-ой ячейке число 0 в левой клетке меньше числа 1 в правой клетке, т.е. дуга (3,5) не насыщена в прямом направлении. Поэтому, в соответствии с шагом 3 алгоритма, вершина 5 получает метку 3 и становятся помеченной, но не просмотренной, т.е. переходят из состояния 1 в состояние 2, что и записывается в нижнюю строку таблицы 1.3. Поскольку сток не помечен, то, в соответствии с шагом 5 алгоритма, надо проверить вершины, из которых ведут дуги в вершину 3. Но поскольку в 3-ем столбце заполнены ячейки (1,3) и (2,3) то в неё ведут две дуги из уже помеченных вершин 1 и 2. Поэтому все операции шага 5 просто не выполняются. Для завершения итерации осталось скопировать данные из предпоследней строки таблицы на те же самые пустые позиции в нижней строке, не меняя ничего в уже заполненных позициях в нижней строке.

Таблица 1.3. Итерация 3 расстановки меток по заданному потоку

	1		2		3		4		5		6	
1			2	3	0	2						
2					1	10	1	12				
3							0	8	0	1		
4											2	2
5							0	5			0	4
	2		1		1		1		1		1	
	3		2	1	2	1	1		1		1	
	3		3	1	2	1	2	2	1		1	
	3		3	1	3	1	2	2	2	3	1	

Итерация 4. Результаты итерации будут записываться в нижнюю строку таблицы 1.4, которая сначала отличается от таблицы 1.3 только добавленной нижней строкой, в которой ничего не записано. Рассматриваем предпоследнюю строку. Из всех помеченных и не просмотренных вершин (т.е. вершин в состоянии 2)выбирается вершина с минимальным номером. В состоянии 2 есть две вершины: 4 и 5, поэтому выбирается вершина с ме́ньшим номером 4. В соответствии с шагом 2 алгоритма эта вершина переходит из состояния 2 в состояние 3, и число 3 записывается в нижнюю строку, в левую клетку ячейки 4. Далее, в соответствии с шагом 3 алгоритма, рассматривается единственная вершина 6, в которую ведёт дуга из вершины 4. В 6-ой ячейке 4-ой строки число 2 в левой клетке равно числу 2 в правой клетке, т.е. дуга (4,6) насыщена в прямом направлении. Поэтому, в соответствии с шагом 3 алгоритма, никакие операции этого шага 3 не выполняются. Поскольку сток не помечен, то, в соответствии с шагом 5 алгоритма, надо проверить вершины, из которых ведут дуги в вершину 4. Но поскольку в 4-ом столбце заполнены ячейки (2,4), (3,4) и (5,4) то в неё ведут три дуги из уже помеченных вершин 2, 3 и 5. Поэтому все операции шага 5 просто не выполняются. Для завершения итерации осталось скопировать данные из предпоследней строки таблицы на те же самые пустые позиции в нижней строке, не меняя ничего в уже заполненных позициях в нижней строке, которая в данном случае только одна. Заметим, что на этой итерации ни одна новая вершина не получила метки, однако, как и на каждой итерации, ровно одна вершина сменила состояние 2 на 3.

Таблица 1.4. Итерация 4 расстановки меток по заданному потоку

	1		2		3		4		5		6	
1			2	3	0	2						
2					1	10	1	12				
3							0	8	0	1		
4											2	2
5							0	5			0	4
	2		1		1		1		1		1	
	3		2	1	2	1	1		1		1	
	3		3	1	2	1	2	2	1		1	
	3		3	1	3	1	2	2	2	3	1	
	3		3	1	3	1	3	2	2	3	1	

Итерация 5. Результаты итерации будут записываться в нижнюю строку таблицы 1.5, которая сначала отличается от таблицы 1.4 только добавленной нижней строкой, в которой ничего не записано. Рассматриваем предпоследнюю строку. Из всех помеченных и не просмотренных вершин (т.е. вершин в состоянии 2) выбирается вершина с минимальным номером. В состоянии 2 есть только одна вершины 5, поэтому она и выбирается. В соответствии с шагом 2 алгоритма эта вершина переходит из состояния 2 в состояние 3, и число 3 записывается в нижнюю строку, в левую клетку ячейки 5. Далее, в соответствии с шагом 3 алгоритма, рассматривается две вершины: 4 и 6, в которые ведут дуги из вершины 5. Вершина 4 уже помечена и далее не рассматривается. В 6-ой ячейке 5-ой строки число 0 в левой клетке меньше числа 4 в правой клетке, т.е. дуга (5,6) не насыщена в прямом направлении. Поэтому, в соответствии с шагом 3 алгоритма, вершина 6 получает метку 5 и меняет состояние 1 на 2, что и заносится в 6-ую ячейку нижней строки. Поскольку сток 6 теперь помечен, то, в соответствии с шагом 4, алгоритм расстановки меток прекращает работу. Для завершения его работы осталось скопировать данные из предпоследней строки таблицы на те же самые пустые позиции в нижней строке, не меняя ничего в уже заполненных позициях в нижней строке. Окончательные метки содержатся в этой строке. Заметим, что в общем случае не все вершины обязаны быть во 2-ом или 3-ьем состоянии и, следовательно, иметь метки.

Таблица 1.5. Итерация 5 расстановки меток по заданному потоку

	1		2		3		4		5		6	
1			2	3	0	2						
2					1	10	1	12				
3							0	8	0	1		
4											2	2

5							0	5			0	4
	2		1		1		1		1		1	
	3		2	1	2	1	1		1		1	
	3		3	1	2	1	2	2	1		1	
	3		3	1	3	1	2	2	2	3	1	
	3		3	1	3	1	3	2	2	3	1	
	3		3	1	3	1	3	2	3	3	2	5

■

Рассмотрим теперь алгоритмы 2 – 4 по отдельности.

2. <u>**Алгоритм построения увеличивающего маршрута**</u>. Начнём с определения. Маршрут μ от источника к стоку называется ***увеличивающим маршрутом***, если все его прямые дуги не насыщены в прямом направлении ($u_j < c_j$), и все его обратные дуги не насыщены в обратном направлении ($u_j > 0$). Алгоритм 2 начинает работу, когда сток сети помечен. Исходной для него является информация из нижней строки последней таблицы в алгоритме 1.

1. Концом маршрута μ является сток n.
2. Пусть уже построен конец маршрута μ от некоторой вершины x до стока n. Если меткой у вершины x является i ($-i$), то предыдущей вершиной маршрута μ является вершина i и дуга (i, x) (дуга (x, i)) является прямой (обратной) дугой строящегося маршрута μ.
3. Если вершина i является источником, то искомый маршрута μ построен; переходим к алгоритму 3 вычисления инкремента маршрута μ. В противном случае возвращаемся на шаг 2 ■

Напомним, что по определению потоковой сети у источника нет входящих дуг, а у стока – выходящих. Поэтому первая и последняя дуги построенного маршрута μ обязательно являются его прямыми дугами.

Пример 8. Применим алгоритм 2 построения увеличивающего маршрута для сети с метками, представленными в последней строке таблицы 1.5. В соответствии с алгоритмом 2 маршрут μ заканчивается в стоке 6. Поскольку у вершины 6 стоит метка 5, предшествующей вершине 6 вершиной маршрута μ является вершина 5. Поскольку у вершины 5 стоит метка 3, предшествующей вершине 5 вершиной является вершина 3. Далее, предшествующей вершине 3 вершиной является источник 1. Поэтому сам увеличивающий маршрут μ от источника к стоку оказывается таким: 1→3→5→6. Заметим, что все его дуги – (1,3), (3,5), (5,6) – являются прямыми дугами построенного маршрута μ ■

Утверждение 4. Маршрут μ от источника к стоку, построенный алгоритмом 2 по меткам, найденным алгоритмом 1, является увеличивающим маршрутом

Доказательство непосредственно следует из правил расстановки меток. Метка i у вершины xставится только в двух случаях: когда дуга (i, x) не насыщена в прямом направлении или когда дуга (x,i) не насыщена в обратном управлении ■

3. <u>**Алгоритм вычисления инкремента увеличивающего маршрута**</u>. Пусть маршрут $\mu = z_1 \to z_2 \to \ldots \to z_k$ является увеличивающим.

1. Положим $D = \max_j c_j$, $i = 0$.
2. $i = i + 1$.
3. Положим

$$D_i = \begin{cases} c_j - u_j, \text{если дуга } v_j = (z_i, z_{i+1}) - \text{ прямаядуга маршрута } \mu, \\ u_j, \text{если дуга } v_j = (z_{i+1}, z_i) - \text{ обратнаядуга маршрута } \mu \end{cases}$$

4. Положим $D = \min\{D, D_i\}$.
5. Если $i<k-1$, возвращаемся к шагу 2; в противном случае алгоритм 3 прекращает работу ■

Найденное значение D является той «добавкой», на <u>которую могут быть увеличены потоки во всех дугах маршрута μ</u> (именно поэтому он назван «увеличивающим»).

Пример 9. Вычислим инкремент увеличивающего маршрута $\mu = x_1 \to x_3 \to x_5 \to x_6$, найденного в примере 8. Положим $D = \max_j c_j = 12$.

Далее:

$i = 1$, $v_j = (x_1, x_3)$, $c_j = 2$, $u_j = 0$, $D_1 = c_j - u_j = 2$, $D = \min(D, D_1) = 2$;

$i = 2$, $v_j = (x_3, x_5)$, $c_j = 1$, $u_j = 0$, $D_2 = c_j - u_j = 1$, $D = \min(D, D_2) = 1$;

$i = 3$, $v_j = (x_5, x_6)$, $c_j = 4$, $u_j = 0$, $D_3 = c_j - u_j = 4$, $D = \min(D, D_3) = 1$.

Таким образом, инкремент увеличивающего маршрута $\mu = x_1 \to x_3 \to x_5 \to x_6$ равен 1. Это значит, что потоки во всех дугах данного маршрута могут быть увеличены на 1 ■

4. **<u>Алгоритм построения нового потока u'</u>**. Новый поток u' определяется по заданному потоку u, построенному увеличивающему маршруту μ и вычисленному инкременту D этого пути следующей формулой:

$$u'_j = \begin{cases} u_j, & \text{если дуга } v_j \text{ не входит в маршрут } \mu, \\ u_j + D, & \text{если дуга } v_j \text{ является прямой дугой маршрута } \mu, \\ u_j - D, & \text{если дуга } v_j \text{ является обратной дугой маршрута } \mu \end{cases} \quad (j = 1, 2, ..., m). \qquad (9)$$

Пример 10. Найдём новый поток u' для сети, рассмотренной в примерах 7 – 9. Все три дуги в маршруте $\mu = x_1 \to x_3 \to x_5 \to x_6$ являются прямыми. «Старые» потоки по этим дугам были равны 0, а новые равны 1. Учитывая нумерацию дуг в данной сети (см. рис.2) и то, что старый поток u = (2, 0, 1, 1, 1, 0, 0, 2, 0), получаем u' = (2, 1, 1, 1, 1, 1, 0, 2, 1), так как в маршрут μ входят дуги v_2, v_6 и v_9■

Утверждение 5.

1. Вектор u' с компонентами, определенными формулой (9), является потоком в сети S;
2. $P(u') = P(u) + D$, (10)

где через $P(u)$ обозначена величина потока u (см. конец раздела 1 перед примером 2) ■

Таким образом, по любому потоку u, не являющемуся максимальным, можно построить поток u' с бóльшей величиной $P(u')$. Именно это построение (состоящее из рассмотренных выше четырёх этапов) и лежит в основе АФФ.

Для завершения описания алгоритма нам понадобится почти очевидное

Утверждение 6. Нулевой вектор u = (0, 0, ..., 0) является потоком в сети ■

Алгоритм Форда-Фалкерсона (АФФ)

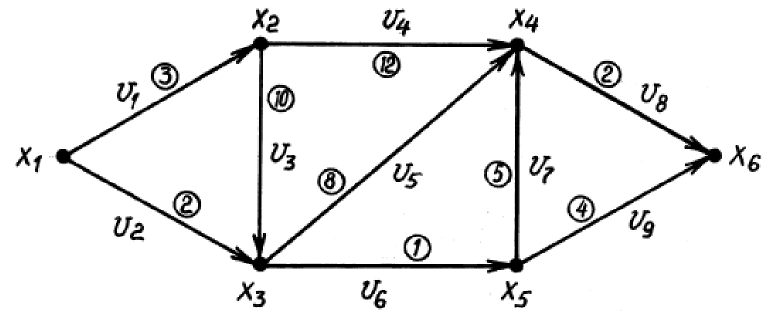
1. Положить u = (0, 0, ..., 0).
2. Выполнить алгоритм расстановки меток по потоку u. При выходе из этого алгоритма на шаге 7 АФФ завершает работу; поток u является максимальным. При выходе из этого алгоритма на шаге 3 перейти на следующий шаг 3 АФФ.
3. Построить увеличивающий маршрут между источником и стоком, исходя из найденных меток.
4. Вычислить инкремент D найденного увеличивающего маршрута, исходя из потока u.
5. Построить новый поток u' по формуле (9), исходя из потока u, маршрута и декремента D.
6. Положить $u = u'$ и вернуться на шаг 2 ■

Утверждение 7. При целочисленных пропускных способностях c_j(j = 1, 2, ..., m) все потоки, построенные АФФ, будут целочисленными, и процесс построения потоков оборвётся за конечное число шагов ■

Это достаточно простое утверждение не означает, что при нецелочисленных пропускных способностях АФФ обязательно будет «циклиться». Более того, при разумном выборе вершины из множества помеченных и не просмотренных вершин на шаге 1 алгоритма расстановки меток можно не только гарантировать конечность АФФ при любых пропускных способностях, но и дать полиномиальную оценку числа необходимых операций.

В качестве начального можно выбирать не обязательно нулевой, но и любой целочисленный поток. Применяя АФФ, через конечное число шагов будет получен максимальный поток (точнее, поток максимальной величины).

Пример 11. Используя АФФ, найдём максимальный поток в той же сети, показанной на рис.2, исходя из потока u = (2, 1, 1, 1, 1, 1, 0, 2, 1), найденного в примере 10. Как и в примере 7, для удобства чтения покажем эту сеть ещё раз.



Как и в примере 7, начнём с инициализации в таблице 2.0. Отличие от таблицы 1.0 состоит только в других значениях потоков в дугах (1,3), (3,5) и (5,6), образующих найденный в примере 8 увеличивающий маршрут. Так как инкремент $D = 1$, то это значит, что в левых клетках ячеек (1,3), (3,5) и (5,6) стоят числа, на 1 бóльшие, чем на тех же местах в «старой» таблице 1.0. Больше эти таблицы ничем не отличаются.

Таблица 2.0. Инициализация для расстановки меток по заданному потоку

	1		2		3		4		5		6	
1			2	3	1	2						
2					1	10	1	12				
3							0	8	1	1		
4											2	2
5							0	5			1	4
	2		1		1		1		1		1	

Итерация 1. Результаты итерации будут записываться в нижнюю строку таблицы 2.1, которая сначала отличается от таблицы 2.0 только добавленной нижней строкой, в которой ничего не записано. Из всех помеченных и не просмотренных вершин (т.е. вершин в состоянии 2) выбирается вершина с минимальным номером. В состоянии 2 есть только одна вершина 1, поэтому она и выбирается. В соответствии с шагом 2 алгоритма эта вершина переходит из состояния 2 в состояние 3, и число 3 записывается в нижнюю строку, в левую клетку ячейки 1. Далее, в соответствии с шагом 3 алгоритма, рассматриваются все вершины, в которые ведут дуги из вершины 1. Этим вершинам соответствуют заполненные ячейки 1-ой строки: 2-ая и 3-ья. Поскольку в обеих ячейках число в левой клетке меньше числа в правой клетке, то это и значит, что обе дуги: $v_1 = (1,2)$ и $v_2 = (1,3)$ не насыщены в прямом направлении. Поэтому, в соответствии с шагом 3 алгоритма, вершины 2 и 3 получают метку 1 и становятся помеченными, но не просмотренными, т.е. переходят из состояния 1 в состояние 2, что и записывается в нижнюю строку таблицы 1.1. Поскольку сток не помечен, то, в соответствии с шагом 5 алгоритма, надо проверить вершины, из которых ведут дуги в вершину 1. Но поскольку заполненных ячеек в 1-ом столбце таблицы 1.1 нет, то, значит, нет и таких вершин, и все операции шага 5 просто не выполняются. Для завершения итерации осталось скопировать данные из предпоследней строки таблицы на те же самые пустые позиции в нижней строке, не меняя ничего в уже заполненных позициях в нижней строке.

Таблица 2.1. Итерации расстановки меток по заданному потоку

	1		2		3		4		5		6	
1			2	3	1	2						
2					1	10	1	12				
3							0	8	1	1		
4											2	2
5							0	5			1	4
И	2		1		1		1		1		1	
1	3		2	1	2	1	1		1		1	
2	3		3	1	2	1	2	2	1		1	
3	3		3	1	3	1	2	2	1		1	
4	3		3	1	3	1	3	2	1		1	

В отличие от примера 7, результаты дальнейших итераций будут записываться в последовательно добавляемые нижние строки той же самой таблицы 2.1. Слева указаны номера итераций; буква И означает инициализацию.

Итерация 2. Выбираем вершину 2. Она переходит в состояние 3. Из неё ведут две дуги в вершины 3 и 4 (см. 2-ую строку таблицы), но вершина 3 уже помечена. Дуга (2,4) не насыщена в прямом направлении и, следовательно, вершина 4 получает метку 2 и переходит в состояние 2. Так как единственной входящей дугой является дуга (1,2) и вершина 1 уже помечена, то других изменений на данной итерации нет.

Итерация 3. Выбираем вершину 3. Она переходит в состояние 3. Из неё ведут две дуги в вершины 4 и 5. Вершина 4 уже помечена, а дуга (3,5) насыщена в прямом направлении (поток 1 равен ограничению 1), так что новых меток от выходящих дуг не появляется. Обе входящие дуги (1,3) и (2,3) выходят из уже помеченных вершин, так что единственное изменение на данной итерации – это переход вершины 3 из состояния 2 в состояние 3.

Итерация 4. Выбираем вершину 4 (как единственную вершину в состоянии 2). Она переходит в состояние 3. Из неё ведёт единственная дуга (4,6) в сток 6, но так как она насыщена в прямом направлении (поток 2 равен ограничению 2), то сток 6 не помечается и не меняет состояния. Далее, в вершину 4 входят 3 дуги: (2,4), (3,4) и (5,4). Вершины 2 и 3 уже помечены, а дуга (5,4) насыщена в обратном направлении (поток в ней равен 0). Поэтому единственным изменением на этой итерации является переход вершины 4 из состояния 2 в состояние 3.

Итерация 5. Вершины в состоянии 2 не существует. В соответствии с шагом 2 АФФ это означает, что исходный поток u= (2, 1, 1, 1, 1, 1, 0, 2, 1) является максимальным. Его величина $P(u)$ = 3, так как сумма потоков в дугах v_1 = (1,2) и v_2 = (1,3) равна 3.

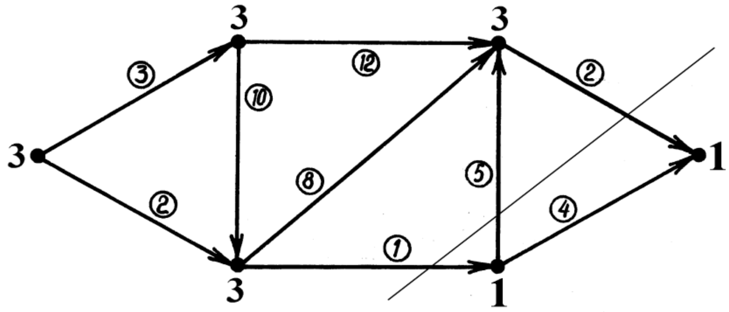


Рис.10

На рис 10 представлена та же самая сеть, но вместо номеров вершин указаны их финальные метки, полученные в результате 4-ёх итераций алгоритма расстановки меток (см. нижнюю строку таблицы 2.4). Естественно, что дуги, соединяющие вершины с разными метками, образуют разрез. Пропускная способность этого разреза, т.е. сумма пропускных способностей его прямых дуг, равна 3, что полностью согласуется с утверждением 3 ■

Пример 12. Рассмотрим потоковую сеть, показанную на рис.11, и найдём в ней макси-

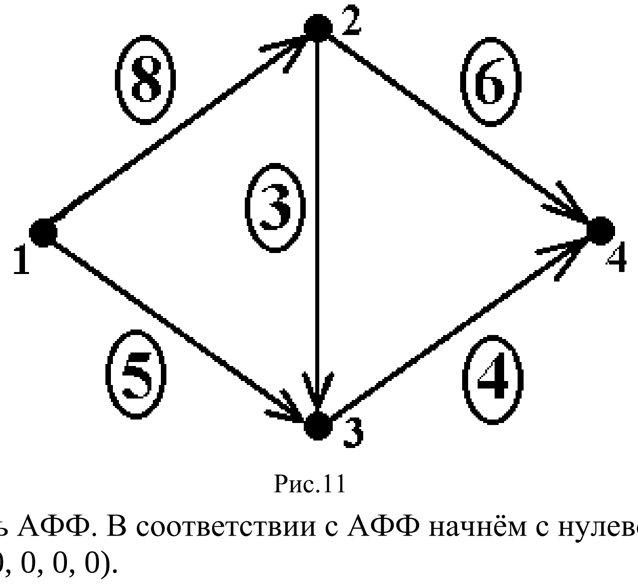


Рис.11

мальный поток, пользуясь АФФ. В соответствии с АФФ начнём с нулевого потока.

1. Положим u = (0, 0, 0, 0, 0).

2. Расставим метки, как это делалось в примерах 7 и 11. Инициализация представлена в таблице 3.0:

Таблица 3.0. Инициализация для расстановки меток по заданному потоку

	1		2		3		4	
1			0	8	0	5		
2					0	3	0	6
3							0	4
И	2		1		1		1	

Результаты последовательных итераций показаны в нижних строках таблицы 3.1.

Таблица 3.1. Итерации расстановки меток по заданному потоку 1

	1		2		3		4	
1			0	8	0	5		
2					0	3	0	6
3							0	4
И	2		1		1		1	
1	3		2	1	2	1	1	
2	3		3	1	2	1	2	2

Поскольку сток (вершина 4) помечен, метки построены. Они приведены в правых клетках нижней строки таблицы 3.1.

3. Построим увеличивающий маршрут (см. алгоритм 2). Он таков: 1→2→4.

4. Найдём инкремент (см. алгоритм 3). Положим $D = 8$; тогда $D_1 = 8$, $D_2 = 6$.

5. Найдём новый поток (см. алгоритм 4). Положим $u' = (6, 0, 0, 6, 0)$ и перейдём на шаг 2.

2. Найдём метки, как это делалось в примерах 7 и 11 и при 1-ой итерации АФФ (см. таблицы 3.0 и 3.1). Инициализация представлена в таблице 4.0:

Таблица 4.0. Инициализация для расстановки меток по заданному потоку

	1		2		3		4	
1			6	8	0	5		
2					0	3	6	6
3							0	4
И	2		1		1		1	

Результаты последовательных итераций показаны в нижних строках таблицы 4.1.

Таблица 4.1. Итерации расстановки меток по заданному потоку 1

	1		2		3		4	
1			6	8	0	5		
2					0	3	6	6
3							0	4
И	2		1		1		1	
1	3		2	1	2	1	1	
2	3		3	1	2	1	1	
3	3		3	1	3	1	2	3

Поскольку сток оказался помеченным, то метки построены. Они приведены в последней строке таблицы 4.1.

3. Построим увеличивающий маршрут (см. алгоритм 2). Он таков: 1→3→4.

4. Найдём инкремент (см. алгоритм 3). Положим $D = 5$; тогда $D_1 = 5$, $D_2 = 4$.

5. Найдём новый поток (см. алгоритм 4). Положим $u' = (6, 4, 0, 6, 4)$ и перейдём на шаг 2.

2. Найдём метки, как это делалось в примерах 7, 11 и при двух итерациях АФФ (см. таблицы 3.1 и 4.1).Инициализация представлена в таблице 5.0:

Таблица 5.0. Инициализация для расстановки меток по заданному потоку

	1		2		3		4	
1			6	8	4	5		
2					0	3	6	6
3							4	4
И	2		1		1		1	

Результаты последовательных итераций показаны в нижних строках таблицы 5.1.

Таблица 5.1. Итерации расстановки меток по заданному потоку 1

	1		2		3		4	
1			6	8	4	5		
2					0	3	6	6
3							4	4
И	2		1		1		1	
1	3		2	1	2	1	1	

2	3		3	1	2	1	1	
3	3		3	1	3	1	1	

Нет ни одной вершины в состоянии 2, и сток не помечен. В соответствии с АФФ это означает, что поток u = (6, 4, 0, 6, 4), найденный перед последним проходом алгоритма расстановки меток, является максимальным потоком, т.е. потоком, величина которого $P(u)$ максимальна. Так как сумма потоков в дугах, выходящих из источника, равна 10, то величина максимального потока $P(u)$ = 10 ■

5. Задания

Задание 1. При нумерации дуг, показанной на рис.2

а. Представить данный поток графически, как сделано в примере 2 (см. рис.3).

б. Найти метки по данному потоку (см. пример 7)

в. Построить увеличивающий маршрут для меток, найденных в задании 1б (см. пример 8)

г. Найти инкремент увеличивающего маршрута, найденных в задании 1в (см. пример 9)

д. Найти новый поток по результатам заданий 1б – 1г (см. пример 10)

е. Исходя из потока, найденного в задании 1д, найти максимальный поток с помощью АФФ (см. пример 11).

Варианты начальных потоков таковы:

u^{01} = (0, 0, 0, 0, 0, 0, 0, 0, 0);
u^{02} = (1, 0, 1, 0, 1, 0, 0, 1, 0);
u^{03} = (0, 1, 0, 0, 1, 0, 0, 1, 0);
u^{04} = (0, 1, 0, 0, 0, 1, 0, 0, 1);
u^{05} = (1, 1, 0, 1, 1, 0, 0, 2, 0);
u^{06} = (1, 2, 0, 1, 1, 1, 0, 2, 1);
u^{07} = (3, 0, 1, 2, 0, 1, 0, 2, 1);
u^{08} = (0, 2, 0, 0, 2, 0, 0, 2, 0);
u^{09} = (1, 2, 0, 1, 1, 1, 0, 2, 1);
u^{10} = (1, 1, 1, 0, 2, 0, 0, 2, 0) ■

Задание 2. Используя АФФ, найти максимальный поток в сети с графом на рис.11, и данными пропускными способностями. Нумерация дуг такова: (1,2), (2,4), (1,3), (3,4), (2,3). Для образца см. пример 12.

Варианты пропускных способностей таковы:

01) 1,2,3,8,5
02) 6,4.7,2,5
03) 5,3,2,1,8
04) 7,6,5,4,2
05) 6,2,5,8,4
06) 4,2,3,8,5
07) 6,4,7,3,5
08) 2,8,6,4,5
09) 5,3,2,2,8
10) 7,6,5,4,3
11) 3,8,6,4,5
12) 6,2,4,8,4
13) 2,2,3,8,5
14) 6,5,7,2,5
15) 5,4,2,2,8
16) 8,6,5,4,3
17) 1,4,2,3,4
18) 5,3,6,3,7■

Задание 3. Используя АФФ, найти максимальный поток в сети с графом, показанным на рис.1, и данными пропускными способностями.

Варианты пропускных способностей таковы:

01) (2,7,3,1,2,1,4,3,5,6,2,7).
02) (3,4,8,4,5,9,3,4,8,2,7,3).
03) (1,9,4,7,8,3,5,5,3,2,5,2).
04) (9,6,4,8,3,2,3,4,2,5,2,7,).
05) (4,5,8,9,2,3,2,6,3,4,2,2).
06) (9,2,7,1,8,9,6,1,8,3,7,2).
07) (5,8,7,8,5,2,1,2,3,8,1,9).
08) (1,3,2,2,3,4,8,5,2,1,7,9).
09) (4,5,8,3,2,3,8,7,2,2,3,1).
10) (2,5,2,9,4,6,8,1,1,7,4,4).
11) (5,8,1,4,5,3,4,2,5,3,1,2).
12) (1,4,3,5,8,1,9,1,3,2,1,8).
13) (9,6,1,8,3,2,3,3,4,2,5,5).
14) (4,6,8,1,9,7,4,8,7,3,1,2) ■

6. Предметный указатель

Глава 8. Кратчайшие пути

1. Понятие кратчайшего пути

Задачи поиска путей минимальной общей длины, часто называемых кратчайшими путями, являются одними из наиболее исследованных и востребованных задач дискретной оптимизации. Это связано с широкой распространённостью разнообразных прикладных задач, которые сводятся к поиску кратчайших путей. При этом в исходных постановках в качестве длин дуг или рёбер могут выступать расстояния, времена, стоимости, штрафы, убытки – словом, любая величина, которую желательно минимизировать и которая обладает свойством аддитивности или другими «декомпозиционными» свойствами, позволяющими использовать аналогичные модифицированные алгоритмы (пример такой модификации рассмотрен в разделе 4). Из множества методов нахождения кратчайших путей в главу включены два метода: алгоритм Дейкстры и алгоритм Флойда-Уоршолла. Одно из важных различий между ними состоит в том, что первый на каждой итерации находит кратчайший путь от заданной вершины в некоторую другую. Поэтому кратчайший путь между двумя фиксированными вершинами зачастую находится гораздо быстрее, чем кратчайшие пути для всех пар вершин. Второй, напротив, в рамках одной процедуры находит кратчайшие пути сразу для всех пар вершин, что требуется в некоторых приложениях, но при этом вплоть до последней итерации нет гарантии, что кратчайший путь найден для любой заранее указанной пары вершин. Однако алгоритм Флойда-Уоршолла «попутно» решает важную задачу о нахождении цикла отрицательной длины, к которой сводится хорошо известная задача о назначении (подробнее о ней говорится в следующей главе 9).

Для упрощения обозначений в данной главе рассматриваются простые ориентированные и неориентированные графы без петель. Напомним, что простым называется граф (орграф), любые две вершины которого соединены не более чем одним ребром (двумя противоположно направленными дугами). Все понятия и результаты главы легко переносятся с неориентированных графов на ориентированные и наоборот. Переносятся они и на случай графов общего вида (с «параллельными» дугами или рёбрами), которые иногда называются мультиграфами (в этой терминологии графами называются только простые графы).

С каждой дугой (ребром) (x, y) заданного орграфа (графа) G ассоциировано число $l(x, y)$, называемое ***длиной*** дуги (ребра). Содержательно это число может быть расстоянием, стоимостью, пропускной способностью и т.д. Удобно считать, что рассматриваемые орграфы (графы) являются полными, т.е. любые две различные вершины соединены двумя противоположно ориентированными дугами (одним ребром), но для реально отсутствующих дуг $l(x, y) = \infty$.

С каждым путём P в графе (орграфе) G связано неотрицательное число $L(P)$ (далее называемое ***длиной пути*** P). В этом и следующих двух разделах предполагается, что длина пути равна сумме длин входящих в него рёбер (дуг). Для любых двух вершин графа G ***минимальным или кратчайшим путём*** называется любой соединяющий их путь P с минимальной длиной $L(P)$. Подробнее: путь P, соединяющий заданные вершины a и b, называется минимальным или кратчайшим, если для любого другого пути P', соединяющего те же самые вершины a и b, верно, что $L(P) \le L(P')$. В случаях, когда длины дуг могут быть отрицательными, задача нахождения кратчайшего пути в такой общей постановке может не иметь решения. В этих случаях обычно рассматриваются не все пути, а только простые пути или цепи (см. раздел 6-3), т.е. пути, проходящие через любую вершину или дугу не более одного раза.

Основными задачами, рассматриваемыми и решаемыми в настоящей главе, являются следующие:

1) ***для заданной вершины найти кратчайшие пути, соединяющие её со всеми остальными вершинами графа G*;**

2) ***для всех пар вершин графа G найти соединяющие их кратчайшие* пути.**

Конечно, решение первой задачи, применённое ко всем вершинам графа, даёт решение второй задачи, а решение второй задачи даёт решение первой для всех вершин. Различие состоит в алгоритмах, непосредственно направленных на решение либо первой, либо второй задачи. Одним из наиболее эффективных методов решения первой задачи является алгоритм Дейкстры (АД), а второй задачи – алгоритм Флойда-Уоршолла (АФУ).

Материал главы организован следующим образом. В разделе 2 вводится в рассмотрение и излагается АД. В разделе 3 вводится в рассмотрение и излагается алгоритм АФУ. Здесь же в отдельном подразделе устанавливается возможность нахождения циклов отрицательной длины с помощью АФУ. В разделе 4 рассматривается другая зависимость длины пути от входящих в него рёбер (дуг). Именно, считается, что длиной пути является длина того ребра (той дуги), которая является максимальной среди длин всех рёбер (дуг), входящих в данный путь.

Как и в других главах, изложение сопровождается иллюстрирующими примерами; в конце главы даются опирающиеся на эти примеры задания.

2. Алгоритм Дейкстры

Идея алгоритма построения дерева кратчайших путей и определения их весов, предложенного в 1959 г. Е. Дейкстрой, такова. На каждом этапе одна новая вершина включается в множество S отмеченных вершин, для которых кратчайшие пути из начальной вершины a уже найдены. Тогда на следующем этапе к множеству S добавляется вершина b с самым коротким путем из a, все вершины которого, кроме b, содержатся в S.

Предполагается, что неориентированный граф $G = \langle X, E\rangle$ содержит n вершин. Без ограничения общности можно считать, что все вершины задаются номерами от 0 до n–1 и что начальная вершина имеет номер 0. Длина ребра (i, j) между вершинами i и j обозначена через $l(i, j)$; если нет ребра, соединяющего эти вершины, то $l(i, j) = \infty$. Длина пути $L(P)$ равна сумме длин входящих в него рёбер. Для того, чтобы на рисунках не перепутывались длины рёбер и номера вер-шин, вершины на них обозначены как X0, X1, и т.д. вместо 0, 1, и т.д.

Алгоритм Дейкстры (АД). Рассматриваемый алгоритм основан на расстановке ***постоянных и временных меток*** у вершин графа G. Метка является вещественным числом или символом ∞.

Как постоянная, так и временная метка у любой вершины x представляет собой длину некоторого пути из начальной вершины в данную вершину x. Если метка является символом ∞, то это значит, что никакой путь из начальной вершины в данную вершину x ещё не определён. На каждой итерации одна из временных меток становится постоянной и далее уже не меняется (она равна длине искомого кратчайшего пути в данную вершину). Постоянные метки обозначены буквой P, временные – буквой T. Через c обозначена последняя (на данный момент) помеченная вершина, через $R(x)$ – вершина, предшествующую вершине x на кратчайшем пути из начальной вершины 0 в вершину x.

Шаг 0 (Инициализация). Положить

$P(0) = 0$, $c = 0$, $T(x) = \infty$ (для всех вершин $x \neq 0$); $R(x)$ при инициализации не задаётся, поскольку предшествующие вершины ещё не определены).

Шаг *i*. Выполнить следующие операции:

А. Для каждой вершины x с временной меткой сделать следующее:

1) положить

$$Z = P(c) + l(c, x); \quad (1)$$

2) если $Z<T(x)$, положить $T(x) = Z$, $R(x) = c$. В противном случае $T(x)$ и $R(x)$ не меняются.

Б. Найти вершину x с минимальной временной меткой (если несколько вершин имеют од-ну и ту же минимальную временную метку, можно выбрать любую из них); положить $c = x$; положить в качестве постоянной метки $P(c)$ её временную метку $T(c)$.

В. Если ещё остались вершины с временными метками, положить $i = i+1$ и перейти к шагу i. В противном случае переходим к шагу ***F***.

Шаг*F*. На этом шаге определяются кратчайшие пути из начальной вершины 0 во все вершины $x\neq 0$. Кратчайший путь из начальной вершины 0 в любую другую вершину x строится следующим образом: вершиной, предшествующей x на искомом пути, является вершина $y = R(x)$; вершиной, предшествующей y на искомом пути, является вершина $z = R(y)$, и т.д., вплоть до начальной вершины 0. Искомый путь из 0 в x состоит из тех же вершин в обратном порядке ■

При «ручной» реализации АД будет удобно промежуточные результаты записывать в таблицу, строки которой соответствуют итерациям, а столбцы (кроме двух левых) – вершинам. В самом левом столбце будем писать номер итерации, а 2-ой слева столбец выделим для записи очередной помечаемой вершины c и её постоянной метки $P(c)$. Каждую клетку в остальных столбцах таблицы разделим на 3 части, в которые будем записывать текущее значение Z, новое значение временной метки T и новую предыдущую вершину R, или писать старые значения, в зависимости от результата сравнения в пункте 2) шага i-А. После того, как вершина получила постоянную метку, в её клетки <u>ничего не записываем</u>. Числа $l(c,x)$ – длины рёбер – берутся непосредственно из рисунка.

Пример 1. Рассмотрим применение АД для графа, показанного на рис.1. Длины рёбер написаны на рисунке прямо около них. Составим вышеупомянутую таблицу.

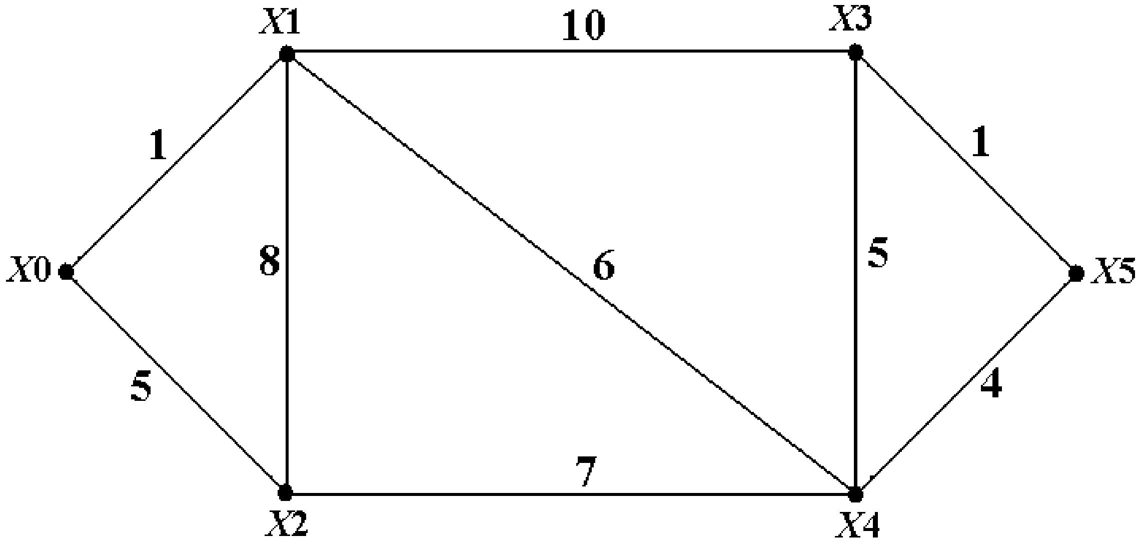


Рис.1. Граф с длинами рёбер

Таблица 1

Итерация	Последняя помеченная		Вершина 0			Вершина 1			Вершина 2			Вершина 3			Вершина 4			Вершина 5		
№	c	P	Z	T	R	Z	T	R	Z	T	R	Z	T	R	Z	T	R	Z	T	R
0	0	0					∞			∞			∞			∞			∞	
1	1	1				1	1	0	5	5	0	∞	∞		∞	∞		∞	∞	
2	2	5							9	5	0	11	11	1	7	7	1	∞	∞	
3	4	7										∞	11	1	12	7	1	∞	∞	
4	3	11										16	11	1				11	11	4
5	5	11																12	11	4

Дадим пояснения к заполнению таблицы 1. На 0-ой итерации (инициализации) в соответствии с АД начальной вершине 0 присваиваем постоянную метку 0 (оба числа заносятся в 0-ую строку в столбце «Последняя помеченная»). Остальные вершины получают временную метку ∞, которая и заносится в 0-ую строку под буквами T. В остальные позиции 0-ой строки в соответствии с АД ничего не заносится.

На 1-ой итерации обрабатываются только столбцы, соответствующие вершинам, имеющим временные метки. Таковыми в этот момент являются все вершины, кроме вершины 0. Вычисляем числа Z для всех этих вершин. Так как на 0-ой итерации $c = 0$, $P(c) = 0$, то имеем (см. граф) $P(0)+l(0,1)=1$, $P(0)+l(0,2)=5$, $P(0)+l(0,x)=\infty$ для $x=3,4,5$. Поэтому в 1-ую строку под знаком Zзаписываются числа 1, 5 и знаки ∞, ∞, ∞. Далее, сравнивая эти значения со значениями T из предыдущей строки, записываем под знаком T в данную строку минимумы, а под знаком R в тех столбцах, где произошли изменения в T – последнюю помеченную вершину из столбца c (в данном случае 0). Выбирая минимальное значение из записанных значений T, получаем 1 в столбце для вершины 1. Поэтому записываем 1 (номер вершины) под c и 1 (длина кратчайшего пути) под P.

На 2-ой итерации обрабатываются только столбцы, соответствующие вершинам 2, 3, 4 и 5. Так как на 1-ой итерации $c=1$, $P(c)=1$, то имеем (см. граф) $P(1)+l(1,2)=9$, $P(1)+l(1,3)=11$, $P(1)+l(1,4)=7$, $P(1)+l(1,5)=\infty$. Поэтому во 2-ую строку под знаком Z записываются числа 9,11,7 и знак ∞. В клетке справа и сверху от этих чисел записаны (под знаком T) предыдущие временные метки: 5,∞,∞,∞. В клетку во 2-ой строке под знаком T записываем новые временные метки – наименьшие значения из этих двух чисел; получаем 5,11,7 и ∞. Под знаком R в 3-ей и 4-ой группах столбцов во 2-ой строке пишется текущий номер $c=1$. Выбирая минимальное значение из записанных значений T, получаем 5 в столбце для вершины 2. Поэтому записываем 2 (номер вершины) под c и 5 (длина кратчайшего пути) под P.

На 3-ьей итерации обрабатываются только столбцы, соответствующие вершинам 3,4 и 5. Так как на 2-ой итерации $c=2$, $P(c)=5$, то имеем (см. граф) $P(2)+l(2,3)=\infty$, $P(2)+l(2,4)=12$, $P(2)+l(2,5) = \infty$. Поэтому в 3-ью строку под знаком Z записываются ∞, 12, ∞. В клетке справа и сверху от этих чисел записаны (под знаком T) предыдущие временные метки: 11, 7,∞. В клетку в 3-ьей строке под знаком T записываем новые временные метки - наименьшие значения из этих пар чисел: 11, 7 и ∞. Под знаком R во всех трёх столбцах ничего не меняется, так как временные метки в них не изменились. Выбирая минимальное значение из записанных значений T, получаем 7 в столбце для вершины 4. Поэтому записываем 4 (номер вершины) под c и 7 (длина кратчайшего пути) под P.

На 4-ой итерации рассматриваются только две вершины: 3 и 5. Новые временные метки для них совпадают. В этом случае можно выбрать любую. Выбираем вершину 3 и в соответствии с этим заполняем 4-ую строку и далее – последнюю, 5-ую строку.

Найдём теперь, в соответствии с шагом F АД, сами кратчайшие пути. Просматриваем вершины в том порядке, в каком они идут во 2-ом столбце (в этом порядке они получают постоянные метки).

Для вершины 1 последнее заполненное значение под знаком R равно 0. Это означает, что предпоследней вершиной на кратчайшем пути из 0 в 1 является начальная вершина 0. Поэтому имеем путь 0→1 длины 1.

Для вершины 2 последнее заполненное значение под знаком R равно 0. Это означает, что предпоследней вершиной на кратчайшем пути из 0 в 2 является начальная вершина 0. Поэтому имеем путь 0→2 длины 5.

Для вершины 4 последнее заполненное значение под знаком R равно 1. Это означает, что предпоследней вершиной на кратчайшем пути из 0 в 4 является вершина 1. Поскольку для вершины 1 кратчайший путь таков: 0→1, то имеем путь 0→1→4 длины 7.

Для вершины 3 последнее заполненное значение под знаком R равно 1. Это означает, что предпоследней вершиной на кратчайшем пути из 0 в 3 является вершина 1. Поскольку для вершины 1 кратчайший путь таков: 0→1, то имеем путь 0→1→3 длины 11.

Для вершины 5 последнее заполненное значение под знаком R равно 4. Это означает, что предпоследней вершиной на кратчайшем пути из 0 в 5 является вершина 4. Поскольку для вершины 4 кратчайший путь таков: 0→1→4, то имеем путь 0→1→4→5 длины 11 ■

Пример 2. Рассмотрим применение АД для графа, показанного на рис.2. Длины рёбер написаны на рисунке прямо около них. Составим вышеупомянутую таблицу 2. Она содержит все ответы, найденные аналогично примеру 1.

Кратчайшие пути из вершины 0 таковы: в вершину 1: 0→1; в вершину 5: 0→5; в вершину 2: 0→1→2; в вершину 3: 0→1→3; в вершину 4: 0→1→4; в вершину 7: 0→1→2→7; в вершину 6: 0→5→6. Длины путей находятся в столбцах «Последний помеченный». Слева стоят номера вершин в порядке получения постоянных меток; справа – длины кратчайших путей в эти вершины■

Замечание. Рассмотренный алгоритм находит кратчайшие пути в неориентированном графе. Однако этот же самый алгоритм находит кратчайшие ориентированные пути в ориентированном графе (см. все определения в разделе 6-3). Просто в описании алгоритма надо все упоминаемые там рёбра заменить на дуги. В частности, $l(c, x)$ будет означать длину дуги (c, x), ведущей из вершины c в вершину x.

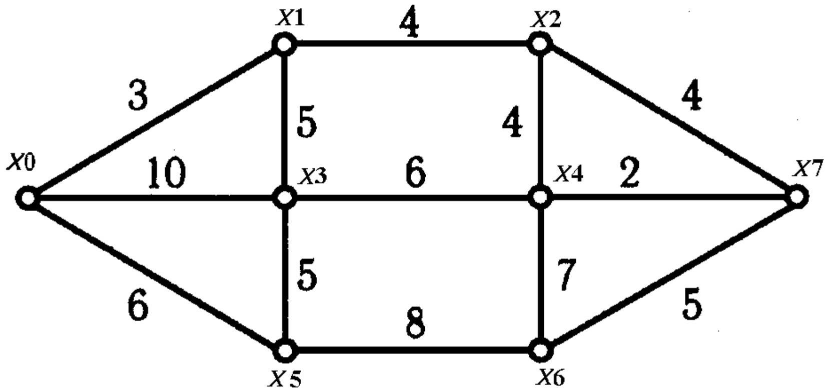


Рис.2

Таблица 2

Итера-ции	Посл помеч		Вершина 0			Вершина 1			Вершина 2			Вершина 3			Вершина 4			Вершина 5			Вершина 6			Вершина 7		
№	c	P	Z	T	R	Z	T	R	Z	T	R	Z	T	R	Z	T	R	Z	T	R	Z	T	R	Z	T	R
0	0	0		0			∞			∞			∞			∞			∞			∞			∞	
1	1	3				3	3	0	∞	∞		10	10	0	∞	∞		6	6	0	∞	∞		∞	∞	
2	5	6							7	7	1	8	8	1	∞	∞		∞	6	0	∞	∞		∞	∞	
3	2	7							∞	7	1	11	8	1	∞	∞					14	14	5	∞	∞	
4	3	8										∞	8	1	11	11	2				∞	14	5	11	11	2
5	4	11													14	11	2				∞	14	5	∞	11	2
6	7	11																			18	14	5	13	11	2
7	6	14																			16	14	5			

3. Алгоритм Флойда-Уоршолла

В этом разделе рассматривается другой способ решения задачи о кратчайших путях для всех пар вершин ориентированного графа. Для упрощения текста будем в пределах этого раздела пользоваться более привычными словами «граф», «путь», «цикл» вместо «орграф», «орпуть», «орцикл» и т.д. В отличие от алгоритма Дейкстры, в АФУ допускаются дуги с отрицательной длиной. Однако циклы с отрицательной длиной запрещаются; точнее, предполагается, что такие циклы в рассматриваемом графе $G = \langle V,A \rangle$ отсутствуют. Если же циклы отрицательной длины присутствуют, то этот факт выясняется уже в процессе выполнения алгоритма

АФУ основан на представлении пути, ведущего из вершины a в вершину b и проходящего через промежуточную вершину c как последовательность двух путей: из a в c и из c в b. Понятно, что если этот путь является кратчайшим среди всех путей того же вида (т.е. ведущих из a в b через c), то указанные части также должны быть кратчайшими. В алгоритме особую роль играют промежуточные вершины с максимальным номером (в некоторой заданной нумерации вершин). ***Промежуточной*** вершиной простого пути $p = \langle v_1, v_2, ..., v_l \rangle$ будем называть любую из вершин $v_2, ..., v_{l-1}$.

Будем считать, что вершины графа $G = \langle V, A \rangle$ занумерованы числами от 1 до n. Длина дуги с началом в вершине i и концом в вершине j обозначена через $l(i, j)$ (как и в других ситуациях, $l(i, j) = 0$, если $i = j$, и $l(i, j) = \infty$, если дуги (i, j) в графе нет). Рассмотрим произвольное $k \le n$. Для данной пары вершин $i, j \in V$ рассмотрим все пути из i в j, у которых все промежуточные вершины принадлежат множеству $\{1, 2, \ldots, k\}$. Пусть p − путь минимальной длины среди всех таких путей. Он является простым путём, поскольку предполагается, что в графе нет циклов отрицательной длины. Как найти длину этого пути, зная длины всех таких (кратчайших) путей для всех пар вершин при ме́ньших k?

Для пути p есть две возможности.

Если вершина k не входит в путь p, все промежуточные вершины пути p содержатся в множестве $\{1, 2, \ldots, k-1\}$. Тогда путь p является кратчайшим путём из i в j, промежуточные вершины которого принадлежат множеству $\{1, 2, \ldots, k-1\}$. Если k является промежуточной вершиной пути p, она разбивает его на два участка p_1 и p_2 (вершина k встречается лишь однажды, так как p − простой путь). Поэтому путь p_1 будет кратчайшим путём из i в k, путь p_2 будет кратчайшим путём из k в j, а промежуточными вершинами на обоих путях будут вершины из множества $\{1, 2, \ldots, k-1\}$. Эти рассуждения лежат в основе излагаемого ниже алгоритма.

Алгоритм Флойда-Уоршолла (АФУ). Алгоритм находит кратчайшие пути для всех пар вершин заданного ориентированного графа G. Для реализации алгоритма вводятся четыре квадратные матрицы размера $n\times n$, где n – число вершин в заданном графе G:

Матрица $D = (d_{ij})$ текущих кратчайших расстояний;

Вспомогательная матрица $H = (h_{ij})$ для пересчёта текущих кратчайших расстояний;

Матрица предшествия $\Pi = (\pi_{ij})$ (матрица предпоследних вершин на текущих кратчайших путях из i в j);

Вспомогательная матрица $\Psi = (\psi_{ij})$ для пересчёта предпоследних вершин.

В процессе работы алгоритма происходит заполнение матриц и изменение их элементов. Алгоритм прекращает работу после n-го шага.

Шаг 0 (Инициализация). Положить

$d_{ij} = l(i, j)$ $(i, j = 1, \ldots, n)$;

$$\pi_{ij}= \begin{cases} \bullet, \text{ если } i = j \text{ или } l(i,j) = \infty, \\ i, \text{ если } i\neq j \text{ и } l(i,j) < \infty \end{cases} (i,j = 1,\ldots,n).$$

Шаг k $(1\le k \le n)$.

1. Для всех $i, j = 1, \ldots, n$ выполнить следующие операции:

   1.1. Положить

   $$z = d_{ik} + d_{kj}, \qquad (2)$$

   1.2. Если $z<d_{ij}$, то положить $h_{ij} = z$ и $\psi_{ij} = \pi_{kj}$. В противном случае положить $h_{ij} = d_{ij}$ и $\psi_{ij} = \pi_{ij}$. Заметим, что из $d_{ik} = \infty$ следует, что $z = \infty$ и неравенство $z<d_{ij}$ не выполняется.

2. Для всех $i, j = 1, \ldots, n$ положить $d_{ij} = h_{ij}$, $\pi_{ij}= \psi_{ij}$.

Шаг F. На этом шаге определяются кратчайшие пути для всех пар вершин. Кратчайший путь из начальной вершины i в любую другую вершину j строится следующим образом: вершиной, предшествующей j на искомом пути, является вершина $p = \pi_{ij}$; вершиной, предшествующей p на искомом пути, является вершина $q = \pi_{ip}$, и т.д., вплоть до начальной вершины i. Искомый путь из i в j состоит из тех же вершин в обратном порядке ■

При «ручной» реализации АФУ будет удобно результаты каждой итерации записывать в последовательные таблицы специального вида – по одной на каждую итерацию. Реализация АФУ состоит в последовательном заполнении таблиц и переносе части результатов в следующую таблицу. Эта схема подробно описана при нахождении всех кратчайших путей в конкретном графе, рассмотренном в следующем примере.

Пример 3. Проиллюстрируем работу АФУ на примере графа, показанного на рис.3. На этом рисунке рядом с вершинами показаны их номера; более крупным шрифтом рядом с дугами показаны их длины.

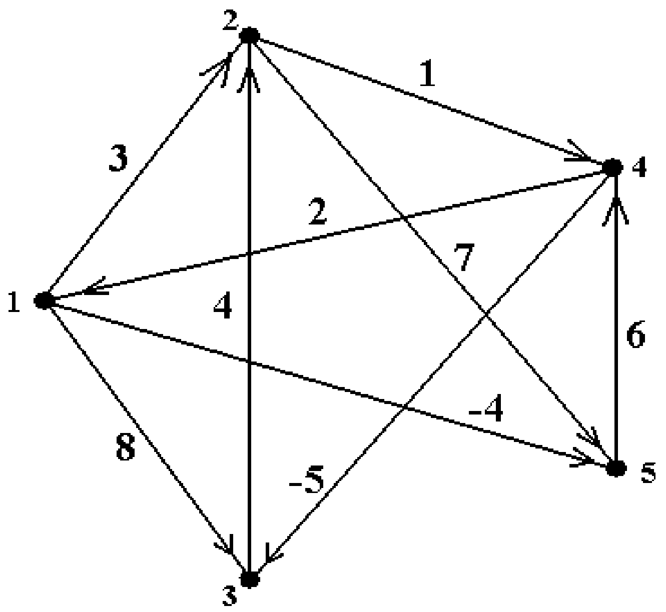


Рис. 3. Ориентированный граф с 5-ью вершинами и 9-ью дугами

Все операции алгоритма являются операциями над данными, записанными в таблицу типа таблицы 3. В эту таблицу входит 5×5 ячеек, каждая из которых состоит из трёх клеток. Эти клетки будут называться левой, средней и правой. Кроме этих 25 ячеек, таблица содержит левый столбец, состоящий из 5-и клеток, и верхнюю строку, состоящую из 5 ячеек, содержащих по 2 клетки каждая (левую и правую). Их роль будет указана ниже.

Таблица 3. Исходная таблица для АФУ

№		1			2			3			4			5		
1																
2																
3																
4																
5																

Инициализация. Она состоит в заполнении таблицы 4.1а, совпадающей с таблицей 3. Последовательно рассматриваются все дуги исходного графа, показанного на рис.3. Если дуга ведёт из вершины i в вершину j, то в среднюю клетку ячейки с индексом (i, j) (т.е. находящуюся на пересечении i-ой строки и j-го столбца) вставляется длина дуги из вершины i в вершину j, а в правую клетку – номер i. После этого во всех ячейках с совпадающими индексами (i, i) в среднюю клетку записывается 0. Наконец, во всех остальных ячейках в средние клетки напишем знак ∞. Больше ничего в таблицу 4.1а не записывается. Все упомянутые данные берутся непосредственно из рис.3. В дальнейших конструкциях рисунок не участвует.

Таблица 4. Результат инициализации

№		1			2			3			4			5		
1			0			3	1		8	1		∞			−4	1
2			∞			0			∞			1	2		7	2
3			∞			4	3		0			∞			∞	
4			2	4		∞			−5	4		0			∞	
5			∞			∞			∞			6	5		0	

Итерация 1. k = 1. На итерации 1 происходит последовательное преобразование данных, полученных при инициализации и представленных в таблице 4. Сначала заполняются левый столбец и верхняя строка. В i-ую клетку левого столбца копируется содержимое из средней

клетки ячейки с индексом (i, k) (номер k равен номеру итерации, и в данном случае k = 1). В j-ую ячейку верхней строки копируется содержимое средней и правой клеток ячейки с индексом (k, j) (k = 1). Результат этой операции приведён в таблице 4.1a. Заметим, что при копировании пустой клетки её копия также будет пустой клеткой.

Таблица 4.1a. Заполнение левого столбца и верхней строки

№		1			2			3			4			5		
		0			3		1	8		1	∞			−4		1
1	0		0			3	1		8	1		∞			−4	1
2	∞		∞			0			∞			1	2		7	2
3	∞		∞			4	3		0			∞			∞	
4	2		2	4		∞			−5	4		0			∞	
5	∞		∞			∞			∞			6	5		0	

Следующей операцией является заполнение левых клеток каждой из 25 ячеек. Правило заполнения достаточно простое. В ячейку с индексом (i, j) вставляется сумма чисел из i-ой клетки левого столбца и левой половины j-ой клетки верхней строки. Операция соответствует пункту 1.1 общего шага АФУ. Заметим, что для любого x (в том числе $x = \infty$) $x + \infty = \infty + x = \infty$. Результаты данной операции представлены в следующей таблице 4.1b.

Таблица 4.1b. Заполнение сумм

№		1			2			3			4			5		
		0			3		1	8		1	∞			−4		1
1	0	0	0		3	3	1	8	8	1	∞	∞		−4	−4	1
2	∞	∞	∞		∞	0		∞	∞		∞	1	2	∞	7	2
3	∞	∞	∞		∞	4	3	∞	0		∞	∞		∞	∞	
4	2	2	2	4	5	∞		10	−5	4	∞	0		−2	∞	
5	∞	∞	∞		∞	∞		∞	∞		∞	6	5	∞	0	

Следующая операция – изменение (в некоторых случаях) записей в средних и правых клетках всех ячеек. Правило также очень простое:

1. Если число в левой клетке меньше числа в средней клетке, то

   1.1. Число из левой клетке записывается в среднюю клетку.

   1.2. В правую клетку записывается номер из правой клетки верхней строки, находящейся в одном столбце с рассматриваемой ячейкой.

2. В противном случае ничего не делается.

В качестве примера рассмотрим ячейку (4,2) в которой число в левой клетке (5) меньше числа в средней клетке (∞). Тогда число 5 надо записать в среднюю клетку вместо ∞, а в правую клетку надо записать номер 1. Легко понять, что описанная операция (выделение минимумов и копирование значений) в точности представляет собой операции пунктов 1.2 и 2 общего шага АФУ. Результаты её выполнения записаны в таблице 4.1c. Заметим, что числа в средних клетках ячейки (i, j) – это длины построенных на рассматриваемой итерации и ранее путей из вершины i в вершину j, а номера в правых клетках – это предпоследние вершины на этих путях.

Таблица 4.1c. Заполнение минимумов и предпоследних вершин

№		1			2			3			4			5		
		0			3		1	8		1	∞			−4		1
1	0	0	0		3	3	1	8	8	1	∞	∞		−4	−4	1
2	∞	∞	∞		∞	0		∞	∞		∞	1	2	∞	7	2
3	∞	∞	∞		∞	4	3	∞	0		∞	∞		∞	∞	
4	2	2	2	4	5	5	1	10	−5	4	∞	0		−2	−2	1
5	∞	∞	∞		∞	∞		∞	∞		∞	6	5	∞	0	

Для завершения итерации 1 осталось сделать следующее. Числа из левых клеток всех ячеек удаляются. Удаляются также все данные из левого столбца и верхней строки. В результате получаем таблицу 4.1d, которая по форме совпадает с таблицей 4, полученной в результате инициализации.

Таблица 4.1d. Результат итерации 1

№		1			2			3			4			5		
1			0			3	1		8	1		∞			−4	1
2			∞			0			∞			1	2		7	2
3			∞			4	3		0			∞			∞	
4			2	4		5	1		−5	4		0			−2	1
5			∞			∞			∞			6	5		0	

<u>Итерация 2</u>. $k = 2$. На итерации 2 происходит последовательное преобразование данных, полученных на итерации 1 и представленных в таблице 4.1d. Естественно, что алгоритм не меняется, поэтому подробных объяснений к каждой таблице, как на итерации 1, не даётся. Выполнение итерации 2 демонстрируется в таблицах 4.2a – 4.2d, аналогичных таблицам 4.1a – 4.1d. Заметим, что в левый столбец копируются данные из 2-го столбца, а в верхнюю строку – из 2-ой строки таблицы 4.2a, так как $k = 2$.

Таблица 4.2a. Заполнение левого столбца и верхней строки

№		1			2			3			4			5		
		∞			0			∞			1		2	7		2
1	3		0			3	1		8	1		∞			−4	1
2	0		∞			0			∞			1	2		7	2
3	4		∞			4	3		0			∞			∞	
4	5		2	4		5	1		−5	4		0			−2	1
5	∞		∞			∞			∞			6	5		0	

Таблица 4.2b. Заполнение сумм

№		1			2			3			4			5		
		∞			0			∞			1		2	7		2
1	3	∞	0		3	3	1	∞	8	1	4	∞	2	10	−4	1
2	0	∞	∞		0	0		∞	∞		1	1	2	7	7	2
3	4	∞	∞		4	4	3	∞	0		5	∞		11	∞	
4	5	∞	2	4	5	5	1	∞	−5	4	6	0		12	−2	1
5	∞	∞	∞		∞	∞		∞	∞		∞	6	5	∞	0	

Таблица 4.2c. Заполнение минимумов и предпоследних вершин

№		1			2			3			4			5		
		∞			0			∞			1		2	7		2
1	3	∞	0		3	3	1	∞	8	1	4	4	2	10	−4	1
2	0	∞	∞		0	0		∞	∞		1	1	2	7	7	2
3	4	∞	∞		4	4	3	∞	0		5	5	2	11	11	2
4	5	∞	2	4	5	5	1	∞	−5	4	6	0		12	−2	1
5	∞	∞	∞		∞	∞		∞	∞		∞	6	5	∞	0	

Таблица 4.2d. Результат итерации 2

№		1			2			3			4			5	
1			0			3	1		8	1		4	2		−4	1
2			∞			0			∞			1	2		7	2
3			∞			4	3		0			5	2		11	2
4			2	4		5	1		−5	4		0			−2	1
5			∞			∞			∞			6	5		0	

<u>Итерация 3</u>. $k = 3$. На итерации 3 происходит последовательное преобразование данных, полученных на итерации 2 и представленных в таблице 4.2d. Выполнение итерации 3 демонстрируется в таблицах 4.3a – 4.3d.

Таблица 4.3a. Заполнение левого столбца и верхней строки

№		1			2			3			4			5		
		∞			4		3	0			5		2	11		2
1	8		0			3	1		8	1		4	2		−4	1
2	∞		∞			0			∞			1	2		7	2
3	0		∞			4	3		0			5	2		11	2
4	−5		2	4		5	1		−5	4		0			−2	1
5	∞		∞			∞			∞			6	5		0	

Таблица 4.3b. Заполнение сумм

№		1			2			3			4			5		
		∞			4		3	0			5		2	11		2
1	8	∞	0		12	3	1	8	8	1	13	4	2	19	−4	1
2	∞	∞	∞		∞	0		∞	∞		∞	1	2	∞	7	2
3	0	∞	∞		4	4	3	0	0		5	5	2	11	11	2
4	−5	∞	2	4	−1	5	1	−5	−5	4	0	0		6	−2	1
5	∞	∞	∞		∞	∞		∞	∞		∞	6	5	∞	0	

Таблица 4.3c. Заполнение минимумов и предпоследних вершин

№		1			2			3			4			5		
		∞			4		3	0			5		2	11		2
1	8	∞	0		12	3	1	8	8	1	13	4	2	19	−4	1
2	∞	∞	∞		∞	0		∞	∞		∞	1	2	∞	7	2
3	0	∞	∞		4	4	3	0	0		5	5	2	11	11	2
4	−5	∞	2	4	−1	−1	3	−5	−5	4	0	0		6	−2	1
5	∞	∞	∞		∞	∞		∞	∞		∞	6	5	∞	0	

Таблица 4.3d. Результат итерации 3

№		1			2			3			4			5		
1			0			3	1		8	1		4	2		−4	1
2			∞			0			∞			1	2		7	2
3			∞			4	3		0			5	2		11	2
4			2	4		−1	3		−5	4		0			−2	1
5			∞			∞			∞			6	5		0	

Итерация 4. k = 4. На итерации 4 преобразование данных показано с ме́ньшей детализацией. Требуемые операции выполняются так же, как и ранее, но промежуточные результаты не записываются в новые таблицы, а последовательно записываются в одни и те же ячейки и клетки. Результаты приводятся в таблице 4.4.

Таблица 4.4. Результат итерации 4

№		1			2			3			4			5		
1			0			3	1		−1	4		4	2		−4	1
2			3	4		0			−4	4		1	2		−1	1
3			7	4		4	3		0			5	2		3	1
4			2	4		−1	3		−5	4		0			−2	1
5			8	4		5	3		1	4		6	5		0	

Итерация 5. k = 5. Эта итерация является последней перед финальным шагом F. В таблице 4.5 содержатся кратчайшие расстояния для всех пар вершин и вершины, предпоследние на каждом таком пути.

Таблица 4.5. Результат итерации 5

№		1			2			3			4			5		
1			0			1	3		−3	4		2	5		−4	1
2			3	4		0			−4	4		1	2		−1	1
3			7	4		4	3		0			5	2		3	1
4			2	4		−1	3		−5	4		0			−2	1
5			8	4		5	3		1	4		6	5		0	

Шаг F. Найдём сами кратчайшие пути, используя правые клетки в ячейках. Напомним, что число в правой клетке в ячейке (i, j) – это номер вершины, предшествующей на кратчайшем пути из i в j вершине j, т.е. номер предпоследней вершины на этом пути.

Для пути из 1 в 2 такой вершиной является вершина 3 (см. таблицу 4.5); далее, для пути из 1 в 3 такой вершиной является вершина 4; для пути из 1 в 4 такой вершиной является вершина 5; наконец, для пути из 1 в 5 такой вершиной является вершина 1. Таким образом, кратчайший путь из вершины 1 в вершину 2 таков: 1→5→4→3→2. По построению, все промежуточные пути: 1→5→4→3; 1→5→4; 1→5 также являются не только кратчайшими путями, но и именно теми кратчайшими путями, которые найдены данным алгоритмом. Все кратчайшие пути из вершины 1 уже найдены. Они таковы: 1→5→4→3→2; 1→5→4→3; 1→5→4; 1→5.

Далее, рассмотрим пути из вершины 2 в другие вершины. На пути из 2 в 1 предпоследней вершиной является вершина 4, а непосредственно перед ней – начальная вершина 2, и весь путь таков: 2→4→1. Далее, на пути из 2 в 3 предпоследней вершиной является 4, и путь таков: 2→4 →3. Путь из 2 в 4 состоит из одной дуги 2→4. Наконец, на пути из 2 в 5 предпоследней вершиной является 1, а путь из 2 в 1 уже известен, поэтому имеем кратчайший путь 2→4→1→5. Все кратчайшие пути из вершины 2 таковы: 2→4→1; 2→4→3; 2→4; 2→4→1→5.

Кратчайшие пути из вершины 3 таковы: 3→5→4→1; 3→2; 3→5→4; 3→5.

Кратчайшие пути из вершины 4 таковы: 4→1; 4→3→2; 4→3; 4→1→5.

Кратчайшие пути из вершины 5 таковы: 5→4→1; 5→4→3→2; 5→4→3; 5→4 ■

3.1. Циклы отрицательной длины. В АФУ предполагалось, что в заданном графе отсутствуют циклы отрицательной длины. При обосновании АФУ это требование существенно, так как само разбиение пути, проходящего через вершину k, на два пути, не содержащих k в качестве промежуточной вершины, возможно только при единственном вхождении kв путь, что, в свою очередь, гарантируется как раз отсутствием циклов отрицательной длины. Поскольку в достаточно сложных случаях сам факт наличия или отсутствия таких циклов не является очевидным, необходим специальный алгоритм, устанавливающий этот факт. Более того, в некоторых практически важных задачах (см. главу 9) требуется найти цикл отрицательной длины или установить их отсутствие безотносительно к нахождению кратчайших путей.

Оказывается, что решение этой проблемы является «побочным продуктом» АФУ. Подробнее. Применяем АФУ к произвольному графу, не выясняя заранее, содержит ли он циклы отрицательной длины. Для этого после каждого шага, начиная с 1-го (k=1), проверяем все числа на диагонали матрицы D (напомним, что после инициализации все они равны 0). Пусть на k-ой итерации хотя бы один элемент (скажем, d_{ii}) оказывается меньше 0. Тогда путь с началом и концом в вершине i, найденный на этой итерации, является циклом с началом и концом в вершине i, длина которого равна отрицательному числу d_{ii}. Это следует просто из описания алгоритма, который на каждом k-ом шаге находит текущие кратчайшие пути для всех пар вершин (включая пары с совпадающими вершинами).

Пример 4. Проиллюстрируем работу АФУ на примере графа, показанного на рис. 4. В нём есть цикл 3→1→2→3 отрицательной длины − 1. В соответствии с описанием «ручной» версии АФУ в примере 3, в результате инициализации получим таблицу 5. Далее, на итерации 1 из таблицы 5 получаем преобразованную таблицу 5.1 точно так же, как это делалось в примере 3, а затем точно так же получаем таблицы 5.2 и 5.3.

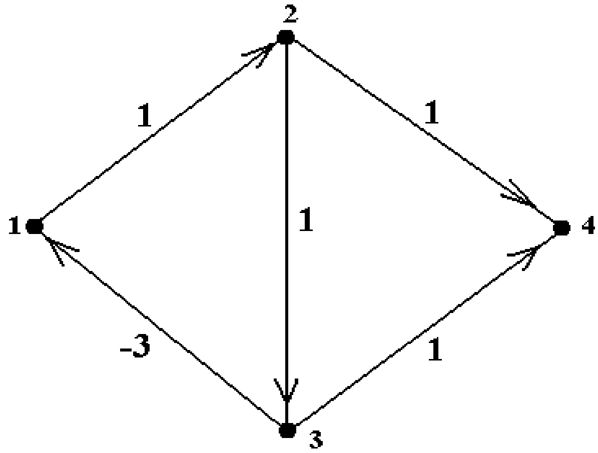


Рис. 4. Граф с отрицательным циклом

После выполнения итерации 3 обнаруживаем путь из вершины 1 в ту же самую вершину 1, на котором предпоследней вершиной является вершина 3, перед ней – вершина 2, а перед вершиной 2 – опять вершина 1. Таким образом, найден цикл 1→2→3→1; длина этого цикла, указанная в средней клетке ячейки (1, 1), равна −1, т.е. найден цикл отрицательной длины. Разумеется, одновременно найдены циклы 2→3→1→2 и 3→1→2→3, отличающиеся от цикла 1→2→3→1 только начальной вершиной; их дли-на также равна − 1.

Таблица 5. Результаты инициализации

№		1			2			3			4		
1			0			1	1		∞			∞	
2			∞			0			1	2		1	2
3			−3	3		∞			0			1	3
4			∞			∞			∞			0	

Таблица 5.1. Результат итерации 1

№		1			2			3			4		
1			0			1	1		∞			∞	
2			∞			0			1	2		1	2
3			−3	3		−2	1		0			1	3
4			∞			∞			∞			0	

Таблица 5.2. Результат итерации 2

№		1			2			3			4		
1			0			1	1		2	2		2	2

2			∞			0			1	2		1	2
3			−3	3		−2	1		−1	2		−1	2
4			∞			∞			∞			0	

Таблица 5.3. Результат итерации 3

№		1			2			3			4		
1			−1	3		0	1		1	2		1	2
2			−2	3		−1	1		0	2		0	2
3			−4	3		−3	1		−2	2		−2	2
4			∞			∞			∞			0	

■

Таким образом, АФУ позволяет не только установить факт наличия или отсутствия циклов отрицательной длины, но и найти по крайней мере один из них.

Замечание. Рассмотренный алгоритм находит кратчайшие пути и циклы отрицательной длины в ориентированном графе. Однако этот же самый алгоритм находит соответствующие пути и циклы в неориентированном графе (см. все определения в разделе 6-3). Просто в описании алгоритма надо все упоминаемые там дуги заменить рёбрами. Например, $l(c, x)$ обозначает длину ребра (c, x) с концами c и x.

4. Минимаксная модификация задачи о кратчайших путях

В некоторых практически и теоретически интересных случаях длиной пути естественно считать значение некоторой функции, выражающей – тем или иным образом – длину пути через длины всех входящих в него дуг или рёбер. Рассмотренная выше длина пути, равная сумме длин всех входящих в него рёбер, далеко не исчерпывает всех таких зависимостей. Не рассматривая – в силу скромных целей настоящего пособия – общую задача нахождения кратчайших путей с обобщёнными функциями длин, сосредоточим внимание лишь на одном варианте такой функции. Именно, определим длину $L(P)$ пути P формулой $L(P) = \max l(p, q)$, (3)

где максимум берётся по всем рёбрам пути P (в ориентированном случае – по всем дугам). Такая постановка называется минимаксной (иногда, имея в виду некоторые экономические интерпретации, её называют задачей на узкие места).

На первый взгляд, минимаксная задача заметно отличается от рассмотренной в предыдущих разделах 2 и 3 традиционной задачи на кратчайшие пути. Тем более удивительно, что решается она теми же самыми алгоритмами – АД и АФУ. Остановимся на этом подробнее.

4.1. Модифицированный алгоритм Дейкстры (МАД) для нахождения минимаксных путей из заданной начальной вершины во все остальные. Структура алгоритма остаётся той же самой. Единственное изменение касается пункта **А** 1) алгоритма. Формула (1) $Z = P(c) + l(c, x)$ заменяется формулой $Z = \max\{P(c), l(c, x)\}$. Больше ничего в АД не меняется.

Пример 5. Рассмотрим применение МАД для графа из примера 1 (этот граф для удобства чтения приводится здесь ещё раз). Составим таблицу 6, аналогичную таблице 1 (как уже говорилось, изменения коснутся только вычисления величины Z). В остальном правила заполнения таблицы не меняются.

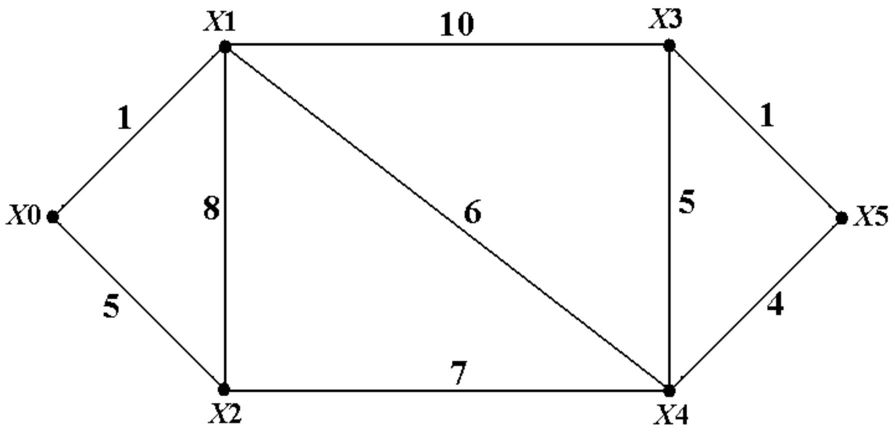


Таблица 6. Пример применения МАД

Итера-ция	Последняя помеченная		Вершина 0			Вершина 1			Вершина 2			Вершина 3			Вершина 4			Вершина 5		
№	*c*	*P*	*Z*	*T*	*R*	*Z*	*T*	*R*	*Z*	*T*	*R*	*Z*	*T*	*R*	*Z*	*T*	*R*	*Z*	*T*	*R*
0	0	0		0			∞			∞			∞			∞			∞	
1	1	1				1	1	0	5	5	0	∞	∞		∞	∞		∞	∞	
2	2	5							8	5	0	10	10	1	6	6	1	∞	∞	
3	4	6										∞	10	1	7	6	1	∞	∞	
4	3	6										6	6	4				6	6	4
5	6	6																6	6	4

Дадим пояснения к заполнению данной таблицы. На 0-ой итерации (инициализации) в соответствии с МАД начальной вершине 0 присваиваем постоянную метку 0 (оба числа заносятся в 0-ую строку в столбце «Последняя помеченная»). Остальные вершины получают временную метку ∞, которая и заносится в 0-ую строку под буквами *T*. В остальные позиции 0-ой строки в соответствии с МАД ничего не заносится.

На 1-ой итерации обрабатываются только столбцы, соответствующие вершинам, имеющим временные метки. Таковыми в этот момент являются все вершины, кроме вершины 0. Вычисляем числа *Z* для всех этих вершин. Так как на 0-ой итерации $c = 0$, $P(c) = 0$, то имеем (см. граф) $\max\{P(0), l(0,1)\} = 1$, $\max\{P(0), l(0,2)\} = 5$, $\max\{P(0), l(0,x)\} = \infty$ для $x = 3,4,5$. Поэтому в 1-ую строку под знаком *Z* записываются числа 1, 5 и знаки ∞, ∞, ∞. Далее, сравнивая эти значения со значениями *T* из предыдущей строки, записываем под знаком *T* в данную строку минимумы, а под знаком *R* в тех столбцах, где произошли изменения в *T* – последнюю помеченную вершину из столбца *c* (в данном случае 0). Выбирая минимальное значение из записанных значений *T*, получаем 1 в столбце для вершины 1. Поэтому записываем 1 (номер вершины) под *c* и 1 (длина кратчайшего пути) под *P*.

На 2-ой итерации обрабатываются только столбцы, соответствующие вершинам 2, 3, 4 и 5. Так как на 1-ой итерации c=1, $P(c)$=1, то имеем (см. граф) $\max\{P(1), l(1,2)\}=8$, $\max\{P(1), l(1,3)\} =10$, $\max\{P(1), l(1,4)\}=6$, $\max\{P(1), l(1,5)\}=\infty$. Поэтому во 2-ую строку под знаком *Z* записываются числа 8,10,6 и знак ∞. В клетках справа и сверху от этих чисел записаны (под знаком *T*) предыдущие временные метки: 5,∞,∞,∞. В клетки во 2-ой строке под знаком *T* записываем новые временные метки – наименьшие значения из этих двух чисел; получаем 5,10,6 и ∞. Под знаком *R* в 3-ей и 4-ой группах столбцов во 2-ой строке пишется текущий номер c=1. Выбирая минимальное значение из записанных значений *T*, получаем 5 в столбце для вершины 2. Поэтому записываем 2 (номер вершины) под *c* и 5 (длина кратчайшего пути) под *P*.

На 3-ьей итерации обрабатываются только столбцы, соответствующие вершинам 3,4 и 5. Так как на 2-ой итерации c=2, $P(c)$=5, то имеем (см. граф) $\max\{P(2), l(2,3)\}=\infty$, $\max\{P(2), l(2, 4)\}=7$, $\max\{P(2), l(2,5) = \infty$. Поэтому в 3-ью строку под знаком *Z* записываются ∞,7,∞. В клетке справа и сверху от этих чисел записаны (под знаком T) предыдущие временные метки: 10,6,∞. В клетку в 3-ьей строке под знаком T записываем новые временные метки – наименьшие значения из этих пар чисел; получаем 10,6 и ∞. Под знаком R во всех трёх столбцах ничего не меняется, так как временные метки в них не изменились. Выбирая минимальное значение из записанных значений T, получаем 6 в столбце для вершины 4. Поэтому записываем 4 (номер вершины) под c и 6 (длина кратчайшего пути) под P.

На 4-ой итерации рассматриваются только две вершины: 3 и 5. Новые временные метки для них совпадают. В этом случае можно выбрать любую. Выбираем вершину 3 и в соответствии с этим заполняем 4-ую строку и далее – последнюю, 5-ую строку.

Найдём теперь, в соответствии с шагом F МАД, сами кратчайшие пути. Просматриваем вершины в том порядке, в каком они идут во 2-ом столбце (в этом порядке они получают постоянные метки).

Для вершины 1 последнее заполненное значение под знаком R равно 0. Это означает, что предпоследней вершиной на кратчайшем пути из 0 в 1 является начальная вершина 0. Поэтому имеем путь 0→1 длины 1.

Для вершины 2 последнее заполненное значение под знаком R равно 0. Это означает, что предпоследней вершиной на кратчайшем пути из 0 в 2 является начальная вершина 0. Поэтому имеем путь 0→2 длины 5.

Для вершины 4 последнее заполненное значение под знаком R равно 1. Это означает, что предпоследней вершиной на кратчайшем пути из 0 в 4 является вершина 1. Поскольку для вершины 1 кратчайший путь таков: 0→1, то имеем путь 0→1→4 длины 6.

Для вершины 3 последнее заполненное значение под знаком R равно 4. Это означает, что предпоследней вершиной на кратчайшем пути из 0 в 3 является вершина 4. Поскольку для вершины 4 кратчайший путь таков: 0→1→4, то имеем путь 0→1→4→3 длины 6.

Для вершины 5 последнее заполненное значение под знаком R равно 4. Это означает, что предпоследней вершиной на кратчайшем пути из 0 в 5 является вершина 4. Поскольку для вершины 4 кратчайший путь таков: 0→1→4, то имеем путь 0→1→4→5 длины 6.

Напомним, что в данном случае длиной пути является не сумма длин рёбер, а длина максимального (в данном пути) ребра. Поэтому кратчайшие пути в одном и том же графе могут отличаться. Запишем их для сравнения рядом:

i	Кратчайший путь из вершины 0 в вершину i (сумма длин рёбер)	Длина пути	Кратчайший путь из вершины 0 в вершину i (длина максимального ребра)	Длина пути
1	0→1	1	0→1	1
2	0→2	5	0→2	5
3	0→1→3	11	0→1→4→3	6
4	0→1→4	7	0→1→4	6
5	0→1→4→5	11	0→1→4→5	6

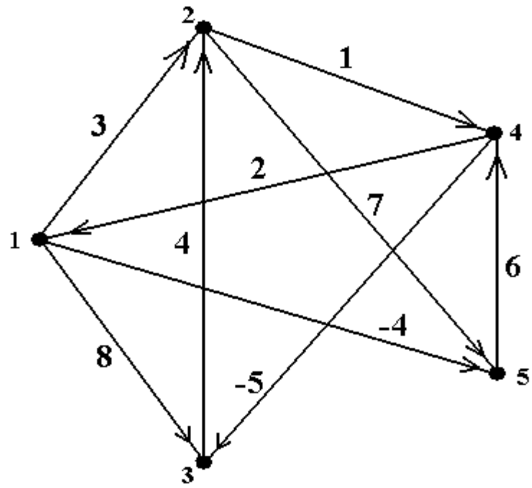
Таким образом, даже в этом простейшем случае решения при различных определениях длины не совпадают ■

4.2. Модифицированный алгоритм Флойда-Уоршолла (МАФУ) для нахождения минимаксных путей между всеми парами вершин. Структура алгоритма остаётся той же самой. Единственное изменение касается пункта 1.1 общего шага k алгоритма. Формула (2) заменяется формулой (4):

$$z = \max\{d_{ik}, d_{kj}\} \qquad (4)$$

Больше ничего в АФУ не меняется.

Пример 6. Рассмотрим применение МАФУ для графа из примера 3 (этот граф для удобст-



ва чтения приводится здесь ещё раз). Результат инициализации совпадает с результатом инициализации, представленным в таблице 4. Приведём здесь эту таблицу ещё один раз:

Таблица 7. Результат инициализации

№		1			2			3			4			5		
1			0			3	1		8	1		∞			−4	1
2			∞			0			∞			1	2		7	2
3			∞			4	3		0			∞			∞	
4			2	4		∞			−5	4		0			∞	
5			∞			∞			∞			6	5		0	

Сами правила заполнения таблиц, подробно описанные в примере 3, не меняются, кроме одного пункта: операция суммирования заменяется на операцию взятия максимума из тех же пар чисел. Ниже приводятся результаты всех 5-и последовательных итераций.

Таблица 7.1. Результат итерации 1

№		1			2			3			4			5		
1			0			3	1		8	1		∞			−4	1
2			∞			0			∞			1	2		7	2
3			∞			4	3		0			∞			∞	
4			2	4		3	1		−5	4		0			2	1
5			∞			∞			∞			6	5		0	

Таблица 7.2. Результат итерации 2

№		1			2			3			4			5		
1			0			3	1		8	1		3	2		−4	1
2			∞			0			∞			1	2		7	2
3			∞			4	3		0			4	2		7	2
4			2	4		3	1		−5	4		0			2	1
5			∞			∞			∞			6	5		0	

Таблица 7.3. Результат итерации 3

№		1		2		3		4		5	
1		0		3	1	8	1	3	2	−4	1
2		∞		0		∞		1	2	7	2
3		∞		4	3	0		4	2	7	2
4		2	4	3	1	−5	4	0		2	1
5		∞		∞		∞		6	5	0	

Таблица 7.4. Результат итерации 4

№		1		2		3		4		5	
1		0		3	1	3	4	3	2	−4	1
2		2	4	0		1	4	1	2	2	1
3		4	4	4	3	0		4	2	4	1
4		2	4	3	1	−5	4	0		2	1
5		6	4	6	1	6	4	6	5	0	

Таблица 7.5. Результат итерации 5

№		1		2		3		4		5	
1		0		3	1	3	4	3	2	−4	1
2		2	4	0		1	4	1	2	2	1
3		4	4	4	3	0		4	2	4	1
4		2	4	3	1	−5	4	0		2	1
5		6	4	6	1	6	4	6	5	0	

Шаг *F*. Найдём сами кратчайшие пути, используя правые клетки в ячейках. Напомним, что число в правой клетке в ячейке (i, j) – это номер вершины, предшествующей на кратчайшем пути из i в j вершине j, т.е. номер предпоследней вершины на этом пути.

Для пути из 1 в 2 такой вершиной является вершина 1 (см. таблицу 7.5); далее, для пути из 1 в 3 такой вершиной является вершина 4; для пути из 1 в 4 такой вершиной является вершина 2; наконец, для пути из 1 в 5 такой вершиной является вершина 1. Таким образом, кратчайшие пути из вершины 1 в другие вершины таковы: 1→2; 1→2→4→3; 1→2→4;1→5.

Кратчайшие пути из вершины 2 таковы: 2→4→1; 2→4→3; 2→4; 2→4→1→5.

Кратчайшие пути из вершины 3 таковы: 3→2→4→1; 3→2; 3→2→4; 3→2→4→1→5.

Кратчайшие пути из вершины 4 таковы: 4→1; 4→1→2; 4→3; 4→1→2→5.

Кратчайшие пути из вершины 5 таковы: 5→4→1; 5→4→1→2; 5→4→3; 5→4.

Напомним, что за длину пути в данном случае принимается длина его самой длинной дуги. Сравнивая найденные в данном примере пути с обычными кратчайшими путями из примера 3, убеждаемся, что во многих случаях пути получаются совершенно различными. Например, кратчайший путь из вершины 1 в вершину 2 в рассматриваемом минимаксном случае таков: 1→2, в то время как в традиционной постановке соединяющий те же две вершины путь с минимальной суммарной длиной таков: 1→5→4→3→2 (см. рисунок) ■

5. Задания

Задание 1а. Используя АД, найти кратчайшие пути между вершиной 0 и остальными вершинами в графе рис.5, когда длиной пути считается сумма длин его рёбер. См. пример 1 для образца. В приведённых ниже наборах длин рёбер порядок чисел соответствует номерам рёбер на рис.5.

Варианты наборов длин рёбер таковы:

01) 1,5,8,9,6,7,5,1,4
02) 2,8,1,5,1,9,4,6,3
03) 1,4,7,5,4,8,3,5,8
04) 2,3,1,8,5,9,5,1,7
05) 4,3,1,2,1,7,2,9,3
06) 1,8,6,1,8,9,7,4,2
07) 5,2,1,8,3,9,2,4,1
08) 1,9,7,4,6,1,9,5,2
09) 4,1,2,9,3,8,1,5,4
10) 2,4,7,9,8,1,6,2,7
11) 1,2,8,9,3,2,1,4,3
12) 7,2,5,9,2,8,1,1,4
13) 8,5,4,9,1,5,7,4,1
14) 4,2,3,1,4,1,9,8,5
15) 3,6,4,5,8,1,1,7,3■

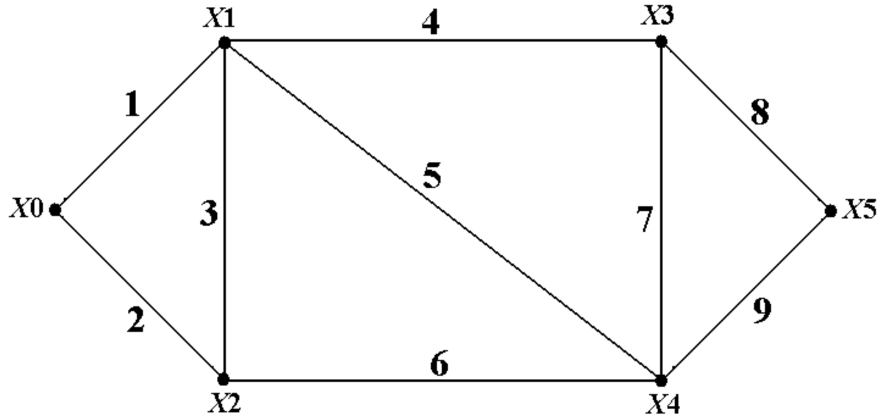


Рис.5

Задание 1б. Используя МАД, найти кратчайшие пути между вершиной 0 и остальными вершинами в графе рис.5, когда длиной пути считается считается длина его максимального ребра. См. пример 5 для образца.

Варианты наборов длин рёбер и их нумерация такие же, как в задании 1а■

Задание 2а. Используя АД, найти кратчайшие пути между вершиной 0 и остальными вершинами в графе рис.6, когда длиной пути считается сумма длин его рёбер. См. пример 2 для образца. В приведённых ниже наборах длин рёбер порядок чисел соответствует номерам рёбер на рис.6.

Варианты наборов длин рёбер таковы:

01) 1,8,6,5,4,8,5,9,6,1,7,4,3
02) 8,3,8,9,1,4,1,1,2,3,8,1,9
03) 4,5,1,3,2,1,9,4,1,5,4,2,3
04) 1,4,4,9,1,2,1,8,3,1,3,2,9
05) 8,1,8,4,1,7,1,2,1,1,8,3,4
06) 8,1,3,5,9,7,1,2,1,1,4,3,4
07) 5,8,5,2,1,7,4,3,8,3,8,9,1
08) 4,1,1,2,3,8,1,9,4,5,1,3,2
09) 1,2,4,1,2,8,9,9,5,7,3,1,7
10) 1,8,1,4,2,1,4,2,4,3,9,7,9
11) 8,1,7,1,1,6,1,7,2,7,8,5,4
12) 3,4,5,1,9,9,9,8,3,6,3,8,1
13) 5,4,1,6,7,9,8,9,7,4,5,1,8
14) 1,2,8,5,3,6,1,9,7,1,1,6,1
15) 9,2,7,2,5,3,4,5,5,8,4,3,6
16) 1,3,1,9,2,6,4,8,5,2,7,5,4■

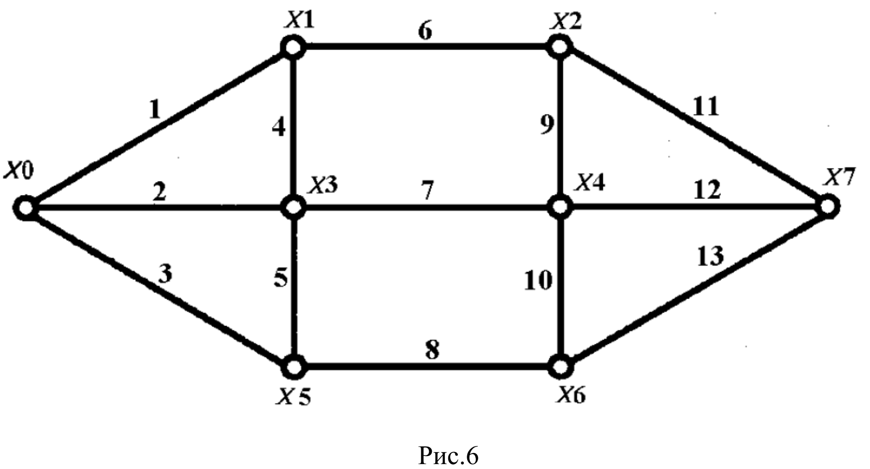
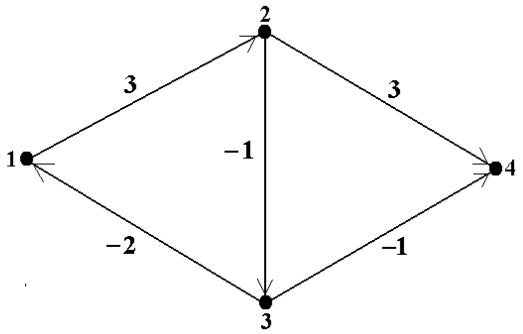


Рис.6

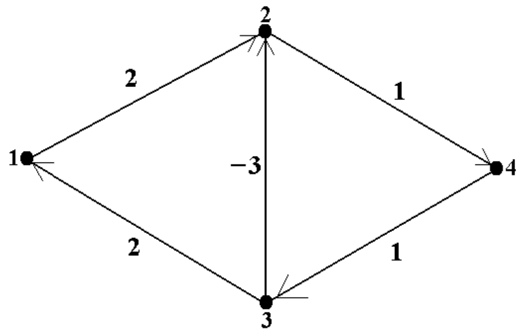
Задание 2б. Используя МАД, найти кратчайшие пути между вершиной 0 и остальными вершинами в графе рис.6, когда длиной пути считается считается длина его максимального ребра. См. пример 5 для образца.

Варианты наборов длин рёбер и их нумерация такие же, как в задании 1а■

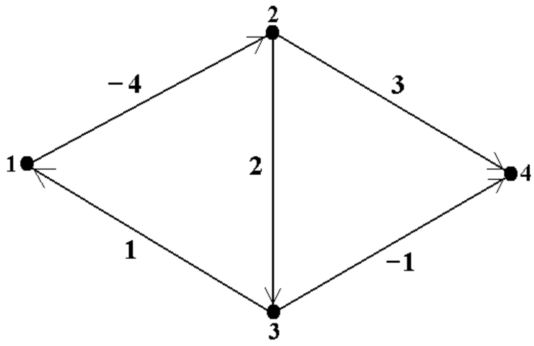
Задание 3. Используя АФУ, в заданном ориентированном графе найти кратчайшие пути между всеми парами вершин, если циклы отрицательной длины не существуют, или найти такой цикл. Для образца см. примеры 3 и 4.



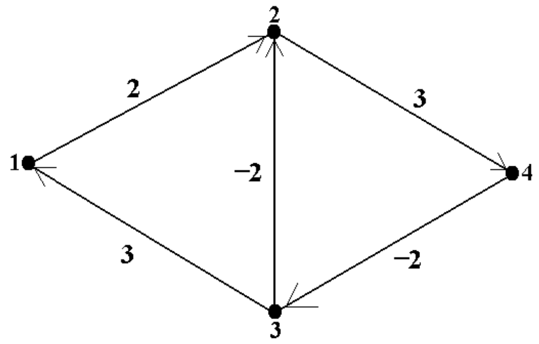
01



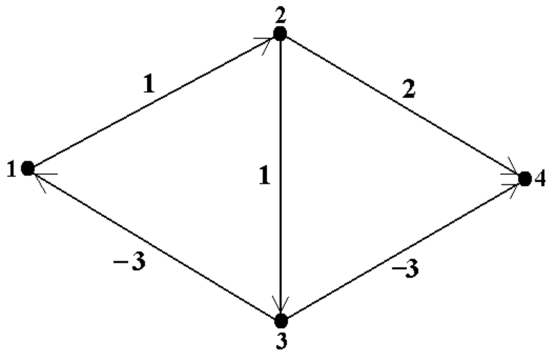
02



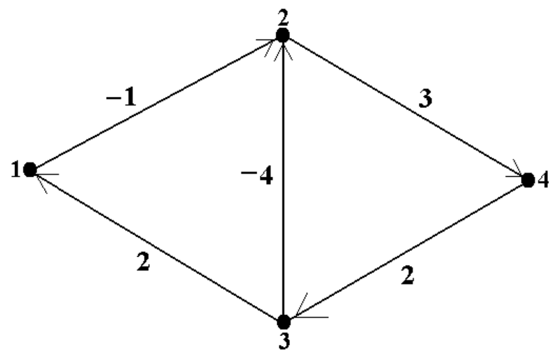
03



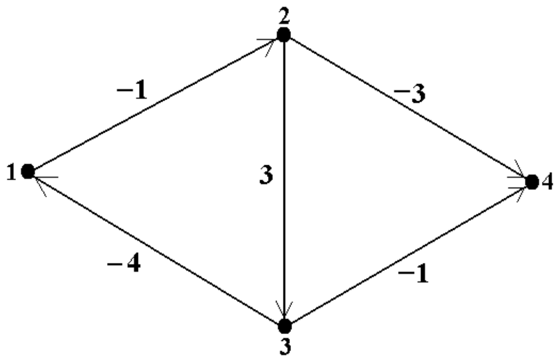
04



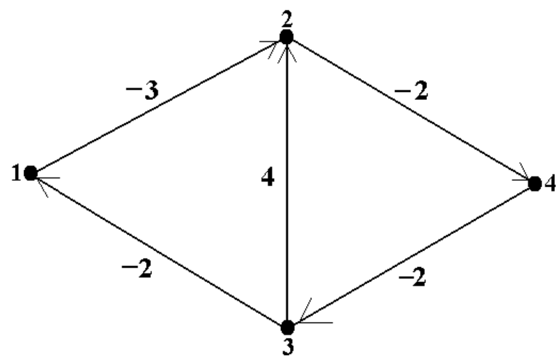
05



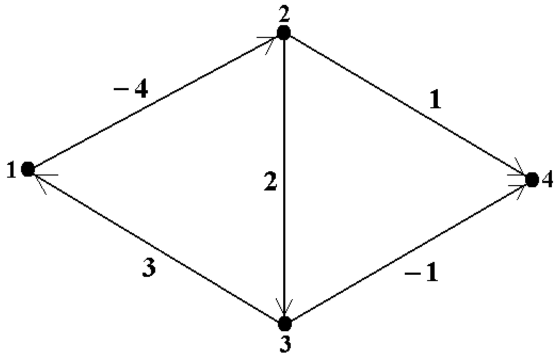
06



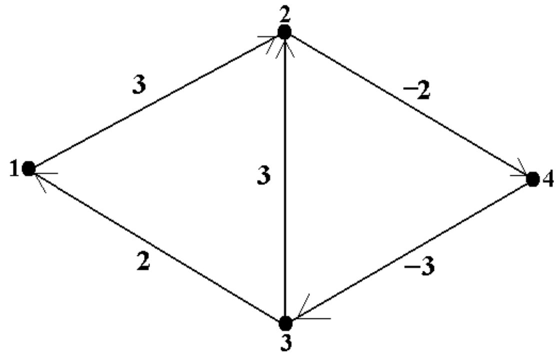
07



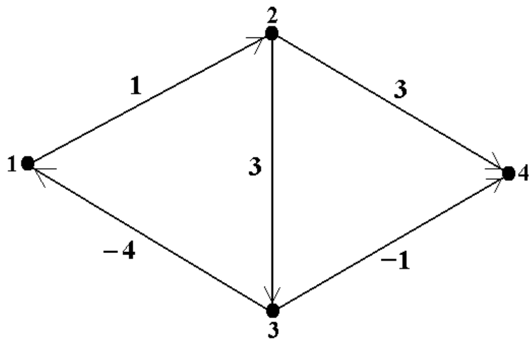
08



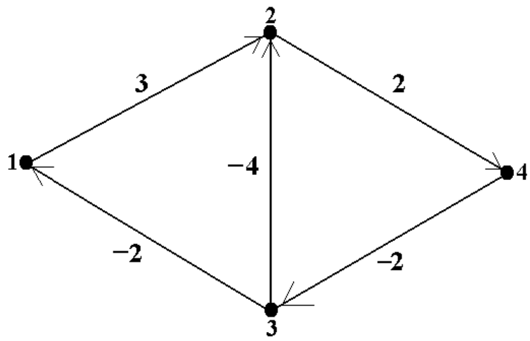
09



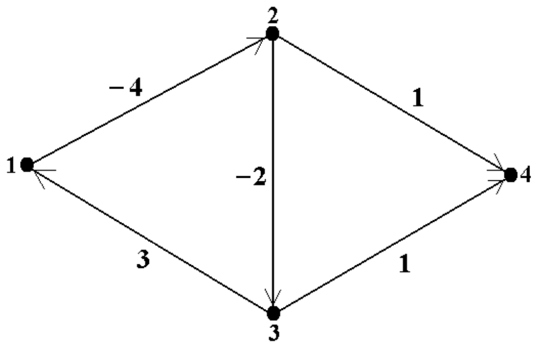
10



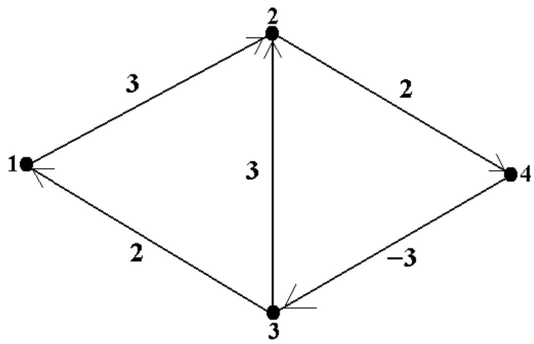
11



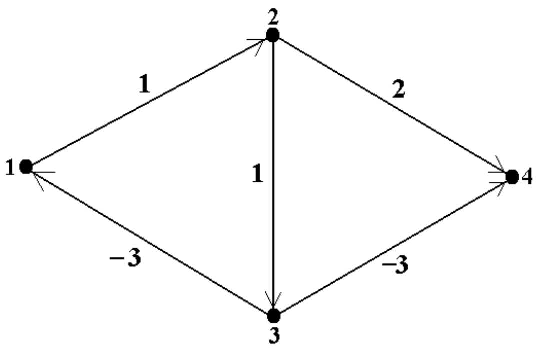
12



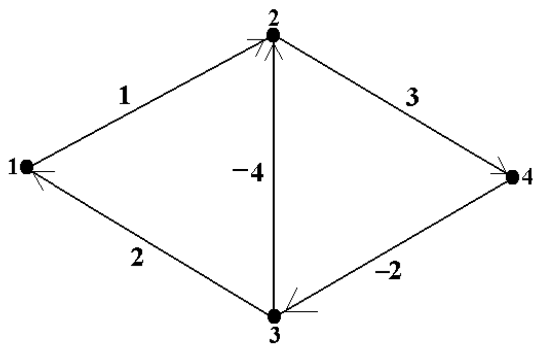
13



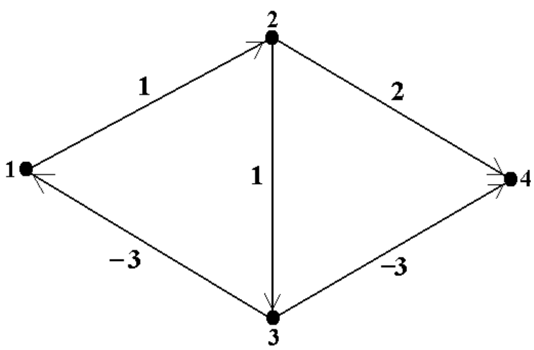
14



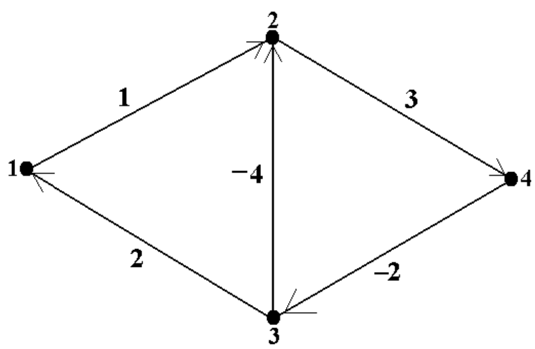
15



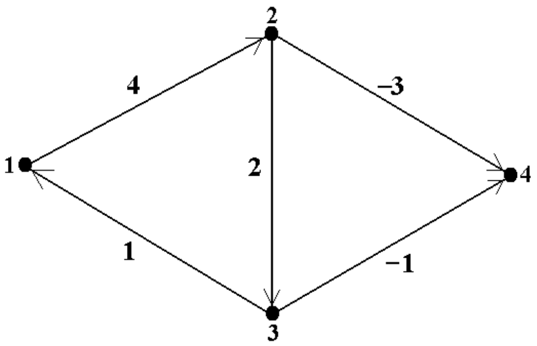
16



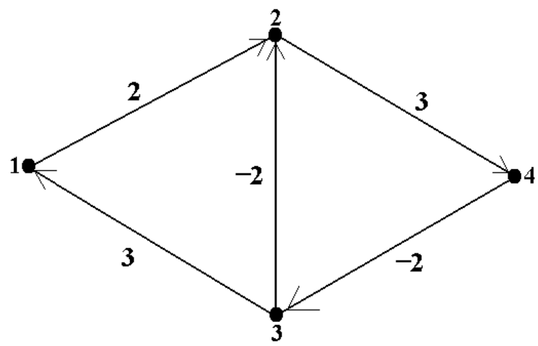
17



18



19



20

■

Задание 4. Используя АФУ, в графе на рис.7 найти кратчайшие пути между всеми парами вершин. Для образца см. пример 3. В приведённых ниже наборах длин дуг порядок чисел соответствует номерам на рис.7.

Варианты наборов длин дуг таковы:

01) 1,8,6,5,4,8,5,9,3	09) 2,1,1,4,3,4,5,8,5	17) 7,2,7,8,5,4,3,4,5
02) 6,1,7,4,3 8,3,8,9	10) 2,1,7,4,3,8,3,8,9	18) 1,9,9,9,8,3,6,3,8,
03) 4,1,1,2,3,8,1,9,4	11) 4,1,1,2,3,8,1,9,4	19) 1,5,4,1,6,7,9,8,9
04) 5,1,3,2,1,9,4,1,5	12) 1,1,2,3,8,1,9,4,5	20) 7,4,5,1,8,1,2,8,5
05) 4,2,3,1,4,4,9,1,2	13) 1,3,2,1,2,4,1,2,8	21) 3,6,1,9,7,1,1,6,1
06) 1,8,3,1,3,2,9,8,1	14) 9,9,5,7,3,1,7,1,8	22) 9,2,7,2,5,3,4,5,5
07) 8,4,1,7,1,2,1,1,8,	15) 1,4,2,1,4,2,4,3,9,	23) 8,4,3,6,1,3,1,9,2
08) 3,4,8,1,3,5,9,7,1,	16) 7,9,8,1,7,1,1,6,1	24) 6,4,8,5,2,7,5,4,1

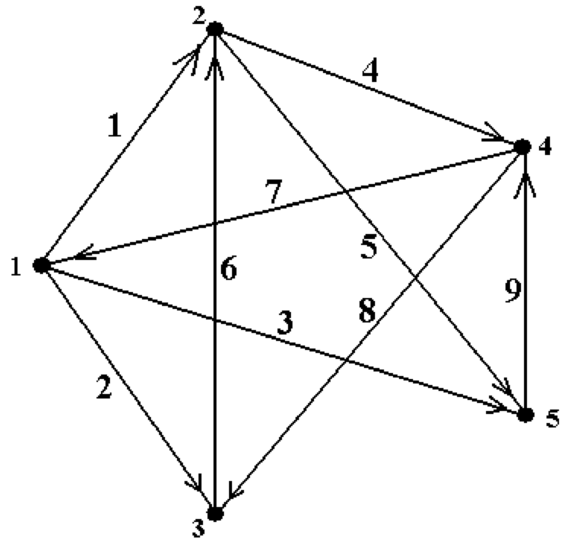


Рис.7

6. Предметный указатель

Глава 9. Паросочетания

В некоторых случаях вершины графа распадаются на две части, так что все имеющиеся рёбра соединяют только вершины из разных частей. Напомним, что соответствующий граф называется двудольным, а любое множество рёбер двудольного графа, которые не имеют общих вершин, называется паросочетанием (см. раздел 6-6). Многие практически важные задачи сводятся к поиску паросочетаний, обладающих теми или иными свойствами. Две оптимизационных задачи поиска паросочетаний рассматриваются в данной главе; ещё одна – но уже не оптимизационная – задача поиска паросочетаний, удовлетворяющих некоторым специальным условиям, рассмотрена в разделе 13-1 пособия.

1. Максимальные паросочетания

Максимальным называется паросочетание, содержащее максимально возможное число рёбер. Известно несколько различных методов поиска максимального паросочетания в заданном двудольном графе. Здесь будет рассмотрен потоковый метод построения максимального паросочетания. Идея метода такова.

1. По исходному двудольному графу строится потоковая сеть (см. главу 7). Для этого добавляется две вершины, которых не было в исходном графе – источник s и сток t. В каждую вершину 1-ой доли входит одна новая дуга, выходящая из источника. Из каждой вершины 2-ой доли выходит одна новая дуга, входящая в сток. Все «старые» рёбра графа ориентируются в направлении от 1-ой доли ко 2-ой доле (напомним, что в исходном графе любое ребро соединяло две разные доли). Положим пропускные способности всех дуг построенной сети равными 1. Таким образом, потоковая сеть полностью определена.

2. Найдём алгоритмом Форда-Фалкерсона (АФФ) максимальный поток в построенной сети. В силу утверждения 7-7 все компоненты потока – целые числа. А поскольку пропускные способности всех дуг равны 1, то отсюда следует, что поток в любой дуге равен либо 0, либо 1.

3. Рассмотрим все дуги, выходящие из 1-ой доли и входящие во 2-ую долю, в которых потоки равны 1. Если хотя бы две таких дуги выходят из одной и той же вершины, то в силу «закона сохранения» (сумма потоков во всех входящих в вершину дугах равна сумме потоков во всех выходящих из вершины дугах) сумма потоков во входящих дугах не меньше 2. Но в любую вершину 1-ой доли по построению входит только одна дуга, поток в которой не может быть больше 1. Полученное противоречие доказывает, что все дуги, поток в которых равен 1, выходят из разных вершин. Совершенно аналогично доказывается, что все дуги, поток в которых равен 1, входят в разные вершины.

4. Поскольку все рассмотренные дуги, потоки в которых равны 1, не имеют общих концов, то соответствующие им в исходном двудольном графе рёбра образуют паросочетание. Легко понять, что из максимальности потока следует максимальность построенного по нему указанным образом паросочетания. Действительно, если найденное паросочетание не является максимальным, то существует паросочетание с бо́льшим числом рёбер. По этому новому паросочетанию построим поток, величина которого будет равна числу рёбер в нём. Но это значит, что его величина больше величины максимального потока, что невозможно. Это и доказывает максимальность построенного ранее паросочетания.

Пример 1. Проиллюстрируем введённые понятия и рассуждения. Рассмотрим двудольный граф, показанный на рис.6-32. Этот же граф показан здесь на рис.1. Построенная по нему описанным выше образом потоковая сеть S показана на рис.2. Найденный с помощью АФФ максимальный поток в сети S условно показан на рис.3. Жирным выделены все дуги, потоки в которых равны 1; потоки во всех остальных дугах равны 0. Таким образом, максимальное паросочетание в исходном двудольном графе на рис.1 состоит из следующих рёбер: (1, 8), (3, 9), (4, 7), (5, 10), (6, 11). Число рёбер в максимальном паросочетании равно 5. Без сложных рассужде-ний легко понять, что это число не может превосходить 5, поскольку 1-ая доля содержит только

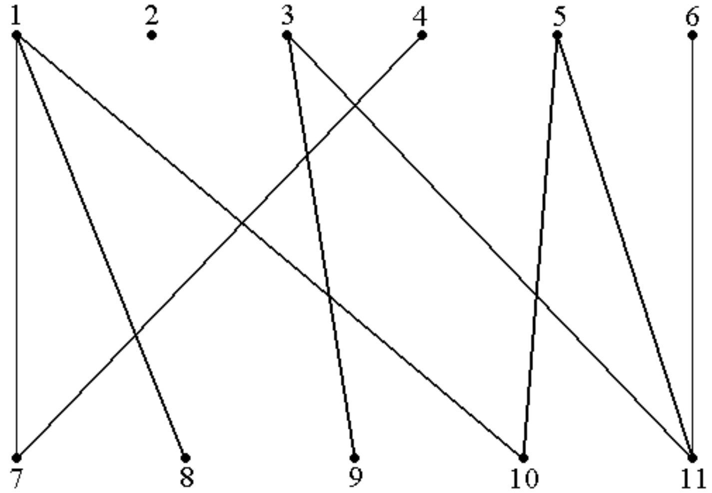


Рис.1. Исходный двудольный граф

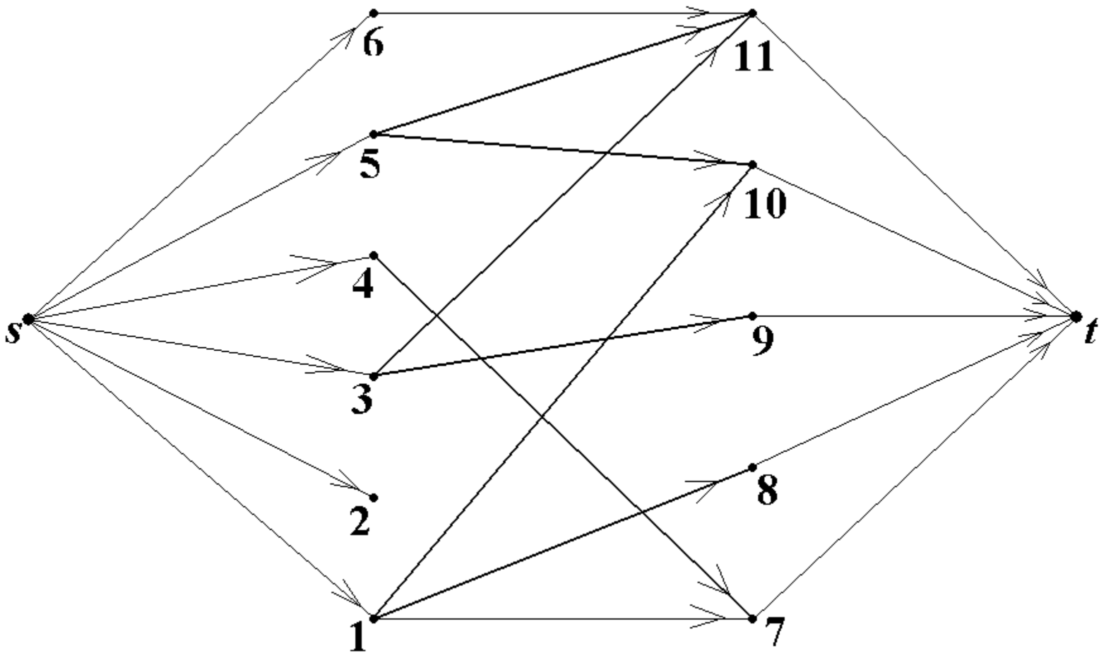


Рис.2. Сеть, построенная по графу

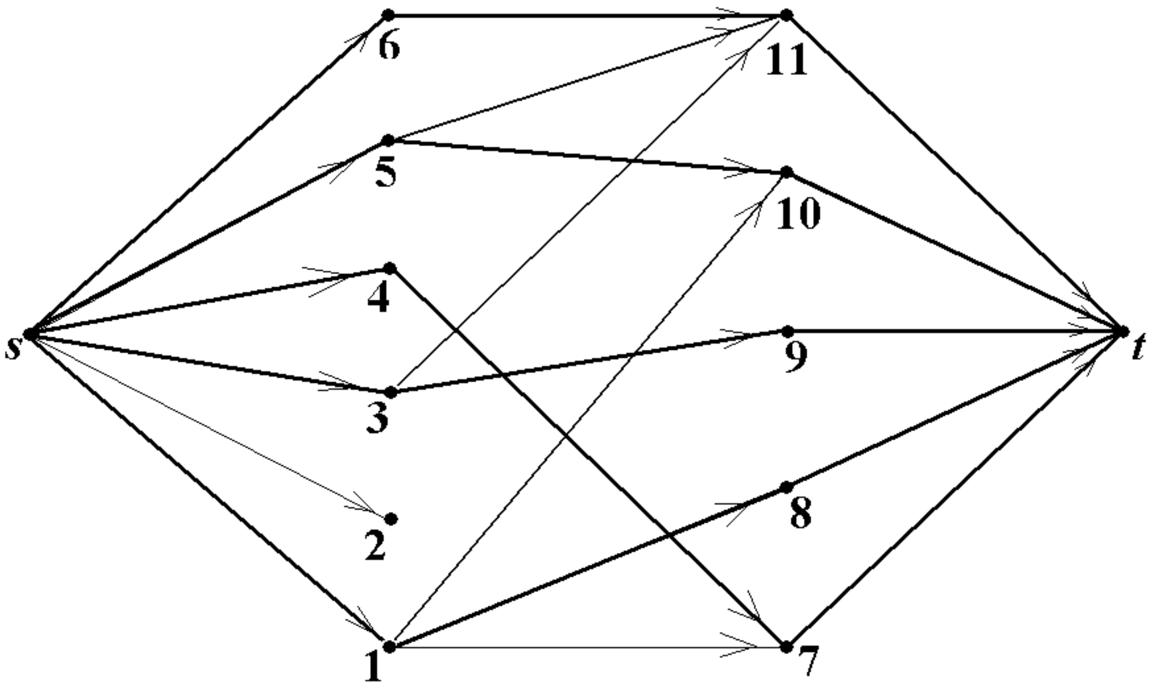


Рис.3. Максимальный поток в сети

5 вершин, которые могут быть соединены с вершинами из 2-ой доли. Однако даже в этом простом случае не так просто «увидеть глазами» паросочетание, содержащее 5 рёбер ■

Таким образом, описанный подход позволяет решить простейшую из рассматриваемых в курсе задач на паросочетания – задачу поиска максимального паросочетания.

■

2. Задача назначения

Естественным обобщением рассмотренной в разделе 1 задачи поиска максимального паросочетания является задача поиска ***максимального взвешенного паросочетания***. Суть дела в том, что каждому ребру заданного двудольного графа сопоставлено некоторая положительное число – ***вес ребра***. Рассматриваемая задача состоит в поиске паросочетания с максимальным суммарным весом. Легко понять, что рассмотренная в разделе 1 задача является частным случаем последней задачи, когда веса всех рёбер равны 1.

Во многих случаях вершины 1-ой доли интерпретируются как исполнители, вершины 2-ой доли – как работы, а наличие соединяющего их ребра с весом a – как возможность выполнения данной работы данным исполнителем с эффективностью (доходом) a. При такой интерпретации задача поиска максимального взвешенного паросочетания представляет собой поиск такого распределения задач по исполнителям, при котором суммарная эффективность будет максимальной. Именно поэтому рассматриваемая задача чаще называется ***задачей назначения*** (работ исполнителям). Настоящий раздел и посвящён этой задаче.

Прежде чем двигаться дальше, сформулируем утверждение, связывающее циклы и паросочетания в произвольных двудольных графах.

Пусть $G(V, E)$ – двудольный граф, $V = X \cup Y$, где X и Y – множества вершин первой и второй доли графа G, C – произвольный цикл в графе G. Нетрудно понять, что число рёбер в любом цикле двудольного графа чётно. Обозначим через C_1 множество рёбер цикла, взятых через одно, а через C_2 – множество остальных рёбер этого цикла (также взятых через одно). Пусть P – произвольное паросочетание в графе G. Определим множество рёбер P' следующим образом:

$$P' = (P \setminus C_1) \cup C_2. \qquad (1)$$

Имеет место

Утверждение 1. Для любого паросочетания P и любого цикла C множество P' также является паросочетанием.

Утверждение 1 почти очевидно. Действительно, при удалении ребра из цикла и добавлении следующего ребра число инцидентных их общей вершине рёбер останется равным 1. Поэтому новое множество рёбер останется паросочетанием ■

Каждому ребру (v,w) графа G сопоставлен неотрицательный вес $c(v,w) \geq 0$. Задачей о ***максимальном взвешенном паросочетании*** в двудольном графе G называется задача нахождения в нём паросочетания, суммарный вес рёбер которого максимален.

Алгоритм решения этой задачи, как и большинства других задач оптимизации на графах, состоит в последовательном переходе от одних паросочетаний к другим, с бóльшим суммарным весом (т.е. к лучшим с рассматриваемой точки зрения), до тех пор, пока не будет достигнуто решение. Ниже будет описан один этап алгоритма – переход от произвольного паросочетания к следующему.

Алгоритм 1 улучшения заданного паросочетания. Исходное паросочетание P состоит из рёбер (x_1, y_1), (x_2, y_2), …, (x_n, y_n).

1. По заданному двудольному графу G и паросочетанию P построим ориентированный взвешенный граф G^* следующим образом:

- множество вершин графа G^* совпадает с множеством вершин $X \cup Y$ графа G;
- каждое ребро $(x_k, y_k) \in P$ определяет дугу (x_k, y_k) графа G^* с тем же весом $c(x_k, y_k)$, каким обладает ребро (x_k, y_k) в исходном графе;
- каждое ребро $(x_k, y_k) \notin P$ определяет дугу (y_k, x_k) графа G^* с весом $-\,c(x_k, y_k)$.

2. С помощью алгоритма Флойда-Уоршолла находим в графе G^* орцикл отрицательной стоимости (если таковой существует). Отсутствие таких орциклов в графе G^* означает, что исходное взвешенное паросочетание P максимально. Алгоритм улучшения прекращает работу.

3. Разобьём множество дуг найденного орцикла C на два подмножества C_1 и C_2, где все дуги, принадлежащие C_1, ведут из X в Y, а все дуги, принадлежащие C_2, ведут из Y в X. Более того, если перейти от дуг (отменой ориентации) обратно к рёбрам G, то, по построению, все рёбра из C_1 содержатся в исходном паросочетании P, а все рёбра из C_2 не содержатся в нём. Также по построению, веса всех дуг из C_1 совпадают с исходными весами дуг в графе G, а веса всех дуг из C_2 равны исходным весам со знаком −. Поскольку сумма весов во всех дугах цикла отрицательна, то это означает, что в исходном графе сумма весов у всех дуг, входящих в множество C_2, <u>строго больше</u>, чем сумма весов во всех дугах, входящих в множество C_1.

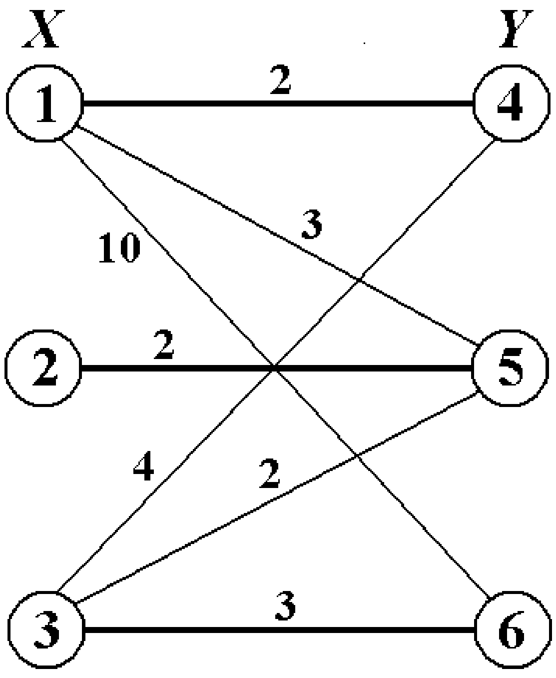
4. Рассмотрим цикл C в исходном графе G, и два его подмножества чередующихся рёбер C_1 и C_2, которые были определены на шаге 3. Определим по исходному паросочетанию P и множествам рёбер C_1 и C_2 новое паросочетание P' формулой (1):

$P' = (P \setminus C_1) \cup C_2$.

На шаге 3 было отмечено, что все рёбра из C_1 входят в P, а все рёбра из C_2 не входят в P. Так как суммарный вес всех дуг, входящих в C_2, по построению строго больше, чем суммарный вес всех дуг, входящих в C_1, то <u>ровно настолько же</u> вес нового паросочетания P' больше, чем вес исходного паросочетания P. Таким образом, новое паросочетание имеет бóльший вес, чем исходное ■

Алгоритм построения максимального взвешенного паросочетания сводится к двум этапам. На 1-ом этапе строится начальное паросочетание (например, потоковым алгоритмом из раздела 1). 2-ой этап состоит в многократном применении только что описанного алгоритма 1 – до тех пор, пока на шаге 2 этого алгоритма будет установлено отсутствие циклов отрицательной длины.

Пример 2. Проиллюстрируем алгоритм построения максимального взвешенного паросочетания. Исходный взвешенный двудольный граф G показан на рис.4.1. Веса рёбер указаны рядом с ними; номера вершин указаны в кружках. Рёбра начального паросочетания P = {(1,4), (2,5), (3,6)} на рис.4а выделены. Вес выделенного паросочетания равен 7.



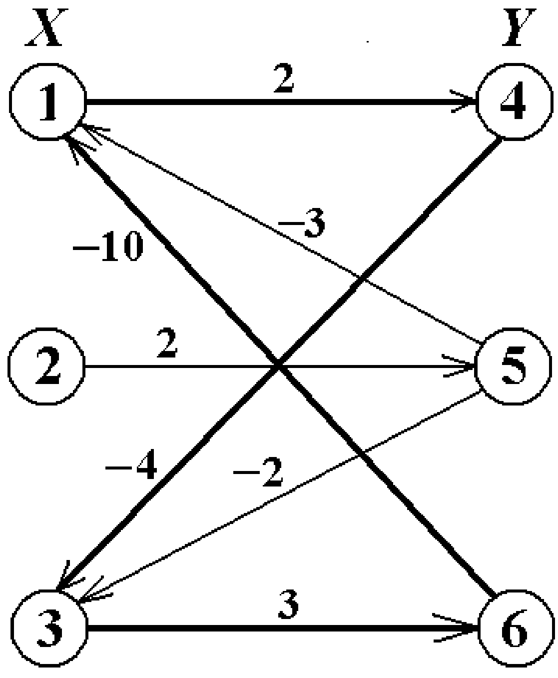
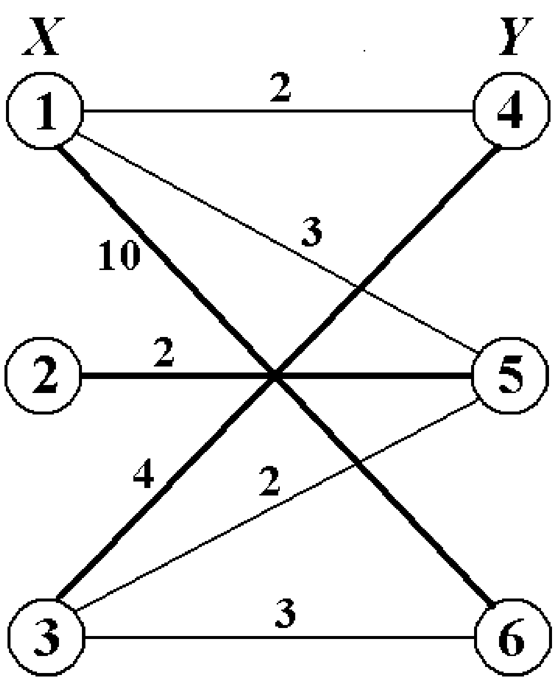


Рис.4a Рис.4b

На рис.4b показан построенный по исходному паросочетанию P рграф G^*. В соответствии с алгоритмом 1 все рёбра исходного паросочетания превращаются в дуги с положительными весами, ориентированные от доли X к доле Y, а все остальные рёбра превращаются в дуги с отрицательными весами, ориентированные от доли Y к доле X. В графе G^* существует орцикл $1\to4\to3\to6\to1$ с отрицательным весом $2 - 4 + 3 - 10 = - 9$. В этом орцикле $C_1 = \{(1,4), (3,6)\}$, $C_2 = \{(6,1), (4,3)\}$. Возвращаясь к исходному графу и отменяя ориентации, получаем $C_2 = \{(1,6), (3, 4)\}$.

Далее, в соответствии с формулой (1), получим новое паросочетание $P' = (P\setminus C_1) \cup C_2 = (\{(1,4), (2,5), (3,6)\} \setminus \{(1,4), (3,6)\}) \cup \{(1,6), (3,4)\} = \{(2,5), (1,6), (3,4)\}$ (см. определения и алгоритмы выполнения теоретико-множественных операций в разделе 2-3). Это новое паросочетание выделено на рис.4c. Строя по нему орграф G^*, как указано на шаге 1 алгоритма 1, приходим к орграфу, показанному на рис.4d.



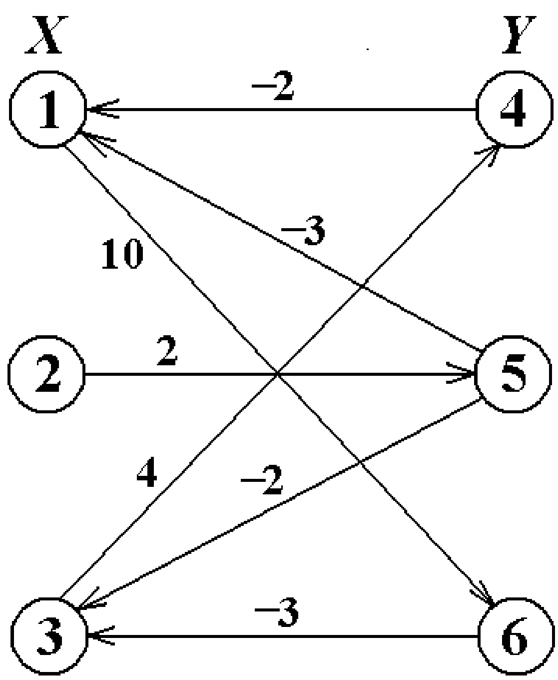
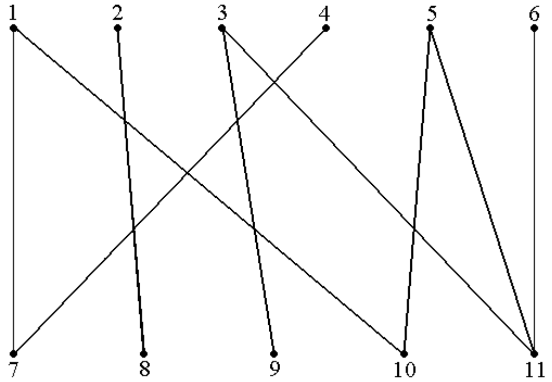


Рис.4c Рис.4d

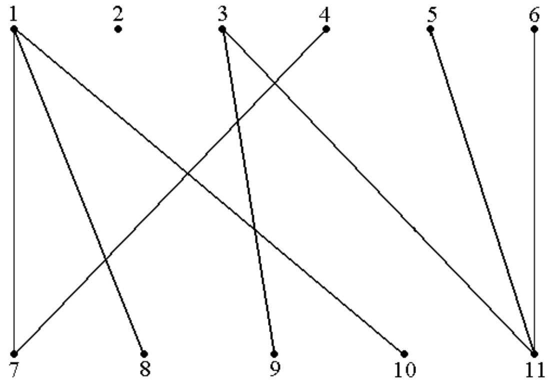
Можно непосредственно убедиться (не прибегая к помощи алгоритма Флойд-Уоршолла), что в последнем орграфе орциклы отрицательной длины отсутствуют. Поэтому последнее паросочетание, показанное на рис.4с, будет максимальным. Его вес равен 16. Оно превосходит вес 7 предыдущего паросочетания на 9 – в соответствии с теорией, как раз настолько, насколько сумма весов рёбер из множества C_2 = {(1,6), (3,4)}, равная 14, больше суммы весов рёбер из множества C_1 = {(1,4), (3,6)}, равной 5 ■

3. Задания

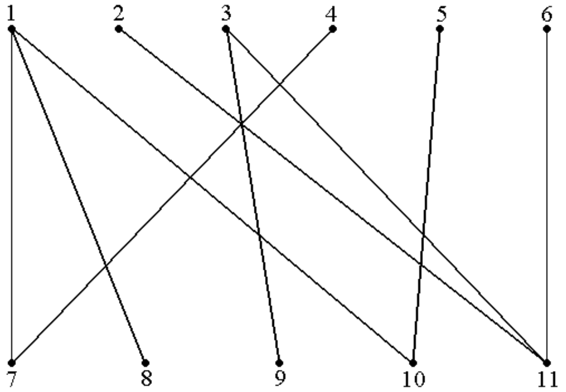
Задание 1. Представить заданный двудольный граф в виде потоковой сети; найти с помощью АФФ максимальный поток в полученной сети; поток представить графически, как на рис.3, и указать максимальное паросочетание в виде перечня его рёбер. Для образца использовать пример 1 и примеры 7-7 – 7-11.



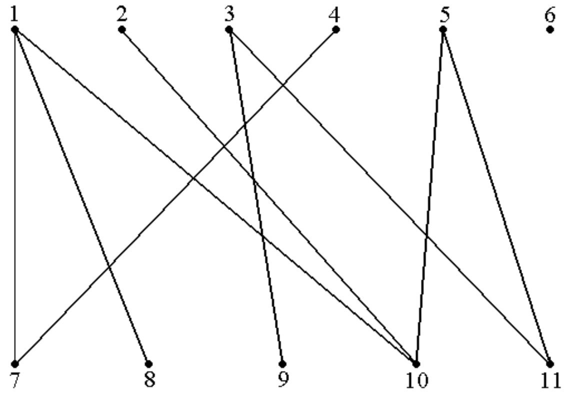
01



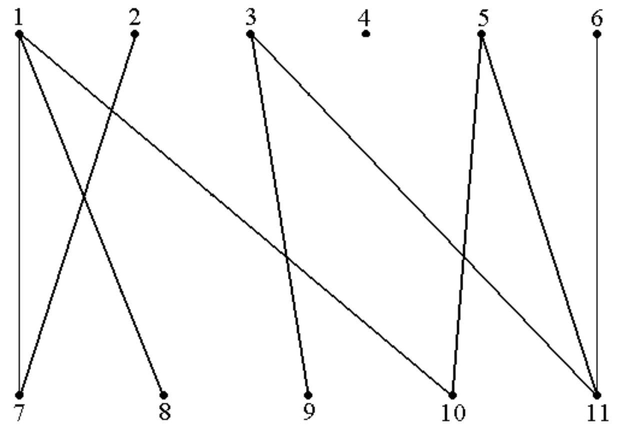
02



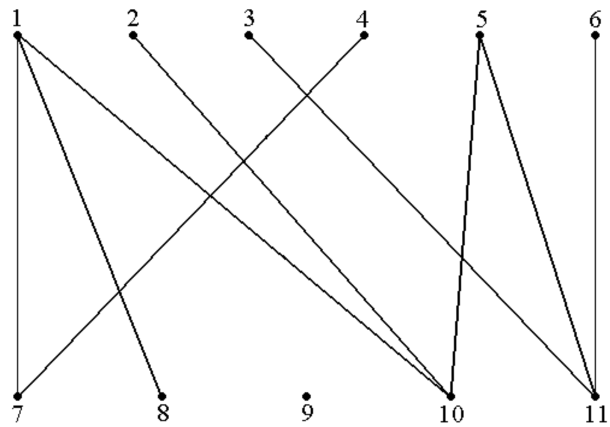
03



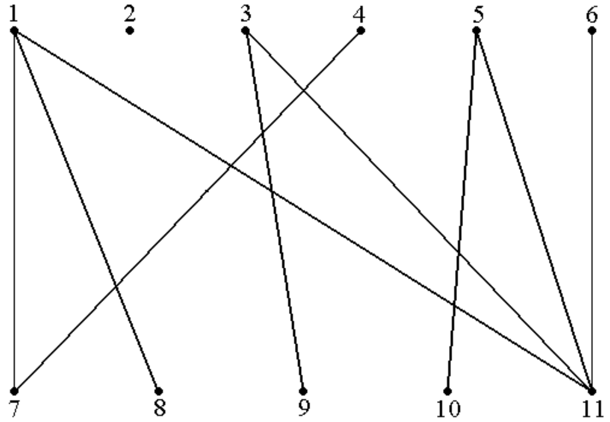
04



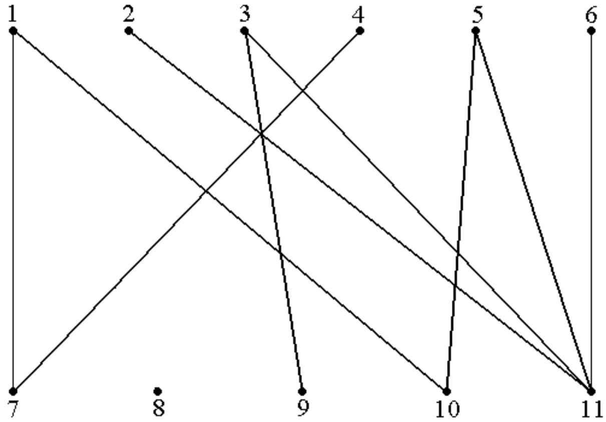
05



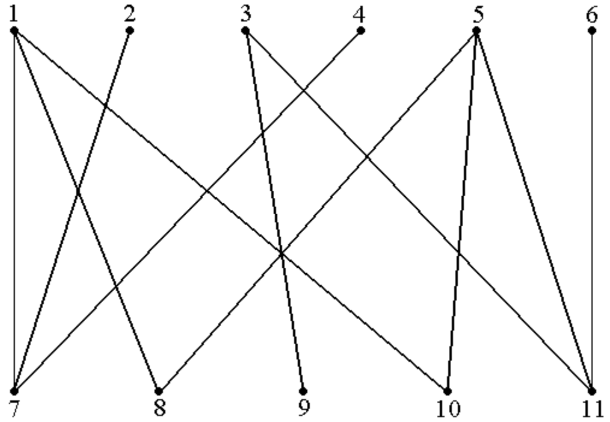
06



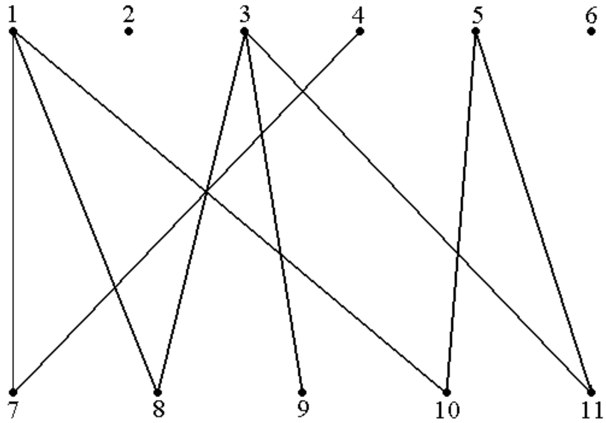
07



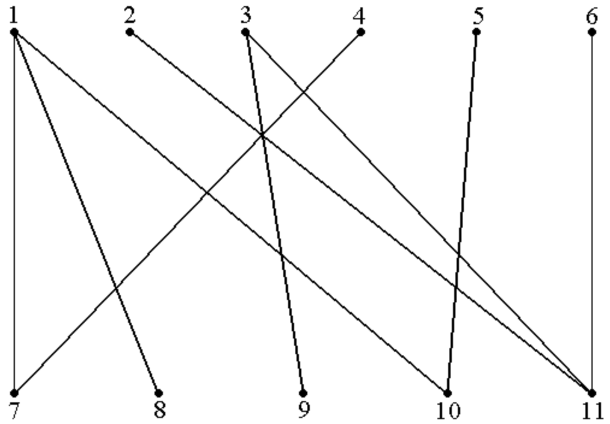
08



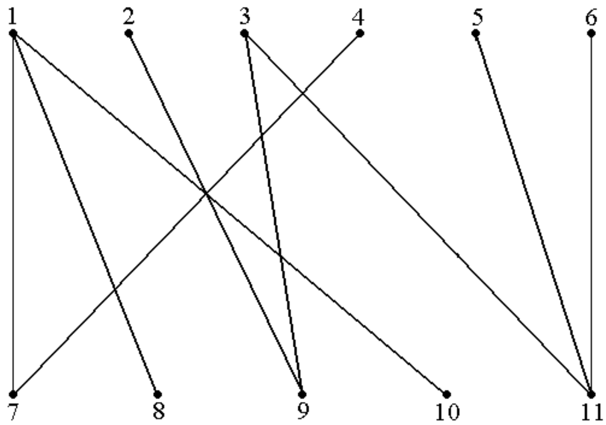
09



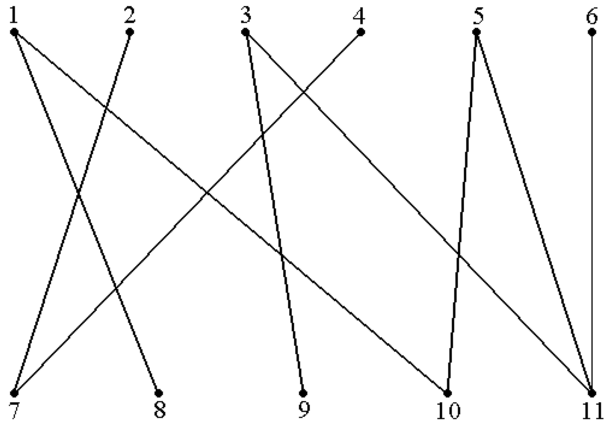
10



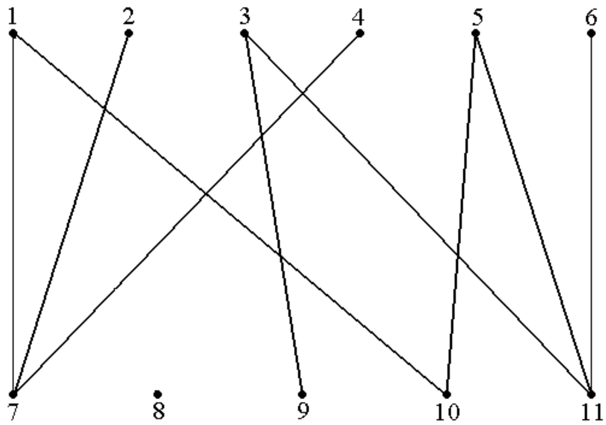
11



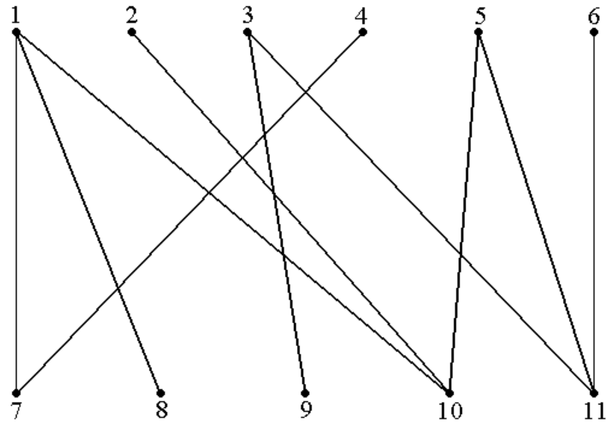
12



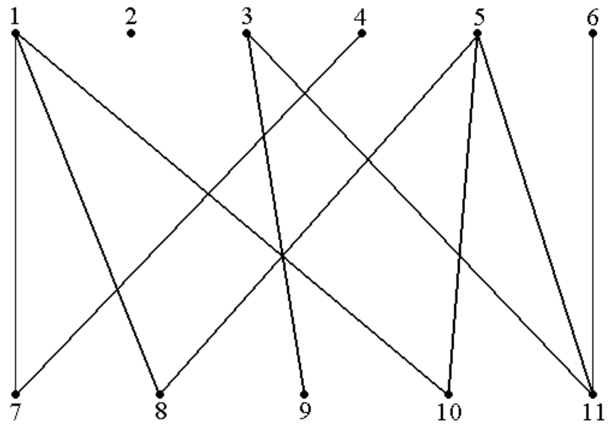
13



14



15



16 ■

Задание 2. Во взвешенном двудольном графе рис.5 найти максимальное взвешенное паросочетание. В качестве образца использовать пример 2 и алгоритм 1 улучшения. Поиск цикла отрицательной длины можно делать «глазами», но все остальные операции должны осуществляться формально.

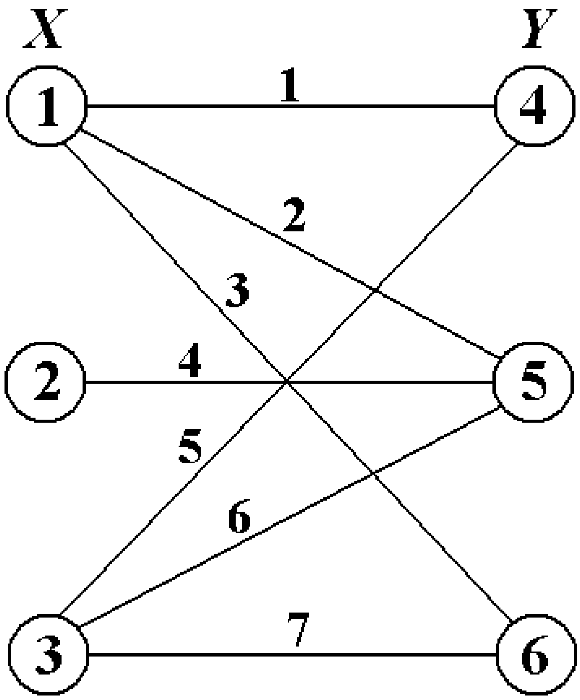


Рис.5

Варианты наборов весов таковы:

01)9,2,2,4,3,3,2	07) 6,8,5,3,2,2,7	13) 2,4,2,2,3,3,5
02) 1,3,6,2,5,9,4	08) 2,3,7,4,4,6,5	14) 2,3,5,2,4,6,3
03) 2,5,1,3,9,7,4	09) 2,3,8,2,6,6,3	15) 2,5,7,6,4,2,3
04) 1,6,2,2,7,1,8	10) 2,9,1,6,5,3,4	16) 7,3,6,2,8,2,5
05) 2,2,3,8,4,5,3	11) 4,3,8,4,7,2,3	17) 5,5,4,6,4,2,3
06) 6,3,7,2,8,5,3	12) 2,3,5,9,4,6,3	18) 7,3,6,2,8,4,5

Порядок чисел соответствует номерам рёбер на рис.5 ■

4. Предметный указатель

Глава 10. Многошаговая оптимизация

1. Основные понятия и постановка задачи

Во многих случаях воздействие на управляемую систему является не однократным актом, а процессом, развивающимся во времени. Задачи оптимизации в подобных случаях обладают четырьмя характерными свойствами, которые, собственно говоря, и выделяют их из всего многообразия задач оптимизации. Свойства эти таковы:

1. Рассматриваемой системе естественно сопоставляется некоторое множество S её ***состояний***, так что функционирование системы фактически состоит в переходах из одних состояний в другие.

2. Для каждого состояния $s \in S$ определено множество Q_s возможных воздействий на рассматриваемую систему, находящуюся в данном состоянии s. Под воздействием ***частного управления*** $v \in Q_s$ система переходит из состояния s в новое состояние $s' \in S$, которое однозначно определяется «старым» состоянием s и частным управлением $v \in Q_s$ по формуле

$$s' = \varphi(s,v), \tag{1}$$

где $\varphi(s,v)$ – так называемая ***функция перехода***, определяющая переходы системы между состояниями.

3. При переходе системы из состояния s в состояние $s' = \varphi(s,v)$ под действием частного управления v доход от функционирования системы равен $\psi(s,v)$, где функция $\psi(s,v)$ называется ***функцией дохода***. Функция дохода, как и функция перехода, определена при всех $s \in S$ и при всех $v \in Q_s$ при заданном состоянии s. При $\psi(s,v) < 0$ доход превращается в убыток.

4. Управление $u = (v_1, v_2, \ldots, v_N)$ является последовательностью частных управлений$v_1, v_2, \ldots, v_N$, под действием которых система последовательно переходит из заданного ***начального состояния*** s_0 в состояния

$$s_1 = \varphi(s_0,v_1),\ s_2 = \varphi(s_1,v_2),\ \ldots,\ s_N = \varphi(s_{N-1},v_N). \tag{2}$$

При этом общий доход от управления $u = (v_1, v_2, \ldots, v_N)$ является суммой доходов от частных управлений:

$$\psi(u) = \psi(s_0,v_1) + \psi(s_1,v_2) + \ldots + \psi(s_{N-1},v_N), \tag{3}$$

где $s_1, s_2, \ldots, s_N$ определяются формулами (2).

Таким образом, на первый план выходит многошаговость всех процессов и аддитивность доходов по шагам, выражаемая формулой (3).

Основная задача многошаговой оптимизации состоит в максимизации дохода $\psi(u)$ при заданном начальном состоянии s_0 и числе шагов N. Эта задача является дискретной при конечности множества S состояний и множеств управлений Q_s при всех $s \in S$. Далее в главе будут рассматриваться только дискретные задачи многошаговой оптимизации.

Всякую дискретную задачу многошаговой оптимизации можно представить как задачу оптимизации на графе. Именно, определим орграф G следующим образом. Множеством вершин орграфа G является конечное множество состояний S. Рассмотрим любую вершину $s \in S$. Каждому управлению $v \in Q_s$, под действием которого система переходит из состояния s в состояние $s' = \varphi(s,v)$, сопоставим дугу графа G с началом в вершине s и концом в вершине s'. За вес этой дуги примем величину $\psi(s,v)$. При этом исходная задача многошаговой оптимизации перейдёт в задачу поиска ориентированного пути с максимальным суммарным весом при заданном начальном состоянии s_0 и числе шагов N. Именно последняя задача и рассматривается в настоящей главе.

В разделе 1 дано краткое описание и постановка задачи многошаговой оптимизации; в разделе 2 приведён общий метод решения общей задачи многошаговой оптимизации – динамическое программирование; в разделе 3 рассмотрены некоторые модификации общей задачи, которые проиллюстрированы развёрнутыми примерами важных классов задач: о замене машины и о критическом пути в сетевом графике.

2. Принцип оптимальности и метод динамического программирования

Дадим более подробное описание рассматриваемой общей задачи многошаговой оптимизации. Предполагается, что задан простой ориентированный взвешенный граф $G(V, A)$ с множеством вершин V и множеством дуг A. Предполагается также, что каждой дуге (x, y) сопоставлено число $w(x, y)$, интерпретируемое как длина, стоимость, вес данной дуги. Далее будет использоваться общий термин «вес». Рассматриваются все ориентированные пути ***длины*** N (т.е. состоящие из N последовательных дуг) с началом в заданной вершине x^0. Для каждого такого пути P определяется его вес $W(P)$ как сумма весов входящих в него дуг. Путь P называется ***оптимальным***, если для любого другого пути P' длины N с началом в вершине x^0 выполняется неравенство $W(P) \geq W(P')$. В тех случаях, когда необходимо явно указать на длину и начальную вершину оптимального пути, будет использоваться обозначение $P(x^0,N)$.

Очевидным образом определение оптимального пути переносится на случай минимизации (вместо максимизации) веса. Заметим, однако, что в рассмотренных в главе 8 задачах на кратчайшие пути не предполагалась, что число дуг в рассматриваемых путях фиксировано. Для существования решения там требовались дополнительные условия типа неотрицательности весов дуг или неотрицательности циклов. В данном случае при фиксированной длине пути множество путей всегда конечно и, следовательно, задача оптимизации (и на максимум, и на минимум) всегда имеет решение. Заметим также, что в общем случае оптимальный путь (далее для определённости будем рассматривать задачу максимизации) может не быть простым путём или цепью (см. определения в разделе 2.3). Это означает, что он может проходить несколько раз через одни и те же вершины и дуги.

Далее в этом разделе излагается хорошо известный метод поиска пути максимального веса при заданном начальном состоянии x^0 и заданной длине N. Этот метод называется методом ***динамического программирования***. Идея метода состоит в сведении задач длины N к семейству задач длины $N - 1$, и т.д., вплоть до непосредственно решаемых задач длины 1. Введём необходимые формальные обозначения.

Обозначим задачу поиска максимального пути длины r с начальной вершиной x через $Z(r, x)$, её решение (т.е. соответствующий максимальный путь) – через $P(x,r)$, а вес этого пути – через $W(x,r)$. Предположим, что для любой вершины x графа G задача оптимизации $Z(r, x)$ решена, то есть известны и пути $P(x,r)$, и их веса $W(x,r)$. Рассмотрим теперь фиксированную вершину x^0 и любой путь P длины $r+1$ с началом в вершине x^0. Следующей (т.е. второй) вершиной на пути P является некоторая вершина x. Вес $W(P)$ пути P равен сумме двух слагаемых: веса $w(x^0, x)$ первой дуги пути P и веса $W(P')$, где P' – оставшаяся часть пути P от вершины x до его конца, длина которого равна r, т.е.

$$W(P) = w(x^0, x) + W(P'). \tag{4}$$

Поскольку длина $W(x,r)$ оптимального пути из x длины r предполагается известной, то из (4) сразу следует, что

$$W(P) \leq w(x^0, x) + W(x,r), \tag{5}$$

а поскольку вершина x была выбрана произвольно, то из (5) следует, что для веса оптимального пути с начальной вершиной x^0 и длиной $r+1$ выполняется равенство

$$W(x^0, r+1) = \max_x \{w(x^0, x) + W(x, r)\}. \tag{6}$$

Формула (6) является, по сути, формулой перехода от размерности r к бóльшей размерности $r+1$. Конечно, она позволяет найти не только оптимальные веса путей, но и сами эти пути, добавляя к началу известных путей длины r дугу (x^0, x), где x максимизирует правую часть (6). Наконец, при $r = 1$ оптимальные пути $P(x,1)$ ищутся непосредственно из условия максимизации веса одной дуги с началом в вершине x:

$$W(x,1) = \max_y w(x, y). \tag{7}$$

Как и ранее, рассматриваемые графы будем предполагать полными, а веса реально отсутствующих дуг будем считать равными $-\infty$ (в задачах минимизации $+\infty$).

Метод, основанный на проведённых простых рассуждениях, называется методом динамического программирования, а основное соотношение (6) называется уравнением Беллмана, по имени американского математика, первым сформулировавшим этот метод именно как общий метод решения многошаговых оптимизационных задач (не только дискретных) и решившим на его основе многие важные прикладные задачи.

Прежде чем идти дальше, приведём грубую оценку выигрыша, который может дать динамическое программирование. Число вершин в орграфе G обозначим через m. Для простоты предположим, что орграф является полным, т.е. из каждой вершины выходят дуги во все вершины, включая её саму (т.е. имеются петли при вершинах). Общее число путей длины N равно, очевидно, m^N. В то же время число операций на каждом переходе к следующей размерности имеет порядок m^2, а общее число операций при N шагах имеет порядок Nm^2. При весьма скромных значениях $m = 20$ и $N = 10$ для числа операций при методе динамического программирования имеем оценку 10*400 = 4000, в то время как при прямом переборе нужно проверить $20^{10} \approx 10^{16}$ вариантов.

Как и в других случаях, для ручной реализации метода Беллмана целесообразно использовать специальные таблицы, которые будут последовательно заполняться. Для упрощения изложения такие таблицы будут рассмотрены в следующем примере.

Пример 1. Рассмотрим орграф, показанный на рис.8-3. Здесь он перерисован как рис.1. В качестве начальной вершины x^0 возьмём вершину 1. Положим длину искомого максимального пути $N = 6$.

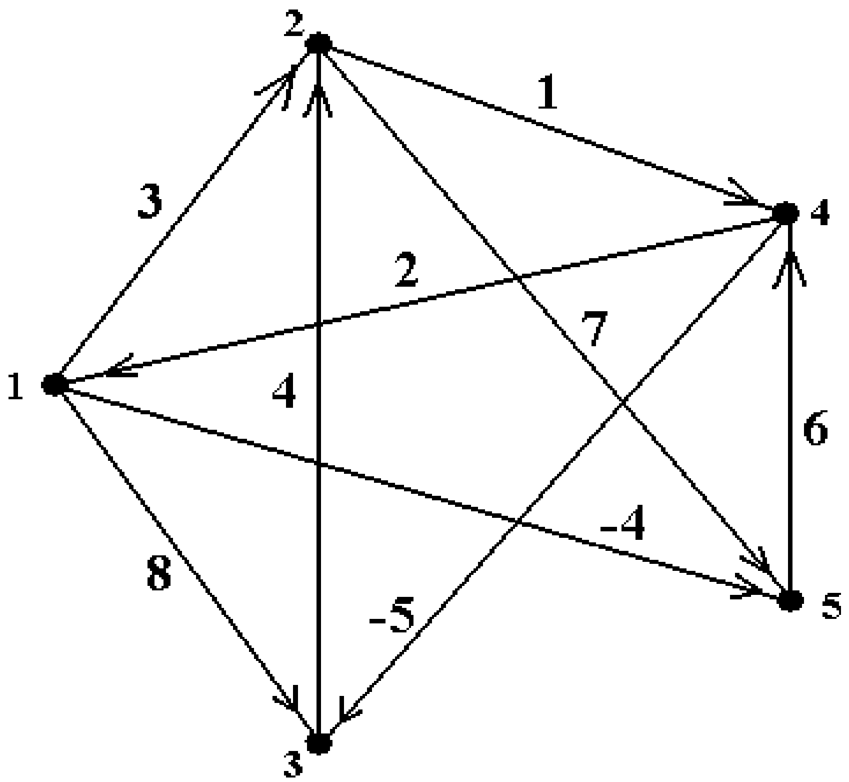


Рис.1

1. <u>Инициализация</u> и <u>1-ая итерация</u>. Сформируем таблицу, состоящую из 5-и строк (по числу вершин) и 12-и столбцов. В левой части таблицы представлен исходный граф. В ячейку

Таблица 1.1. Инициализация и итерация 1

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞													8	3
2	3	−∞	4	−∞	−∞													7	5
3	8	−∞	−∞	−5	−∞													4	2
4	−∞	1	−∞	−∞	6													2	1
5	−4	7	−∞	−∞	−∞													6	4

(i, j) на пересечении i-ой строки и j-го столбца записан вес дуги из вершины j в вершину i (именно в таком порядке). Эта часть таблицы далее не меняется. В правом столбце в i-ую ячейку слева записаны максимальный вес дуги, выходящей из i-ой вершины, а слева – номер вершины, в которую входит эта дуга. Так, в 1-ой ячейке слева записан максимальный вес дуги, выходящей из вершины 1 (8), а справа – конец этой дуги (вершина 3); во 2-ой ячейке – вес 7 и номер 5, и т.д. С учётом формулы (7) можно сказать, что 1-ый справа столбец содержит всю информацию о максимальных путях длины 1. Этот столбец далее называется столбцом итерации 1

2-ая итерация.

Таблица 1.2. Итерация 2.1

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	−∞	−∞									12	3	8	3
2	3	−∞	4	−∞	−∞	3	10											7	5
3	8	−∞	−∞	−5	−∞	8	12											4	2
4	−∞	1	−∞	−∞	6	−∞	−∞											2	1
5	−4	7	−∞	−∞	−∞	−4	2											6	4

В левый вспомогательный столбец скопируем 1-ый столбец левой части таблицы. В нём будут находиться расстояния от вершины 1 до остальных вершин. В правый вспомогательный столбец запишем сумму чисел левого вспомогательного столбца и левого столбца итерации 1. Максимальное число (12) из записанных в правый вспомогательный столбец запишем в левую клетку 1-ой ячейки итерации 2, а в правую клетку запишем номер строки этого максимального числа (3). Эти операции соответствуют уравнению Беллмана (6). После этого стираем содержимое вспомогательных столбцов и повторяем те же операции для 2-го столбца левой части таблицы:

Таблица 1.2. Итерация 2.2

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	−∞	−∞									12	3	8	3
2	3	−∞	4	−∞	−∞	−∞	−∞									13	5	7	5
3	8	−∞	−∞	−5	−∞	−∞	−∞											4	2
4	−∞	1	−∞	−∞	6	1	3											2	1
5	−4	7	−∞	−∞	−∞	7	13											6	4

Повторяем те же действия для 3-го, 4-го и 5-го столбцов. Получаем:

Таблица 1.2. Итерация 2.3

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	−∞	−∞									12	3	8	3
2	3	−∞	4	−∞	−∞	4	11									13	5	7	5
3	8	−∞	−∞	−5	−∞	−∞	−∞									11	2	4	2
4	−∞	1	−∞	−∞	6	−∞	−∞											2	1
5	−4	7	−∞	−∞	−∞	−∞	−∞											6	4

Таблица 1.2. Итерация 2.4

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	2	10									12	3	8	3
2	3	−∞	4	−∞	−∞	−∞	−∞									13	5	7	5
3	8	−∞	−∞	−5	−∞	−5	−1									11	2	4	2
4	−∞	1	−∞	−∞	6	−∞	−∞									10	1	2	1
5	−4	7	−∞	−∞	−∞	−∞	−∞											6	4

Таблица 1.2. Итерация 2.5

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	−∞	−∞									12	3	8	3
2	3	−∞	4	−∞	−∞	−∞	−∞									13	5	7	5
3	8	−∞	−∞	−5	−∞	−∞	−∞									11	2	4	2

4	−∞	1	−∞	−∞	6	6	8									10	1	2	1
5	−4	7	−∞	−∞	−∞	−∞	−∞									8	4	6	4

В результате найдены все максимальные пути длины 2. Из 1-ой вершины имеем путь 1→3→2 с весом 12, из 2-ой вершины – путь 2→5→4 с весом 13, из 3-ей – путь 3→2→5 с весом 11, из 4-ой – путь 4→1→3 с весом 10, из 5-ой – путь 5→4→2 с весом 8.

<u>3-ья итерация</u>. Повторяем операции 2-ой итерации с той разницей, что теперь прибавляются элементы левого столбца итерации 2:

Таблица 1.3. Итерация 3.1

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	−∞	−∞							19	3	12	3	8	3
2	3	−∞	4	−∞	−∞	3	16									13	5	7	5
3	8	−∞	−∞	−5	−∞	8	19									11	2	4	2
4	−∞	1	−∞	−∞	6	−∞	−∞									10	1	2	1
5	−4	7	−∞	−∞	−∞	−4	4									8	4	6	4

Таблица 1.3. Итерация 3.2

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	−∞	−∞							19	3	12	3	8	3
2	3	−∞	4	−∞	−∞	−∞	−∞							15	5	13	5	7	5
3	8	−∞	−∞	−5	−∞	−∞	−∞									11	2	4	2
4	−∞	1	−∞	−∞	6	1	11									10	1	2	1
5	−4	7	−∞	−∞	−∞	7	15									8	4	6	4

Таблица 1.3. Итерация 3.3

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	−∞	−∞							19	3	12	3	8	3
2	3	−∞	4	−∞	−∞	4	17							15	5	13	5	7	5
3	8	−∞	−∞	−5	−∞	−∞	−∞							17	2	11	2	4	2
4	−∞	1	−∞	−∞	6	−∞	−∞									10	1	2	1
5	−4	7	−∞	−∞	−∞	−∞	−∞									8	4	6	4

Таблица 1.3. Итерация 3.4

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	2	14							19	3	12	3	8	3
2	3	−∞	4	−∞	−∞	−∞	−∞							15	5	13	5	7	5
3	8	−∞	−∞	−5	−∞	−5	6							17	2	11	2	4	2
4	−∞	1	−∞	−∞	6	−∞	−∞							14	1	10	1	2	1
5	−4	7	−∞	−∞	−∞	−∞	−∞									8	4	6	4

Таблица 1.3. Итерация 3.5

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	−∞	−∞							19	3	12	3	8	3
2	3	−∞	4	−∞	−∞	−∞	−∞							15	5	13	5	7	5
3	8	−∞	−∞	−5	−∞	−∞	−∞							17	2	11	2	4	2
4	−∞	1	−∞	−∞	6	6	16							14	1	10	1	2	1
5	−4	7	−∞	−∞	−∞	−∞	−∞							16	4	8	4	6	4

<u>4-ая итерация</u>:

Таблица 1.4. Итерация 4.1

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	−∞	−∞					18	2	19	3	12	3	8	3
2	3	−∞	4	−∞	−∞	3	18							15	5	13	5	7	5

3	8	−∞	−∞	−5	−∞	8	25							17	2	11	2	4	2
4	−∞	1	−∞	−∞	6	−∞	−∞							14	1	10	1	2	1
5	−4	7	−∞	−∞	−∞	−4	12							16	4	8	4	6	4

Таблица 1.4. Итерация 4.2

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	−∞	−∞					18	2	19	3	12	3	8	3
2	3	−∞	4	−∞	−∞	−∞	−∞					23	5	15	5	13	5	7	5
3	8	−∞	−∞	−5	−∞	−∞	−∞							17	2	11	2	4	2
4	−∞	1	−∞	−∞	6	1	15							14	1	10	1	2	1
5	−4	7	−∞	−∞	−∞	7	23							16	4	8	4	6	4

Таблица 1.4. Итерация 4.3

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	−∞	−∞					18	2	19	3	12	3	8	3
2	3	−∞	4	−∞	−∞	4	19					23	5	15	5	13	5	7	5
3	8	−∞	−∞	−5	−∞	−∞	−∞					19	2	17	2	11	2	4	2
4	−∞	1	−∞	−∞	6	−∞	−∞							14	1	10	1	2	1
5	−4	7	−∞	−∞	−∞	−∞	−∞							16	4	8	4	6	4

Таблица 1.4. Итерация 4.4

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	2	21					18	2	19	3	12	3	8	3
2	3	−∞	4	−∞	−∞	−∞	−∞					23	5	15	5	13	5	7	5
3	8	−∞	−∞	−5	−∞	−5	12					19	2	17	2	11	2	4	2
4	−∞	1	−∞	−∞	6	−∞	−∞					21	1	14	1	10	1	2	1
5	−4	7	−∞	−∞	−∞	−∞	−∞							16	4	8	4	6	4

Таблица 1.4. Итерация 4.5

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	−∞	−∞					18	2	19	3	12	3	8	3
2	3	−∞	4	−∞	−∞	−∞	−∞					23	5	15	5	13	5	7	5
3	8	−∞	−∞	−5	−∞	−∞	−∞					19	2	17	2	11	2	4	2
4	−∞	1	−∞	−∞	6	6	20					21	1	14	1	10	1	2	1
5	−4	7	−∞	−∞	−∞	−∞	−∞					20	4	16	4	8	4	6	4

5-ая итерация:

Таблица 1.5. Итерация 5.1

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	−∞	−∞			26	2	18	2	19	3	12	3	8	3
2	3	−∞	4	−∞	−∞	3	26					23	5	15	5	13	5	7	5
3	8	−∞	−∞	−5	−∞	8	27					19	2	17	2	11	2	4	2
4	−∞	1	−∞	−∞	6	−∞	−∞					21	1	14	1	10	1	2	1
5	−4	7	−∞	−∞	−∞	−4	16					20	4	16	4	8	4	6	4

Таблица 1.5. Итерация 5.2

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	−∞	−∞			26	2	18	2	19	3	12	3	8	3
2	3	−∞	4	−∞	−∞	−∞	−∞			27	5	23	5	15	5	13	5	7	5
3	8	−∞	−∞	−5	−∞	−∞	−∞					19	2	17	2	11	2	4	2
4	−∞	1	−∞	−∞	6	1	22					21	1	14	1	10	1	2	1
5	−4	7	−∞	−∞	−∞	7	27					20	4	16	4	8	4	6	4

Таблица 1.5. Итерация 5.3

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	−∞	−∞			26	2	18	2	19	3	12	3	8	3
2	3	−∞	4	−∞	−∞	4	27			27	5	23	5	15	5	13	5	7	5
3	8	−∞	−∞	−5	−∞	−∞	−∞			27	2	19	2	17	2	11	2	4	2
4	−∞	1	−∞	−∞	6	−∞	−∞					21	1	14	1	10	1	2	1
5	−4	7	−∞	−∞	−∞	−∞	−∞					20	4	16	4	8	4	6	4

Таблица 1.5. Итерация 5.4

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	2	20			26	2	18	2	19	3	12	3	8	3
2	3	−∞	4	−∞	−∞	−∞	−∞			27	5	23	5	15	5	13	5	7	5
3	8	−∞	−∞	−5	−∞	−5	14			27	2	19	2	17	2	11	2	4	2
4	−∞	1	−∞	−∞	6	−∞	−∞			20	1	21	1	14	1	10	1	2	1
5	−4	7	−∞	−∞	−∞	−∞	−∞					20	4	16	4	8	4	6	4

Таблица 1.5. Итерация 5.5

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	−∞	−∞			26	2	18	2	19	3	12	3	8	3
2	3	−∞	4	−∞	−∞	−∞	−∞			27	5	23	5	15	5	13	5	7	5
3	8	−∞	−∞	−5	−∞	−∞	−∞			27	2	19	2	17	2	11	2	4	2
4	−∞	1	−∞	−∞	6	6	27			20	1	21	1	14	1	10	1	2	1
5	−4	7	−∞	−∞	−∞	−∞	−∞			27	4	20	4	16	4	8	4	6	4

6-ая итерация. На 6-ой итерации рассматриваемые операции делаются только с 1-ым столбцом, поскольку требуется найти максимальный путь с началом в вершине 1:

Таблица 1.6. Итерация 6

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	−∞	−∞	−∞	2	−∞	−∞	−∞	35	3	26	2	18	2	19	3	12	3	8	3
2	3	−∞	4	−∞	−∞	3	30			27	5	23	5	15	5	13	5	7	5
3	8	−∞	−∞	−5	−∞	8	35			27	2	19	2	17	2	11	2	4	2
4	−∞	1	−∞	−∞	6	−∞	−∞			20	1	21	1	14	1	10	1	2	1
5	−4	7	−∞	−∞	−∞	−4	23			27	4	20	4	16	4	8	4	6	4

Последняя таблица определяет путь длины 6 с началом в вершине 1. Последовательность вершин, образующих искомый путь, выделена. Его вес равен 35. Слева от выделенных вершин указан вес пути от предыдущей вершины до конца. Сам путь таков: 1→3→2→5→4→1→3. Он не является ни простым путём, ни цепью, поскольку дважды проходит по дуге 1→3.

Напомним, что метод находит максимальный путь для произвольной дискретной многошаговой задачи с аддитивной функцией. Граф здесь только иллюстрирует метод ■

Пример 2. Для того же графа, той же начальной вершины и той же длины пути $N = 6$ найдём путь минимального веса. Отличие от предыдущего примера будет состоять только в том, что на итерации 1 будет выбираться дуга не с максимальным, а с минимальным весом; далее в правом вспомогательном столбце будем искать не максимальный, а минимальный элемент. Конечно, вес отсутствующих дуг равен ∞ (а не −∞). Промежуточные таблицы опущены; демонстрируется только результат каждой итерации.

Таблица 2.1. Инициализация и итерация 1

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	∞	∞	∞	2	∞													−4	5
2	3	∞	4	∞	∞													1	4
3	8	∞	∞	−5	∞													4	2
4	∞	1	∞	∞	6													−5	3
5	−4	7	∞	∞	∞													6	4

Таблица 2.2. Итерация 2

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	∞	∞	∞	2	∞											2	5	−4	5
2	3	∞	4	∞	∞											−4	4	1	4
3	8	∞	∞	−5	∞											5	2	4	2
4	∞	1	∞	∞	6											−2	1	−5	3
5	−4	7	∞	∞	∞											1	4	6	4

Таблица 2.3. Итерация 3

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	∞	∞	∞	2	∞									−3	5	2	5	−4	5
2	3	∞	4	∞	∞									−1	4	−4	4	1	4
3	8	∞	∞	−5	∞									0	2	5	2	4	2
4	∞	1	∞	∞	6									−1	1	−2	1	−5	3
5	−4	7	∞	∞	∞									4	4	1	4	6	4

Таблица 2.4. Итерация 4

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	∞	∞	∞	2	∞							0	5	−3	5	2	5	−4	5
2	3	∞	4	∞	∞							0	4	−1	4	−4	4	1	4
3	8	∞	∞	−5	∞							3	2	0	2	5	2	4	2
4	∞	1	∞	∞	6							−5	3	−1	1	−2	1	−5	3
5	−4	7	∞	∞	∞							5	4	4	4	1	4	6	4

Таблица 2.5. Итерация 5

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	∞	∞	∞	2	∞					1	5	0	5	−3	5	2	5	−4	5
2	3	∞	4	∞	∞					−4	4	0	4	−1	4	−4	4	1	4
3	8	∞	∞	−5	∞					4	2	3	2	0	2	5	2	4	2
4	∞	1	∞	∞	6					−2	3	−5	3	−1	1	−2	1	−5	3
5	−4	7	∞	∞	∞					1	4	5	4	4	4	1	4	6	4

Таблица 2.6. Итерация 6

	1	2	3	4	5	всп		6		5		4		3		2		1	
1	∞	∞	∞	2	∞			−3	5	1	5	0	5	−3	5	2	5	−4	5
2	3	∞	4	∞	∞					−4	4	0	4	−1	4	−4	4	1	4
3	8	∞	∞	−5	∞					4	2	3	2	0	2	5	2	4	2
4	∞	1	∞	∞	6					−2	3	−5	3	−1	1	−2	1	−5	3
5	−4	7	∞	∞	∞					1	4	5	4	4	4	1	4	6	4

Последняя таблица определяет минимальный путь длины 6 с началом в вершине 1. Его вес равен −3. Слева от выделенных вершин указан вес пути от предыдущей вершины до конца. Сам путь таков: 1→5→4→3→2→4→3. Он не является ни простым путём, ни цепью, поскольку дважды проходит по дуге 4→3. Заметим, что простой путь 1→5→4→3→2, являющийся частью найденного пути, также является минимальным путём, соединяющим вершины 1 и 2. Этот же путь был найден алгоритмом Флойда-Уоршолла в примере 8-3■

3. Модификация основной постановки

В данном разделе рассмотрим модификации общей задачи многошаговой оптимизации и соответствующие варианты алгоритма динамического программирования.

1. В ряде случаев кроме начального состояния s^0 и числа шагов N задаётся финальное состояние s^*. Это означает, что $\varphi(s_{N-1},v_N) = s^*$. При переходе к графу это означает, что искомый максимальный путь должен кончаться в соответствующей состоянию s^* вершине x^*.

Модификация алгоритма в этом случае практически очевидна. На итерации 1 в правом столбце для каждой вершины x слева пишется вес дуги (x, x^*), а справа – номер x (напомним, что все вершины графа занумерованы числами 1, 2, …, n). Начиная с итерации 2, алгоритм полностью совпадает с рассмотренным в примере 1 (см. таблицы 1.2–1.6).

2. Иногда требуется найти не оптимальный путь длины N, а оптимальный путь длины <u>не более N</u>, с заданной или не заданной конечной вершиной. В этих случаях применяются точно те же самые алгоритмы. Ведь суть динамического программирования как раз в том, что последовательно находятся оптимальные пути <u>из всех</u> вершин длины 1, длины 2, …, длины N. После этого надо просмотреть числа из строки, соответствующей начальному состоянию, и выбрать из них максимальное или минимальное. В примере 1 путём с максимальным весом среди всех путей длины не более 6-и с началом в вершине 1 оказывается путь длины 6 (см. таблицу 1.6). А в примере 2 на минимизацию оптимальным при тех же условиях оказывается не путь длины 6, вес которого равен –3, а путь длины 1, вес которого равен –4. Естественно, это может случиться только в случае, когда имеются дуги с весами разных знаков. В противном случае вес любого пути при добавлении дуги увеличивается (если веса всех дуг положительны) или уменьшается (если веса всех дуг отрицательны). В этих случаях такие постановки не имеют смысла.

3. В некоторых ситуациях естественно связывать доходы не с переходом из состояния в состояние, а с самими состояниями. Однако эту ситуацию легко свести к предыдущей, положив в построенном графе вес каждой дуги равным доходу от вершины, являющейся концом данной дуги. Тогда вес любого пути станет равным суммарному весу всех его вершин, кроме начальной. Поскольку этот вес один и тот же для всех путей с фиксированным начальным состоянием, это не влияет на определение оптимального пути.

4. Если не задавать число шагов N в многошаговом процессе, то тем самым не будет задаваться и число дуг (т.е. длина пути) в соответствующем графе. Понятно, что если отсутствуют какие бы то ни было требования к графу и ⁄ или к виду искомых путей, то задача оптимизации просто может не иметь решения. В частности, если в графе есть цикл положительной длины, а пути могут содержать циклы, понятно, что вес пути при многократном прохождении такого положительного цикла может быть сделан сколь угодно большим. В то же время ограничения на вид пути (например, запрет на повторяющиеся вершины или дуги) просто «не встраиваются» в алгоритм динамического программирования (в том смысле, что найденные при ограничениях пути не обязаны быть оптимальными). Наиболее естественно в этих случаях выглядят такие условия на граф, при котором длины любых путей будут ограниченными (напомним, что речь в этой главе идёт только об ориентированных графах и путях). Именно такие постановки рассмотрены в следующем разделе.

3.1. Ацикличные задачи многошаговой оптимизации. Под такими задачами будут пониматься задачи, в которых под действием любого многошагового управления никогда нельзя вернуться в начальное состояние. Такие задачи часто связаны с тем, что время, которое всегда течёт в одну сторону, является одним из параметров, определяющих состояние системы. Соответствующий граф будет ациклическим. В силу утверждения 6-7 множество вершин графа распадается на подмножества (которые можно называть слоями), такие, что любая дуга ведёт из слоя с бóльшим номером в слой с мéньшим номером. Для упрощения обозначений будем считать, что 0-ой слой состоит из единственной вершины x^* и что рассматриваются пути, оканчивающиеся в этой вершине. Обычно такая вершина соответствует окончанию рассматриваемого многошагового процесса. Напомним, что число шагов N в данной модификации не задаётся.

Обозначим через $W(x)$ вес оптимального пути, ведущего из вершины x в вершину x^*. Основное уравнение Беллмана можно записать в виде

$$W(x) = \max_y\{w(x, y) + W(y)\}, \qquad (8)$$

где максимум берётся по всем вершинам y, в которые ведёт дуга из вершины x. Учитывая, что все дуги ведут в слои с мéньшими номерами, знание оптимальных путей для всех вершин из слоёв V_r, V_{r-1}, …, V_1 в вершину x^*, образующую слой V_0, позволяет по формуле (8) легко найти оптимальные пути для всех вершин из слоя V_{r+1}. Можно сказать, что в отличие от общего случая, переход здесь делается не по шагам для всех вершин, а по слоям, которых обычно значительно меньше, чем вершин. Соответственно, таблицы, заполнение которых позволяет найти оптимальный путь, также меняются и становятся значительно более обозримыми.

3.1.1. Сетевое планирование. В этом разделе мы остановимся на одном важном классе прикладных задач – так называемых задачах сетевого планирования. Оно используется при проведении крупных разработок, включающих в себя выполнение целого комплекса взаимосвязанных работ (например, строительство, разработка новых технических систем и изделий, проведение ремонта и реконструкции предприятий и агрегатов и т.п.). Сетевые методы основаны на наглядном представлении выполняемого комплекса работ в виде ориентированного графа, дуги которого изображают выполняемые работы, а вершины – события, представляющие собой завершение одних работ и начало других. Последовательность дуг в таком графе определяет порядок, в котором выполняются работы.

На рис.2 приведён пример сетевого графика, на котором показаны работы, необходимые для изготовления некоторого прибора. Сетевой график даёт наглядное представление о порядке выполнения работ, что даёт возможность наиболее рационально распорядиться имеющимися ресурсами. Сам граф отражает ограничения, связанные с выполнением данного комплекса работ. Ясно, что производство деталей может быть начато только после подготовки чертежей и т.д.

Итак, у нас уже есть сетевой график. Более того, будем считать, что уже определены (например, при помощи экспертов) длительности отдельных работ. Возникает вопрос – чему равна длительность выполнения всего комплекса? Этот вопрос не является простым, поскольку в реальных ситуациях число дуг и вершин в сетевом графике может достигать нескольких сотен. Для ответа на поставленный вопрос возможно использование динамического программирования.

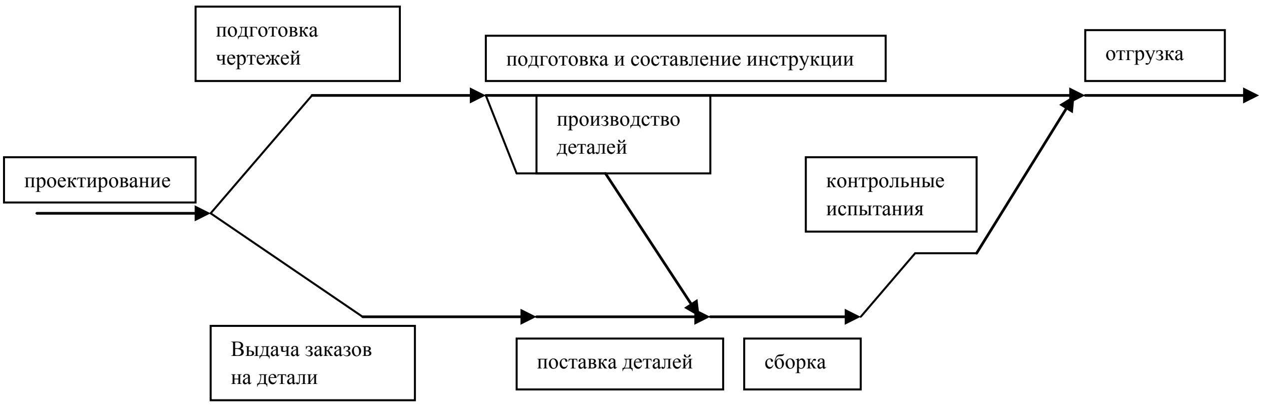


Рис.2

Проведём дальнейшую формализацию задачи. Естественно считать, что граф G, описывающий последовательность работ, является ациклическим, поскольку наличие цикла означает, например, что работа Б должна выполняться после окончания работы А, работа В – после работы Б, а работа А – после работы В. Можно считать также, что в графе G есть ровно одна вершина, в которую не входит ни одна дуга, и ровно одна вершина, из которой не выходит ни одна дуга. Эти вершины соответствуют началу и завершению всего комплекса работ. Каждой дуге графа приписано положительное (не обязательно целое) число (длина дуги), которое интерпретируется как длительность операции (работы), условно представляемой данной дугой.

Рассмотрим любой ориентированный путь P из начальной вершины x^0 в конечную вершину x^*. Рассмотрим сумму длин дуг, образующих путь P, и обозначим её через $T(P)$. Так как операции, соответствующие дугам одного и того же пути, могут выполняться только последовательно (по построению сетевого графика), то общая длительность выполнения всего комплекса не меньше, чем $T(P)$. Поскольку в качестве P был взят произвольный путь, то
$\mathrm{T}_{\text{об}} \geq \max_{\mathrm{P}} T(P)$

Таким образом, возникает задача поиска пути максимальной длины (такой путь в сетевом графике называется ***критическим***). Эта задача в общем виде рассматривалась в начале раздела 3.1. Начальная вершина x^0 и конечная вершина x^* – это входная и выходная вершины сетевого графика. Число дуг в пути не фиксируется. Основная идея заключается в разделении множества вершин графа на слои. При этом начальный слой состоит из единственной выходной вершины, а последний слой состоит из единственной входной вершины. Основой для перехода от слоя к слою служит уравнение Беллмана (8).

Пример 3. Рассмотрим сетевой график, показанный на рис. 3. Пунктирная линия не соответствует никакой конкретной работе и её длина равна 0. Вводятся такие «псевдоработы» только для того, чтобы правильно отражать последовательность реальных работ. В рассматриваемом случае включение псевдоработы Р означает, что работа З не может быть начата до окончания работы В. Использование псевдоработ не меняет последовательности любых других операций и не изменяет длины критического пути.

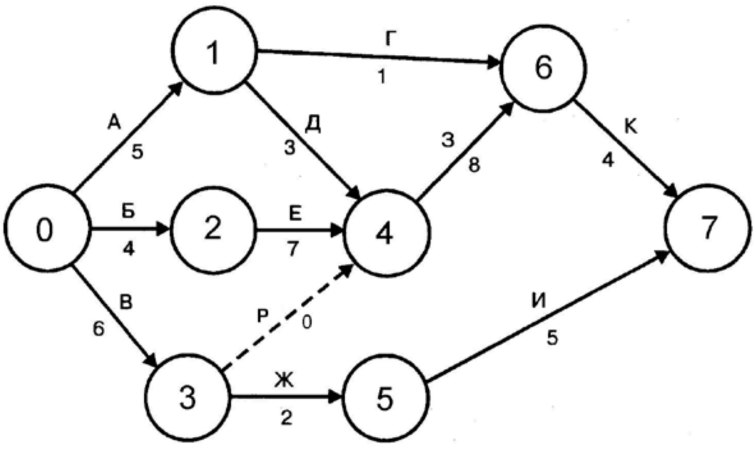


Рис.3. Пример сетевого графика

Построим разбиение множества вершин графа, показанного на рис.3, в соответствии с доказательством утверждения 6-7:

1. Положим $G_0 = G$, V_0={7}.

2. Удалим из графа G_0 вершину 7 и ведущие в неё дуги И и К. Оставшийся граф G_1 показан на рис.4a. В этом графе из вершин 5 и 6 не выходит дуг, и в соответствии с процедурой из упомянутого доказательства V_1={5,6}.

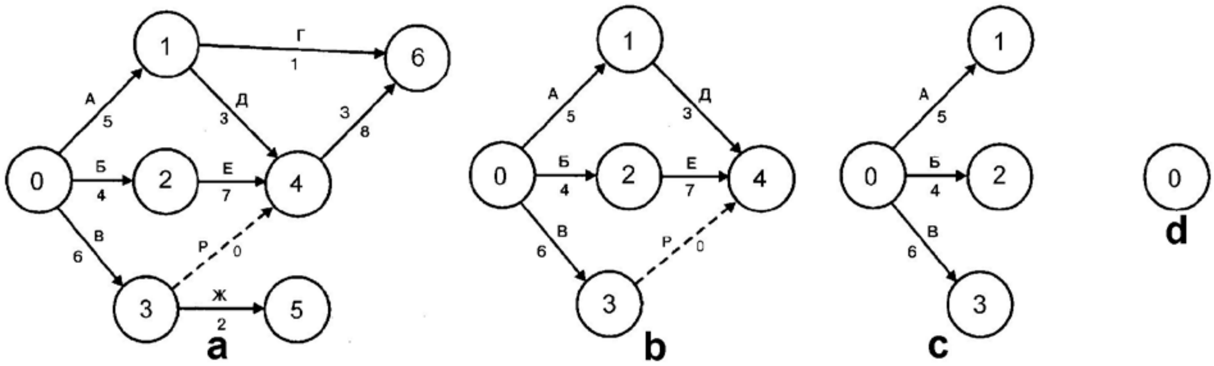


Рис.4

3. Удалим из графа G_1 (см. рис.4a) вершины 5, 6 и ведущие в них дуги Ж, З и Г. Оставшийся граф G_2 показан на рис.4b. В этом графе из вершины 4 не выходит дуг, и в соответствии с той же процедурой V_2={4}.

4. Удалим из графа G_2 (см. рис.4b) вершину 4 и ведущие в неё дуги Р, Е и Д. Оставшийся граф G_3 показан на рис.4c. В этом графе из вершин 1, 2, 3 не выходит дуг, и в соответствии с процедурой V_3={1, 2, 3}.

5. Удалим из графа G_3 (см. рис.4c) вершины 1, 2, 3 и ведущие в них дуги А, Б и В. Оставшийся граф G_4 показан на рис.4d. В этом графе имеется только одна вершина 0 и в соответствии с той же процедуройV_4={0}.

6. Процедура окончена, поскольку других вершин больше нет. Построено следующее разбиение:

V_0={7}, V_1={5, 6},V_2= {4}, V_3={1, 2, 3}, V_4={0}, (9)

и любая дуга ведёт из слоя с бóльшим номером в слой с мéньшим номером. Задача состоит в том, чтобы определить путь максимальной длины из вершины 0 в вершину 7.

Инициализация и итерация 1. Составим таблицу, столбцы которой соответствуют вершинам заданного графа G. Расположим эти столбцы слева направо в соответствии с разбиением (9): сначала идёт столбец, соответствующий вершине 0, потом (в произвольном порядке) – столбцы, соответствующие вершинам 1, 2, 3 и т.д., вплоть до вершины 7. Число ячеек в каждом столбце равно числу дуг, выходящих из данной вершины, плюс одна строка (на следующую вершину в искомом максимальном пути).

Таблица 3.1. Нахождение критического пути. Инициализация и итерация 1

0		1		2		3		4		5		6		7
5	1	3	4	7	4	0	4	8	6	5	7	4	7	
4	2	1	6			2	5			5	7	4	7	
6	3													

Каждая ячейка соответствует одной дуге, выходящей из вершины. В правую верхнюю клетку ячейки пишется номер вершины, в которую идёт дуга; в левую верхнюю – её длина. Так, из вершины 3 выходят две дуги: в вершину 4 длины 0 (псевдоработа) и в вершину 5 длины 2. Эти данные полностью описывают исходный сетевой график и далее не меняются.

На итерации 1 определяются максимальные пути из вершин из 1-го слоя. По самой конструкции ясно, что такие пути совпадают с дугами, ведущими из этих вершин в конечную вершину 7. Длина максимального пути и следующая вершина на нём пишутся в нижние две клетки каждого столбца. Поэтому на итерации 1 туда просто копируются числа из верхних двух клеток.

Итерация 2. На итерации 2 определяются максимальные пути для вершин из слоя $V_2 = \{4\}$.

Таблица 3.2. Итерация 2

0		1		2		3		4		5		6		7
5	1	3	4	7	4	0	4	8	6	5	7	4	7	
								12	6					
4	2	1	6			2	5	12	6	5	7	4	7	
6	3													

В левую нижнюю клетку из верхней ячейки столбца 4 пишутся сумма длины 8 (от вершины 4 до вершины 6) и длины 4 (максимальной длины от вершины 6 до конечной вершины 7, взятой из левой нижней клетки в столбце 6). В правую нижнюю клетку этой же ячейки пишется номер вершины 6. Поскольку других дуг, выходящих из вершины 4, нет, то эти же два числа копируются в нижнюю строчку столбца 4. Тем самым уже найден максимальный путь из вершины 4 в вершину 7: 4→6→7.

Итерация 3. На итерации 3 определяются максимальные пути для вершин из слоя $V_3 = \{1,2, 3\}$.

Таблица 3.3. Итерация 3

0		1		2		3		4		5		6		7
5	1	3	4	7	4	0	4	8	6	5	7	4	7	
		15	4	19	4	12	4	12	6					
4	2	1	6	19	4	2	5	12	6	5	7	4	7	
		5	6			7	5							
6	3	15	4			12	4							

Начинаем для определённости с правого (из этого слоя) столбца 3. 1-ая сверху ячейка в этом столбце соответствует дуге (3,4) длины 0. В левую нижнюю клетку этой же ячейки записываем сумму длины 0 и длины 12 из нижней строчки столбца 4 (максимальной длины пути из 4 в 7). Далее, в следующей сверху ячейке столбца 3 записываем сумму длины 2 дуги (3,5) и длины 5 максимального пути из вершины 5 в вершину 7 (взятую из нижней строки столбца 5). Копируем в нижнюю строку столбца 5 ту из двух полученных строк – (12,4) и (7,5) – в которой длина (1-ое число) больше. Поэтому копируем (12,4). Заметим, что именно здесь реализуется уравнение Беллмана (8).

Столбец 2 содержит только одну ячейку, которая соответствует дуге (2,4) длины 7. Записываем в нижнюю половину этой ячейки длину 19 = 7+12 и вершину 4; ту же пару копируем в нижнюю строку.

Столбец 1 содержит две ячейки, соответствующие дугам (1,4) длины 3 и (1,6) длины 1. Записываем суммы и выбираем максимальную из них для записи в нижнюю строку.

Итерация 4. На итерации 4 определяются максимальные пути для вершин из последнего слоя $V_4 = \{0\}$. Столбец 1 содержит 3 ячейки, соответствующие дугам (0,1), (0,2) и (0,3). В верхнюю ячейку запишем длину 20 = 5 + 15 и номер 1. В следующую ячейку запишем сумму 23 = 4 + 19 и номер 2. В нижнюю ячейку запишем сумму 18 = 6 + 12 и номер 3. В нижнюю строку запишем пару с максимальной длиной 23.

На этом итерации закончены. Искомый путь выделен подсветкой в нижних строчках. Он таков: 0→2→4→6→7. Его длина равна 23 ■

Таблица 3.4. Итерация 4

0		1		2		3		4		5		6		7
5	1	3	4	7	4	0	4	8	6	5	7	4	7	
15	1	15	4	19	4	12	4	12	6					
4	2	1	6	19	4	2	5	12	6	5	7	4	7	
23	2	5	6			7	5							
6	3	15	4			12	4							
18	3													
23	2													

Особенно простым алгоритм динамического программирования становится в случае, когда каждая дуга ведёт из слоя в слой с предыдущим номером, а число выходящих дуг всегда равно 2. Именно такой случай рассмотрен в следующем подразделе.

3.1.2. Задача о замене машины. В настоящем разделе приводится конкретный пример, иллюстрирующий введенные понятия и рассматриваемую модификацию алгоритма динамического программирования. Одной из основных проблем индустрии является проблема замены старого парка машин новым, устаревших орудий современными устройствами. Оборудование со временем изнашивается как в буквальном смысле слова, так и «морально», т.е. устаревает по сравнению с более современным модернизированным оборудованием. Наступает момент, когда большие затраты на новое оборудование, убытки вследствие остановки производства и расходы на переподготовку кадров – всё это компенсируется увеличением производительности и уменьшением производственных затрат.

Содержательно задача о замене состоит в определении оптимальной политики модернизации и замены оборудования, т.е. в определении последовательности замен, максимизирующей суммарную прибыль.

Введем упрощающие предположения. Предположим, что у нас имеется только одна машина, которая приносит ежегодно некоторый доход, требует определённого ухода и может быть в любой момент продана и заменена новой машиной, аналогичной старой, но более эффективной. Для определенности будем считать, что замена может быть осуществлена один раз в год (именно, в начале каждого года). Предполагается также, что доход и затраты на содержание известным образом зависят от года выпуска машины и срока её службы. Наконец, стоимость замены старой машины на новую известным образом зависит от года замены (в основном от цены новой машины).

Рассмотрим процесс, длящийся 6 лет. Доход и затраты на содержание новых машин, появляющихся к началу каждого из этих 6 лет, в зависимости от возраста машины (т.е. числа полных лет, которые она уже проработала), приведены в таблице 4 (в условных единицах). Мы начинаем процесс, имея машину, называемую исходной, прослужившую уже 3 года; её характеристики представлены в таблице 5. Затраты на замену машины на новую (в зависимости от года замены) представлены в таблице 6.

Общий (суммарный) доход равен сумме доходов, полученных на каждом году работы. Доход за каждый год может быть получен из дохода (указанного в таблицах 4 и 5) путём вычитания затрат на содержание машины и (в случае замены в начале этого года) затрат на замену.

Таблица 4. Данные о новых машинах

1. Машина, новая в начале 1-го года

Возраст машины	0	1	2	3	4	5
Доход от работы	120	115	115	110	105	100
Затраты на содержание	10	10	15	15	20	20

2. Машина, новая в начале 2-го года

Возраст машины	0	1	2	3	4
Доход от работы	125	120	110	110	105
Затраты на содержание	10	10	10	15	15

3. Машина, новая в начале 3-го года

Возраст машины	0	1	2	3
Доход от работы	135	125	110	105
Затраты на содержание	10	10	10	10

4. Машина, новая в начале 4-го года

Возраст машины	0	1	2
Доход от работы	140	135	125
Затраты на содержание	5	10	10

5. Машина, новая в начале 5-го года

Возраст машины	0	1
Доход от работы	150	140
Затраты на содержание	5	10

6. Машина, новая в начале 6-го года

Возраст машины	0
Доход от работы	155
Затраты на содержание	5

Таблица 5. Данные об исходной машине

Возраст машины	3	4	5	6	7	8
Доход от работы	60	60	50	50	50	45
Затраты на содержание	55	55	55	60	60	60

Таблица 6. Данные о заменах

Год замены машины	1	2	3	4	5	6
Затраты на замену	80	90	100	110	120	130

Выбор оптимальной последовательности замен можно осуществить в результате применения метода динамического программирования. Для этого, в соответствии с тем, что излагалось в начале этой главы, надо определить:

множество S состояний;

множества Q_s воздействий для всех $s \in S$;

функцию переходов $\varphi(s,v)$;

функцию дохода $\psi(s,v)$;

начальное состояние $s^0 \in S$;

число шагов N.

Определим указанные параметры задачи.

1. Определение множества состояний S является самой важным при построении подобных моделей. В данном случае состояниями естественно считать пары вида (t,τ), где t – номер только что начавшегося года (t = 1, 2, ..., 6), τ – возраст машины на начало года этого года t. Именно в начале года принимается основное решение – продолжать работать на старой машине или заменить её на новую. Обратим внимание на следующее. В последний (шестой) год есть состояние (6,8). Это состояние соответствует исходной машине, у которой в начале 6-го года возраст 8 лет, так как она проработала 3 года до начала процесса и 5 лет после начала процесса. Есть пара (6,5) (она соответствует машине, новой в начале 1-го года, у которой в начале 6-го года возраст 5 лет), пара (6,4) (она соответствует машине, новой в начале 2-го года, у которой в начале 6-го года возраст 4 года) и т.д. Состояний (6,7) и (6,6) в множестве S просто нет! Точно также нет состояний (5,6) и (5,5), (4,5) и (4,4) и т.д. Обратим внимание, что пар вида $(t,0)$ также нет. В начале года у нас ещё нет новой машины, поэтому возраст имеющейся не меньше 1-го года. Если же мы приняли решение о покупке новой машины, то через год её возраст также будет не меньше 1-го года. Наконец, необходимо ввести «финальное» состояние (обозначим его через s^*), соответствующее завершению процесса после окончания 6-го года.

2. Множество воздействий Q_s для любого состояния $s = (t,\tau) \in S$, кроме состояния s^*, состоит из двух элементов: С и Н (оставить старую машину на весь t-ый год или заменить её в начале t-го года новой машиной). Для состояния s^* множество Q_s пусто.

3. Функция переходов $\varphi(s,v)$ определяется формулами

для $t = 1, ..., 5$ $\varphi((t,\tau),\text{С}) = (t+1, \tau+1)$ – старая машина через год становится на год старше;

$\varphi((t,\tau),\text{Н}) = (t+1,1)$ $(t = 1, ..., 5)$ – возраст новой машины через год становится равным 1;

для $t = 6$ $\varphi((6,\tau),\text{С}) = \varphi((6,\tau),\text{Н}) = s^*$ – поскольку 6-ой год является последним, то и соответствующее состояние совпадает с s^*.

Множество состояний и функцию переходов будет удобно рассматривать в специальной таблице 7, в которой ячейки, соответствующие состояниям, разделены на 4 клетки. При выборе управления С переход из любого состояния (t,τ) при $t < 6$ происходит в соседнюю справа ячейку, а при выборе управления Н – в нижнюю ячейку соседнего справа столбца. При $t = 6$ переход возможен только в финальное состояние s^*. Назначение других клеток будет объяснено далее.

Таблица 7. Функция переходов

6	5	4	3	2	1	0
(1,3)	(2,4)	(3,5)	(4,6)	(5,7)	(6,8)	s^*
	(2,1)	(3,2)	(4,3)	(5,4)	(6,5)	
		(3,1)	(4,2)	(5,3)	(6,4)	
			(4,1)	(5,2)	(6,3)	
				(5,1)	(6,2)	
					(6,1)	

4. Функция доходов $\psi(s,v)$. Для определения функции доходов используем данные из таблиц 4 – 6. Числа $\psi((t,\tau),\text{С})$ (т.е. доход при сохранении старой машины) будем записывать в правые верхние клетки каждой ячейки, в которых в левой верхней клетке уже записаны состояния (t,τ). В верхней строчке таблицы записаны состояния (1,3), (2,4), …, (6,8). Все они соответствуют исходной машине, возраст которой равен 3 в начале 1-го года, равен 4 в начале 2-го года, и т.д. вплоть до начала 6-го года, когда её возраст равен 8. Но доходы и расходы на содержание исходной машины в зависимости от возраста даны в таблице 5. Остаётся только вычесть расходы из доходов и разность записать в правые верхние клетки ячеек 1-ой строки (см. 1-ую строку таблицы 8).

Далее, во 2-ой сверху строке таблицы 8 записаны состояния (2,1), (3,2), …, (6,5). Все они соответствуют машине, возраст которой равен 1 в начале 2-го года, равен 2 в начале 3-го года, и т.д. вплоть до начала 6-го года, когда её возраст равен 5. Конечно, речь идёт о машине, которая была новой в начале 1-го года, имела возраст 1 в начале 2-го года и т.д. Доходы и затраты на содержание в зависимости от возраста даны в таблице 4.1. Вычтем из доходов расходы и запишем разности в правые верхние клетки ячеек 2-ой строки (см. 2-ую строку таблицы 8).

Точно так же заполняем правые верхние клетки ячеек 3-ей, 4-ой, 5-ой и 6-ой строк таблицы 8.

Таблица 8. Построение функция доходов при управлении С

6		5		4		3		2		1		0
(1,3)	5	(2,4)	5	(3,5)	–5	(4,6)	–10	(5,7)	–10	(6,8)	–15	s^*
		(2,1)	105	(3,2)	100	(4,3)	95	(5,4)	85	(6,5)	80	
				(3,1)	110	(4,2)	100	(5,3)	95	(6,4)	90	
						(4,1)	115	(5,2)	100	(6,3)	95	
								(5,1)	125	(6,2)	115	
										(6,1)	130	

Числа $\psi((t,\tau)$, Н) (доход при замене старой машины на новую) будем записывать в правые нижние клетки каждой ячейки, в которых в левой верхней клетке уже записаны состояния (t,τ). Решение Н в начале t-го года приводит к следующей алгебраической сумме. Доходы от работы новой машины идут со знаком плюс, а расходы на её содержание и расходы на замену, идут со знаком минус. Важно, что все три составляющие зависят только от года t. Составим вспомогательную таблицу следующего вида, основываясь только на заданных таблицах 4 и 6:

Год t	1	2	3	4	5	6
Доход от работы машины, новой в году t	120	125	135	140	150	155
Затраты на её содержание	10	10	10	5	5	5
Затраты на замену	80	90	100	110	120	130
Суммарный доход	30	25	25	25	25	20

Суммарный доход от машины в году t вставляется во все ячейки t-го столбца таблицы 9, в правые нижние клетки ячеек (t = 1,2,3,4,5,6). Поэтому таблица 9 содержит всю информацию о доходах в зависимости от состояния и управления.

Заметим, что все предшествующие рассуждения позволили только прийти к разумному представлению исходной информации в задаче о замене машины. Можно сказать, что пока осуществлена только инициализация. И теперь можно перейти к самой многошаговой оптимизации, исходя из построенной таблицы 9.

Таблица 9. Исходная информация о многошаговой задаче

6		5		4		3		2		1		0
(1,3)	5	(2,4)	5	(3,5)	–5	(4,6)	–10	(5,7)	–10	(6,8)	–15	s^*
	30		25		25		25		25		20	
		(2,1)	105	(3,2)	100	(4,3)	95	(5,4)	85	(6,5)	80	
			25		25		25		25		20	
				(3,1)	110	(4,2)	100	(5,3)	95	(6,4)	90	
					25		25		25		20	
						(4,1)	115	(5,2)	100	(6,3)	95	
							25		25		20	
								(5,1)	125	(6,2)	115	
									25		20	
										(6,1)	130	
											20	

Определим оставшиеся два общих параметра.

5. Начальное состояние $s^0 = (1, 3)$ (по условию мы начинаем процесс при исходной машине, прослужившей 3 года).

6. Описанный процесс начинается в начале 1-го года, а заканчивается в начале 7-го года, т.е. сразу после окончания 6-го года. В каждом следующем состоянии (t,τ) 1-ая компонента t

увеличивается на 1 независимо от выбранного управления С или Н. Поэтому любой из возможных процессов состоит из 6 шагов, т.е. $N = 6$.

<u>Итерация 1</u>. Скопируем таблицу 9. В столбце 1 в левую нижнюю клетку каждой ячейки занесём максимум из двух чисел, стоящих в правом столбце ячейки. Запишем рядом (справа) букву С, если число сверху было больше, чем число снизу. Запишем рядом (справа) букву Н, если число сверху было меньше, чем число снизу. В случае равенства запишем любую букву.

Таблица 10.1. Результат итерации 1

6		5		4		3		2		1		0
(1,3)	5	(2,4)	5	(3,5)	–5	(4,6)	–10	(5,7)	–10	(6,8)	–15	s^*
	30		25		25		25		25	20 Н	20	
		(2,1)	105	(3,2)	100	(4,3)	95	(5,4)	85	(6,5)	80	
			25		25		25		25	80 С	20	
				(3,1)	110	(4,2)	100	(5,3)	95	(6,4)	90	
					25		25		25	90 С	20	
						(4,1)	115	(5,2)	100	(6,3)	95	
							25		25	95 С	20	
								(5,1)	125	(6,2)	115	
									25	115 С	20	
										(6,1)	130	
										130 С	20	

Легко понять, что таблица 10.1 содержит решение задачи за один шаг. Для любого из состояний предпоследнего уровня (из столбца 1) найдено оптимальное решение за один шаг – работать на старой машине или заменить её на новую, а также посчитан максимально возможный доход, который (вместе с соответствующим решением) написан в левой нижне клетке всех ячеек 1-го стобца.

<u>Итерация 2</u>. Скопируем таблицу 10.1. На этой итерации мы будем заполнять свободные левые нижние клетки во 2-ом столбце таблице. Суть дела такова. В каждом состоянии мы можем выбрать одно из двух воздействий (С или Н), получить соответствующие доходы (они указаны в правом столбце ячейки – верхнее число при выборе С, нижнее число – при выборе Н). Известно, в какие состояния из 1-го столбца мы переходим при выборе С или Н, и самое главное – известно, какие максимальные доходы мы можем получить в каждом из этих состояний. Поэтому мы можем выбрать то воздействие, которое даст суммарный наилучший результат.

Рассмотрим состояние (5,7) из 2-го столбца. Если мы выберем С, то перейдём в состояние (6,8), в котором максимальный доход (за один шаг) уже посчитан и равен 20. В сумме получается $-10 + 20 = 10$. Если мы выберем Н, то перейдём в состояние (6,1), в котором максимальный доход (за один шаг) уже посчитан и равен 130. В сумме получается $25 + 130 = 155$. Так как $155 > 10$, то выбираем Н и записываем в левую нижнюю клетку 155 Н. Аналогично, для каждого состояния сравниваем две суммы: 1) число в правой верхней клетке ячейки + число в левой нижней клетке ячейки справа от данной и 2) число в правой нижней клетке ячейки и число в левой нижней клетке последней ячейки в соседнем справа столбце. Имеем для состояния (5,4) $85 + 80$ и $25 + 130$. Так как $165 > 155$, то записываем под состоянием (5,4) 165 С. Далее, для состояния (5,3) имеем $95+90 > 25 + 130$; для (5,2) имеем $100+95 > 25 + 130$;для (5,1) имеем $125+115 > 25 + 130$. Результаты записаны во 2-ом столбце.

Таблица 10.2. Результат итерации 2

6		5		4		3		2		1		0
(1,3)	5	(2,4)	5	(3,5)	–5	(4,6)	–10	(5,7)	–10	(6,8)	–15	s*
	30		25		25		25	155 Н	25	20 Н	20	
		(2,1)	105	(3,2)	100	(4,3)	95	(5,4)	85	(6,5)	80	
			25		25		25	165 С	25	80 С	20	
				(3,1)	110	(4,2)	100	(5,3)	95	(6,4)	90	
					25		25	185 С	25	90 С	20	
						(4,1)	115	(5,2)	100	(6,3)	95	
							25	195 С	25	95 С	20	
								(5,1)	125	(6,2)	115	
								240 С	25	115 С	20	
										(6,1)	130	
										130 С	20	

Итерация 3. Скопируем таблицу 10.2. На этой итерации мы будем заполнять свободные левые нижние клетки в 3-ем столбце таблице, исходя только из того, что уже записано во 3-ем и 2-ом столбцах. Имеем –10 + 155 < 25 + 240, т.е. записываем 265 Н. Далее, имеем 95 + 165 < 25 + 240, т.е. записываем 265 Н. Далее, имеем 100 + 185 > 25 + 240, т.е. записываем 285 Н. Далее, имеем 115 + 195 > 25 + 240, т.е. записываем 310 С.

Таблица 10.3. Результат итерации 3

6		5		4		3		2		1		0
(1,3)	5	(2,4)	5	(3,5)	–5	(4,6)	–10	(5,7)	–10	(6,8)	–15	s*
	30		25		25	265 Н	25	155 Н	25	20 Н	20	
		(2,1)	105	(3,2)	100	(4,3)	95	(5,4)	85	(6,5)	80	
			25		25	265 Н	25	165 С	25	80 С	20	
				(3,1)	110	(4,2)	100	(5,3)	95	(6,4)	90	
					25	285 Н	25	185 С	25	90 С	20	
						(4,1)	115	(5,2)	100	(6,3)	95	
						310 С	25	195 С	25	95 С	20	
								(5,1)	125	(6,2)	115	
								240 С	25	115 С	20	
										(6,1)	130	
										130 С	20	

Итерации 4 – 6 . Скопируем таблицу 10.3. Будем представлять все три итерации в одной таблице, что возможно, поскольку столбцы заполняются последовательно справа налево. Для 4-го столбца имеем –5 + 265 < 25 + 310, так что записываем 335 Н. Далее, 100 + 265 > 25 + 310, так что записываем 365 С. Далее, 110 + 285 > 25 + 310, так что записываем 395 С. Для 5-го столбца имеем 5 + 335 < 25 + 395, так что записываем 420 Н. Далее, 105 + 365 > 25 + 395, так что записываем 470 С. Наконец, для 6-го столбца 5 + 420 < 30 + 470, так что записывем 500 Н.

Таблица 10.4. Результат итераций 4 – 6

6		5		4		3		2		1		0
(1,3)	5	(2,4)	5	(3,5)	–5	(4,6)	–10	(5,7)	–10	(6,8)	–15	s^*
500 Н	30	420 Н	25	335 Н	25	265 Н	25	155 Н	25	20 Н	20	
		(2,1)	105	(3,2)	100	(4,3)	95	(5,4)	85	(6,5)	80	
		470 С	25	365 С	25	265 Н	25	165 С	25	80 С	20	
				(3,1)	110	(4,2)	100	(5,3)	95	(6,4)	90	
				395 С	25	285 Н	25	185 С	25	90 С	20	
						(4,1)	115	(5,2)	100	(6,3)	95	
						310 С	25	195 С	25	95 С	20	
								(5,1)	125	(6,2)	115	
								240 С	25	115 С	20	
										(6,1)	130	
										130 С	20	

Таким образом, таблица 10.4 даёт полное решение рассматриваемой задачи о замене машины. Максимальный доход равен 500. Правильная стратегия замен в данном случае такова. В начале 1-го года меняем машину на новую, на которой работаем до начала 4-го года. В начале 4-го года ещё раз меняем машину на новую, на которой работаем до конца рассматриваемого 6-летнего периода

Заметим, что число состояний в данном случае равно 21, и использовать общую схему, т.е. заполнять вручную таблицы с 21-ой строкой и ещё бóльшим числом столбцов, практически нереально. Здесь существенно использована специфика примера, приводящая к разделению множества состояний на слои. Как и в других примерах, связанных с заполнением таблиц, повторное заполнение похожих таблиц здесь делается только с демонстрационными целями. Конечно, все данные можно последовательно записывать в одну и ту же таблицу, продвигаясь от правых столбцов к левым ■

4. Задания

Задание 1а. В графе на рис.08-7 (см. также рис.1) методом динамического программирования найти путь, состоящий из 5-и дуг, с максимальной суммарной длиной (суммой длин входящих в него дуг) и началом в вершине 2. Длины дуг взять из задания 8-4. Для образца см. пример 1 ■

Задание 1б. В графе на рис.08-7 (см. также рис.1) методом динамического программирования найтипуть, состоящий из 5-и дуг, с минимальной суммарной длиной (суммой длин входящих в него дуг) и началом в вершине 2. Длины дуг взять из задания 8-4. Для образца см. пример 2■

Задание 2. Решить описанным в разделе 3.1.2 методом задачу о замене машины при данных о новых машинах из таблицы 4. Данные об исходной машине и затратах на замену взять из таблиц 5 и 6. Для образца см. раздел 3.1.2.

Таблица 5.01. Данные об исходной машине

Возраст машины	3	4	5	6	7	8
Доход от работы	60	60	50	50	50	45
Затраты на содержание	55	55	55	60	60	60

Таблица 6.01. Данные о заменах

Год замены машины	1	2	3	4	5	6
Затраты на замену	120	130	140	150	150	160

Таблица 5.02. Данные об исходной машине

Возраст машины	3	4	5	6	7	8
Доход от работы	80	70	50	50	50	40
Затраты на содержание	50	55	60	60	70	75

Таблица 6.02. Данные о заменах

Год замены машины	1	2	3	4	5	6
Затраты на замену	110	115	120	125	130	145

Таблица 5.03. Данные об исходной машине

Возраст машины	3	4	5	6	7	8
Доход от работы	60	60	50	50	50	45
Затраты на содержание	50	60	60	65	70	75

Таблица 6.03. Данные о заменах

Год замены машины	1	2	3	4	5	6
Затраты на замену	110	120	120	125	130	140

Таблица 5.04. Данные об исходной машине

Возраст машины	3	4	5	6	7	8
Доход от работы	80	70	50	50	50	40
Затраты на содержание	55	55	55	60	60	60

Таблица 6.04. Данные о заменах

Год замены машины	1	2	3	4	5	6
Затраты на замену	110	110	120	125	130	145

Таблица 5.05. Данные об исходной машине

Возраст машины	3	4	5	6	7	8
Доход от работы	80	70	50	50	50	40
Затраты на содержание	50	55	60	65	70	70

Таблица 6.05. Данные о заменах

Год замены машины	1	2	3	4	5	6
Затраты на замену	120	130	140	150	150	160

Таблица 5.06. Данные об исходной машине

Возраст машины	3	4	5	6	7	8
Доход от работы	80	70	60	50	50	40
Затраты на содержание	55	55	55	60	60	60

Таблица 6.06. Данные о заменах

Год замены машины	1	2	3	4	5	6
Затраты на замену	120	130	135	145	150	160

Таблица 5.07. Данные об исходной машине

Возраст машины	3	4	5	6	7	8
Доход от работы	60	60	50	50	50	45
Затраты на содержание	50	55	60	60	70	75

Таблица 6.07. Данные о заменах

Год замены машины	1	2	3	4	5	6
Затраты на замену	115	130	140	150	150	155

Таблица 5.08. Данные об исходной машине

Возраст машины	3	4	5	6	7	8
Доход от работы	60	60	50	50	50	45
Затраты на содержание	55	55	55	60	60	60

Таблица 6.08. Данные о заменах

Год замены машины	1	2	3	4	5	6
Затраты на замену	110	115	120	125	130	145

Таблица 5.09. Данные об исходной машине

Возраст машины	3	4	5	6	7	8
Доход от работы	75	65	60	60	55	55
Затраты на содержание	45	45	50	50	50	60

Таблица 6.09. Данные о заменах

Год замены машины	1	2	3	4	5	6
Затраты на замену	90	110	120	150	160	175

Таблица 5.10. Данные об исходной машине

Возраст машины	3	4	5	6	7	8
Доход от работы	70	70	50	50	40	40
Затраты на содержание	45	65	65	65	70	70

Таблица 6.10. Данные о заменах

Год замены машины	1	2	3	4	5	6
Затраты на замену	115	120	130	130	150	155

Таблица 5.11. Данные об исходной машине

Возраст машины	3	4	5	6	7	8
Доход от работы	70	65	60	60	55	55
Затраты на содержание	50	65	65	70	70	70

Таблица 6.11. Данные о заменах

Год замены машины	1	2	3	4	5	6
Затраты на замену	110	120	130	130	150	150

Таблица 5.12. Данные об исходной машине

Возраст машины	3	4	5	6	7	8
Доход от работы	70	70	50	50	40	40
Затраты на содержание	45	45	50	50	50	65

Таблица 6.12. Данные о заменах

Год замены машины	1	2	3	4	5	6
Затраты на замену	110	120	130	130	150	150

Таблица 5.13. Данные об исходной машине

Возраст машины	3	4	5	6	7	8
Доход от работы	70	70	50	50	40	40
Затраты на содержание	50	65	65	65	70	70

Таблица 6.13. Данные о заменах

Год замены машины	1	2	3	4	5	6
Затраты на замену	90	110	120	150	160	165

Таблица 5.14. Данные об исходной машине

Возраст машины	3	4	5	6	7	8
Доход от работы	70	70	50	50	40	40
Затраты на содержание	45	45	50	50	50	65

Таблица 6.14. Данные о заменах

Год замены машины	1	2	3	4	5	6
Затраты на замену	90	110	120	150	160	170

Таблица 5.15. Данные об исходной машине

Возраст машины	3	4	5	6	7	8
Доход от работы	70	65	60	60	55	55
Затраты на содержание	50	65	65	65	70	70

Таблица 6.15. Данные о заменах

Год замены машины	1	2	3	4	5	6
Затраты на замену	90	110	120	150	160	175

Таблица 5.16. Данные об исходной машине						
Возраст машины	3	4	5	6	7	8
Доход от работы	70	65	60	60	55	55
Затраты на содержание	45	45	50	50	50	65

Таблица 6.16. Данные о заменах						
Год замены машины	1	2	3	4	5	6
Затраты на замену	110	120	130	130	150	150

■

5. Предметный указатель

Литература к части 2

1. Арис Р. Дискретное динамическое программирование. – М.: Мир, 1969, 172 с.
2. Гудман С., Хидетниеми С. Введение в анализ и разработку алгоритмов. – М.: Мир, 1981, 368 с.
3. Дехтярь М.И. Лекции по дискретной математике. – М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2009, 259 с. ISBN 978-5-94774-714-0.
4. Котов В.М., Соболевская Е.П. Разработка и анализ алгоритмов: теория и практика. – Минск: БГУ, 2009. – 251 с. ISBN 978-985-518-107-2.
5. Кузнецов О.П. Дискретная математика для инженера. − 5-е изд., стер. – СПб: Изд-во «Лань», 2007. – 400 с.
6. Робертс Ф.С. Дискретные математические модели с приложениями к социальным, биологическим и экологическим задачам. − М.: Наука. Гл. ред. физ.-мат. лит., 1986. − 496 с.
7. Форд Л, Фалкерсон Д. Потоки в сетях. – М.: Мир, 1966, 236 с.

Часть 3. ВЗАИМОДЕЙСТВИЯ: КОНФЛИКТЫ И СОТРУДНИЧЕСТВО

В книге Г. Коэна «Обо всё можно договориться» утверждается: «Успешный подход к переговорам заключается в выяснении того, что в действительности нужно противной стороне, и в доведении до её сознания того, каким образом она могла бы добиться желаемого, не мешая мне получить своё». Эта цитата определяет очень разумный подход к взаимодействию участников с несовпадающими интересами, фокусирующийся на сотрудничестве**,** а не на соперничестве. Мы видим, что взаимодействия самого различного рода, в которых в той или иной степени присутствуют и конфликты, и сотрудничество, встречаются постоянно − и в нашей повседневной жизни, и в профессиональной деятельности, и в окружающей нас действительности. Деление наследства или имущества при разводе, согласование различных точек зрения при принятии решений, спортивные соревнования и творческие конкурсы, выделение квот партиям или территориям в выборном органе, распределение различных видов работ между сотрудниками, доступ к компьютерным ресурсам при работе операционных систем и пользовательских программ, борьба за существование между хищниками и жертвами, переговоры о разоружении между супердержавами − всё это примеры взаимодействий, естественно сочетающих конфликты и сотрудничество.

Важность и распространённость таких взаимодействий уже с давних пор привлекали внимание представителей различных видов профессиональной деятельности − математиков, управленцев, психологов, политологов, социологов, военных и других специалистов.

На многочисленные ситуации такого типа естественно смотреть с двух разных точек зрения, которые условно можно назвать внутренней и внешней. С <u>внутренней точки зрения</u> мы ассоциируем себя с одним из участников взаимодействия (или же на самом деле являемся им), пытаясь добиться наилучших для этого участника результатов, при том, что действия других участников (также преследующих свои цели) от нас не зависят. В силу такой точки зрения эта проблематика ориентирована больше на конфликты, чем на сотрудничество. Она относится к теории игр, является хорошо и подробно исследованной и имеет многочисленные приложения в самых разных реальных ситуациях. С <u>внешней точки зрения</u> мы ассоциируем себя не с одним из участников взаимодействия, а с теми людьми и/или ведомствами, которые организуют это взаимодействие и разрабатывают правила, в рамках которых оно и проистекает. При этом предполагается, что результат взаимодействия всех участников, которые преследуют свои <u>индивидуальные цели</u>, будет достаточно <u>разумным</u> с более общей точки зрения, которая больше ориентирована на сотрудничество, чем на конфликт. Люди и организации, заинтересованные в результатах подобных взаимодействия и разрабатывающих их правила, иногда называются ***метаигроками***, что подчёркивает их особую – по сравнению с остальными участниками – роль.

В отличие от теории игр, общей теории взаимодействия с указанной точки зрения нет, однако есть важные результаты, полученные в рамках отдельных направлений. Достаточно известным примером такого рода постановок является синтез механизмов в институциональной экономике, когда игроками являются экономические агенты, а в роли метаигрока, определяющего правила их взаимодействия, выступает государство (в лице соответствующих властных структур). Однако с такой точки зрения можно рассмотреть и многие другие процессы и явления в самых различных сферах. Примером такой постановки является и выработка правил агрегации индивидуальных предпочтений, и разработка систем пропорционального представительства, и правила справедливого дележа, и алгоритмы предоставления вычислительных ресурсов процессам в современных ЭВМ, и многие другие ситуации, связанные с разработкой оптимальных или хотя бы разумных правил взаимодействия участников. Некоторые из этих вопросов, наряду с более традиционными постановками теории игр, и составляют материал данной части пособия. В него входят три главы. Глава 11 «Матричные игры» посвящена самому базовому и простому классу игр – конечным играм двух участников с нулевой суммой. В главе 12 «Другие игровые модели» рассмотрены – на самом простом уровне – другие классы игр: биматричные и позиционные. Наконец, в главе 13 «Организация взаимодействия» представлены результаты анализа взаимодействия с упомянутой внешней точки зрения в трёх достаточно известных и практически интересных ситуациях.

Глава 11. Матричные игры

1. Понятие матричной игры

В этой главе будет рассмотрена простейшая модель конфликтных ситуаций – матричные игры двух лиц. Изложим эту модель, учитывая, что некоторые её свойства присущи значительно более сложным конфликтам. Сначала дадим содержательное описание моделируемой ситуации.

В игре участвуют два игрока (скажем, А и В). У каждого игрока имеется конечное число возможных действий, которые обычно называются ***стратегиями***. Природа этих действий может быть самой разнообразной, но сама по себе она не имеет значения. Каждый игрок выбирает одну из своих стратегий независимо от другого игрока. Реально это значит, что игрок А не знает, какую стратегию выбирает игрок В, а игрок В не знает, какую стратегию выбирает игрок А. После того, как оба выбора сделаны, игрок А получает выигрыш, зависящий от обеих выбранных стратегий. Если выигрыш – положительное число, то игрок В платит игроку А соответствующую сумму (в заранее обговорённой валюте); в противном случае игрок А платит игроку В. Игрок А стремится максимизировать свой выигрыш, а игрок В стремится минимизировать свой проигрыш.

При формальном описании игры в силу конечности множеств стратегий у обоих игроков без ограничения общности можно считать, что сами стратегии задаются натуральными числами от 1 до m у игрока А и от 1 до n у игрока В. Поэтому выигрыш игрока А можно задавать ***платёжной матрицей*** Q размера m×n: если игроки выбрали стратегии $i\in\{1, ..., m\}$ и $j\in\{1, ..., n\}$, то выигрыш игрока А равен числу q_{ij} – элементу матрицы Q, стоящему на пересечении i-ой строки и j-го столбца. Напомним, что проигрыш игрока В считается равным выигрышу игрока А. Можно считать, что выигрыш игрока В при выборе стратегий $i\in\{1, ..., m\}$ и $j\in\{1, ..., n\}$ равен числу $-\,q_{ij}$. Игры, в которых суммарный выигрыш всех игроков равен нулю, называются ***играми с нулевой суммой***. Таким образом, матричная игра двух лиц является – по определению – игрой с нулевой суммой.

2. Решение игры

В матричной игре с платёжной матрицей Q игрок А стремится максимизировать свой выигрыш при любом выборе стратегии игроком В. Предположим, что он выбирает стратегию $i\in\{1, ..., m\}$. Поскольку игрок А не знает, какую именно стратегию (т.е. какой индекс j) выбрал игрок В, то всё, что он может себе гарантировать – это выигрыш в размере

$$g_i = \min_{j\in\{1,\dots,n\}} q_{ij}. \qquad (1)$$

Естественно, что игрок А выберет такую стратегию $i\in\{1, ..., m\}$, которая максимизирует его гарантированный выигрыш g_i. Поэтому он выбирает стратегию i^*, для которой

$$g_{i^*} = \max_{i\in\{1,\dots,m\}} \min_{j\in\{1,\dots,n\}} q_{ij}. \qquad (2)$$

Заметим, что при любом $i\in\{1, ..., m\}$ выражение $\min_{j\in\{1,\dots,n\}} q_{ij}$, входящее в (2), не зависит от j. Поэтому значение g_{i^*} зависит только от платёжной матрицы Q. Любая стратегия i^*, удовлетворяющая условию (2), называется ***оптимальной стратегией*** игрока А (оптимальная стратегия i^* может определяться неоднозначно, так как g_i может быть максимальным при различных $i\in\{1, ..., m\}$).

Проведённые рассуждения показывают, что именно должен делать игрок А, чтобы добиться максимально возможного гарантированного выигрыша. Величина g_{i^*}, которая определяется только по платёжной матрице Q, называется ***нижней ценой игры*** и обозначается через V^-.

Игрок B стремится минимизировать свой проигрыш. Предположим, что он выбирает стратегию $j\in\{1, ..., n\}$. Поскольку игрок B не знает, какую именно стратегию выбрал игрок A, то всё, что он может себе гарантировать – это проигрыш в размере

$$f_j= \max_{i\epsilon\{1,...,m\}} q_{ij}. \qquad (3)$$

Естественно, что игрок B выберет такую стратегию $j\in\{1, ..., n\}$, которая минимизирует его гарантированный проигрыш f_j. Поэтому он выбирает стратегию j^*, для которой

$$f_{j^*}= \min_{j\in\{1,...,n\}} \max_{i\epsilon\{1,...,m\}} q_{ij}. \qquad (4)$$

Как и значение g_{i^*}, значение f_{j^*} зависит только от платёжной матрицы Q. Любая стратегия j^*, удовлетворяющая условию (4), называется ***оптимальной стратегией*** игрока B (оптимальная стратегия j^* может определяться неоднозначно, так как f_j может быть минимальным при различных $j\in\{1, ..., n\}$).

Проведённые рассуждения показывают, что именно должен делать игрок B, чтобы добиться минимально возможного гарантированного проигрыша. Величина f_{j^*}, которая определяется только по платёжной матрице Q, называется ***верхней ценой игры*** и обозначается через V^+.

В силу теоремы о минимаксе для любой матрицы Q имеет место неравенство

$$\max_{i\in\{1,...,m\}} \min_{j\epsilon\{1,...,n\}} q_{ij}\leq \min_{j\in\{1,...,n\}} \max_{i\epsilon\{1,...,m\}} q_{ij} \quad (5)$$

или, с учётом (2) и (4),

$$V^- \leq V^+.$$

Таким образом, гарантированный выигрыш игрока A не может быть больше гарантированного проигрыша игрока B.

В теории игр особенно важны те случаи, когда нижняя и верхняя цены игры совпадают. Общее значение $V= V^-= V^+$. называется **ценой игры**. При этом имеет место следующее

Утверждение 1. 1) $V^-=V^+$ тогда и только тогда, когда некоторый элемент $q_{i^*j^*}$ платёжной матрицы Q является одновременно максимальным в её i^*-ой строке и минимальным в её j^*-ом столбце, т.е. для любых $i\in\{1, ..., m\}$ и $j\in\{1, ..., n\}$

$$q_{ij^*}\leq q_{i^*j^*} \leq q_{i^*j}. \qquad (6)$$

2) цена игры V совпадает с $q_{i^*j^*}$; 3) стратегии i^* и j^* являются оптимальными ■

Пара стратегий $\langle i^*, j^*\rangle$, удовлетворяющих требованиям утверждения 1, может определяться неоднозначно. Однако для всех таких пар $\langle i^*, j^*\rangle$ элементы $q_{i^*j^*}$ платёжной матрицы Q равны одному и тому же числу – цене игры V. Всякий элемент $q_{i^*j^*}$ платёжной матрицы Q, который является одновременно максимальным в i^*-ой строке и минимальным в j^*-ом столбце, называется ***седловой точкой*** матрицы Q.

Заметим, что в силу неравенств (6) любому игроку оказывается невыгодным менять оптимальную стратегию, если другой игрок будет придерживаться своей оптимальной стратегии. Для игрока A это следует из 1-го неравенства, для игрока B – из 2-го неравенства. Именно в силу разумности одновременного выбора игроками в данной ситуации своих оптимальных стратегий, любая пара стратегий $\langle i^*, j^*\rangle$, удовлетворяющих требованиям утверждения 1, называется ***решением игры.***

Решения существуют далеко не у всех игр рассматриваемого типа. Как выбирать стратегии в случае отсутствия седловых точек у платёжной матрицы, будет показано в разделе 4 данной главы.

Пример 1. Рассмотрим матричную игру двух лиц со следующей платёжной матрицей:

$$Q = \begin{pmatrix} 7 & 6 & 5 & 4 \\ 1 & 8 & 2 & 3 \\ 8 & 1 & 3 & 2 \end{pmatrix}. \qquad (7)$$

Найдём, в соответствии с формулами (1) и (2), оптимальные стратегии игрока A. По формуле (1)

$g_1= \min_{1\leq j\leq 4} q_{1j}$ = min{7, 6, 5, 4} = 4;

$g_2= \min_{1\leq j\leq 4} q_{2j}$ = min{1, 8, 2, 3} = 1;

$g_3= \min_{1\leq j\leq 4} q_{3j}$ = min{8, 1, 3, 2} = 1.

По формуле (2) g_{i^*}= max{4, 1, 1}, откуда следует, что оптимальная стратегия $i^* = 1$ и нижняя цена игры $V^-= 4$.

Найдём, в соответствии с формулами (3) и (4), оптимальные стратегии игрока B. По формуле (3)
$f_1 = \max_{1\le i\le 3} q_{i1}$ = max{7, 1, 8} = 8;
$f_2 = \max_{1\le i\le 3} q_{i2}$ = max{6, 8, 1} = 8;
$f_3 = \max_{1\le i\le 3} q_{i3}$ = max{5, 2, 3} = 5;
$f_4 = \max_{1\le i\le 3} q_{i4}$ = max{4, 3, 2} = 4.
По формуле (4) f_{j^*} = min{8, 8, 5, 4} = 4, откуда следует, что оптимальная стратегия j^* = 4 и верхняя цена игры V^+= 4.

Таким образом, нижняя цена игры совпадает с верхней ценой. Как и должно быть по утверждению 1, цена игры 4 равна $q_{i^*j^*} = q_{14}$. Пара стратегий ⟨1, 4⟩ является решением игры и седловой точкой матрицы Q■

Пример 2. Рассмотрим матричную игру двух лиц со следующей платёжной матрицей:

$$Q = \begin{pmatrix} 7 & 6 & 5 & 4 \\ 1 & 8 & 2 & 5 \\ 8 & 1 & 3 & 2 \end{pmatrix}, \qquad (8)$$

все элементы которой, кроме q_{24}, совпадают с соответствующими элементами платёжной матрицы из примера 1.

Найдём, в соответствии с формулами (1) и (2), оптимальные стратегии игрока A. По формуле (1)
$g_1 = \min_{1\le j\le 4} q_{1j}$ = min{7, 6, 5, 4} = 4;
$g_2 = \min_{1\le j\le 4} q_{2j}$ = min{1, 8, 2, 5} = 1;
$g_3 = \min_{1\le j\le 4} q_{3j}$ = min{8, 1, 3, 2} = 1.
По формуле (2) g_{i^*}= max{4, 1, 1}, откуда следует, что оптимальная стратегия i^* = 1 и нижняя цена игры V^-= 4.

Найдём, в соответствии с формулами (3) и (4), оптимальные стратегии игрока B. По формуле (3)
$f_1 = \max_{1\le i\le 3} q_{i1}$ = max{7, 1, 8} = 8;
$f_2 = \max_{1\le i\le 3} q_{i2}$ = max{6, 8, 1} = 8;
$f_3 = \max_{1\le i\le 3} q_{i3}$ = max{5, 2, 3} = 5;
$f_4 = \max_{1\le i\le 3} q_{i4}$ = max{4, 5, 2} = 5.
По формуле (4) f_{j^*} = min{8, 8, 5, 5} = 5, откуда следует, что множеством оптимальных стратегий j^* является множество {3, 4} и верхняя цена игры V^+ = 5.
Таким образом, в данном случае, в отличие от примера 1, нижняя цена игры не равна верхней цене. Поэтому для данной платёжной матрицы Q решений игры, как и седловых точек, нет ■

Для любой матричной игры верно следующее

Утверждение 2. При прибавлении любого числа a (положительного или отрицательного) ко всем элементам платёжной матрицы Q множества оптимальных стратегий обоих игроков не изменяются, а нижняя и верхняя цена игры получены из исходных прибавлением того же числа a ■

В силу утверждения 2 существование решения игры также не зависит от прибавлении ко всем элементам платёжной матрицы Q произвольной константы.

3. Удаление стратегий

Количество строк и/или столбцов платёжной матрицей может быть уменьшено за счёт удаления доминируемых и дублирующих стратегий. Стратегия s игрока A называется ***доминируемой*** его стратегией t, если все элементы s-ой строки платёжной матрицы не больше, чем соответствующие элементы её t-ой строки. Игроку A нет смысла выбирать доминируемую стратегию s, поскольку при выборе доминирующей стратегии t он выигрывает не меньше, независимо от стратегии игрока B. Стратегия s игрока B называется ***доминируемой*** его стратегией t, если все элементы s-го столбца платёжной матрицы не меньше, чем соответствующие элементы её t-го столбца. Игроку B нет смысла выбирать доминируемую стратегию s, поскольку при выборе доминирующей стратегии t он проигрывает не больше, независимо от стратегии игрока A. Стратегия s(любого игрока) называется ***дублирующей*** его стратегию t, если соответствующие этим стратегиям строки (или столбцы) совпадают. Очевидно, что из нескольких совпадающих стратегий достаточно оставить только одну, удалив все другие. Обратите особое внимание на подчёркнутые различия в определении доминируемых стратегий для строчного (A) и столбцового (B) игроков.

Пример 3. Рассмотрим матричную игру двух лиц со следующей платёжной матрицей

$$\begin{array}{c c}
 & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 1 & 2 & 3 & 4 & 4 & 7 \\ 7 & 6 & 5 & 4 & 4 & 8 \\ 1 & 8 & 2 & 3 & 3 & 6 \\ 8 & 1 & 3 & 2 & 2 & 5 \end{pmatrix}
\end{array}$$

В данном случае для игрока A стратегия 1 является доминируемой стратегией 2, для игрока B стратегия 6 является доминируемой стратегиями 3, 4 и 5, а стратегии 4 и 5 являются дублирующими друг друга. После удаления указанных доминируемых стратегий и одной из дублирующих стратегий останется следующая матрица, в которой доминируемых и дублирующих стратегий уже нет:

$$\begin{pmatrix} 7 & 6 & 5 & 4 \\ 1 & 8 & 2 & 3 \\ 8 & 1 & 3 & 2 \end{pmatrix} \blacksquare$$

Утверждение 3. Удаление доминируемых и дублирующих стратегий из платёжной матрицы не влияет на значение нижней и верхней цены игры, заданной данной матрицей. Множества оптимальных стратегий также не меняются (за исключением удаления дублирующих) ■

Таким образом, при удалении из платёжной матрицы «лишних» строк и столбцов суть игры не меняется, в то время как размерность матрицы может значительно уменьшиться. В примере 3 вместо матрицы размера 4×6 получена матрица размера 3×4.

Пример 4. Рассмотрим матричную игру двух лиц со следующей платёжной матрицей:

16	12	10	8	22
21	28	18	38	35
33	14	17	21	13

В данной матрице 1-ая строка доминируема 2-ой строкой, поскольку все элементы 2-ой строки больше, чем соответствующие (стоящие в тех же столбцах) элементы 1-ой строки. После удаления 1-ой строки получаем следующую матрицу:

22	28	18	38	33
33	14	17	21	13

В новой матрице 1-ый и 4-ый столбцы доминируемы 3-им столбцом. После их удаления останется матрица

28	18	33
14	17	13

в которой 1-ая строка доминирует 2-ую строку. После удаления 2-ой строки останется единственная строка ⟨28, 18, 33⟩, откуда следует, что игроку B надо выбрать 2-ой столбец, содержащий минимальный проигрыш 18, т.е. исходная матрица сведена к матрице размера 1×1, или одному числу 18. Возвращаясь к исходной матрице, видим, что пара стратегий ⟨2, 3⟩ является решением данной игры, а её цена равна 18.

Начнём теперь в исходной платёжной матрице с удаления доминируемых столбцов, а не строк, как делали выше. После удаления доминируемого 1-го столбца получаем

12	10	8	22
28	18	38	35
14	17	21	13

Далее, удаляя доминируемые 1-ую и 3-ью строку, получаем единственную строку ⟨28, 18, 38, 33⟩, и выбирая в ней минимальный проигрыш 18, получаем тот же ответ - матрицу размером 1×1, т.е. число 18 ■

Результат примера 4 не случаен. Имеет место

Утверждение 4. При любом порядке удаления доминируемых и дублирующих стратегий получаем одну и ту же матрицу, в которой удаляемых строк и столбцов больше нет ■

4. Смешанное расширение матричных игр

Исследование в матричных играх начинается с нахождения её цены. Если матричная игра имеет решение, то нахождением цены игры (и соответствующих оптимальных стратегий) исследование игры и заканчивается. Но при нахождении оптимальных стратегий в любом случае определяется нижняя и верхняя цена данной игры (см. формулы (2) и (4)). Игрок А не должен надеяться на выигрыш бóльший, чем верхняя цена игры, и может быть уверен в получении выигрыша не меньше нижней цены игры. Соответственно, игрок В не должен надеяться на проигрыш мéньший, чем нижняя цена игры, и может быть уверен, что его проигрыш не больше верхней цены игры. В этом общем случае новые решения матричных игр следует искать в возможности многократного повторения игр и применения стратегий случайно, с определённой вероятностью.

Модификация исходной матричной игры, в которой её стратегии могут выбираться игроками уже не однозначно (как в разделах 1 – 3), а вероятностным образом, называется ***смешанным расширением*** данной матричной игры. Если игрок А выбирает стратегию i с вероятностью x_i (i = 1, ..., m), а игрок В – стратегию j с вероятностью y_j(j = 1, ..., n), то говорят, что игроки А и В используют ***смешанные стратегии*** $x = (x_1, ..., x_m)$ и $y = (y_1, ..., y_n)$. Стратегии исходной игры обычно называются ***чистыми стратегиями***, а рассмотренные в разделе 2 решения исходной игры – решениями в чистых стратегиях. Чистая стратегия является частным случаем смешанной стратегии, когда все вероятности, кроме вероятности выбора данной чистой стратегии, равны 0, а вероятность выбора данной чистой стратегии равна 1.

Таким образом, в смешанном расширение матричной игры множество стратегий игрока А – это бесконечное множество вероятностных распределений на конечном множестве исходных чистых стратегий $\{1, ..., m\}$. Проще говоря, это множество всех векторов $x = (x_1 , ..., x_m)$, у которых все координаты неотрицательны и их сумма равна 1. Координата x_i вектора x является вероятностью выбора чистой стратегии i. Аналогично определяется множество стратегий игрока В – множество всех векторов $y = (y_1 , ..., y_n)$, у которых все координаты неотрицательны и их сумма равна 1. В отличие от множеств чистых стратегий, множества смешанных стратегий бесконечны.

Пусть игроки А и В выбрали стратегии x и y. Тогда средний выигрыш игрока А определяется формулой

$$g(x,y) = \sum_{i=1}^{m} \sum_{j=1}^{n} x_i y_j q_{ij}. \quad (9)$$

Как и ранее, игрок А стремится максимизировать свой средний выигрыш $g(x,y)$, а игрок В – минимизировать его, т.е. минимизировать свой средний проигрыш. Как и ранее, игрок А не знает, какую стратегию y выбирает игрок В. Повторяя рассуждения из начала раздела 2, приходим к определению оптимальной смешанной стратегии x^* игрока А и нижней цены игры в смешанных стратегияхV^-:

смешанной оптимальной стратегией x^* игрока А называется любая смешанная стратегия игрока А, на которой достигается внешний максимум по множеству всех смешанных стратегий в следующем выражении (максимине):

$$\max_x \min_y g(x,y), \quad (10a)$$

где $g(x,y)$ определяется формулой (9), x^* – любая смешанная стратегия игрока А, на которой достигается внешний максимум в (10а), и $V^- = \min_y g(x^*,y)$. Обозначим через X^* множество всех оптимальных стратегий игрока А. Как и в разделе 2,V^- и X^* определяются только по платёжной матрице Q.

Аналогично определяется минимакс

$$\min_y \max_x g(x,y), \quad (10b)$$

смешанная оптимальная стратегияy^* игрока В, множество Y^* всех оптимальных стратегий игрока В и верхняя цена игры в смешанных стратегиях $V^+ = \max_x g(x,y^*)$, определяемые только по платёжной матрице Q.

Как и в случае чистых стратегий, для смешанных стратегий имеет место неравенство

$$V^- \le V^+; \quad (11)$$

если $V^- = V^+$, то число V, равное их общему значению, называется ***ценой игры в смешанных стратегиях***.

Решением матричной игры в смешанных стратегиях называется пара стратегий $\langle x^*, y^*\rangle$, удовлетворяющая условиям

$$g(x, y^*) \le g(x^*, y^*) \le g(x^*, y), \quad (12)$$

где $g(x,y)$ определяется формулой (9), x и y – произвольные стратегии игроков А и В.

Для смешанных стратегий имеет место утверждение, аналогичное утверждению 1 для чистых стратегий:

Утверждение 5. 1) $V^- = V^+$ тогда и только тогда, когда для некоторой пары стратегий $\langle x^*, y^*\rangle$ выполняются условия (12); 2) стратегии x^* и y^*, удовлетворяющие (12), являются оптимальными; 3) цена игры $V = g(x^*, y^*)$ ■

Принципиальное отличие случая смешанных стратегий от случая чистых стратегий формулируется в следующем виде:

Утверждение 6. Для любой матричной игры существует решение в смешанных стратегиях ■

В частности, в силу утверждений 6 и 5, в любой матричной игре в смешанных стратегиях
$V^- = V^+$. (13)

Напомним, что решение игры в чистых стратегиях существует не всегда, что и продемонстрировано в примере 2.

Пример 5. Рассмотрим матричную игру двух лиц со следующей платёжной матрицей:
$Q = \begin{pmatrix} 2 & 3 \\ 4 & 1 \end{pmatrix}$

В этом случае $\max_i \min_j qij = \max\{2, 1\} = 2$; $\min_j \max_i q_{ij} = \min\{4, 3\} = 3$, и поскольку
$\max_i \min_j q_{ij} < \min_j \max_i q_{ij}$,
то чистых оптимальных стратегий нет.

Рассмотрим смешанные стратегии $x^* = (0{,}75;\ 0{,}25)$ и $y^* = (0{,}5;\ 0{,}5)$. Проверим их на оптимальность. Для этого подсчитаем выражение $g(x^*, y^*)$. В соответствии с формулой (9) имеем
$g(x^*, y^*) = \sum_{i=1}^{m}\sum_{j=1}^{n} x_i^* y_j^* q_{ij} = x_1^* y_1^* 2 + x_1^* y_2^* 3 + x_2^* y_1^* 4 + x_2^* y_2^* 1 = 0{,}75\cdot 0{,}5\cdot 2 + 0{,}75\cdot 0{,}5\cdot 3 +$
$0{,}25\cdot 0{,}5\cdot 4 + 0{,}25\cdot 0{,}5\cdot 1 = 0{,}75 + 1{,}125 + 0{,}5 + 0{,}125 = 2{,}5.$ (14)

Рассмотрим теперь выражение $g(x, y^*)$ для произвольной стратегии $x = (x_1,x_2)$. Из 3-его члена в равенствах (14) после простых выкладок получаем
$g(x, y^*) = x_1\cdot 2{,}5 + x_2\cdot 2{,}5 = (x_1 + x_2)\cdot 2{,}5 = 2{,}5$ (15a)
(так как сумма вероятностей равна 1).

Рассмотрим теперь выражение $g(x^*, y)$ для произвольной стратегии $y = (y_1,y_2)$. Из 3-его члена в равенствах (14) после простых выкладок получаем
$g(x^*, y) = y_1\cdot 2{,}5 + y_2\cdot 2{,}5 = (y_1 + y_2)\cdot 2{,}5 = 2{,}5$ (15b)
(так как сумма вероятностей равна 1).

Формулы (14), (15a) и (15b) влекут для указанных стратегий $x^* = (0{,}75;\ 0{,}25)$ и $y^* = (0{,}5;\ 0{,}5)$ и для любых стратегий $x = (x_1,x_2)$ и $y = (y_1,y_2)$ двойное неравенство
$g(x, y^*) \le g(x^*, y^*) \le g(x^*, y)$,
совпадающее с (12). Поэтому по определению пара стратегий $\langle x^*, y^*\rangle$ является решением в смешанных стратегиях данной матричной игры, не обладающей решением в чистых стратегиях ■

Пример 6. Рассмотрим матричную игру двух лиц со следующей платёжной матрицей:
$Q = \begin{pmatrix} 2 & 3 \\ 1 & -14 \end{pmatrix}$

В этом случае $\max_i \min_j q_{ij} = \max\{2, -14\} = 2$; $\min_j \max_i q_{ij} = \min\{2, 3\} = 2$, и поскольку максимин равен минимаксу, пара чистых стратегий $\langle 1, 1\rangle$ является решением данной игры в чистых стратегиях (см. раздел 2).

Напомним, что чистая стратегия является частным случаем смешанной. Положим $x^* = (1;\ 0)$ и $y^* = (1;\ 0)$. Пара смешанных стратегий $\langle x^*, y^*\rangle$ является <u>той же самой</u> парой чистых стратегий $\langle 1, 1\rangle$. При этом, с учётом матрицы Q и вида стратегий x^* и y^*, имеем
$g(x^*, y^*) = 2.$ (16)

Рассмотрим теперь выражение $g(x, y^*)$ для произвольной стратегии $x = (x_1,x_2)$. В данном случае, с учётом матрицы Q и того, что $y^* = (1;\ 0)$, получаем
$g(x, y^*) = x_1\cdot 2 + x_2\cdot 1 = x_1\cdot 2 + (1-x_1) = x_1+1 \le 2.$ (17a)
Аналогично, для $x^* = (1;\ 0)$ и произвольной стратегии $y = (y_1,y_2)$, получаем
$g(x^*, y) = y_1\cdot 2 + y_2\cdot 3 = y_1\cdot 2 + (1-y_1)\cdot 3 = 3-y_1 \ge 2.$ (17b)

Формулы (16), (17a) и (17b) влекут для указанных стратегий $x^* = (0{,}75;\ 0{,}25)$ и $y^* = (0{,}5;\ 0{,}5)$, и для любых стратегий $x = (x_1,x_2)$ и $y = (y_1,y_2)$ двойное неравенство
$g(x, y^*) \le g(x^*, y^*) \le g(x^*, y)$,

совпадающее с (12). Поэтому, по определению, исходная пара чистых стратегий $\langle x^*, y^*\rangle$, которая была решением данной игры в чистых стратегиях, является решением этой же игры и в смешанных стратегиях ■

Результат примера 6 не случаен. Имеет место (почти очевидное)

Утверждение 7. В любой матричной игре решение в чистых стратегиях является решением в смешанных стратегиях ■

К сожалению, поиск решений матричных игр в смешанных стратегиях (если у них нет решений в чистых стратегиях) является достаточно сложным с вычислительной точки зрения и здесь не рассматривается. Однако для отдельных классов матричных игр удаётся найти решения в смешанных стратегиях значительно более простыми способами, которые рассматриваются в следующем разделе.

5. Решение игр размерности 2×2 и 2×*n* в смешанных стратегиях

5.1. Игра размерности 2×2. Рассмотрим произвольную матричную игру со следующей платёжной матрицей размерности 2×2 общего вида:

$$Q = \begin{pmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{pmatrix}. \tag{18}$$

Для поиска решений в смешанных стратегиях в данной игре рассматриваются два взаимоисключающих случая.

Случай 1. У игры есть решение в чистых стратегиях. Тогда в силу утверждения 7 это же самое решение (представленное в виде пары смешанных стратегий, как в примере 6) является решением игры в смешанных стратегиях. На этом поиск решений в смешанных стратегиях закончен.

Прежде чем перейти к случаю 2 – отсутствию решения в чистых стратегиях – сформулируем некоторые простые, но нужные для дальнейшего утверждения.

Утверждение 8. Пусть в матрице (18) есть доминируемые или дублирующие строки или столбцы. Тогда в матричной игре с этой платёжной матрицей есть решение в чистых стратегиях ■

Заметим, что условие утверждения 8 является достаточным, но не необходимым. В игре с матрицей $Q = \begin{pmatrix} 2 & 3 \\ 1 & -9 \end{pmatrix}$ есть решение в чистых стратегиях, а доминируемых или дублирующих строк и столбцов нет.

Утверждение 9. Пусть в матрице (18) совпадают элементы одного столбца или одной строки. Тогда в матричной игре с этой платёжной матрицей есть решение в чистых стратегиях ■

Напомним определение функции sgn(x):

$$\operatorname{sgn}(x) = \begin{cases} 1, \text{если} \, x > 0 \\ -1, \text{если} \, x < 0 \\ 0, \text{если} \, x = 0 \end{cases}$$

Утверждение 10. Пусть в матрице (18)

$$\operatorname{sgn}(q_{11}-q_{21}) = -\operatorname{sgn}(q_{22}-q_{12}) \tag{19a}$$

или

$$\operatorname{sgn}(q_{11}-q_{12}) = -\operatorname{sgn}(q_{22}-q_{21}). \tag{19b}$$

Тогда в матричной игре с этой платёжной матрицей есть решение в чистых стратегиях ■

Случай 2. У игры нет решения в чистых стратегиях. Рассмотрим выражение (9) для выигрыша игрока А в игре размерности 2×2:

$$g(x,y) = x_1 \cdot y_1 \cdot q_{11} + x_1 \cdot y_2 \cdot q_{12} + x_2 \cdot y_1 \cdot q_{21} + x_2 \cdot y_2 \cdot q_{22}. \tag{20}$$

Учитывая, что $x_1 + x_2 = 1$, $y_1 + y_2 = 1$, положим $x = x_1$, $y = y_1$ и отождествим стратегии игроков с однозначно определяющими их вероятностями x и y. Представим (20) (после простых преобразований) в следующем виде:

$$g(x, y) = mx + n, \tag{21a}$$

где

$$m = (q_{11}-q_{12}-q_{21}+q_{22}) \cdot y + (q_{12} - q_{22}), \tag{22a}$$

$$n = (q_{21}-q_{22}) \cdot y + q_{22}. \tag{23a}$$

Выражение (20) является, в соответствии с определениями, проигрышем игрока B в той же самой игре. Представим (20) в виде, аналогичном (21a):

$g(x, y) = sy+ t,$ (21b)

где

$s = (q_{11}–q_{12}–q_{21}+q_{22})\cdot x+(q_{21} – q_{22}),$ (22b)

$t= (q_{12}–q_{22})\cdot x+q_{22}.$ (23b)

Положим

$q_1 = q_{11} – q_{12} – q_{21} + q_{22},$ (24)

$q_2 = q_{22}–q_{12},$ (25)

$q_3 = q_{22}–q_{21}.$ (26)

Имеетместо

Утверждение 11. Пусть в игре с платёжной матрицей (18) нет решений в чистых стратегиях. Тогда

1) $q_1 \neq 0, q_2 \neq 0, q_3 \neq 0;$ (27)

2) $\mathrm{sgn}(q_2) = \mathrm{sgn}(q_3);$ (28)

3) $0 < q_2/q_1 < 1, 0 < q_3/q_1 < 1$ ■ (29)

Все три части утверждения 11 являются следствиями утверждений 8 – 10. В частности, вторые неравенства в формулах (29) следуют из того, что $q_1 = q_{11} – q_{21}+ q_2 = q_{11} – q_{12} + q_3$ и утверждения 10.

Перейдём к явному построению решения игры в смешанных стратегиях. Начнём с игрока A. Для любой стратегии y игрока B положим

$\varphi_A(y) = \{x \mid (\forall x') (g_A(x', y) \leq g_A(x, y))\}.$ (30a)

Другими словами, $\varphi_A(y)$ – это <u>множество</u> всех стратегий x игрока A, которые максимизирует его выигрыш $g_A(x, y)$ при данном y.

Положим

$\Phi_A = \{(x, y) \mid y\in[0, 1], x\in\varphi_A(y)\}.$ (31a)

Множество Φ_A назовём ***графиком игрока*** A (см. общее определение графика в гл. 1.3)

Поскольку при любом фиксированном y функция (21a) линейна по x, и при этом $0 \leq x \leq 1$, то максимум достигается при $m \neq 0$ на концах отрезка [0, 1] (т.е. при одной из двух чистых стратегий), а при $m = 0$ – при всех стратегиях, поскольку в этом случае $g(x, y)$ просто не зависит от x. Представим эти рассуждения геометрически, нарисовав график Φ_A игрока A. Из (22a) и (23a) следует, что

$g(x, 0) = –x\cdot q_2 + q_{22}.$

Поэтому

$$\varphi(0) = \begin{cases}\{0\}, \text{если} q_2 > 0\\ \{1\}, \text{если} q_2 < 0\end{cases}.$$ (32a)

Далее, представим (22a) с учётом обозначений (24) – (26) как

$m = q_1\cdot y– q_2.$

Будем увеличивать y от 0 до 1. В силу 3-ей части утверждения 11 q_1 и q_2 имеют одинаковые знаки. Поэтому при $y= y^* = q_2/q_1$ число $m = 0$, $g(x, y^*)$ от x не зависит и, следовательно, множеством $\varphi_A(y^*)$ является отрезок [0, 1]. Далее, при $y>y^*$ число m меняет знак на противоположный. Поэтому, если $\varphi(0) = \{0\}$, то $\varphi(1) = \{1\}$; если же $\varphi(0) = \{1\}$, то $\varphi(1) = \{0\}$. Оба случая (при $q_2< 0$ и при $q_2 > 0$) представлены на рис.1.

Теперь перейдём к игроку B. Для любой стратегии x игрока A положим

$\psi_B(x) = \{y \mid (\forall y') (g_B(x, y) \leq g_B(x, y'))\}.$ (30b)

Другими словами, $\psi_B(x)$ – это множество всех стратегий y игрока B, которые минимизируют его выигрыш $g_B(x, y)$ при данном x.

Определим ***график игрока*** B формулой, аналогичной формуле (31a):

$\Psi_B = \{(x, y) \mid x\in[0, 1], y\in\psi_B(x)\}.$ (31b)

Далее практически повторим все рассуждения относительно графика Φ_A применительно к графику Ψ_B (учитывая минимизацию, а не максимизацию). В частности, получим

$$\psi(0) = \begin{cases}\{1\}, \text{если} q_3 < 0\\ \{0\}, \text{если} q_3 > 0\end{cases}$$ (32b)

(различие (32a) и (32b) определяется минимизацией вместо максимизации).

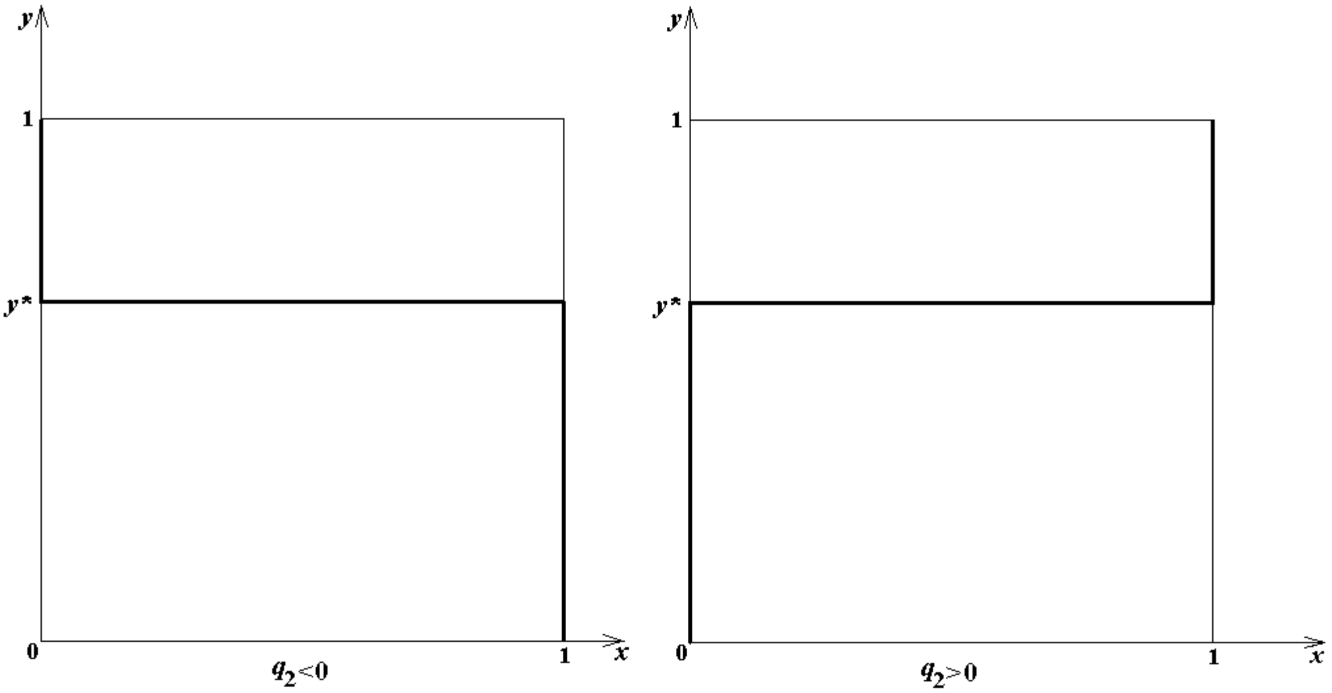


Рис.1

Как и выше, положим $x^* = q_3/q_1$. При $x = x^*$ коэффициент sпри y в (21b) обращается в 0, так что минимум достигается при любом y от 0 до 1; в этой точке знак sменяется и, соответственно, $\psi(x)$ меняет значение с $\{0\}$ на $\{1\}$ или наоборот. Оба случая представлены на рис.2 в тех же координатах, что и на рис.1.

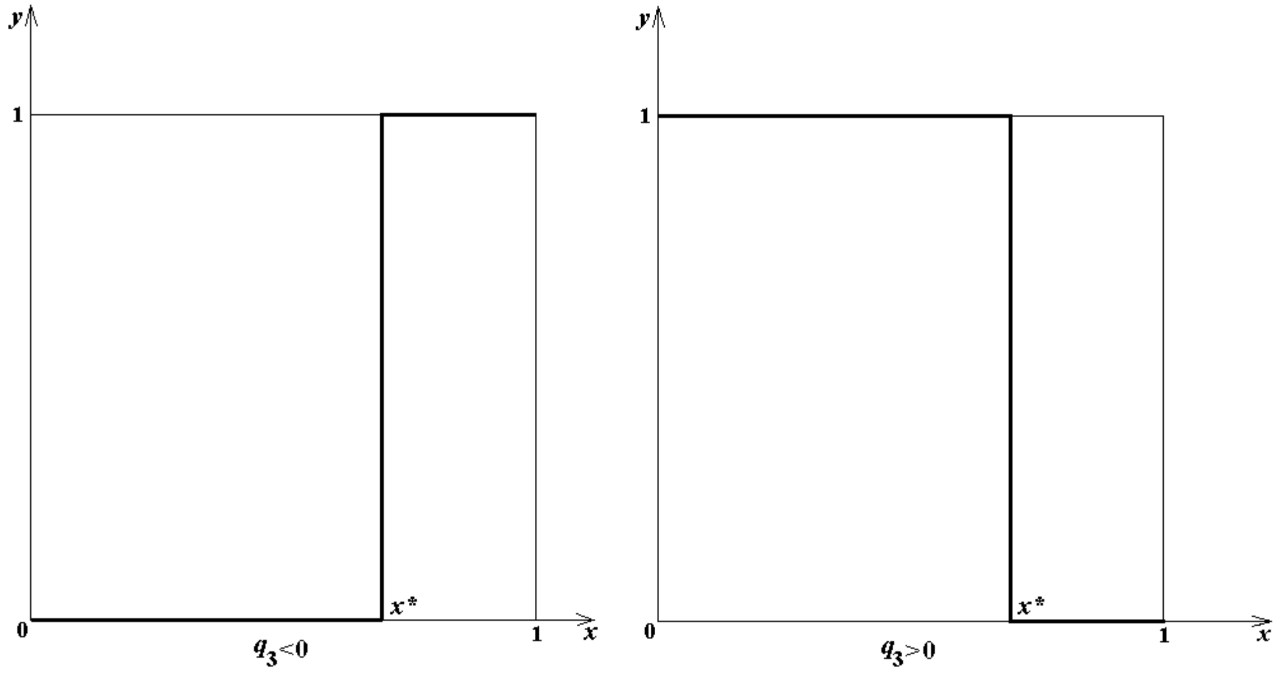


Рис.2

В силу формулы (28), $q_2< 0$ тогда и только тогда, когда $q_3< 0$. Поэтому при изображении обоих графиков в одних и тех же координатах имеется два случая: $q_2< 0$, $q_3< 0$ и $q_2> 0$, $q_3> 0$. Других вариантов не может быть. Эти графики показаны на рис. 3. Из формулы (29) следует, что q_1 имеет тот же знак, что q_2 иq_3. Поэтому левой и правой части рис.3 соответствуют $q_1< 0$ и $q_1> 0$.

Суть дела состоит в том, что найденные смешанные стратегии x^* и y^* образуют решение исходной матричной игры с платёжной матрицей (18), не имеющей решений в чистых стратегиях. Действительно, по построению $\varphi_A(y)$ (см. формулу (30a)) для любого yи для любого $x'\in\varphi_A(y)$ выполняется условие

$$(\forall x)g(x, y) \le g(x', y). \tag{33a}$$

В частности, (33a) верно при $y = y^*$ и $x' = x^*$, так как при $y = y^*$ x' можно выбрать произвольно.

Поэтому из (33a) следует, что для любого x
$g(x, y^*) \le g(x^*, y^*)$. (34a)
Аналогично, по построению $\psi_B(x)$ для любого x и для любого $y' = \psi(x)$ выполняется условие
$(\forall y)g(x, y') \le g(x, y)$. (33b)
В частности, (33b) верно при $y' = y^*$ и $x = x^*$, так как при $x = x^*$ y' можно выбрать произвольно.

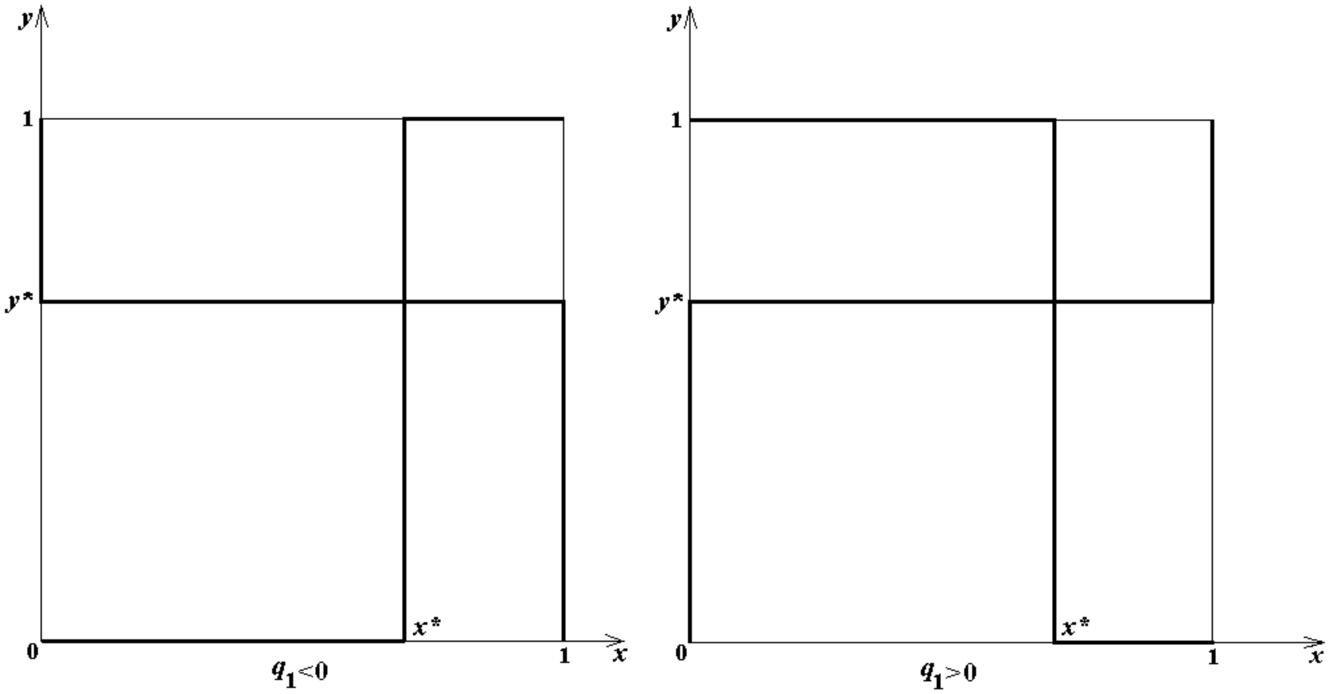


Рис.3

Поэтому из (33b) следует, что для любого y
$g(x^*, y^*) \le g(x^*, y)$. (34b)
Неравенства (34a) и (34b) вместе совпадают с двойным неравенством (12), т.е. пара стратегий $\langle x^*, y^*\rangle$ по определению является решением игры в смешанных стратегиях.

Если «принять на веру» утверждения 8 – 11 (доказательства которых достаточно просты), то вместе с предыдущими рассуждениями они доказывают существование решения в смешанных стратегиях любой игры размерности 2×2 для произвольной платёжной матрицы (18). Конечно, этот факт сам по себе непосредственно следует из общего утверждения 6. Однако оно является типичной «теоремой существования», не объясняющей, как находить само решение.

Для игр размерности 2×2 ситуация другая. Числа x^* и y^* определены в явном виде:
$x^* = q_3/q_1$, $y^* = q_2/q_1$,
где числа q_1, q_2 и q_3 выражаются через элементы исходной платёжной матрицы (18) по формулам (24) – (26). Переходя обратно от задания смешанной стратегии одним числом (вероятностью выбора 1-ой стратегии) к её заданию парами чисел (см. текст после формулы (20)), окончательно получаем

$$x_1^* = \frac{q_{22}-q_{21}}{q_{11}-q_{12}-q_{21}+q_{22}}, x_2^* = 1-x_1^*,\ x^* = (x_1^*, x_2^*); \qquad (35a)$$

$$y_1^* = \frac{q_{22}-q_{12}}{q_{11}-q_{12}-q_{21}+q_{22}}, y_2^* = 1-y_1^*,\ y^* = (y_1^*, y_2^*). \qquad (35b)$$

При взгляде на формулы (35) возникает вопрос – что делать, если входящий в них делитель (24) окажется равным 0? Ведь на 0 делить нельзя! При этом число q_1, определяемое формулой (24), действительно может быть равным 0 – например, в платёжной матрице $\begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix}$. Однако для платёжных матриц без седловой точки (а только такие матрицы здесь и рассматриваются) выражение (24) не равно 0 (см. 1-ую часть утверждения 11). Так что на 0 делить не придётся.

Таким образом, полностью решена задача нахождения решения матричной игры размерности 2×2 для произвольной платёжной матрицы (18). Если у игры есть решение в чистых стратегиях, то оно и является искомым решением. В том и только том случае, если такого решения нет, можно найти решение в смешанных стратегиях по формулам (35).

Пример 7. Рассмотрим матричную игру двух лиц со следующей платёжной матрицей: $Q=\begin{pmatrix} 6 & 2 \\ 5 & 8 \end{pmatrix}$. Найдём её решение. Поскольку у данной игры нет решения в чистых стратегиях, то воспользуемся формулами (35), положив $q_{11} = 6$, $q_{12} = 2$, $q_{21} = 5$, $q_{22} = 8$. Получим

$x_1^* = \dfrac{q_{22}-q_{21}}{q_{11}-q_{12}-q_{21}+q_{22}} = \dfrac{8-5}{6-2-5+8} = \dfrac{3}{7}$; $y_1^* = \dfrac{q_{22}-q_{12}}{q_{11}-q_{12}-q_{21}+q_{22}} = \dfrac{8-2}{6-2-5+8} = \dfrac{6}{7}$;

$x_2^* = 1-x_1^* = 1-\dfrac{3}{7} = \dfrac{4}{7}$; $y_2^* = 1-y_1^* = 1-\dfrac{6}{7} = \dfrac{1}{7}$; $x^* = (\frac{3}{7}, \frac{4}{7})$; $y^* = (\frac{6}{7}, \frac{1}{7})$ ■

5.2. Игра размерности 2×*n*. Рассмотрим произвольную матричную игру со следующей платёжной матрицей размерности $2\times n$ общего вида:

$$Q = \begin{pmatrix} q_{11} & q_{12} & \cdots & q_{1n} \\ q_{21} & q_{22} & \cdots & q_{2n} \end{pmatrix}. \tag{36}$$

Оказывается, что для игр с платёжной матрицей вида (36) решение может быть найдено графо-аналитическим методом. Для простоты решение излагается для $n = 4$ в следующих двух примерах.

Пример 8. Найдём решение и цену игры с платёжной матрицей $Q = \begin{pmatrix} 2 & 4 & 6 & 3 \\ 4 & 8 & 4 & 6 \end{pmatrix}$. Прежде всего проверим наличие (или отсутствие) решения в чистых стратегиях. В данном случае
$\max_i\min_j q_{ij}=\max\{2, 4\} = 4$; $\min_j\max_i q_{ij} = \min\{4, 8, 6, 6\} = 4$.
Так как $\max_i\min_j q_{ij}=\min_j\max_i q_{ij} ==q_{21}$, то пара чистых стратегий $\langle 2, 1\rangle$ является решением данной игры. Как и в примере 6, представим эти чистые стратегии как смешанные: $x = (0, 1)$; $y = (1, 0, 0, 0)$. Ценой игры является общее значение максимина и минимакса, равное $q_{21} = 4$ ■

Пример 9. Найти решение и цену игры с платёжной матрицей $Q = \begin{pmatrix} 6 & 4 & 6 & 3 \\ 5 & 8 & 4 & 6 \end{pmatrix}$. В этом случаета же самаяпроверка даёт
$\max_i\min_j q_{ij} = \max\{3, 4\} = 4$; $\min_j\max_i q_{ij} = \min\{6, 8, 6, 6\} = 6$.
Поскольку в данном случае $\max_i\min_j q_{ij} < \min_j\max_i q_{ij}$ то решений в чистых стратегиях нет.
Поэтому можно и нужно искать решение игры в смешанных стратегиях описанным ниже графо-аналитическим методом.

Игрок A имеет только 2 чистых стратегии. Его задача состоит в максимизации своего выигрыша в зависимости от своей смешанной стратегии (x_1, x_2)

$$v(x_1, x_2) = \min_{j=1,2,3,4} (q_{1j}\, x_1 + q_{2j}\, x_2), \tag{37}$$

где минимум берётся по 4-ём чистым стратегиям игрока B.
Так как $x_2 = 1 - x_1$, то, обозначая x_1 через x, из (40) получаем выражение

$$v(x) = \min_{j=1,2,3,4} \{(q_{1j}-q_{2j})\, x + q_{2j}\}. \tag{38}$$

Таким образом, $v(x)$ является минимумом четырёх линейных функций одной переменной x; можно начертить графики этих функций и затем максимизировать их минимум $v(x)$ графическим методом.

Нетрудно построить графики функций $(q_{1j}-q_{2j})\, x + q_{2j}$, если заметить, что они должны проходить через точки $(0, q_{2j})$ и $(1, q_{1j})$. Эти графики изображены на рис.4. В данном случае имеем четыре прямых линии, нижняя огибающая которых $v(x)$ выделена жирным. Высшая точка функции $v(x)$ находится на пересечении прямых $y = 2x + 4$ и $y = -3x + 6$, соответствующих 3-му и 4-му столбцам матрицы Q (напомним, что j-ая прямая по построению проходит через точки $(0, q_{2j})$ и $(1, q_{1j})$). Поскольку в данном случае ломаная $v(x)$ состоит только из двух звеньев, соединённых в точке D, то указанные два столбца находятся сразу. Если же точек, где соединяются звенья ломаной, несколько, то надо взять ту из них, у которой ордината максимальна. Именно нахождение такой точки и делается графически. Большой точности в построениях не требуется, поскольку всё, что нужно – это найти те два столбца, которые соответствуют двум линиям, определяющим эту точку.

Далее для вычисления оптимальных смешанных стратегии надо рассмотреть 2×2 матрицу P, образованную найденными выше двумя столбцами. В данном случае это 3-ий и 4-ый столбцы, а сама матрица имеет вид:

$P = \begin{pmatrix} 6 & 3 \\ 4 & 6 \end{pmatrix}$

(см. исходную матрицу Q). Для этой матрицы найдём решение в смешанных стратегиях способом, подробно описанном в примере 7. Имеем в данном случае q_{11}= 6, q_{12}= 3,q_{21}= 4, q_{22}= 6. Поэтому в силу формул (35)

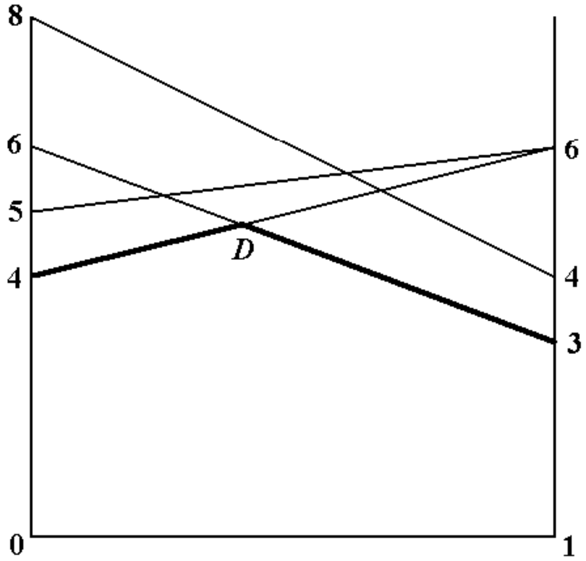


Рис.4

$x_1^* = \frac{q_{22}-q_{21}}{q_{11}-q_{12}-q_{21}+q_{22}} = \frac{6-4}{6-3-4+6}$=0,4;$y_1^* = \frac{q_{22}-q_{12}}{q_{11}-q_{12}-q_{21}+q_{22}} = \frac{6-3}{6-3-4+6}$=0,6;

x_2^* = 1–x_1^* = 1 – 0,4 =0,6; y_2^* = 1–y_1^* = 1 –0,6 = 0,4.

Теперь вспоминаем, что в исходной матрице размера 2×4 у игрока A есть две чистых стратегии: выбор 1-й или 2-й строки. В данном случае имеем

x^* = (0,4; 0,6).

У игрока B есть четыре чистых стратегии, соответствующие выбору одного из 4-ёх столбцов заданной матрицы. Однако активными (т.е. отличными от 0) являются только те две стратегии, которые соответствуют выбранным (с помощью рисунка) столбцам. В нашем случае это столбцы 3 и 4. Вероятность выбора 3-его и 4-го столбца была найдена выше: 0,6 и 0,4. Сама оптимальная стратегия игрока B имеет вид

y^* = (0; 0; 0,6; 0,4).

Цена игры V с исходной матрицей Q совпадает с ценой игры с матрицей P. В силу 3-ьей части утверждения 5 и формулы (9) получаем

$V = x_1^*y_1^*\cdot 6 + x_1^*y_2^*\cdot 3 + x_2^*y_1^*\cdot 4 + x_2^*y_2^*\cdot 6 = 0{,}4\cdot 0{,}6\cdot 6 + 0{,}4\cdot 0{,}4\cdot 3 + 0{,}6\cdot 0{,}6\cdot 4 + 0{,}6\cdot 0{,}4\cdot 6$ = 1,44 + 0,48 + 1,44 + 1,44 = 4,8 ■

Во многих случаях удаётся упростить описанный выше поиск решения матричных игр с платёжной матрицей размерности 2×n за счёт предварительного упрощения платёжной матрицы способами, описанными в разделе 3. Проиллюстрируем это в следующем примере.

Пример 10. Рассмотрим матричную игру двух лиц со следующей платёжной матрицей размера 2×6: $Q = \begin{pmatrix} 6 & 5 & 3 & 7 & 1 & 3 \\ 7 & 2 & 4 & 2 & 4 & 3 \end{pmatrix}$. В данной матрице 1-ый, 3-ий и 4-ый столбцы доминируемы 2-ым, 5-ым и снова 2-ым столбцами соответственно. После их удаления остаётся матрица $P = \begin{pmatrix} 5 & 1 & 3 \\ 2 & 4 & 3 \end{pmatrix}$, которая больше не упрощается. Номера её столбцов в исходной матрице таковы: (2, 5, 6). Далее применим описанный в примере 9 графо-аналитический метод поиска решения к матрице P. Входящие в формулу (38) три линейные функции должны проходить через точки (0, q_{2j}) и (1, q_{1j}) (j = 1, 2, 3). С учётом матрицы Pэти точки таковы:

(0, 2) и (1, 5); (0, 4) и (1, 1); (0, 3) и (1, 3).

Изобразим соответствующие отрезки на рис.5. Функция $v(x)$, определённая формулой (38), показана жирной линией. Эта линия состоит из двух отрезков, пересекающихся в точке D. Сами отрезки соответствуют 1-му и 2-му столбцам матрицы *P*. Матрица *R*, образованная этими столбцами, такова: $R = \begin{pmatrix} 5 & 1 \\ 2 & 4 \end{pmatrix}$. Аналогично тому, как это делалось в примере 9, находим

$$x_1^* = \frac{q_{22}-q_{21}}{q_{11}-q_{12}-q_{21}+q_{22}} = \frac{4-2}{5-1-2+4} = 0{,}33; \quad y_1^* = \frac{q_{22}-q_{12}}{q_{11}-q_{12}-q_{21}+q_{22}} = \frac{4-1}{5-1-2+4} = 0{,}5;$$

$x_2^* = 1 - x_1^* = 1 - 0{,}33 = 0{,}67$; $y_2^* = 1 - y_1^* = 1 - 0{,}5 = 0{,}5$.

Поскольку рассмотренные два столбца были 2-ым и 5-ым в исходной матрице *Q*, получаем $x^* = (0{,}33;\ 0{,}67)$ и $y^* = (0;\ 0{,}5;\ 0;\ 0;\ 0{,}5;\ 0)$ ■

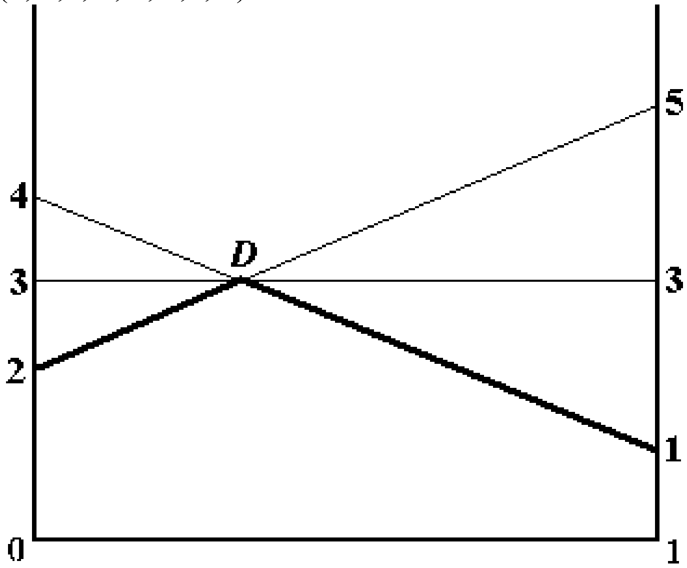


Рис.5

6. Задания

Задание 1. В игре с данной платёжной матрицей найти оптимальные стратегии обоих игроков, нижнюю и верхнюю цены игры. Указать решение игры, если оно есть; в противном случае явно указать на его отсутствие. В качестве образца см. примеры 1 и 2.

Варианты платёжных матриц для задания 1:

15	20	25	19	17
15	21	18	30	19
27	10	19	20	25

01

20	25	15	25	10
19	20	9	27	14
37	20	19	19	15

02

14	16	30	25	18
19	18	15	24	15
10	27	26	6	-99

03

10	27	15	34	21
18	15	30	20	15
19	25	23	13	29

04

21	19	12	14	17
25	34	18	19	25
20	12	14	29	10

05

17	12	10	8	22
22	28	18	38	33
33	14	17	21	13

06

17	8	23	15	24
32	20	26	22	21
23	19	37	11	28

07

11	18	14	26	12
27	10	12	13	21
13	21	35	23	29

08

10	11	12	23	27
12	28	25	20	14
11	22	17	19	10

09

10	27	37	14	30
12	14	27	12	22
16	24	14	25	24

10

22	15	24	234	-23
24	22	26	22	15
14	0	13	24	22

11

25	19	16	25	19
27	10	8	18	30
15	20	30	15	21

12

16	8	15	20	30
18	24	22	18	965
12	21	30	25	15

13

25	19	21	14	25
18	30	19	15	20
30	25	30	15	210

14

8	13	24	24	16
30	15	27	26	8
22	22	31	13	30

15

27	12	14	25	8
14	25	10	27	20
22	18	12	14	17

16

15	20	25	19	17
15	21	18	30	19
27	10	19	20	25

17

20	25	15	25	10
19	20	9	27	14
37	20	19	19	15

18

14	16	30	25	18
-19	18	15	24	15
10	27	26	6	-90

19

30	27	15	34	21
18	15	30	20	15
19	25	23	13	29

20

21	19	12	14	17
25	14	18	19	25
20	12	14	29	10

21

17	12	10	80	22
22	28	18	38	33
33	13	17	21	13

22

17	8	27	15	24
32	20	26	22	21
23	19	37	11	28

23

11	18	14	26	12
27	10	12	13	21
13	21	35	27	29

24

10	11	12	13	27
12	28	25	20	14
11	22	17	19	30

25

10	27	15	34	21
18	-15	30	20	15
19	25	23	-13	29

26

21	-19	12	14	17
25	34	-18	19	25
20	12	14	-29	10

27

16	8	15	20	30
18	24	12	18	965
12	21	-30	25	15

28

25	19	21	14	25
18	30	-19	15	20
30	25	30	15	10

29

8	-13	24	24	16
30	15	-27	26	48
22	22	31	13	30

30

Задание 2. Удалить доминируемые стратегии в данной матрице. В качестве образца см. примеры 3 и 4.

Варианты платёжных матриц для задания 2:

20	25	15	25	10
19	20	9	27	14
37	20	19	19	15

01

14	16	30	25	18
19	18	15	24	15
10	27	26	6	-99

02

17	12	10	8	22
22	28	18	38	33
33	14	17	21	13

03

17	8	23	15	24
32	20	26	22	21
23	19	37	11	28

04

10	11	12	23	27
12	28	25	20	14
11	22	17	19	10

05

27	12	14	25	8
14	25	10	27	20
22	18	12	14	17

06

10	27	37	14	30
12	14	27	12	22
16	24	14	25	24

07

22	15	24	234	-23
24	22	26	22	15
14	0	13	24	22

08

25	19	16	25	19
27	10	8	18	30
15	20	30	15	21

09

16	8	15	20	30
18	24	22	18	965
12	21	30	25	15

10

25	19	21	14	25
18	30	19	15	20
30	25	30	15	210

11

8	13	24	24	16
30	15	27	26	8
22	22	31	13	30

12

20	25	15	25	10
19	20	9	27	14
37	20	19	19	15

13

14	16	30	25	18
-19	18	15	24	15
10	27	26	6	-90

14

21	19	12	14	17
25	14	18	19	25
20	12	14	29	10

15

17	12	10	80	22
22	28	18	38	33
33	13	17	21	13

16

17	8	27	15	24
32	20	26	22	21
23	19	37	11	28

17

11	18	14	26	12
27	10	12	13	21
13	21	35	27	29

18

10	11	12	13	27
12	28	25	20	14
11	22	17	19	30

19

10	27	15	34	21
18	-15	30	20	15
19	25	23	-13	29

20

21	-19	12	14	17
25	34	-18	19	25
20	12	14	-29	10

21

16	8	15	20	30
18	24	12	18	965
12	21	-30	25	15

22

25	19	21	14	25
18	30	-19	15	20
30	25	30	15	10

23

8	-13	24	24	16
30	15	-27	26	48
22	22	31	13	30

24■

Задание 3. Найти решения в данной матричной игре размера 2×2. В качестве образца см. пример 7.

Варианты платёжных матриц для задания 3:

$\begin{pmatrix}3&2\\2&7\end{pmatrix}$ 01	$\begin{pmatrix}3&2\\1&4\end{pmatrix}$ 02	$\begin{pmatrix}3&4\\5&1\end{pmatrix}$ 03	$\begin{pmatrix}7&0\\4&6\end{pmatrix}$ 04	$\begin{pmatrix}1&7\\3&2\end{pmatrix}$ 05	$\begin{pmatrix}4&0\\2&5\end{pmatrix}$ 06	$\begin{pmatrix}6&9\\3&4\end{pmatrix}$ 07
$\begin{pmatrix}1&4\\2&1\end{pmatrix}$ 08	$\begin{pmatrix}1&9\\6&3\end{pmatrix}$ 09	$\begin{pmatrix}1&9\\8&2\end{pmatrix}$ 10	$\begin{pmatrix}0&8\\7&4\end{pmatrix}$ 11	$\begin{pmatrix}3&9\\5&1\end{pmatrix}$ 12	$\begin{pmatrix}1&8\\2&1\end{pmatrix}$ 13	$\begin{pmatrix}2&0\\1&8\end{pmatrix}$ 14
$\begin{pmatrix}3&1\\5&4\end{pmatrix}$ 15	$\begin{pmatrix}5&2\\2&7\end{pmatrix}$ 16	$\begin{pmatrix}3&2\\0&7\end{pmatrix}$ 17	$\begin{pmatrix}7&4\\5&3\end{pmatrix}$ 18	$\begin{pmatrix}7&3\\2&6\end{pmatrix}$ 19	$\begin{pmatrix}1&6\\4&3\end{pmatrix}$ 20	$\begin{pmatrix}3&0\\2&5\end{pmatrix}$ 21
$\begin{pmatrix}3&1\\0&4\end{pmatrix}$ 22	$\begin{pmatrix}6&9\\8&5\end{pmatrix}$ 23	$\begin{pmatrix}1&9\\5&4\end{pmatrix}$ 24	$\begin{pmatrix}1&9\\9&2\end{pmatrix}$ 25	$\begin{pmatrix}1&3\\4&2\end{pmatrix}$ 26	$\begin{pmatrix}6&2\\5&8\end{pmatrix}$ 27	$\begin{pmatrix}0&2\\7&0\end{pmatrix}$ 28■

Задание 4. Найти решения в данной матричной игре размера 2×4. В качестве образца см. примеры 8 и 9.

Варианты платёжных матриц для задания 4:

$\begin{pmatrix}1&9&4&6\\4&3&2&8\end{pmatrix}_{01}$	$\begin{pmatrix}4&7&5&1\\1&2&3&4\end{pmatrix}_{02}$	$\begin{pmatrix}1&7&2&9\\2&2&5&4\end{pmatrix}_{03}$	$\begin{pmatrix}4&3&2&1\\2&1&3&4\end{pmatrix}_{04}$	$\begin{pmatrix}1&9&2&1\\2&5&3&4\end{pmatrix}_{05}$
$\begin{pmatrix}4&3&3&4\\3&9&1&2\end{pmatrix}_{06}$	$\begin{pmatrix}2&8&1&5\\3&4&7&2\end{pmatrix}_{07}$	$\begin{pmatrix}3&5&4&2\\1&2&5&4\end{pmatrix}_{08}$	$\begin{pmatrix}4&7&5&1\\1&2&6&4\end{pmatrix}_{09}$	$\begin{pmatrix}1&9&4&6\\4&3&7&4\end{pmatrix}_{10}$
$\begin{pmatrix}5&9&4&6\\4&3&2&8\end{pmatrix}_{11}$	$\begin{pmatrix}4&7&5&6\\1&2&3&4\end{pmatrix}_{12}$	$\begin{pmatrix}1&7&2&9\\2&6&5&4\end{pmatrix}_{13}$	$\begin{pmatrix}4&3&2&1\\2&7&3&4\end{pmatrix}_{14}$	$\begin{pmatrix}1&9&2&1\\7&5&3&4\end{pmatrix}_{15}$
$\begin{pmatrix}4&3&3&4\\3&9&8&2\end{pmatrix}_{16}$	$\begin{pmatrix}2&8&1&5\\7&4&7&2\end{pmatrix}_{17}$	$\begin{pmatrix}3&5&4&2\\6&2&5&4\end{pmatrix}_{18}$	$\begin{pmatrix}4&7&5&1\\8&2&3&4\end{pmatrix}_{19}$	$\begin{pmatrix}1&9&4&6\\4&3&2&4\end{pmatrix}_{20}$
$\begin{pmatrix}1&6&4&6\\4&3&2&8\end{pmatrix}_{21}$	$\begin{pmatrix}2&7&5&1\\1&2&3&4\end{pmatrix}_{22}$	$\begin{pmatrix}1&7&2&3\\2&2&5&4\end{pmatrix}_{23}$	$\begin{pmatrix}4&3&2&8\\2&1&3&4\end{pmatrix}_{24}$	$\begin{pmatrix}1&9&2&6\\2&5&3&4\end{pmatrix}_{25}$
$\begin{pmatrix}4&9&4&6\\1&3&2&8\end{pmatrix}_{26}$	$\begin{pmatrix}4&2&5&1\\1&7&3&4\end{pmatrix}_{27}$	$\begin{pmatrix}1&7&5&9\\2&2&2&4\end{pmatrix}_{28}$	$\begin{pmatrix}4&3&2&4\\2&1&3&1\end{pmatrix}_{29}$	$\begin{pmatrix}1&9&2&4\\2&5&3&1\end{pmatrix}_{30}$
$\begin{pmatrix}4&3&1&4\\3&9&3&2\end{pmatrix}_{31}$	$\begin{pmatrix}2&4&1&5\\3&8&7&2\end{pmatrix}_{32}$	$\begin{pmatrix}1&5&4&2\\3&2&5&4\end{pmatrix}_{33}$	$\begin{pmatrix}1&7&5&1\\4&2&6&4\end{pmatrix}_{34}$	$\begin{pmatrix}1&3&4&6\\4&9&7&4\end{pmatrix}_{35}$
$\begin{pmatrix}5&9&2&6\\4&3&4&8\end{pmatrix}_{36}$	$\begin{pmatrix}4&7&5&4\\1&2&3&6\end{pmatrix}_{37}$	$\begin{pmatrix}1&7&2&4\\2&6&5&9\end{pmatrix}_{38}$	$\begin{pmatrix}4&3&3&1\\2&7&2&4\end{pmatrix}_{39}$	$\begin{pmatrix}1&5&2&1\\7&9&3&4\end{pmatrix}_{40}$
$\begin{pmatrix}3&3&3&4\\4&9&8&2\end{pmatrix}_{41}$	$\begin{pmatrix}2&4&1&5\\7&8&7&2\end{pmatrix}_{42}$	$\begin{pmatrix}3&5&5&2\\6&2&4&4\end{pmatrix}_{43}$	$\begin{pmatrix}4&7&5&4\\8&2&3&1\end{pmatrix}_{44}$	$\begin{pmatrix}1&9&4&4\\4&3&2&6\end{pmatrix}_{45}$
$\begin{pmatrix}1&6&2&6\\4&3&4&8\end{pmatrix}_{46}$	$\begin{pmatrix}2&2&5&1\\1&7&3&4\end{pmatrix}_{47}$	$\begin{pmatrix}2&7&2&3\\1&2&5&4\end{pmatrix}_{48}$	$\begin{pmatrix}4&2&5&1\\8&7&3&4\end{pmatrix}_{49}$	$\begin{pmatrix}1&9&3&6\\2&5&2&4\end{pmatrix}_{50}$■

Задание 5. Упростить платёжную матрицу и найти решение и цену игры с упрощённой или исходной (если не упрощается) платёжной матрицей. В качестве образца см. пример 10

Варианты платёжных матриц те же, что в задании 4 ■

6. Предметный указатель

Глава 12. Другие игровые модели

1. Биматричные игры
2. Позиционные игры
3. Задания
4. Предметный указатель

В этой главе будут рассмотрены несколько различных конфликтных ситуаций, для анализа которых используется общий термин – игра. При всех различиях общим для моделируемых ситуаций является

1) наличие двух или более участников (игроков) с несовпадающими, хотя и не обязательно противоположными, интересами;

2) формальное описание взаимодействия игроков, т.е. правил игры;

3) зависимость выигрыша каждого игрока от стратегий (действий, «ходов») остальных игроков.

Из большого числа разнообразных классов игр рассмотрены – на самом простом уровне – два из них: биматричные и позиционные игры. Описания указанных классов игр и их решений (в том или ином смысле) приводятся в соответствующих разделах данной главы.

1. Биматричные игры

Из многих видов игр ***биматричные игры*** в наибольшей степени похожи на рассмотренные в главе 11 матричные игры (см. раздел 11-1). Как и там, в игре участвуют два игрока А и В, у каждого из которых имеется конечное число стратегий. Игрок выбирает одну из своих стратегий независимо от другого игрока. После того, как оба выбора сделаны, каждый игрок получает выигрыш или платит проигрыш, которые зависят от обеих выбранных стратегий. В отличие от матричной игры, здесь не предполагается, что выигрыш одного игрока равен проигрышу другого. Таким образом, не предполагается, что биматричная игра является игрой с нулевой суммой. Вопрос, кто платит выигрыш и кто получает проигрыш, несмотря на свою важность, в рамках данной модели не рассматривается (организатор игры? рынок? начальство?....). Важно, что соблюдаются правила, т.е. игроки получают выигрыши и платят проигрыши.

В силу конечности множеств стратегий у обоих игроков без ограничения общности можно считать, что сами стратегии задаются натуральными числами от 1 до m у игрока А и от 1 до n у игрока В. Поэтому выигрыш игрока А можно задавать ***платёжной матрицей*** A размера $m\times n$, а выигрыш игрока В – ***платёжной матрицей*** B того же размера. Если игроки выбрали стратегии $i\in\{1, ..., m\}$ и $j\in\{1, ..., n\}$, то выигрыш игрока А равен числу a_{ij} – элементу матрицы A, стоящему на пересечении её i-ой строки и j-го столбца, а выигрыш игрока В равен числу b_{ij} – элементу матрицы B, стоящему на пересечении её i-ой строки и j-го столбца.

Пример 1. ***Игра Аумана***. Дадим описание этой игры. Каждый участник может обратиться к организатору игры с одной из двух возможных просьб, которая будет обязательно выполнена. Просьбы таковы:

1. Дать сто долларов другому игроку;
2. Дать один доллар мне.

Выбор просьбы и есть выбор стратегии. Запишем обе платёжные матрицы:

$A = \begin{pmatrix} 100 & 0 \\ 101 & 1 \end{pmatrix},\ \ B = \begin{pmatrix} 100 & 101 \\ 0 & 1 \end{pmatrix}.$

Проанализируем все 4 возможности. Если оба игрока выбирают 1-ые стратегии, то у каждого будет $ 100, что и выражено числом 100 в верхнем левом углу обеих матриц. Если игрок А выбрал 1-ую стратегию, а игрок В – 2-ую стратегию, то игрок А не получает ничего, а игрок В –$ 101 (число a_{12} = 0, а число b_{12} = 101). Если же игрок А выбрал 2-ую стратегию, а игрок В – 1-ую стратегию, то число a_{21} = 101, а число b_{21} = 0. Наконец, если оба игрока выбирают 2-ые стратегии, то у каждого будет $ 1, что и выражено числом 1 в правом нижнем углу обеих матриц.

В данной игре, как и во многих других биматричных играх, более наглядным является представление двух платёжных матриц в виде одной матрицы с парой чисел в каждой позиции i, j, где 1-ое (2-ое) число равно выигрышу игрока А (В) при выборе ими стратегий i и j. В данном случае такая матрица принимает вид:

$$\begin{pmatrix} (100,100) & (0,101) \\ (101,0) & (1,1) \end{pmatrix} \blacksquare \qquad (1)$$

1.1. Равновесие Нэша. Рассмотренные в главе 11 матричные игры можно рассматривать как частный случай биматричных игр, в которых $B = -A$. Введём для биматричных игр понятие ***равновесия Нэша***. Оно не только является обобщением понятия решения игры для матричных игр, но и является одним из центральных понятий теории игр в самом широком смысле. Содержательно равновесием Нэша является такой набор стратегий разных игроков, при котором никому <u>не выгодно</u> менять свою стратегию, если <u>все остальные</u> игроки свои стратегии <u>не меняют</u>. Напомним, что то же самое верно для решений матричных игр (см. текст между утверждением 1 и примером 1 в разделе 11-2).

Дадим формальное определение равновесия Нэша (в чистых стратегиях) для рассматриваемого класса биматричных игр. Пара стратегий $\langle i^*, j^*\rangle$ называется равновесием Нэша, если

$$(\forall i)(\forall j)\ ((a_{ij^*} \le a_{i^*j^*}) \wedge (b_{i^*j} \le b_{i^*j^*})). \qquad (2)$$

Другими словами, если игрок А воспользуется вместо стратегии i^* любой другой стратегией i, а игрок В воспользуется стратегией j^*, то в силу неравенства $a_{ij^*} \le a_{i^*j^*}$ игрок А не может выиграть больше, чем при использовании «равновесной» стратегииi^*. То же самое верно и для игрока В (в силу неравенства $b_{i^*j} \le b_{i^*j^*}$). Именно потому, что никто не будет отступать от своей стратегии, такие пары называются равновесиями, а входящие в них стратегии – ***равновесными***.

Пример 2. Рассмотрим важный частный случай, когда у каждого игрока есть две стратегии. Таким образом, всего имеется 4 пары чистых стратегий: $\langle 1, 1\rangle$, $\langle 1, 2\rangle$, $\langle 2, 1\rangle$, $\langle 2, 2\rangle$. Поскольку каждый игрок может поменять свою стратегию 1 или 2 только на одну стратегию (2 или 1), то условия равновесия (2) записываются только для одного $i \ne i^*$ и одного $j \ne j^*$ для каждой из этих пар:

для $\langle 1, 1\rangle$: $(a_{21} \le a_{11}) \wedge (b_{12} \le b_{11})$; (2.11)

для $\langle 1, 2\rangle$: $(a_{22} \le a_{12}) \wedge (b_{11} \le b_{12})$; (2.12)

для $\langle 2, 1\rangle$: $(a_{11} \le a_{21}) \wedge (b_{22} \le b_{21})$; (2.21)

для $\langle 2, 2\rangle$: $(a_{12} \le a_{22}) \wedge (b_{21} \le b_{22})$. (2.22)

Суть этих условий в том, что при изменении только одной стратегии у игрока, поменявшего её, выигрыш не может увеличиться – он может остаться тем же или уменьшиться.

Рассмотрим игру Аумана из примера 1, платёжная матрица которой имеет вид (1).

Для того, чтобы пара стратегий $\langle 1, 1\rangle$ была равновесием Нэша, по определению требуется выполнение условий (2.11). В силу (1) $a_{21} = 101$, $a_{11} = 100$, т.е. уже 1-ое условие из (2.11) не выполняется. Поэтому пара стратегий $\langle 1, 1\rangle$ не является равновесием Нэша.

Для того, чтобы пара стратегий $\langle 1, 2\rangle$ была равновесием Нэша, по определению требуется выполнение условий (2.12). В силу (1) $a_{22} = 1$, $a_{12} = 0$, т.е. уже 1-ое условие из (2.12) не выполняется. Поэтому пара стратегий $\langle 1, 2\rangle$ не является равновесием Нэша.

Для того, чтобы пара стратегий $\langle 2, 1\rangle$ была равновесием Нэша, по определению требуется выполнение условий (2.21). В силу (1) $a_{11} = 100$, $a_{21} = 101$, т.е. 1-ое условие из (2.21) выполняется. Также в силу (1) $b_{21} = 0$, $b_{22} = 1$, т.е. 2-ое условие из (2.21) не выполняется. Поэтому пара стратегий $\langle 2, 1\rangle$ не является равновесием Нэша.

Для того, чтобы пара стратегий $\langle 2, 2\rangle$ была равновесием Нэша, по определению требуется выполнение условий (2.22). В силу (1) $a_{22} = 1$, $a_{12} = 0$, т.е. 1-ое условие из (2.22) выполняется. Также в силу (1) $b_{22} = 1$, $b_{21} = 0$, т.е. 2-ое условие из (2.22) также выполняется. Поэтому <u>пара стратегий $\langle 2, 2\rangle$ является равновесием Нэша.</u>

Заметим, что при выборе обоими игроками 1-ых стратегий оба получают выигрыш 100 – в 100 раз больше, чем они получают в единственном равновесии Нэша $\langle 2, 2\rangle$. Но при выборе 2-ой стратегии каждый игрок гарантирует себе хотя бы выигрыш 1 <u>независимо от выбора другого игрока</u>, а при выборе 1-ой стратегии <u>такой гарантии нет</u>. Именно в гарантии некоторых, пусть и небольших, выигрышей обоих игроков, суть равновесия Нэша. «Кто не рискует, тот не ужинает с коньячком, а кто рискует, тот вообще не ужинает.» ■

Как и в матричных играх, можно определить оптимальные стратегии для обоих игроков. ***Оптимальной чистой стратегией*** игрока А называется любая стратегия i^*, которая максимизирует зависящее только от i выражение

$\min_{j\in\{1,\ldots,n\}} a_{ij}$. (3a)

Оптимальной чистой стратегией игрока В называется любая стратегия j^*, которая максимизирует зависящее только от j выражение

$\min_{i\in\{1,\ldots,m\}} b_{ij}$. (3b)

Утверждение 1. Пусть пара стратегий $\langle i^*, \ j^*\rangle$ является равновесием Нэша. Тогда стратегии i^* и j^* оптимальны ■

Нетрудно видеть, что, как в матричных играх может не быть решений в чистых стратегиях, так в бимаричных играх может не быть равновесий Нэша в чистых стратегиях.

Пример 3. Рассмотрим биматричную игру со следующей платёжной матрицей: $Q=\begin{pmatrix}0{,}8 & 2{,}6\\ 3{,}5 & 1{,}7\end{pmatrix}$. Проверяя, как в примере 2, каждую из 4-ёх пар чистых стратегий ⟨1, 1⟩, ⟨1, 2⟩, ⟨2, 1⟩, ⟨2, 2⟩ на выполнение условий (2.11) – (2.22), убеждаемся, что ни одна из них не является равновесием Нэша ■

1.2. Смешанное расширение биматричных игр. Понятие смешанного расширения определяется так же, как и для матричных игр. Если игрок А выбирает стратегию i с вероятностью x_i (i = 1, ..., m), а игрок В – стратегию j с вероятностью y_j(j = 1, ..., n), то говорят, что игроки А и В используют ***смешанные стратегии*** = (x_1, ..., x_m) и y = (y_1, ..., y_n). Чистая стратегия является частным случаем смешанной стратегии, когда все вероятности, кроме вероятности выбора данной чистой стратегии, равны 0, а вероятность выбора данной чистой стратегии равна 1.

Пусть игроки А и В выбрали стратегии x и y. Тогда средние выигрыши игроков А и В определяются формулами

$g_{\mathrm{A}}(x,y) =\sum_{i=1}^{m}\sum_{j=1}^{n} x_i y_j a_{ij}$, (4a)

$g_{\mathrm{B}}(x,y) =\sum_{i=1}^{m}\sum_{j=1}^{n} x_i y_j b_{ij}$.(4b)

Определим ***равновесие Нэша в смешанных стратегиях*** и ***оптимальные смешанные стратегии***, по аналогии с соответствующими понятиями в чистых стратегиях для рассматриваемого класса биматричных игр. Пара стратегий $\langle x^*, y^*\rangle$ называется равновесием Нэша в смешанных стратегиях, если

$(\forall x)(\forall y)\ ((g_{\mathrm{A}}(x,y^*) \le g_{\mathrm{A}}(x^*,y^*) \wedge (g_{\mathrm{B}}(x^*,y) \le g_{\mathrm{B}}(x^*,y^*))$. (5)

Оптимальной смешанной стратегией игрока Аназывается любая стратегия x^*, которая максимизирует зависящее только от x выражение

$\min_y g_{\mathrm{A}}(x,y)$. (6a)

Оптимальной смешанной стратегией игрока Вназывается любая стратегия y^*, которая максимизирует зависящее только от y выражение

$\min_x g_{\mathrm{B}}(x,y)$. (6b)

Утверждение 2. Пусть пара стратегий $\langle x^*, \ y^*\rangle$ является равновесием Нэша. Тогда стратегии x^* и y^* оптимальны ■

Принципиальное отличие случая смешанных стратегий от случая чистых стратегий для биматричных игр является аналогом того же отличия для матричных игр (см. утверждение 11-6):

Утверждение 3. Для любой биматричной игры существует равновесие Нэша в смешанных стратегиях ■

Напомним, что равновесие Нэша в чистых стратегиях существует не всегда, что и продемонстрировано в примере 3.

1.3. Нахождение равновесий Нэша в биматричных играх размерности 2×2. Как и нахождение решений для матричных игр, нахождение равновесий Нэша в смешанных стратегиях для произвольных биматричных игр является вычислительно достаточно сложной задачей – во всяком случае, выходящей за скромные рамки настоящего пособия. Поэтому мы ограничимся случаем, когда у каждого из двух игроков имеется всего две стратегии (игры размерности 2×2). Именно такой игрой является игра Аумана, проанализированная в примерах 1 и 2. В отличие от матричных игр размерности 2×2, описанный ниже способ позволяет находить все равновесия Нэша – как в чистых, так и в смешанных стратегиях.

Рассматриваемые игры задаются парой матриц $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ и $B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$. У каждого игрока есть две чистых стратегии, которым соответствует выбор 1-ым игроком одной из двух строк матрицы A, а 2-ым игроком – одного из двух столбцов матрицы B. При выборе игроками пары стратегий i и j игрок А получает выигрыш a_{ij}, а игрок В – выигрыш b_{ij} ($i, j = 1, 2$). Представим их средние выигрыши (4) в следующем виде:

$g_A(x, y) = (a_{11} - a_{12} - a_{21} + a_{22})xy + (a_{12} - a_{22})x + (a_{21} - a_{22})y + a_{22};$ (7a)

$g_B(x, y) = (b_{11} - b_{12} - b_{21} + b_{22})xy + (b_{12} - b_{22})x + (b_{21} - b_{22})y + b_{22},$ (7b)

где x и y– вероятности выбора 1-ых чистых стратегий игроками А и В (см. для сравнения формулы (11-20) – (11-23)).

Дальнейшие рассуждения во многом аналогичны рассуждениям из раздела 11-5. Положим

$a_1 = (a_{11} - a_{12} - a_{21} + a_{22}),$ (8a)

$a_2 = a_{22} - a_{12};$ (9a)

$b_1 = (b_{11} - b_{12} - b_{21} + b_{22}),$ (8b)

$b_2 = b_{22} - b_{21}.$ (9b)

Представим теперь средние выигрыши (7) в виде:

$g_A(x, y) = a_1xy - a_2x + (a_{21} - a_{22})y + a_{22},$ (10a)

$g_B(x, y) = b_1xy - b_2y + (b_{12} - b_{22})x + b_{22}.$ (10b)

Положив

$u = a_1y - a_2,\ v = (a_{21} - a_{22})y + a_{22},$ (11a)

$s = b_1x - b_2,\ \ t = (b_{12} - b_{22})x + b_{22}.$ (11b)

запишем (10) в виде

$g_A(x, y) = ux + v,$ (12a)

$g_B(x, y) = sy+t,$ (12b)

где u и v не зависят от x,s и t не зависят от y. Поэтому при любом <u>фиксированном</u> $y$$g_A(x, y)$ является линейной функцией от x, при любом <u>фиксированном</u> $x$$g_B(x, y)$ является линейной функцией от y. При этом коэффициент u, в силу (11a), является линейной функцией от y, а коэффициент s, в силу (11b), является линейной функцией от x. Положим также

$y_0 = a_2/a_1,$ (13a)

$x_0 = b_2/b_1.$ (13b)

Из формул (11a), (13a) и (11b), (13b) следует, что при $a_1 \neq 0$ значение $u = a_1y_0 - a_2 = 0$, и при $b_1 \neq 0$ значение$s = b_1x_0 - b_2 = 0$. Графики линейных функций $u = a_1y - a_2$ в зависимости от a_1 и a_2, которые понадобятся для явного построения равновесия Нэша, показаны на рис.1. Графики линейных функций $s = b_1x - b_2$ не приводятся, так как являются <u>точно такими же</u>.

Перейдём к явному построению равновесий Нэша. Начнём с игрока А. Для любой стратегии y (напомним, что y – это вероятность выбора 1-ой чистой стратегии игроком В, так что $0 \leq y \leq 1$) положим

$\varphi_A(y) = \{x \mid (\forall x') (g_A(x', y) \leq g_A(x, y))\}.$ (14a)

Другими словами, $\varphi_A(y)$ – это множество всех стратегий x игрока А, которые максимизирует его выигрыш $g_A(x, y)$ при данном y.

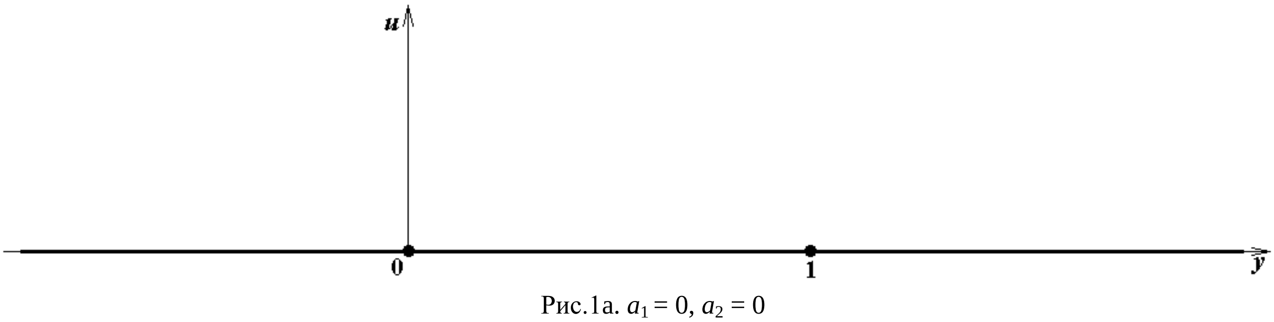


Рис.1а. $a_1 = 0$, $a_2 = 0$

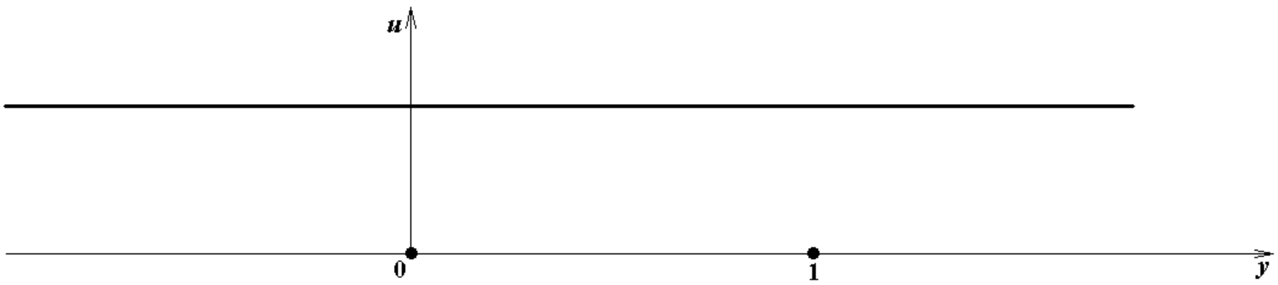


Рис.1b. $a_1 = 0$, $a_2 < 0$

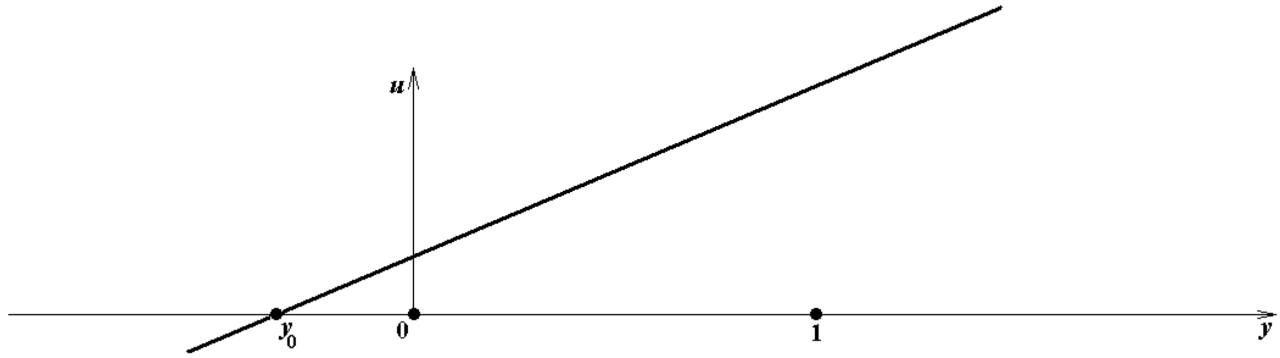


Рис.1c. $a_1 > 0$, $a_2 < 0$

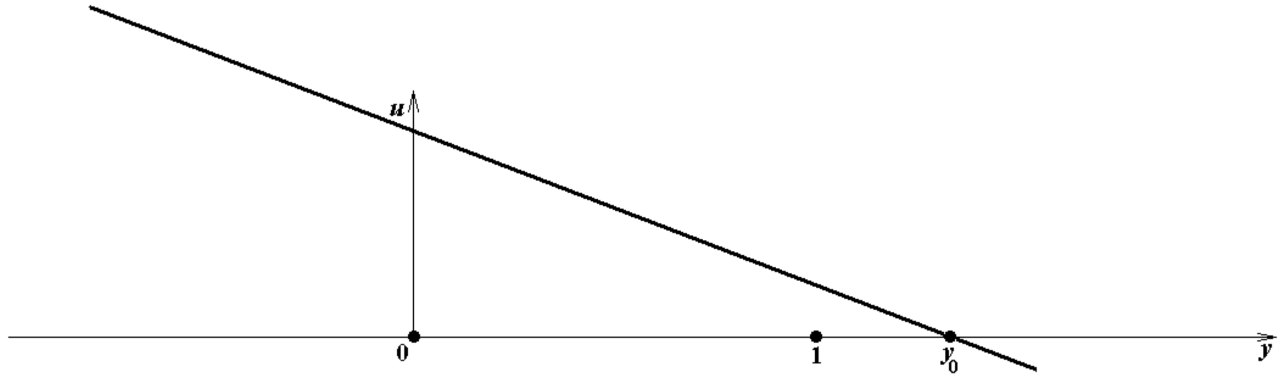


Рис.1d. $a_2 < a_1 < 0$

Рис.1e. $a_1 = 0$, $a_2 > 0$

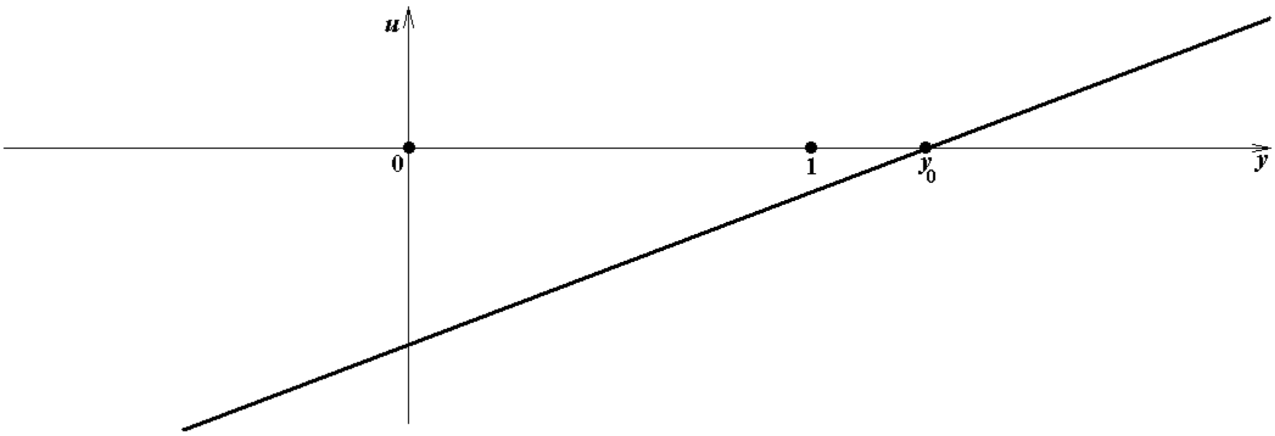


Рис.1f. $a_2 > a_1 > 0$

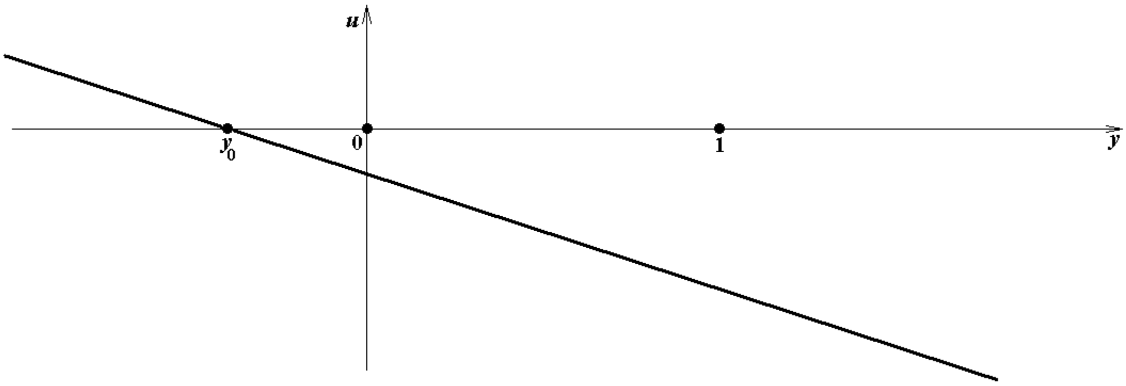
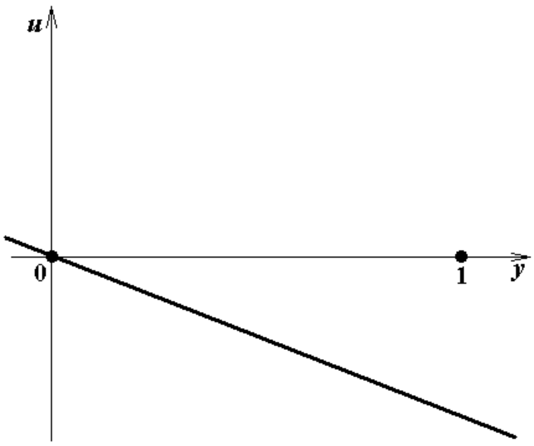


Рис.1g. $a_1 < 0$, $a_2 > 0$



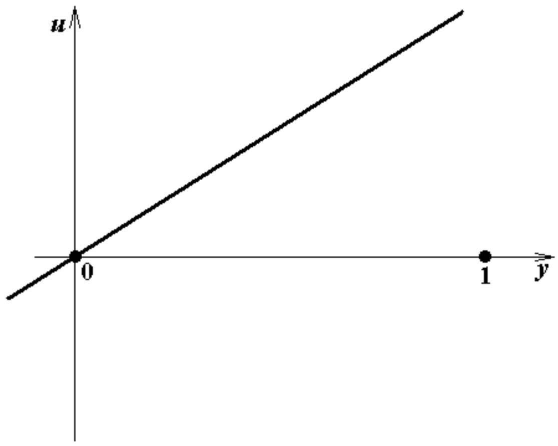
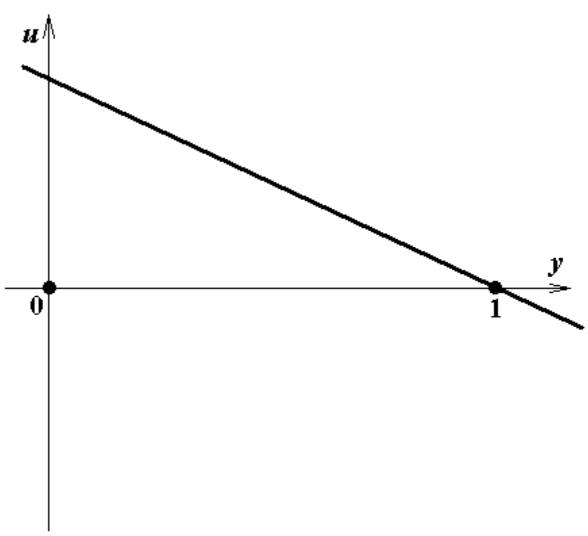


Рис.1h. $a_1 < 0$, $a_2 = 0$

Рис.1i. $a_1 > 0$, $a_2 = 0$



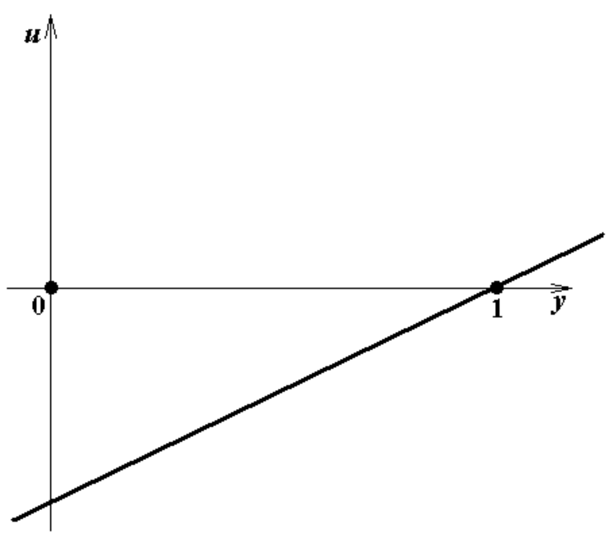


Рис.1j. $a_1 < 0$, $a_2 = a_1$ Рис.1k. $a_1 > 0$, $a_2 = a_1$

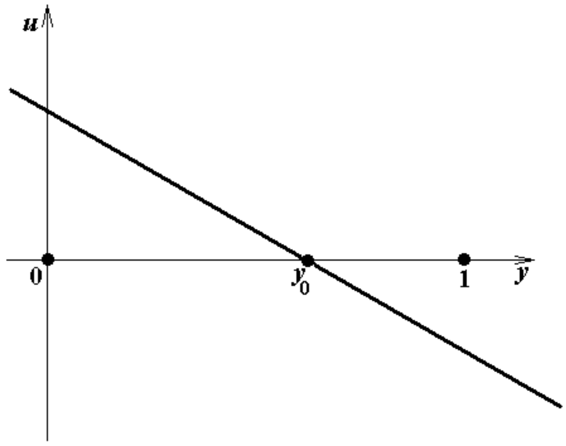


Рис.1*l*. $a_1 < a_2 < 0$

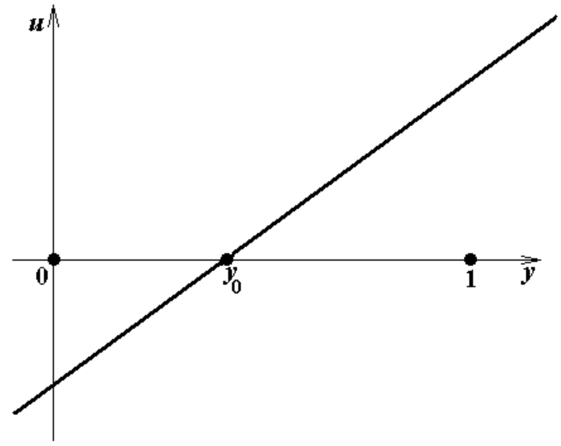


Рис.1m. $a_1 > a_2 > 0$

Имеет место

Утверждение 4. Множество $\varphi_A(y)$ определяется формулой

$$\varphi_A(y) = \begin{cases} \{0\}, \text{если } u < 0, \\ \{1\}, \text{если } u > 0, \\ [0,1], \text{если } u = 0. \end{cases} \qquad (15)$$

Доказательство. При любом фиксированном y линейная по x функция $g_A(x, y) = ux + v$ (см. (12a)), достигает максимума на отрезке [0, 1] при $x = 0$, если $u < 0$, и при $x = 1$, если $u > 0$. При $u = 0$ $g_A(x, y) \equiv v$, т.е. она не зависит от x. Поэтому её максимум достигается в <u>любой точке</u> отрезка [0, 1], что завершает доказательство ■

Положим

$$\Phi_A = \{(x, y) \mid y\in[0, 1], x\in\varphi_A(y)\}. \qquad (16a)$$

Множество Φ_A назовём ***графиком игрока*** А. Поскольку графики функции $u = a_1y - a_2$ при любых a_1 и a_2 представлены на рис.1, то в силу формул (15) и (16a) по ним легко строятся и графики игрока А также при любых a_1 и a_2. Остановимся на этом подробнее. При $a_1 = a_2 = 0$ имеем $u \equiv 0$ (см. рис. 1a), и, следовательно, $\varphi_A(y) = [0, 1]$ при всех y, что показано на рис.2a (график состоит <u>из всех точек</u> единичного квадрата). В следующих 3-ёх случаях (рис.1b, 1c и 1d) при $0 \le y\le 1$ имеем $u(y) > 0$, откуда, в силу (15), $\varphi_A(y) \equiv \{1\}$, что показано на рис.2b. В следующих 3-ёх случаях (рис.1e, 1f и 1g) при $0 \le y\le 1$ имеем$u(y) < 0$, откуда, в силу (15), $\varphi_A(y) \equiv \{0\}$, что показано на рис.2c.

Далее, для функции $u = a_1y - a_2$ в случае, показанном на рис.1h, $u(0) = 0$, $u(y) < 0$ при $y> 0$. В силу (15)$\varphi_A(0) = [0, 1]$, $\varphi_A(y) = 0$ при $y> 0$, что показано на рис.2d. Аналогично, в случае, показанном на рис.1i, $\varphi_A(0) = [0, 1]$, $\varphi_A(y) = 1$ при $y> 0$, что показано на рис.2e. В случае, показанном на рис.1j, $u(1) = 0$, $u(y) > 0$ при $y< 1$, откуда $\varphi_A(1) = [0, 1]$, $\varphi_A(y) = 1$ при $y< 1$, что показано на рис.2f. Аналогично, в случае, показанном на рис.1k, получаем график на рис.2g.

Наконец, в последних двух случаях, показанных на рис.1*l* и 1m – и только в этих случаях – получаем зигзаги, показанные на рис.2h и 2i, подобные представленным рис.11-1.

Таким образом, на рис.2 представлены <u>все возможные</u> варианты графиков игрока А в зависимости от a_1 и a_2.

Перейдём теперь к игроку В и проведём для него аналогичные рассуждения. Для любой стратегии xигрока А положим

$$\psi_B(x) = \{y \mid (\forall y')\ (g_B(x, y') \le g_B(x, y))\}. \qquad (14b)$$

Другими словами, $\psi_B(x)$ – это множество всех стратегий y игрока В, которые максимизирует его выигрыш $g_B(x, y)$ при данном x.

Определим график игрока В формулой, аналогичной формуле (16a):

$$\Psi_B = \{(x, y) \mid x\in[0, 1], y\in \psi_B(x)\}. \qquad (16b)$$

Повторяя все те рассуждения, которые были сделаны для графика игрока А, применительно к графику игрока В, совершенно аналогично приходим к 9 графикам, показанным на рис.3.

Из определения равновесия Нэша и формул (14a) и (14b) сразу следует

Утверждение 5. Любая точка (x^*, y^*), принадлежащая пересечению графиков Φ_A и Ψ_B, является равновесием Нэша ■

Имеет место утверждение, завершающее построение равновесия Нэша:

Утверждение 6. Для любых параметров a_1, a_2, b_1 и b_2 пересечение графиков Φ_A и Ψ_B непусто.

Для доказательства утверждения надо рассмотреть все пары графиков, в которых 1-ый график является одним из 9 графиков рис.2, а 2-ой график – одним из 9 графиков рис.3. Заметим, что каждый из графиков содержит не менее 2-ух вершин единичного квадрата. Поэтому в случаях, когда один из графиков содержит не менее 3-ёх вершин, пересечение обязательно есть. Таким образом, осталось рассмотреть только случаи, когда каждый из графиков содержит только две вершины. Это графики на рисунках 2b, 2c, 2h, 2i и на рисунках 3b, 3c, 3h, 3i. Во всех этих 16-и случаях, кроме двух пар: ⟨2h, 3i⟩ и ⟨2i, 3h⟩ наличие хотя бы одной общей вершины очевидно. В последних двух случаях имеется одна внутренняя общая точка, как это показано на рис. 11-1.3 ■

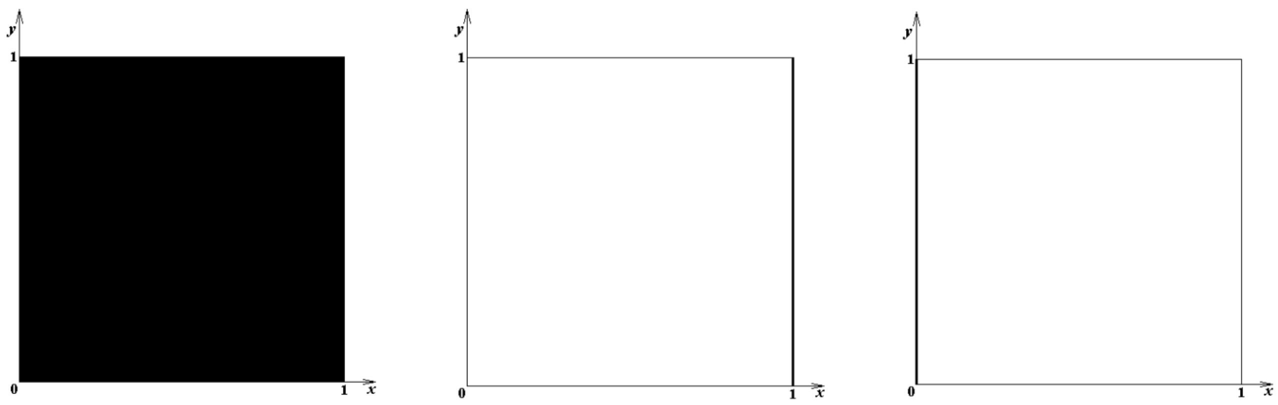


Рис.2a. $a_1 = 0$, $a_2 = 0$Рис.2b. $a_1 = 0$, $a_2 < 0$; $a_1 > 0$, Рис.2c. $a_1 = 0$, $a_2 > 0$; $a_2 > a_1 > 0$;
$a_2 < 0$; $a_2 < a_1 < 0$ $a_1 < 0$, $a_2 > 0$

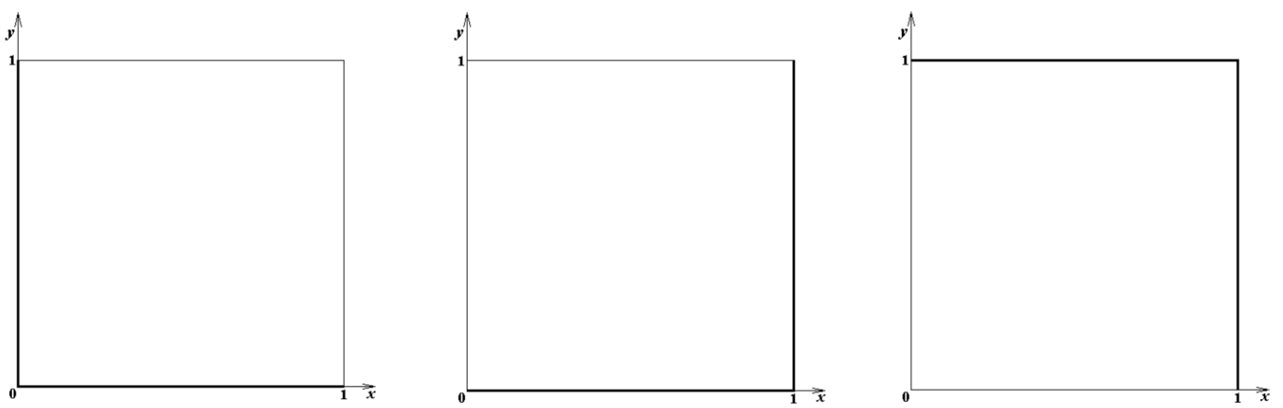


Рис.2d. $a_1 < 0$, $a_2 = 0$ Рис.2e. $a_1 > 0$, $a_2 = 0$Рис.2f. $a_1 < 0$, $a_2 = a_1$

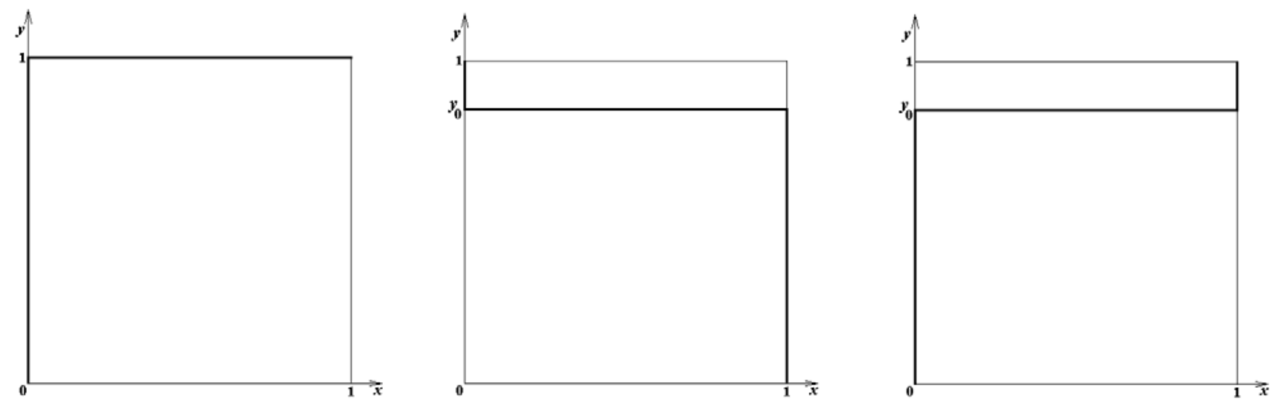
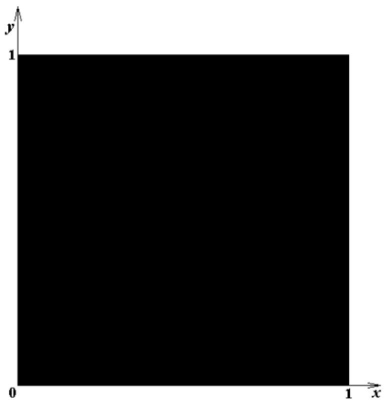
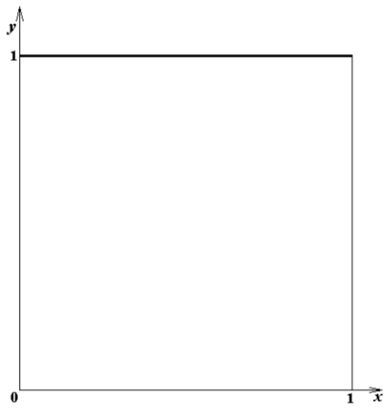


Рис.2g. $a_1 > 0$, $a_2 = a_1$Рис.2h. $a_1 < a_2 < 0$ Рис.2i. $a_1 > a_2 > 0$





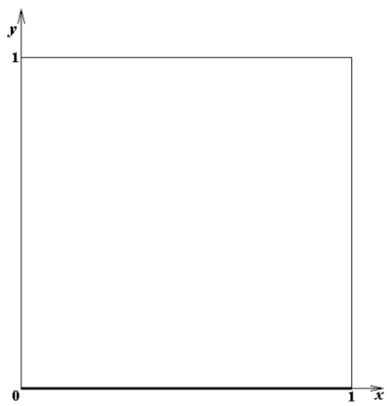
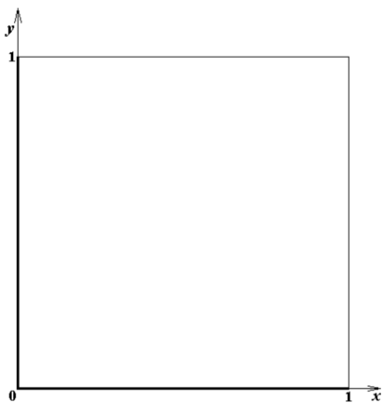
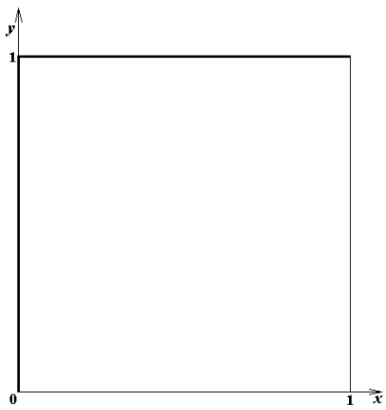


Рис.3a. $b_1 = 0$, $b_2 = 0$ Рис.3b. $b_1 = 0$, $b_2 < 0$;$b_1 > 0$, Рис.3c. $b_1 = 0$, $b_2 > 0$; $b_2 > b_1 > 0$;
$b_2 < 0$; $b_2 < b_1 < 0$ $b_1 < 0$, $b_2 > 0$





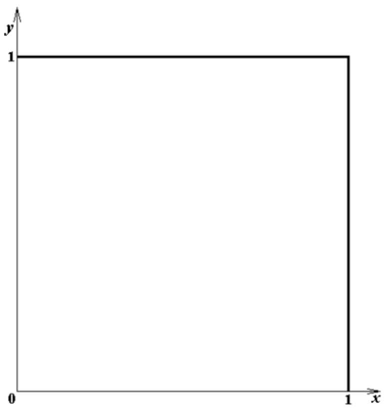
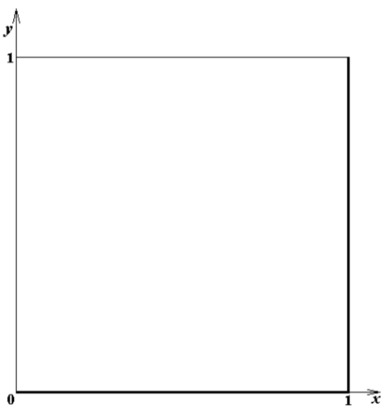
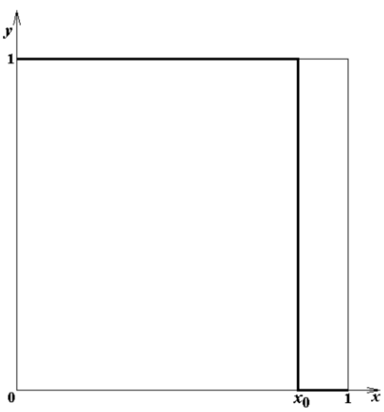


Рис.3d. $b_1 < 0$, $b_2 = 0$ Рис.3e. $b_1 > 0$, $b_2 = 0$Рис.3f. $b_1 < 0$, $b_2 = b_1$





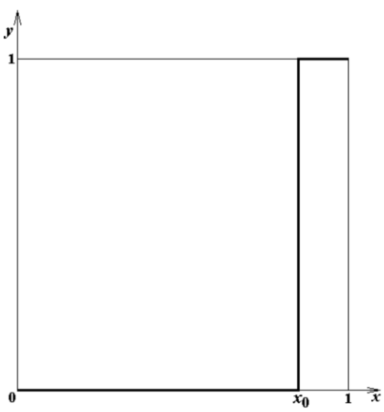


Рис.3g. $b_1 > 0$, $b_2 = b_1$Рис.3h. $b_1 < b_2 < 0$ Рис.3i. $b_1 > b_2 > 0$

Для анализа биматричных игр требуется найти все равновесия Нэша и выигрыши обоих игроков в этих равновесиях. Для этого надо выполнить следующий

Алгоритм 1. Анализ биматричной игры размерности 2×2.

1. По платёжным матрицам A и B обоих игроков, пользуясь формулами (8) и (9), найти параметры a_1, a_2, b_1,b_2.

2. По параметрам a_1 и a_2, определить график игрока А, пользуясь рис.2.

3. По параметрам b_1 и b_2, определить график игрока В, пользуясь рис.3.

4. Найти пересечение найденных графиков.

5. Для каждой точки (x^*, y^*) из указанного пересечения, определить смешанные стратегии $x = (x^*, 1 - x^*)$, $y = (y^*, 1 - y^*)$. Все они представляют собой искомые равновесия Нэша.

6. Выигрыш игрока А в равновесии Нэша определяется формулой (10a), игрока В – формулой (10b), в которые подставляются координаты x и y данного равновесия (вероятности выбора игроками своих 1-ых чистых стратегий). Заметим, что если данное равновесие является равновесием в чистых стратегиях, то ничего вычислять не надо: выигрыши игроков равны соответствующим элементам платёжной матрицы. Например, если равновесием является пара стратегий$\langle 1, 1\rangle$, то выигрыш игрока А равен a_{11}, игрока В –b_{11}, и т.д.■

Для иллюстрации работы алгоритма 1 рассмотрим несколько примеров.

Пример 4. Рассмотрим игру Аумана из примера 1. Её платёжная матрица задаётся формулой (1). В примере 2 было установлено, что в данной игре существует единственное равновесие $\langle 2,\ 2\rangle$ в чистых стратегиях. Это равновесие при записи в виде смешанной стратегии принимает вид $x = (0,\ 1)$, $y = (0,\ 1)$. Убедимся, что никаких других равновесий у игры Аумана нет. В данном случае

$a_{11} = 100$, $a_{12} = 0$, $a_{21} = 101$, $a_{22} = 1$;

$b_{11} = 100$, $b_{12} = 101$, $b_{21} = 0$, $b_{22} = 1$.

По формулам (8) и (9) получаем

$a_1 = a_{11} - a_{12} - a_{21} + a_{22} = 100 - 0 - 101 + 1 = 0$;

$a_2 = a_{22} - a_{12} = 1 - 0 = 1$;

$b_1 = b_{11} - b_{12} - b_{21} + b_{22} = 100 - 101 - 0 + 1 = 0$;

$b_2 = b_{22} - b_{21} = 1 - 0 = 1$.

График игрока А при $a_1 = 0$, $a_2 > 0$ показан на рис.2c; график игрока В при $b_1 = 0$, $b_2 > 0$ показан на рис.3c. Оба графика и их пересечение показаны на рис.4. Единственной общей точкой является точка (0, 0), которой и соответствует единственное равновесие Нэша $x = (0,\ 1)$, $y = (0,\ 1)$.

Найдём выигрыши игроков. В соответствии с шагом 6 алгоритма 1 имеем $g_{\mathrm{A}}(0,\ 0) = a_{22} = 1$;$g_{\mathrm{B}}(0,\ 0) = b_{22} = 1$■

Пример 5. Рассмотрим биматричную игру из примера 3 с платёжной матрицей $Q = \begin{pmatrix} 0{,}8 & 2{,}6 \\ 3{,}5 & 1{,}7 \end{pmatrix}$, которая не имеет равновесий Нэша в чистых стратегиях. Подсчитаем a_1, a_2, b_1, b_2 по формулам (8), (9):

$a_1 = a_{11} - a_{12} - a_{21} + a_{22} = 0 - 2 - 3 + 1 = -4$;

$a_2 = a_{22} - a_{12} = 1 - 2 = -1$;

$b_1 = b_{11} - b_{12} - b_{21} + b_{22} = 8 - 6 - 5 + 7 = 4$;

$b_2 = b_{22} - b_{21} = 7 - 5 = 2$.

Вид графика игрока А при $a_1 = -4$, $a_2 = -1$ ($a_1 < a_2 < 0$) показан на рис.2h; вид графика игрока В при $b_1 = 4$, $b_2 = 2$ ($b_1 > b_2 > 0$) показан на рис.3i. В данном случае $y_0 = 0{,}25$; $x_0 = 0{,}5$. Оба графика и их пересечение показаны на рис.5. Единственной общей точкой графиков является точка (0,5; 0,25), которой и соответствует единственное равновесие Нэша $x = (0{,}5;\ 0{,}5)$, $y = (0{,}25;\ 0{,}75)$.

Выигрыши игроков определяются по формулам (10):

$g_{\mathrm{A}}(0{,}5;\ 0{,}25) = -4\cdot 0{,}5\cdot 0{,}25 + 1\cdot 0{,}5 + (3 - 1)\cdot 0{,}25 + 1 = -0{,}5 + 0{,}5 + 0{,}5 + 1 = 1{,}5$;

$g_{\mathrm{B}}(0{,}5;\ 0{,}25) = 4\cdot 0{,}5\cdot 0{,}25 - 2\cdot 0{,}25 + (6 - 7)\cdot 0{,}5 + 7 = 0{,}5 - 0{,}5 - 0{,}5 + 7 = 6{,}5$ ■

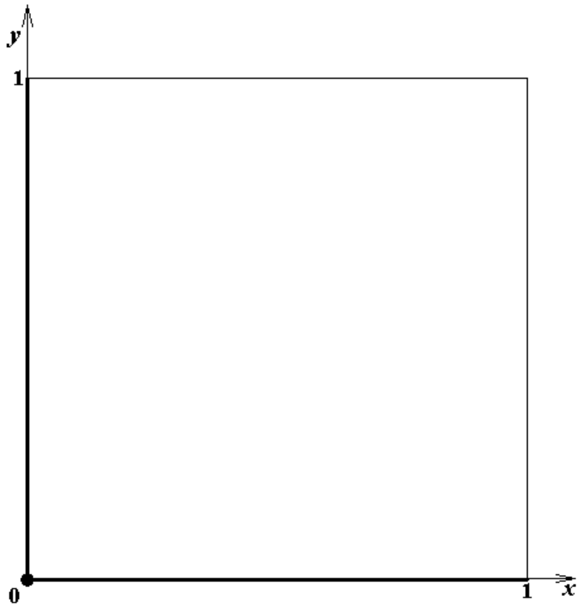


Рис.4. Равновесие в чистых стратегиях.

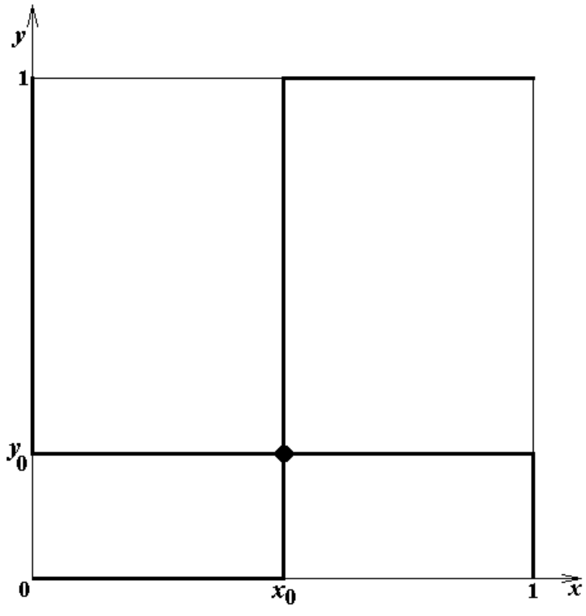


Рис.5. Равновесие в смешанных стратегиях.

Пример 6. Игра «***Семейный спор***». Рассмотрим биматричную игру со следующей платёжной матрицей $Q = \begin{pmatrix} 1{,}2 & 0{,}0 \\ 0{,}0 & 2{,}1 \end{pmatrix}$. Подсчитаем a_1, a_2, b_1, b_2 по формулам (8), (9):

$a_1 = a_{11} - a_{12} - a_{21} + a_{22} = 1 - 0 - 0 + 2 = 3$;

$a_2 = a_{22} - a_{12} = 2 - 0 = 2;$
$b_1 = b_{11} - b_{12} - b_{21} + b_{22} = 2 - 0 - 0 + 1 = 3;$
$b_2 = b_{22} - b_{21} = 1 - 0 = 1.$
Вид графика игрока А при $a_1 = 3$, $a_2 = 2$ ($a_1 > a_2 > 0$) показан на рис.2i; вид графика игрока В при $b_1 = 3$, $b_2 = 1$ ($b_1 > b_2 > 0$) показан на рис.3i. В данном случае $y_0 = ⅔$; $x_0 = ⅓$. Оба графика и их пересечение показаны на рис.6. Таким образом, в данной игре есть два равновесия в чистых стратегиях и одно равновесие в смешанных стратегиях. Выигрыши в паре чистых стратегий $\langle 1, 1\rangle$, соответствующей пересечению графиков в точке (1, 1), равны 1 и 2. Выигрыши в паре чистых стратегий $\langle 2, 2\rangle$, соответствующей пересечению графиков в точке (0, 0), равны 2 и 1. Наконец, выигрыши в паре смешанных стратегий, соответствующей пересечению графиков в точке (⅓,⅔), определяются по формулам (10):
$g_A(⅓,⅔) = 3\cdot⅓\cdot⅔ - 2\cdot⅓ + (0-2)\cdot⅔ + 2 = ⅔ - ⅔ - 2\cdot⅔ + 2 = ⅔;$
$g_B(⅓,⅔) = 3\cdot⅓\cdot⅔ - 1\cdot⅔ + (0-1)\cdot⅓ + 1 = ⅔ - ⅔ - ⅓ + 1 = ⅔.$

В отличие от примеров 4 и 5, в данном примере имеется 3 равновесия Нэша. Здесь впервые в нашем курсе возникает один из центральных вопросов организации коллективного взаимодействия участников (игроков), который в данном случае формулируется так: какое равновесие лучше? В чистых стратегиях имеется два равновесия, в которых один игрок выигрывает 2, а другой 1. Естественно, что игрок А заинтересован в равновесии $\langle 2, 2\rangle$, в котором его выигрыш равен 2, а у игрока В равен 1. По тем же причинам игрок В заинтересован в равновесии $\langle 1, 1\rangle$, в котором его выигрыш равен 2, а у игрока А равен 1. В единственном смешанном равновесии x = (⅓, ⅔),y = (⅔,⅓) выигрыши у обоих игроков равны ⅔,т.е. каждый получает <u>меньше</u>, чем в двух других равновесиях. Но зато они получают <u>поровну</u>, т.е. никто никому не завидует (в отличие от двух других равновесий), и этот вариант часто представляется более справедливым. Вопрос о том, что лучше – бóльший доход или справедливость – вообще не решается в рамках математики (или любой другой науки). Математика (как в данном случае) позволяет правильно описать имеющиеся возможности. А решение принимается другими людьми на основе других соображений ■

Пример 7. Более сложный случай. Рассмотрим биматричную игру со следующей платёжной матрицей $Q = \begin{pmatrix} (2,0) & (3,2) \\ (0,2) & (3,-1) \end{pmatrix}$. Подсчитаем a_1, a_2, b_1,b_2 по формулам (8), (9):
$a_1 = a_{11} - a_{12} - a_{21} + a_{22} = 2 - 3 - 0 - 1 = -2;$
$a_2 = a_{22} - a_{12} = 3 - 3 = 0;$
$b_1 = b_{11} - b_{12} - b_{21} + b_{22} = 0 - 2 - 0 + 3 = 1;$
$b_2 = b_{22} - b_{21} = -1 - 2 = -3.$
График игрока А при $a_1 = -2$, $a_2 = 0$ показан на рис.2d; график игрока В при $b_1 = 1$, $b_2 = -3$ показан на рис.3c. Оба графика и их пересечение показаны на рис.7. В данном случае пересечение состит из целого отрезка [0,1] на оси x-ов.

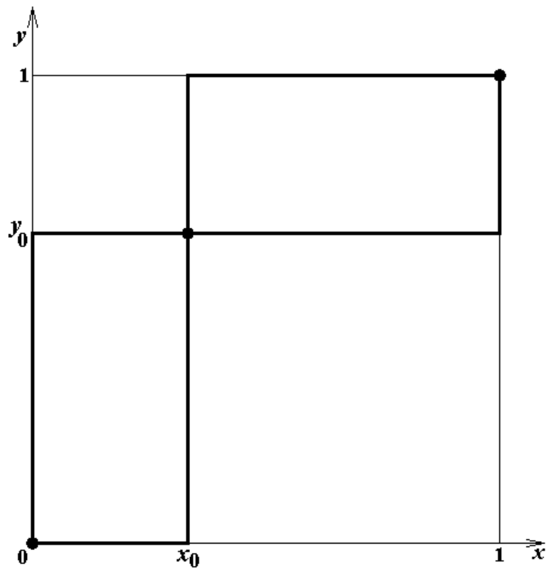


Рис.6.Три равновесия

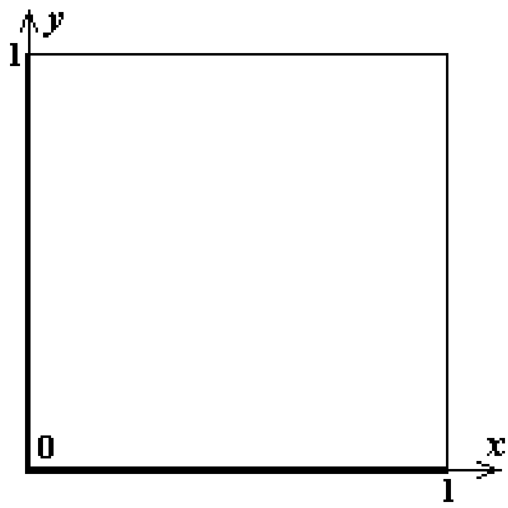


Рис 7. Отрезок равновесий

В соответствии с конструкцией, все эти точки представляют собой равновесия Нэша в рассматриваемой биматричной игре. Стратегия игрока A определяется числом x, $0 \le x \le 1$. Игрок B во всех этих точках выбирает чистую стратегию 2 (так как $y = 0$)Подсчитаем доходы во всех точках этого отрезка. Подставляя любое число x, $0 \le x \le 1$, число $y = 0$ и данные из платёжной матрицы в формулы (10a) и (10b), получаем $g_A(x, y) = 3$, $g_B(x, y) = 3x-1$. Таким образом, при изменении x от 0 до 1 выигрыш игрока B в соответствующем числу x равновесии Нэша линейно растёт от –1 до 2-ух, в то время как выигрыш игрока A во всех этих равновесиях остаётся постоянным ■

Графики игроков совершенно независимы друг от друга, поскольку определяются только по «своим» элементам платёжной матрицы. Поэтому нетрудно понять, просто глядя на рисунки 2 и 3, что пересечение графиков может быть любым отрезком любой длины на любой стороне единичного квадрата, с одним ограничением – такой общий отрезок должен содержать хотя бы одну вершину квадрата. Нетрудно найти и все другие возможные ситуации конечных и бесконечных множеств равновесий Нэша. В частности, может быть два чистых равновесия (в противоположных углах квадрата) и при этом ни одного смешанного равновесия. В любом случае рассуждения и вычисления в соответствии с алгоритмом 1 позволяют проанализировать (с данной точки зрения) любую игру из рассматриваемого класса игр.

2. Позиционные игры

Не давая пока никаких формальных определений, рассмотрим примеры игр другого типа, в которых ходы делаются по очереди, каждый ход и каждое новое (после этого хода) состояние игры немедленно становятся известными обоим игрокам.

Пример 8. *Игра «ним».* В этой игре на стол между двумя игроками выкладывается некоторое количество фишек. При каждом ходе игрок должен разделить одну из имеющихся групп фишек на два непустых подмножества разных размеров. Например, 6 фишек можно разделить на группы 5 и 1 или 4 и 2, но не 3 и 3. В начале игры все фишки образуют одну группу. После 1-го деления появится две группы, после 2-го деления – 3 группы, и т.д. Игрок, который не сможет больше сделать ход, проигрывает. На рис.8 показан граф игры «ним» для 7 фишек. Каждая вершина этого графа соответствует одному из возможных разбиений 7 фишек на группы. Их можно назвать ***позициями в игре***. Позиция ⟨5-1-1⟩ означает, что имеется три группы: в 1-ой группе – 5 фишек, а в двух других – по 1-ой фишке. Дуги соответствуют возможным ходам игрока в данной позиции. В позиции ⟨5-1-1⟩ есть две возможности: разделить группу из 5 фишек на группы из 3-ёх и 2-ух фишек или на группы из 4-ёх и 1-ой фишки. Это значит, что из позиции ⟨5-1-1⟩ можно перейти в позиции ⟨3-2-1-1⟩ и ⟨4-1-1-1⟩. Позиции ⟨2-2-2-1⟩, ⟨2-2-1-1-1⟩ и ⟨2-1-1-1-1-1⟩ называются ***финальными***, поскольку из них никуда нельзя перейти. Позиция ⟨7⟩ называется ***начальной***.

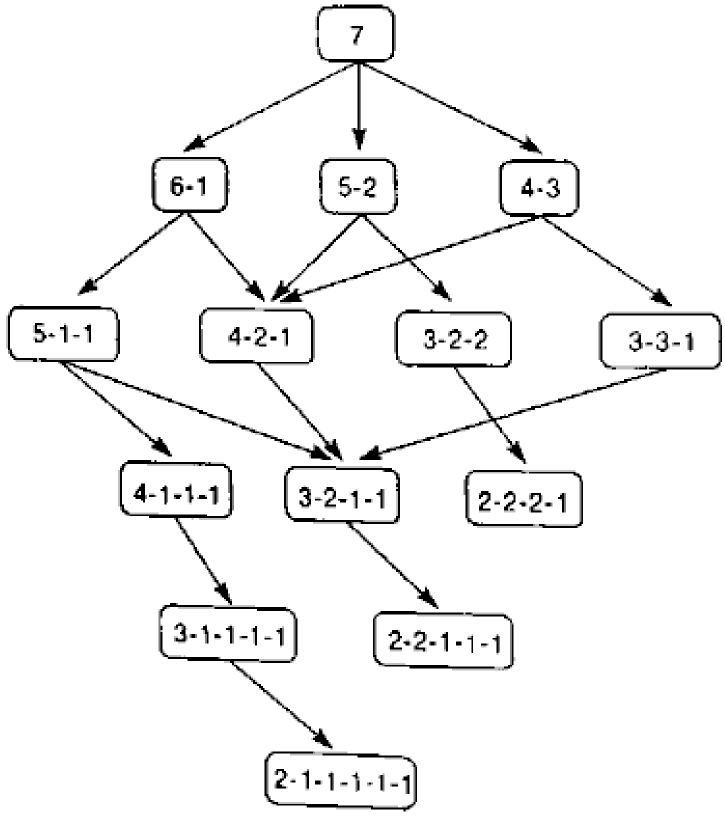


Рис.8. Граф игры «ним» для 7 фишек ■

Пример 9. Игра «***Две кучки***». Имеется две кучки, в каждой из которых имеется 4 фишки. Игрок может сделать один из следующих трёх ходов:

1. Забрать одну фишку из 1-ой кучки.
2. Забрать одну фишку из 2-ой кучки.
3. Забрать по одной фишке из обеих кучек.

Два игрока делают ходы по очереди. Проигрывает игрок, который забирает последнюю фишку. Разумеется, если в процессе игры в одной из кучек не осталось фишек, то у игроков из трёх ходов остаётся только один: забрать одну фишку из той кучки, в которой они ещё остались.

Рассматриваемую игру можно представить графом, показанным на рис.9. Как и в предыдущем примере, вершины соответствуют всем возможным позициям в игре, а дуги – ходам, т.е. переходам между позициями. Пара чисел в скобках у вершины указывает на число предметов в 1-й и 2-й кучке в соответствующей позиции. Игра начинается в вершине (4,4) и заканчивается в вершине (0,0). Из каждой вершины есть три перехода – налево (взят предмет из 1-й кучки), вниз (взят предмет из 2-й кучки) и по диагонали (взяты по одному предмету из обеих кучек).

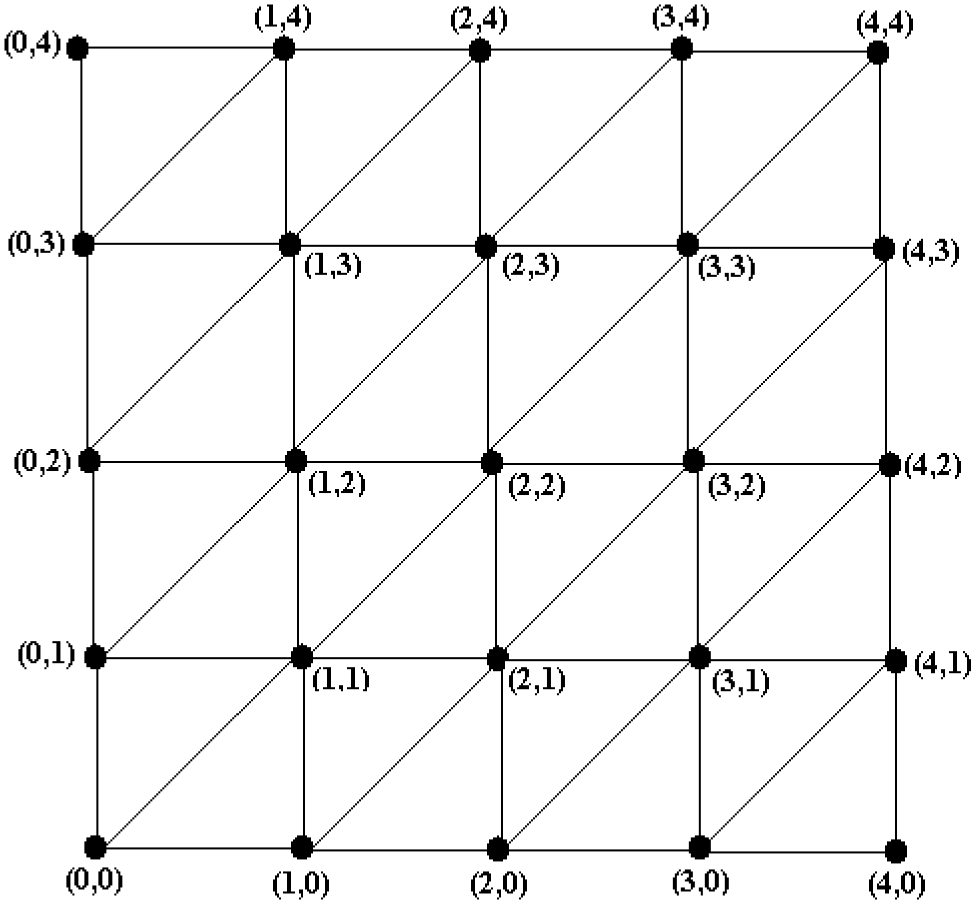


Рис.9. Граф игры «две кучки» ■

Дадим теперь достаточно общее описание игр рассматриваемого типа. Имеется ацикличный (т.е. не содержащий циклов) граф. Одна вершина объявляется ***начальной***. Одна (отличная от начальной) вершина объявляется ***финальной***. Предполагается, что финальная вершина является единственной вершиной в графе, из которой не выходит ни одной дуги. Игра состоит в том, что игроки по очереди переходят из вершины в вершину по дугам графа, причём 1-ая вершина обязательно должна быть начальной. При указанном предположении через конечное число ходов один из игроков обязательно попадает в финальную вершину, на чём игра и заканчивается. Игрок, попавший в финальную вершину, считается победителем. Обычно (см. примеры 8 и 9) вершины графа интерпретируются как позиции в игре, что и позволило дать таким играм естественное название **«*позиционные*».**

На первый взгляд, некоторые условия могут показаться слишком ограничительными. В игре «ним» имеется не одна, а три финальных вершины. В ней игрок, попавший в любую финальную вершину, действительно является победителем, поскольку другой игрок не может ходить. В то же время в игре «две кучки» игрок, попавший в единственную финальную вершину (0,0), является проигравшим, так как он взял последнюю фишку. Но оказывается, что все эти ситуации просто сводятся к описанной. Рассмотрим эти ситуации по отдельности.

1. **Имеется несколько финальных вершин**. Добавим к графу одну вершину z и проведём дуги из каждой финальной вершины в вершину z. Вершина z будет единственной финальной вершиной в новом графе. Если в исходной игре попадание в любую финальную вершину было проигрышем, то в новой игре попадание в новую единственную финальную вершину стало выигрышем. Наоборот, если в исходной игре попадание в любую финальную вершину было выигрышем, то в новой игре попадание в новую единственную финальную вершину стало проигрышем. Понятно, что суть игры совершенно не изменилась, поскольку в новую вершину попадает всегда не тот игрок, который попал в любую из исходных финальных вершин, а как раз другой. Таким образом, всегда можно рассматривать эквивалентную игру с одним финальным состоянием.

2. **Игрок, попавший в единственную финальную вершину, считается проигравшим**. В этом случае также добавляется одна новая финальная вершина z, в которую ведёт одна дуга из старой финальной вершины. Очевидно, что игрок, не попавший в старую финальную вершину, попадает в новую финальную вершину, т.е. является победителем в новой игре. Поэтому проигравшим в новой игре игроком является тот же игрок, который был проигравшим и в старой игре, т.е. новая и старая игра эквивалентны.

Таким образом, мы приходим к описанной выше модели позиционной игры, победитель которой – это игрок, попавший в единственную финальную вершину.

Опишем алгоритм, позволяющий одному из игроков всегда выигрывать в позиционной игре. Суть его в том, что вершинам графа (т.е. позициям игры) приписываются метки 0 или 1. В зависимости от того, какую метку получает начальная вершина, выигрывает начинающий или делающий следующий за ним ход игрок.

Алгоритм 2. Расстановка меток у вершин графа игры.

1. Инициализация. Сопоставим финальной вершине метку 0.

В поочерёдных позиционных играх, в которых ноль стоит в финальной позиции, все остальные нули и единицы ставятся по следующим правилам:

А. Пометим 1-ей все вершины, откуда за один ход <u>можно</u> попасть в вершины с меткой 0.

Б. Пометим 0-ём все вершины, откуда за один ход <u>можно</u> попасть <u>только</u> в вершины с меткой 1.

В. Чередуем шаги А и Б до тех пор, пока не окажется помеченной начальная вершина.

Стоп (алгоритм прекращает работу).

Дадим необходимые пояснения. Тот игрок, который каким бы то ни было образом оказался в вершине с меткой 1, может <u>гарантированно выиграть</u>, потому что он всегда <u>может</u> из 1 перейти в 0, откуда другой игрок может перейти <u>только</u> в 1 и т.д. Таким образом, другой игрок всегда будет попадать в вершины с меткой 1, и поэтому он никогда не придёт в финальную вершину с меткой 0. Следовательно, если в начальной вершине стоит метка 1, то начинающий игрок выигрывает, а если стоит 0, то он проигрывает ■

Пример 10. Рассмотрим игру из примера 9 и проиллюстрируем на этом примере работу алгоритма 2. Прежде всего надо добавить новую финальную вершину z, поскольку в исходной игре из примера 9 игрок, попавший в финальную вершину, проигрывает. Новый граф показан на рисунке 10a. У вершины z, в соответствии с шагом 1 алгоритма 2, поставлена метка 0.

Выполняя шаг А, ставим метку 1 у вершины (0,0), откуда можно попасть в вершину z, у которой уже стоит метка 1 (см. рис.10b). Выполняя шаг 2, ставим метки 0 у вершин (0,1) и (1,0), откуда можно попасть <u>только</u> в вершину (0,0), у которой уже стоит метка 1 (см. рис.10c). Далее, выполняя 2-ой раз шаг А, ставим метку 1 у 5 вершин: (0,2), (1,2), (1,1), (2,1), (2,0), из каждой из которых можно попасть в вершины (0,1) или (1,0), уже помеченных 0-ём (рис.10d). Выполняя 2-ой раз шаг Б, ставим метку 0 у вершин (0,3), (2,2) и (3,0), из которых можно попасть только в одну из вершин (0,2), (1,2), (1,1), (2,1), (2,0), которые уже получили метку 1 (рис.10e).

Дальнейшие аналогичные шаги представлены на рис.10f и 10g, а затем на рис.10h и 10i. При выполнении шага Б начальная вершина (4,4) получает метку 0. Это означает, что начинающий игрок в данной игре проигрывает, если другой игрок будет каждый раз из вершины с меткой 1 переходить в вершину с меткой 0, в соответствии с пояснениями к алгоритму 2.

Пример 11. Рассмотрим модификацию игру «Две кучки» с графом на рис.11. От игры из из примеров 9 и 10 она отличается только тем, что игрок, взявший последнюю фишку, выигрывает. Это означает, что вершина (0,0) является финальной вершиной и в соответствии с шагом 1 алгоритма 2 она получает метку 0. Все метки, поставленные в соответствии с алгоритмом 2, показаны на рис.11. У всех вершин, кроме самого левого столбца и самой нижней строки, все метки на рисунках 11 и 10i полностью совпадают. В частности, совпадают и метки у начальной

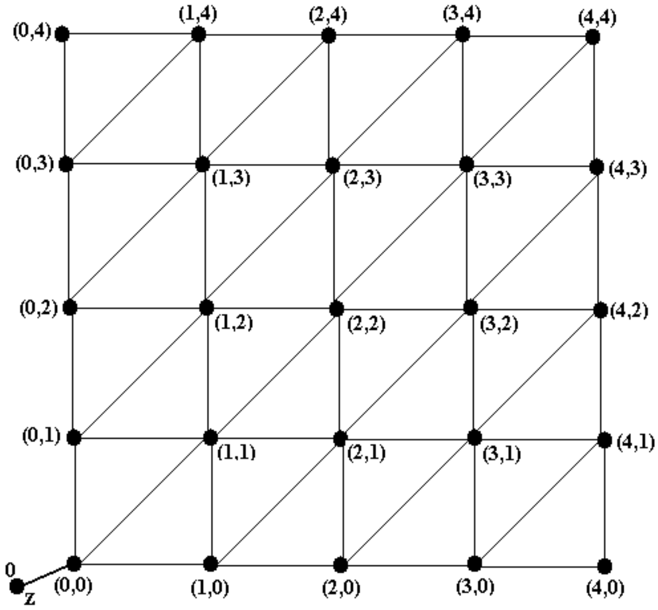


Рис.10а. Новый граф и его инициализация

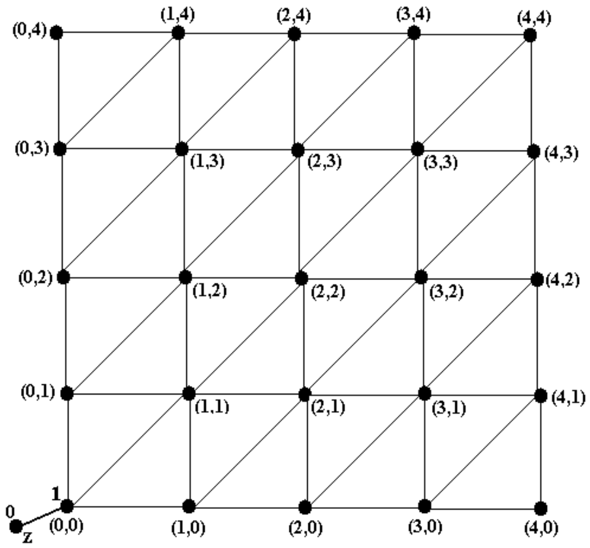


Рис.10b. Шаг А-1

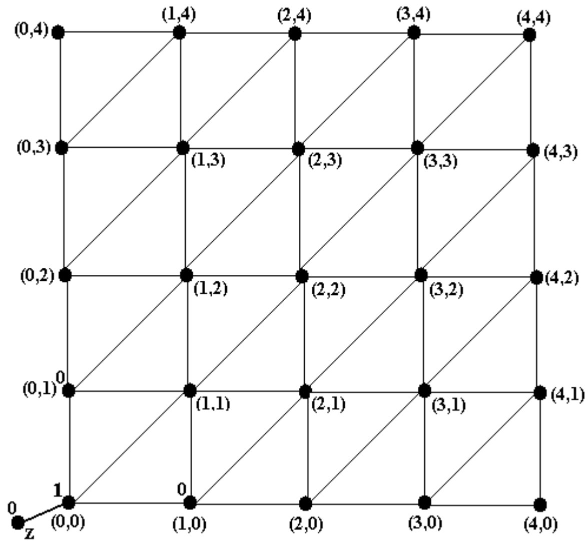


Рис.10c. Шаг Б-1

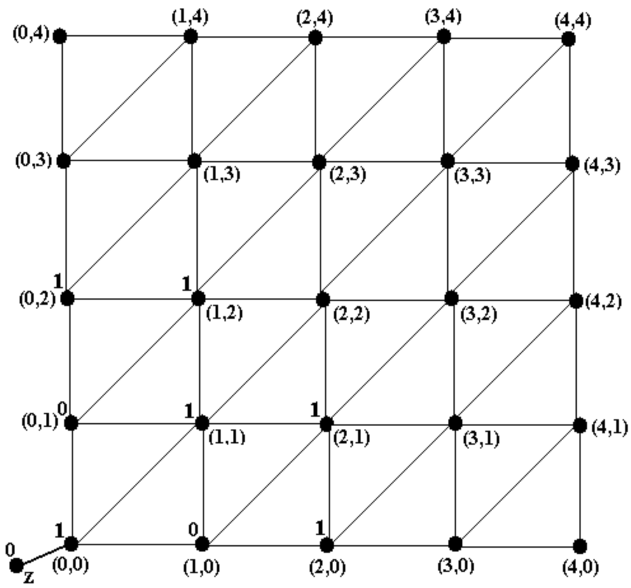


Рис.10d. Шаг А-2

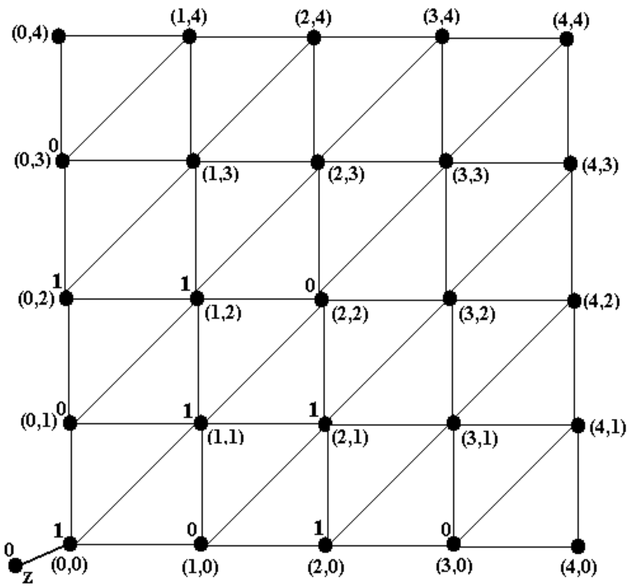


Рис.10e. Шаг Б-2

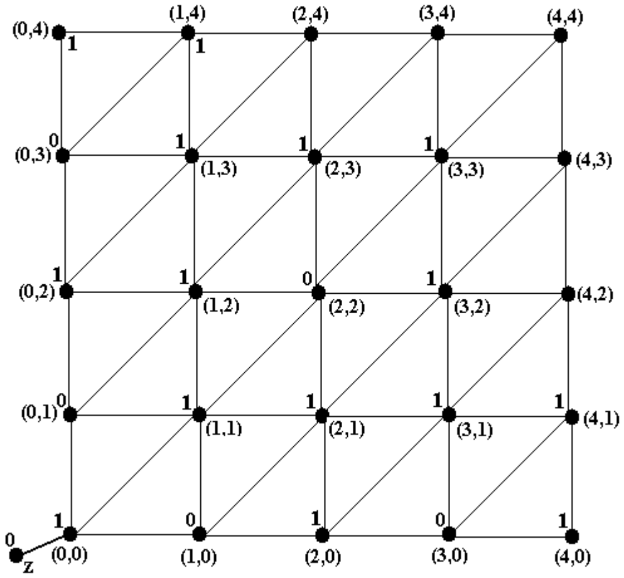


Рис.10f. Шаг А-3

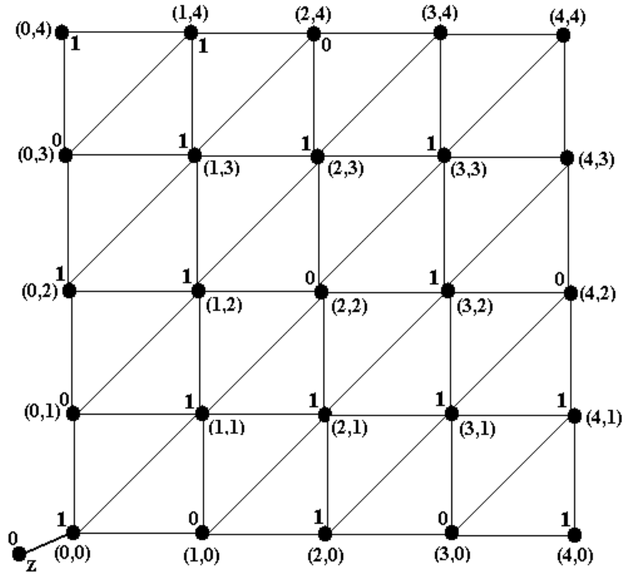


Рис.10g. Шаг Б-3

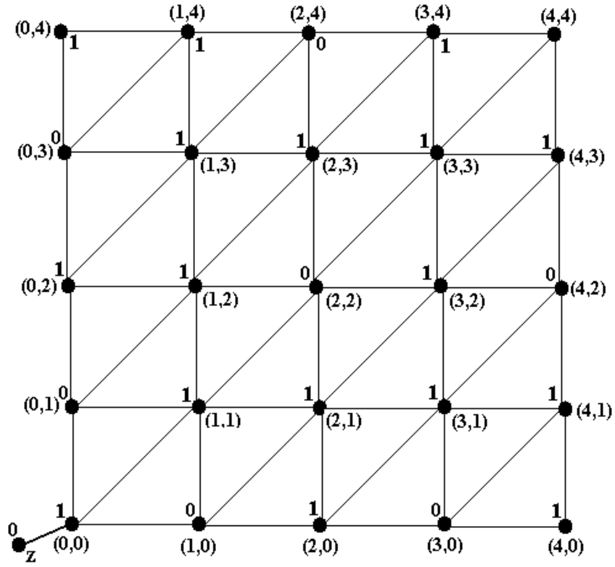


Рис.10h. Шаг А-4

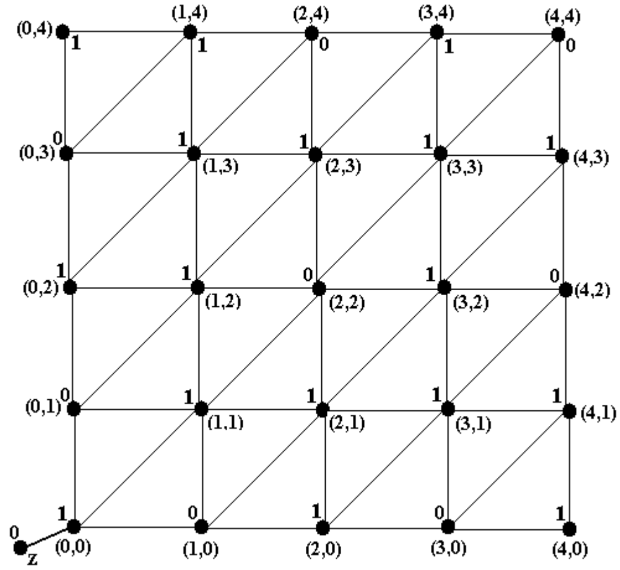


Рис.10i. Шаг Б-4

■

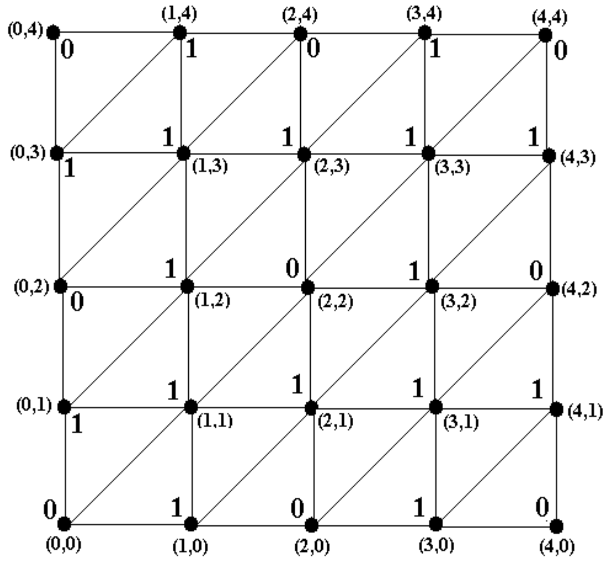


Рис.11. Граф модифицированной игры «Две кучки»

вершины (в обоих случаях 0). Поэтому в ***модифицированной игре «Две кучки»*** начинающий также проигрывает, как и в исходной игре. Это означает, что симметрия при изменении определения победителя отсутствует ■

3. Задания

Задание 1. В данной биматричной игре найти все равновесия Нэша в чистых стратегиях или установить их отсутствие. В качестве образца см. примеры 2 и 3 ■

Варианты платёжных матриц для задания 1:

$\begin{pmatrix}(-9,5) & (2,-2)\\(1,-1) & (-1,1)\end{pmatrix}$ 01	$\begin{pmatrix}(1,5) & (2,4)\\(0,3) & (3,2)\end{pmatrix}$ 02	$\begin{pmatrix}(1,2) & (3,0)\\(0,0) & (2,1)\end{pmatrix}$ 03	$\begin{pmatrix}(5,2) & (4,0)\\(3,3) & (2,1)\end{pmatrix}$ 04
$\begin{pmatrix}(9,-1) & (0,0)\\(0,0) & (8,9)\end{pmatrix}$ 05	$\begin{pmatrix}(2,2) & (0,3)\\(1,5) & (3,2)\end{pmatrix}$ 06	$\begin{pmatrix}(8,-1) & (3,0)\\(0,3) & (7,8)\end{pmatrix}$ 07	$\begin{pmatrix}(2,0) & (3,2)\\(0,2) & (3,5)\end{pmatrix}$ 08
$\begin{pmatrix}(9,-1) & (0,0)\\(0,-2) & (8,9)\end{pmatrix}$ 09	$\begin{pmatrix}(0,3) & (2,5)\\(2,1) & (2,3)\end{pmatrix}$ 10	$\begin{pmatrix}(8,-2) & (0,0)\\(0,1) & (7,8)\end{pmatrix}$ 11	$\begin{pmatrix}(3,1) & (5,3)\\(0,2) & (1,4)\end{pmatrix}$ 12
$\begin{pmatrix}(9,-1) & (3,0)\\(0,3) & (6,7)\end{pmatrix}$ 13	$\begin{pmatrix}(1,5) & (3,4)\\(3,0) & (2,1)\end{pmatrix}$ 14	$\begin{pmatrix}(7,-1) & (0,0)\\(0,-2) & (8,9)\end{pmatrix}$ 15	$\begin{pmatrix}(8,8) & (0,9)\\(9,0) & (1,1)\end{pmatrix}$ 16
$\begin{pmatrix}(5,3) & (4,1)\\(1,3) & (0,2)\end{pmatrix}$ 17	$\begin{pmatrix}(5,6) & (3,8)\\(9,2) & (4,4)\end{pmatrix}$ 18	$\begin{pmatrix}(3,3) & (2,4)\\(4,2) & (1,1)\end{pmatrix}$ 19	$\begin{pmatrix}(-9,5) & (3,-2)\\(1,-1) & (-1,1)\end{pmatrix}$ 20
$\begin{pmatrix}(1,5) & (2,4)\\(0,-3) & (3,2)\end{pmatrix}$ 21	$\begin{pmatrix}(1,2) & (0,0)\\(0,-1) & (2,1)\end{pmatrix}$ 22	$\begin{pmatrix}(5,2) & (4,0)\\(3,-2) & (2,1)\end{pmatrix}$ 23	$\begin{pmatrix}(9,-1) & (0,0)\\(0,-2) & (1,9)\end{pmatrix}$ 24
$\begin{pmatrix}(1,-1) & (2,-2)\\(-9,5) & (-1,1)\end{pmatrix}$ 25	$\begin{pmatrix}(2,4) & (1,5)\\(0,3) & (3,2)\end{pmatrix}$ 26	$\begin{pmatrix}(0,0) & (3,0)\\(1,2) & (2,1)\end{pmatrix}$ 27	$\begin{pmatrix}(2,1) & (4,0)\\(3,3) & (5,2)\end{pmatrix}$ 28
$\begin{pmatrix}(9,-1) & (3,0)\\(0,0) & (8,9)\end{pmatrix}$ 29	$\begin{pmatrix}(2,2) & (0,3)\\(3,2) & (1,5)\end{pmatrix}$ 30	$\begin{pmatrix}(8,-1) & (7,8)\\(0,3) & (3,0)\end{pmatrix}$ 31	$\begin{pmatrix}(0,2) & (3,2)\\(2,0) & (3,5)\end{pmatrix}$ 32
$\begin{pmatrix}(0,0) & (9,-1)\\(0,-2) & (8,9)\end{pmatrix}$ 33	$\begin{pmatrix}(2,3) & (2,5)\\(2,1) & (0,3)\end{pmatrix}$ 34	$\begin{pmatrix}(0,1) & (0,0)\\(8,-2) & (7,8)\end{pmatrix}$ 35	$\begin{pmatrix}(3,1) & (1,4)\\(0,2) & (5,3)\end{pmatrix}$ 36
$\begin{pmatrix}(9,-1) & (3,0)\\(-2,3) & (6,7)\end{pmatrix}$ 37	$\begin{pmatrix}(1,5) & (3,4)\\(2,1) & (3,0)\end{pmatrix}$ 38	$\begin{pmatrix}(0,0) & (7,-1)\\(0,-2) & (8,9)\end{pmatrix}$ 39	$\begin{pmatrix}(1,4) & (0,9)\\(9,0) & (8,8)\end{pmatrix}$ 40
$\begin{pmatrix}(1,3) & (4,1)\\(5,3) & (0,2)\end{pmatrix}$ 41	$\begin{pmatrix}(5,6) & (4,4)\\(9,2) & (3,8)\end{pmatrix}$ 42	$\begin{pmatrix}(3,3) & (2,4)\\(5,2) & (1,1)\end{pmatrix}$ 43	$\begin{pmatrix}(-9,5) & (3,-2)\\(-1,1) & (1,-1)\end{pmatrix}$ 44
$\begin{pmatrix}(2,4) & (1,5)\\(0,-3) & (3,2)\end{pmatrix}$ 45	$\begin{pmatrix}(2,1) & (0,0)\\(0,-1) & (2,5)\end{pmatrix}$ 46	$\begin{pmatrix}(3,-2) & (4,0)\\(2,5) & (2,1)\end{pmatrix}$ 47	$\begin{pmatrix}(9,-1) & (1,9)\\(0,-2) & (0,0)\end{pmatrix}$ 48■

Задание 2. В данной биматричной игре найти все равновесия Нэша в чистых и / или смешанных стратегиях. В качестве образца см. примеры 4 – 7

Варианты платёжных матриц для задания 2:

$\begin{pmatrix}(-6,5) & (2,-2)\\(1,-1) & (-1,1)\end{pmatrix}$ 01	$\begin{pmatrix}(1,3) & (2,4)\\(0,3) & (3,2)\end{pmatrix}$ 02	$\begin{pmatrix}(1,2) & (7,0)\\(0,0) & (2,1)\end{pmatrix}$ 03	$\begin{pmatrix}(5,2) & (4,-2)\\(3,3) & (2,1)\end{pmatrix}$ 04
$\begin{pmatrix}(9,-1) & (0,0)\\(2,0) & (8,9)\end{pmatrix}$ 05	$\begin{pmatrix}(2,2) & (0,3)\\(1,3) & (3,2)\end{pmatrix}$ 06	$\begin{pmatrix}(8,-1) & (3,0)\\(0,3) & (4,8)\end{pmatrix}$ 07	$\begin{pmatrix}(2,0) & (3,2)\\(0,2) & (3,-1)\end{pmatrix}$ 08

$\begin{pmatrix}(7,-1) & (0,0)\\(0,-2) & (8,9)\end{pmatrix}$ 09	$\begin{pmatrix}(0,4) & (2,5)\\(2,1) & (2,3)\end{pmatrix}$ 10	$\begin{pmatrix}(8,-2) & (2,0)\\(0,1) & (7,8)\end{pmatrix}$ 11	$\begin{pmatrix}(3,1) & (5,-1)\\(0,2) & (1,4)\end{pmatrix}$ 12
$\begin{pmatrix}(9,-1) & (3,0)\\(2,3) & (6,7)\end{pmatrix}$ 13	$\begin{pmatrix}(1,5) & (3,4)\\(3,-2) & (2,1)\end{pmatrix}$ 14	$\begin{pmatrix}(7,-1) & (0,0)\\(0,-2) & (6,9)\end{pmatrix}$ 15	$\begin{pmatrix}(8,8) & (0,9)\\(9,0) & (1,5)\end{pmatrix}$ 16
$\begin{pmatrix}(3,3) & (4,1)\\(1,3) & (0,2)\end{pmatrix}$ 17	$\begin{pmatrix}(5,1) & (3,8)\\(9,2) & (4,4)\end{pmatrix}$ 18	$\begin{pmatrix}(3,3) & (6,4)\\(4,2) & (1,1)\end{pmatrix}$ 19	$\begin{pmatrix}(-9,5) & (3,-4)\\(1,-1) & (-1,1)\end{pmatrix}$ 20
$\begin{pmatrix}(1,5) & (2,4)\\(3,-3) & (3,2)\end{pmatrix}$ 21	$\begin{pmatrix}(1,2) & (0,0)\\(0,-2) & (2,1)\end{pmatrix}$ 22	$\begin{pmatrix}(5,2) & (4,0)\\(3,-2) & (5,1)\end{pmatrix}$ 23	$\begin{pmatrix}(9,-1) & (0,0)\\(0,-2) & (1,6)\end{pmatrix}$ 24
$\begin{pmatrix}(3,-1) & (2,-2)\\(-9,5) & (-1,1)\end{pmatrix}$ 25	$\begin{pmatrix}(2,1) & (1,5)\\(0,3) & (3,2)\end{pmatrix}$ 26	$\begin{pmatrix}(0,0) & (1,0)\\(1,2) & (2,1)\end{pmatrix}$ 27	$\begin{pmatrix}(2,1) & (4,-1)\\(3,3) & (5,2)\end{pmatrix}$ 28
$\begin{pmatrix}(9,-1) & (3,0)\\(4,0) & (8,9)\end{pmatrix}$ 29	$\begin{pmatrix}(2,2) & (0,3)\\(3,6) & (1,5)\end{pmatrix}$ 30	$\begin{pmatrix}(8,-1) & (7,8)\\(0,3) & (9,0)\end{pmatrix}$ 31	$\begin{pmatrix}(0,2) & (3,2)\\(2,0) & (3,1)\end{pmatrix}$ 32
$\begin{pmatrix}(4,0) & (9,-1)\\(0,-2) & (8,9)\end{pmatrix}$ 33	$\begin{pmatrix}(2,-2) & (2,5)\\(2,1) & (0,3)\end{pmatrix}$ 34	$\begin{pmatrix}(0,1) & (3,0)\\(8,-2) & (7,8)\end{pmatrix}$ 35	$\begin{pmatrix}(3,1) & (1,0)\\(0,2) & (5,3)\end{pmatrix}$ 36
$\begin{pmatrix}(9,-1) & (3,0)\\(5,3) & (6,7)\end{pmatrix}$ 37	$\begin{pmatrix}(1,5) & (3,4)\\(2,-4) & (3,0)\end{pmatrix}$ 38	$\begin{pmatrix}(0,0) & (7,-1)\\(0,-2) & (5,9)\end{pmatrix}$ 39	$\begin{pmatrix}(1,4) & (0,9)\\(9,0) & (8,4)\end{pmatrix}$ 40
$\begin{pmatrix}(-2,3) & (4,1)\\(5,3) & (0,2)\end{pmatrix}$ 41	$\begin{pmatrix}(5,3) & (4,4)\\(9,2) & (3,8)\end{pmatrix}$ 42	$\begin{pmatrix}(3,3) & (-1,4)\\(5,2) & (1,1)\end{pmatrix}$ 43	$\begin{pmatrix}(-9,5) & (3,4)\\(-1,1) & (1,-1)\end{pmatrix}$ 44
$\begin{pmatrix}(2,4) & (1,5)\\(5,-3) & (3,2)\end{pmatrix}$ 45	$\begin{pmatrix}(2,1) & (0,0)\\(0,3) & (2,5)\end{pmatrix}$ 46	$\begin{pmatrix}(3,-2) & (4,0)\\(2,5) & (-1,1)\end{pmatrix}$ 47	$\begin{pmatrix}(9,-1) & (1,9)\\(0,-2) & (0,5)\end{pmatrix}$ 48■

Задание 3. Найти решение в играх «Две кучки», заданных в следующей таблице:

№	Число фишек в 1-ой кучке	Число фишек во 2-ой кучке	Игрок, взявший последнюю фишку
01	3	4	проигрывает
02	3	4	выигрывает
03	3	5	проигрывает
04	3	5	выигрывает
05	3	6	проигрывает
06	3	6	выигрывает
07	4	5	проигрывает
08	4	5	выигрывает
09	4	6	проигрывает
10	4	6	выигрывает
11	5	5	проигрывает
12	5	5	выигрывает
13	5	6	проигрывает
14	5	6	выигрывает
15	6	6	проигрывает
16	6	6	выигрывает

Ответ должен быть дан в виде графа с помеченными вершинами, как на рис.10i и 11. Должно быть указано, какой игрок (начинающий или второй) выигрывает ■

4. Предметный указатель

Глава 13. Неигровые модели взаимодействия

В этой главе рассматривается взаимодействие участников, которое хотя обычно и не рассматривается в рамках традиционной теории игр, всё же носит в себе черты как конфликта, так и сотрудничества. При таком взаимодействии интересы участников, хотя и не совсем совпадают, всё же не являются полностью противоположными, как в матричных играх двух лиц с нулевой суммой. Ещё более принципиальное отличие состоит в том, что суть дела в рассматриваемых здесь случаях состоит не в достижении оптимального результата отдельными игроками, а в определении таких правил взаимодействия участников, которые представляются разумными. Разумность правил означает, что попытка достижения своих целей каждым участником в отдельности в рамках этих правил приводит к результатам, которые с некоторой общей точки зрения представляются целесообразными и справедливыми. Напомним, что лица или организации, определяющие правила взаимодействия участников и заинтересованные в результатах этого взаимодействия, иногда называются метаигроками.

1. Обобщённые паросочетания

В разделе 6-6 было введено понятие двудольного графа. Далее, в главе 9, рассматривались связанные с такими графами задача о максимальном паросочетании и задача назначения. В этих задачах вершины одной доли графа интерпретируются как исполнители, а вершины другой – как задания, которые могут быть выполнены определёнными исполнителями. При поиске паросочетания требовалось обеспечить максимальную эффективность соответствующего назначения, т.е. сопоставления работ исполнителям. При этом интересы самих исполнителей не только не принимались во внимание, но даже и не рассматривались.

Однако во многих случаях обеим частям двудольного графа (а не только исполнителям) естественно сопоставляются некоторые интересы. Примером может служить распределение выпускников медицинского вуза между больницами, распределение выпускников военной академии между воинскими частями, и т.д. Другим примером служит распределение рукописей между рецензентами в научном издательстве. Конечно, рукописи сами по себе не обладают никакими мнениями относительно рецензентов, однако такими мнениями обладают редакторы издательства, которые понимают, какие рецензенты лучше подходят для работы с той или иной рукописью.

Общепринято рассматривать ситуации, в которых элементы каждой из двух сторон имеют предпочтения относительно элементов другой стороны, в гендерных терминах. Предполагается, что имеются мужчины и женщины: с точки зрения любого мужчины, все женщины ранжированы от лучшей к худшей, а с точки зрения любой женщины, так же ранжированы все мужчины. Все эти ранжировки могут быть совершенно произвольными.

Задача брачного агентства состоит в поиске разумного паросочетания (об оптимальных в такой деликатной ситуации вряд ли стоит говорить). Для того, чтобы формализовать понятие разумности паросочетания, рассмотрим некоторые примеры. Здесь и далее предполагается, что участники могут свободно обмениваться информацией о своих предпочтениях, а брачное агентство, естественно, знает их все.

Пример 1. Пусть имеется трое мужчин и три женщины, т.е. $M = \{m_1, m_2, m_3\}$, $W = \{w_1, w_2, w_3\}$ и предпочтения участников имеют вид:

$P(m_1) = w_2, w_1, w_3$; $P(w_1) = m_1, m_2, m_3$
$P(m_2) = w_1, w_3, w_2$; $P(w_2) = m_3, m_1, m_2$;
$P(m_3) = w_1, w_2, w_3$; $P(w_3) = m_3, m_1, m_2$.

Это означает, что с точки зрения мужчины m_1 лучшей является женщина w_2, за ней следует женщина w_1 и затем – женщина w_3; с точки зрения женщины w_1 лучшим является мужчина m_1, за ним следует мужчина m_2 и затем – мужчина m_3, и т.д. Рассмотрим паросочетание

$$\mu = \begin{pmatrix} m_1 & m_2 & m_3 \\ w_1 & w_2 & w_3 \end{pmatrix}$$

и мужчину m_1. Ему паросочетанием μ предписывается жениться на женщине w_1, при том, что более предпочтительным для него вариантом является женитьба на w_2. В то же время женщина w_2 должна, согласно паросочетанию μ, выйти замуж за m_2, хотя мужчина m_1 для неё более предпочтителен. Но тогда пара (m_1, w_2)откажется принимать условия, предлагаемые паросочетанием μ, поскольку они оба предпочитают друг друга более, чем предлагаемых им партнёров. Если брачное агентство предлагает своим клиентам устроить браки в соответствии с данным паросочетанием μ, то пара (m_1,w_2) не будем следовать советам этого агенства ■

В случаях, аналогичных рассмотренному в примере 1, говорят, что пара (m,w) ***блокирует*** паросочетание μ. Сама такая пара называется ***блокирующей*** паросочетание.

Пример 2. При тех же самых предпочтениях, как в примере 1, рассмотрим паросочетание

$$\nu = \begin{pmatrix} m_1 & m_3 & m_2 \\ w_1 & w_2 & w_3 \end{pmatrix}.$$

Рассмотрим мужчину m_1 и женщину w_2. Поскольку женщине w_2 предписан лучший, по её мнению, мужчина m_1, то пара (m_1,w_2) не является блокирующей данное паросочетание ν. Пара (m_1,w_3) не является блокирующей, поскольку мужчине m_1 предписана женщина w_1, которая в его предпочтениях стоит перед женщиной w_3. Пара (m_3,w_3) не является блокирующей, поскольку мужчине m_3 предписана женщина w_2, которая в его предпочтениях стоит перед женщиной w_3. Далее, рассмотрим пару (m_3,w_1). Она не является блокирующей, поскольку женщине w_1 предписан лучший, по её мнению, мужчина m_1. Пара (m_2,w_1) не является блокирующей по той же причине (женщине w_1 предписан лучший, по её мнению, мужчина m_1). Наконец, пара (m_2,w_2) не является блокирующей, поскольку женщине w_2 предписан лучший, по её мнению, мужчина m_3.

Таким образом, рассмотрены все 6 пар, которые не предписаны друг другу паросочетанием ν (число таких пар всегда равно 6 при 3-ёх мужчинах и 3-ёх женщинах независимо от паросочетания). Ни одна из этих пар не является блокирующей ■

Паросочетание, не содержащее блокирующих пар, называется ***устойчивым***. Пример 2 показывает, что при рассматриваемых предпочтениях паросочетание ν является устойчивым. Устойчивость не означает, что все получают лучших – с их точки зрения – партнёров. В паросочетании ν каждый мужчина получает 2-ую по своему предпочтению женщину, женщины w_1 иw_2 получают лучших – с их точки зрения – партнёров, а женщина w_3 получает мужчину m_2, занимающего последнее место в её предпочтениях. И тем не менее пары, которая блокировала бы это паросочетание, нет.

Естественно возникающие вопросы состоят в следующем. Существует ли хотя бы одно устойчивое паросочетание состояние при любых предпочтениях участников? Как его найти, если оно существует? На 1-ый вопрос положительный ответ даёт

Утверждение 1. При любых предпочтениях участников, задаваемых строгими ранжировками, устойчивые паросочетания существуют ■

Приведём алгоритм построения устойчивого паросочетания. Доказательство утверждения 1 (которое здесь не приводится) как раз и состоит в доказательстве того, что паросочетание, построенное данным алгоритмом, действительно устойчиво. Для упрощения изложения алгоритм демонстрируется на конкретном примере.

Пример 3. Построить устойчивое паросочетание при заданных предпочтениях мужчин и женщин. Пусть имеется трое мужчин и четыре женщины, т.е. $M = \{m_1, m_2, m_3\}$, $W = \{w_1, w_2, w_3, w_4\}$ и предпочтения участников имеют вид:

$P(m_1) = w_2, w_1, w_4, w_3$; $P(w_1) = m_2, m_1, m_3$;
$P(m_2) = w_2, w_3, w_4, w_1$; $P(w_2) = m_3, m_2, m_1$;
$P(m_3) = w_1, w_4, w_3, w_2$; $P(w_3) = m_3, m_1, m_2$;
$P(w_4) = m_1, m_3, m_2$.

Алгоритм состоит из последовательно выполняемых шагов 1, 2, 3, … . Каждый шаг состоит из двух фаз: ***фазы приписывания*** (А) и ***фазы отказа*** (Б). В фазе приписывания каждому мужчине из тех, у кого на данный момент нет пары, приписывается женщина, следующая в его предпочтении сразу после той, которая отказала ему (отвергла его) на предыдущем шаге. (На шаге 1 в фазе приписывания каждому мужчине просто приписывается первая в его предпочтении женщина.)

Если на каком-то шаге после фазы приписывания всем мужчинам приписаны разные женщины, то искомое паросочетание найдено и алгоритм прекращает работу. В противном случае переходим к фазе отказа.

В фазе отказа каждая женщина из тех, которым приписано более одного мужчины, оставляет того мужчину, который предшествует остальным из приписанных ей. Этот мужчина образует с данной женщиной пару, а остальным она отказывает (отвергает их).

Посмотрим, как работает алгоритм в рассматриваемом случае.

Шаг 1.

Фаза А. Приписываем каждому мужчине предпочитаемую им женщину. Получим
$\begin{pmatrix} m_1 & m_2 & m_3 \\ w_2 & w_2 & w_1 \end{pmatrix}$.

Фаза Б. Женщине w_2 приписано двое мужчин: m_1 и m_2. Поскольку в её предпочтении m_2 предшествует m_1, то она отвергает m_1. Получаем «неполное» паросочетание
$\begin{pmatrix} m_1 & m_2 & m_3 \\ & w_2 & w_1 \end{pmatrix}$.

Запомним, что мужчину m_1 отвергла женщина w_2.

Шаг 2.

Фаза А. В данный момент у мужчины m_1 нет пары. В соответствии с алгоритмом приписываем ему женщину, следующую в его списке сразу после отвергшей его женщиныw_2. Таковой является женщина w_1. В результате приписывания получаем
$\begin{pmatrix} m_1 & m_2 & m_3 \\ w_1 & w_2 & w_1 \end{pmatrix}$.

Фаза Б. Женщине w_1 приписано двое мужчин: m_1 и m_3. Поскольку в её предпочтении m_1 предшествует m_3, то она отвергает m_3. Получаем «неполное» паросочетание
$\begin{pmatrix} m_1 & m_2 & m_3 \\ w_1 & w_2 & \end{pmatrix}$.

Запомним, что мужчину m_3 отвергла женщина w_1.

Шаг 3.

Фаза А. В данный момент у мужчины m_3 нет пары. В соответствии с алгоритмом приписываем ему женщину, следующую в его списке сразу после отвергшей его женщиной w_1. Таковой является женщина w_4. В результате приписывания получаем
$\begin{pmatrix} m_1 & m_2 & m_3 \\ w_1 & w_2 & w_4 \end{pmatrix}$.

Поскольку всем мужчинам сопоставлены разные женщины, то искомое паросочетание найдено и алгоритм прекращает работу. В результате мужчина m_2 получает предпочитаемую им женщину, мужчины m_1 и m_3 получают женщин w_1 и w_4, вторых по их предпочтениям. Женщинам w_1, w_2 и w_4 достаются мужчины m_1, m_2 и m_3, вторые по их предпочтениям. Женщина w_3 остаётся одна ■

В рассмотренном примере на 1-ом шаге в фазе приписывания женщины приписывались мужчинам. Можно использовать тот же алгоритм, в котором женщины заменены на мужчин, а мужчины на женщин, т.е. на 1-ом шаге мужчины приписываются женщинам, далее на фазе отказа из нескольких женщин, приписанных одному мужчине, остаётся та, которая предпочтительней остальных, и т.д. Возможно построить несколько устойчивых паросочетаний, некоторые из которых могут быть в каких-то отношениях лучше, чем данное. Однако используемый алгоритм гарантирует устойчивость полученного паросочетания (отсутствие блокирующих пар) при любых исходных данных.

Можно сказать, что брачное агентство является в данном случае метаигроком, который определяет правила взаимодействия участников: они указывают свои предпочтения (в виде строгих ранжировок), а участникам предлагается достаточно разумное устойчивое паросочетание, которое строится на основе полученной от участников информации об их предпочтениях. В большинстве случаев такие разумные (в данном случае – устойчивые) паросочетания определяются неоднозначно, однако вопрос о том, как их оценивать и как выбирать в том или ином смысле «лучшие» или «оптимальные» паросочетания, здесь не рассматривается.

2. Справедливый делёж

В середине 1990-х годов американскими учёными Брамсом и Тэйлором была предложена новая модель разрешения конфликтов (см. литературу в конце данной части). В этой модели конфликт состоит в разногласиях сразу по нескольким вопросам (пунктам), причём важность этих пунктов для различных участников, вообще говоря, различна. Именно эти различия позволяют предложить такой вариант улаживания конфликта, при котором каждый получает то, что его больше устраивает по его собственной оценке. Возвращаясь к цитате, с которой начинается данная часть, можно сказать, что модель справедливого дележа Брамса-Тэйлора, коротко описанная в данном разделе (и подробно в литературе, упомянутой в конце части), является формальной моделью, выражающей этот очень разумный подход, призывающий к сотрудничеству даже при заметно отличающихся взглядах и целях

Пример 4. Раздел наследства. Рассмотрим реальный случай супружеской пары, которая имела собственный дом и коттеджи в штате Мэн. Когда муж умер, жена продала дом и коттеджи и переехала во Флориду. Оставалось несколько предметов, не имевших ценности для неё, но интересовавших двух её сыновей, Брэда и Дика, которые также жили в штате Мэн. Эти предметы перечислены в левой части приводимой ниже таблицы:

	Брэд (%)	Дик (%)
Двенадцатифутовая алюминиевая гребная лодка	6	14
Лодочный мотор в 3 лошадиных силы	6	14
Пианино в хорошем состоянии	17	2
Небольшой персональный компьютер	17	1
Охотничье ружье	4	4
Набор инструментов	6	2
Трактор фирмы «Форд» 1953 года выпуска с прицепным плугом	2	21
Сравнительно старенький крытый грузовичок («пикап»)	8	14
Мопед	17	14
Мопед	17	14
ВСЕГО	100	100

Ради иллюстрации, предположим, что мы можем заглянуть в мысли сыновей и точно узнать, какую долю от общей ценности представляет для них каждый предмет. За основу примем то, что Брэд получил бизнес-образование и интересуется музыкой, а Дик – спортсмен и занимается сельским хозяйством. Таким образом, пианино и компьютер больше привлекают Брэда, чем Дика. С другой стороны, Дика больше привлекают лодка, мотор, трактор и грузовичок. Эти рассуждения позволяют предположить оценки (выраженные в виде процента от общей ценности) различных предметов Брэдом и Диком, приведённые в правой части той же таблицы.

Можно рассмотреть, например такой вариант дележа (вместе с предметом указана и его оценка тем из братьев, которому достаётся данный предмет):

Набор Брэда: пианино (17%), компьютер (17%), ружье (4%), инструменты (6%), мопед (17%) – в сумме 61%.

Набор Дика: лодка (14%), лодочный мотор (14%), трактор (21%), пикап (14%), мопед (14%) – в сумме 77%.

Не обсуждая пока, хорош или плох предложенный делёж и как его улучшить, заметим самое главное: каждый из братьев получает больше половины возможных баллов (100) по своей собственной оценке. Именно идея опоры на собственные оценки каждого участника и лежит в основе предложенного Брамсом и Тэйлором подхода к разрешению конфликтов ■

2.1. Формальные понятия и определения. Дадим формальное описание рассматриваемой модели в общем виде. Предполагается, что всего имеется N пунктов (материальных благ, спорных вопросов, видов работ и пр.), совокупность разногласий по которым и составляет рассматриваемый конфликт между двумя участниками А и В. Натуральные (т.е. целые положительные) числа $a_1, \ldots, a_N$ и $b_1, \ldots, b_N$ представляют собой ***сравнительную важность пунктов***, составляющих конфликт, для участников А и В. Без ограничения общности можно считать, что

$$\sum_{i=1}^{N} a_i = \sum_{i=1}^{N} b_i = 100, \tag{1}$$

интерпретируя эти числа как проценты важности соответствующих пунктов.

Делёж формально представляет собой вектор $x = (x_1, \ldots, x_N)$, где x_i – вещественные числа, такие что $0 \leq x_i \leq 1$ ($i = 1, \ldots, N$). Содержательно x_i представляет собой доля i-го пункта, получаемую при дележе x участником А (при этом участник В получает долю $(1–x_j)$ того же пункта). Если i-й пункт представляет собой материальное благо (типа денег, участка земли и пр.), то понятие доли достаточно ясно. В других случаях оно требует дополнительного соглашения между участниками. Например, при разводе супругов, каждый из которых желает как можно больше общаться с их единственным ребёнком, таким конфликтным пунктом является время, а доля x_i есть просто доля времени, которое ребёнок проводит с родителем А. В США в таких случаях достаточно часто ребёнок проводит учебное время с одним из родителей, а каникулы и праздники – с другим, и при этом доли определяются, как 0,75 и 0,25. В любом случае доли x_i и $1–x_j$ интерпретируются, как степени удовлетворённости участников договорённостью по i-му пункту.

Выигрыши $G_A(x)$ и $G_B(x)$ участников А и В при дележе $x = (x_1, \ldots, x_N)$ определяются формулами

$$G_A(x) = \sum_{i=1}^{N} a_i x_i, \quad G_B(x) = \sum_{i=1}^{N} b_i (1 - x_i). \tag{2}$$

Независимо от «содержания» конфликтных пунктов эти числа всегда лежат между 0 и 100 и по сути дела представляют собой выраженные в 100-балльной шкале степени удовлетворённости участников данным дележом x «в целом». В частности, если всё достаётся участнику А, то $G_A(x) = 100$, $G_B(x) = 0$, а если участнику В, то $G_A(x) = 0$, $G_B(x) = 100$. Вряд ли участник с нулевым выигрышем будет считать такой делёж справедливым, да и вообще он вряд ли на него согласится добровольно. Из формул (2) и (1) непосредственно следует, что выигрыш участника Вс точки зрения участникаА (т.е. по оценкам А)

$$\sum_{i=1}^{N} a_i (1 - x_i) = 100 - G_A(x), \tag{3a}$$

а выигрыш участника Ас точки зрения участникаВ (т.е. по оценкам В)

$$\sum_{i=1}^{N} b_i x_i = 100 - G_B(x). \tag{3b}$$

Брамс и Тэйлор определили несколько важных свойств дележей, основываясь на выигрышах $G_A(x)$ и $G_B(x)$ участников А и В. Положим $G(x) = (G_A(x), G_B(x))$ и назовём вектор $G(x)$ (имеющий две компоненты) ***общим выигрышем*** от дележа x.

Делёж x называется ***свободным от зависти***, если

$$G_A(x) \geq 50 \tag{4a}$$

и

$$G_B(x) \geq 50. \tag{4b}$$

Из формул (4a) и (3a) сразу получаем, что с точки зрения участника А выигрыш участника В не превосходит его выигрыша, т.е. он не завидует тому, что получил участник В при дележе x; по той же причине участник В не завидует участнику А, а сам делёж и назван свободным от зависти.

Делёж x называется ***равноценным***, если

$$G_A(x) = G_B(x), \tag{5}$$

т.е. оба участника в равной степени удовлетворены дележом x.

Делёж x называется ***эффективным по Парето*** (или ***Парето-эффективным***), если общий выигрыш $G(x)$ недоминируем по Парето на множестве всех возможных общих выигрышей. Формально это означает, что для любого другого дележа y из $G_A(x) > G_A(y)$ следует, что $G_B(x) < G_B(y)$, а из $G_B(x) > G_B(y)$ следует, что $G_A(x) < G_A(y)$. Другими словами, результаты дележа x нельзя улучшить для одного участника, не ухудшив для другого, при любом другом дележе y.

2.2. Метод подстраивающегося победителя (ПП). Делёж x называется ***справедливым***, если он является одновременно свободным от зависти, равноценным и Парето-эффективным. Основные естественные вопросы, возникающие при исследовании справедливых дележей, являются такими же, как и при исследовании других формально определённых математических объектов: существуют ли справедливые дележи? если да, то как их находить? какими ещё интересными свойствами (кроме указанных в определениях) они обладают? если они существуют не всегда, то при каких условиях? как их можно модифицировать, чтобы гарантировать существование дележей при новых условиях? и т.д.

В данном случае Брамсом и Тэйлором в рамках сформулированных выше допущений даны исчерпывающие ответы на эти вопросы, Кратко сформулируем эти ответы. Справедливый делёж существует для любых сравнительных важностей $a_1, \ldots, a_N$ и $b_1, \ldots, b_N$. Он легко находится предложенным авторами методом «подстраивающегося победителя» (ПП). Важным свойством найденного методом ПП справедливого дележа является следующее: все пункты (кроме, быть может, одного, определяемого в процессе работы метода ПП) достаются целиком одному или другому участнику, и только один пункт действительно может делиться между ними.

Приведём описание алгоритма ПП. Для упрощения изложения примем соглашение о том, что сумма $\sum_{i=p}^{q} c_i = 0$, если $q<p$ (не может быть отрицательного числа слагаемых).

Предварительный шаг. Пункты, образующие конфликт, упорядочиваются так, чтобы выполнялось условие

$$a_1/b_1 \geq a_2/b_2 \geq \ldots \geq a_N/b_N. \quad (6)$$

Основные шаги

1. Положим $r = 0$.
2. Положим $r = r + 1$.
3. Проверяем условие

$$\sum_{i=1}^{r} a_i \geq \sum_{i=r+1}^{N} b_i. \quad (7)$$

Если (7) выполняется, то сделаем следующее.

3.1. Положим

$$x_r = \left(\sum_{i=r}^{N} b_i - \sum_{i=1}^{r-1} a_i\right)/(a_r+b_r). \quad (8)$$

3.2. Положим $x_i = 1$ ($i = 1, \ldots, r-1$), $x_i = 0$ ($i = r+1, \ldots, N$).

3.3. Стоп. Алгоритм прекращает работу.

В противном случае переходим на шаг 2.

В силу равенства (1) легко понять, что условие (7) обязательно будет выполнено при каком-либо r. Нетрудно проверить, что x_r, определяемое формулой (8), лежит между 0 и 1. Более сложным является центральное в данном разделе

Утверждение 2. Делёж, построенный описанным выше алгоритмом ПП, является справедливым для любых сравнительных важностей $a_1, \ldots, a_N$ и $b_1, \ldots, b_N$. Единственный пункт, который может делиться в построенном платеже – это r-ый пункт, где r– минимальный индекс, удовлетворяющий условию (7) ■

Пример 5. Рассмотрим задачу дележа, данные которой представлены в таблице 1. После

Таблица 1

пп	A	B	a_i / b_i
1	26	12	2,17
2	11	28	0,39
3	27	10	2,70
4	13	21	0,62
5	23	29	0,79

Таблица 2

пп	A	B	a_i / b_i
1(3)	27	10	2,70
2(1)	26	12	2,17
3(5)	23	29	0,79
4(4)	13	21	0,62
5(2)	11	28	0,39

перенумерации пунктов в соответствии с формулой (6) получим задачу, представленную в таблице 2 (в 1-м столбце в скобках указаны исходные номера пунктов).

Далее последовательно проверяем, начиная с $r = 1$, условия (7), используя нумерацию пунктов из таблицы 2. При $r = 1$ имеем

$\sum_{i=1}^{r} a_i = a_1 = 27$, $\sum_{i=r+1}^{N} b_i = b_2 + b_3 + b_4 + b_5 = 12 + 29 + 21 + 28 = 90$. Так как $27 < 90$, то условие (7) не выполняется.

Далее, при $r = 2$,

$\sum_{i=1}^{r} a_i = a_1 + a_2 = 27 + 26 = 53$, $\sum_{i=r+1}^{N} b_i = b_3 + b_4 + b_5 = 29 + 21 + 28 = 78$. Так как $53 < 78$, то условие (7) не выполняется.

Далее, при$r = 3$,

$\sum_{i=1}^{r} a_i = a_1 + a_2 + a_3 = 27 + 26 + 23 = 76$, $\sum_{i=r+1}^{N} b_i = b_4 + b_5 = 21 + 28 = 49$. Так как $76 \geq 49$, то условие (7) выполняется. Поэтому, в соответствии с операциями на шаге 3, получаем

$x_r = \left(\sum_{i=r}^{N} b_i - \sum_{i=1}^{r-1} a_i\right)/(a_r+b_r) = (78 - 53)/(23 + 29) = 25/52 = 0{,}48$.

Таким образом, построенный делёж таков: x = (1; 1; 0,48; 0; 0). Это означает, что 1-ый и 2-ой пункты целиком достаются участнику A, 4-ый и 5-ый пункты – участнику B; 3-ий пункт делится между ними в пропорции 0,48:0,52. Переходя к исходной нумерации пунктов, приведённой в левом столбце таблицы 2, окончательно получаем следующее. Участник A целиком получает 1-ый и 3-ий пункты, участник B – 2-ой и 4-ый пункты, а 5-ый пункт делится между участниками в той же пропорции 0,48:0,52. Поэтому окончательный делёж принимает вид: x = (1; 0; 1; 0; 0,48).

Зная делёж x, можно по формулам (2) подсчитать выигрыши участников. Имеем

$G_A(x) = \sum_{i=1}^{N} a_i x_i =$ 26•1+11•0+27•1+13•0+23•0,48 ≈ 64,06;

$G_B(x) = \sum_{i=1}^{N} b_i(1 - x_i)$ = 12•0+28•1+10•0+21•1+29•0,52 ≈ 64,06.

Примерное равенство выигрышей здесь не случайное совпадение, а следствие утверждения 2. Если считать не в десятичных, а в простых дробях, получится точное равенство. Поскольку построенный делёж является справедливым, то оба участника получают одинаковую сумму баллов просто по определению справедливого дележа. Из формулы (5) ясно также, что построенный делёж свободен от зависти. В силу того же утверждения 2 делёж x Парето-эффективен ■

Пример 6. Рассмотрим задачу дележа, данные которой представлены в таблице 3. После

Таблица 3

пп	A	B	a_i / b_i
1	2	20	2,17
2	3	20	0,39
3	2	20	2,70
4	90	20	0,62
5	3	20	0,79

Таблица 4

пп	A	B	a_i / b_i
1(4)	90	20	4,50
2(2)	3	20	0,15
3(5)	3	20	0,15
4(1)	2	20	0,10
5(3)	2	20	0,10

перенумерации пунктов в соответствии с формулой (6) получим задачу, представленную в таблице 4 (в 1-м столбце в скобках указаны исходные номера пунктов). Далее последовательно проверяем, начиная с r = 1, условия (7), используя нумерацию пунктов из таблицы 4. При r = 1 имеем $\sum_{i=1}^{r} a_i = a_1 = 90$, $\sum_{i=r+1}^{N} b_i = b_2 + b_3 + b_4 + b_5 = 20 + 20 + 20 + 20 = 80$. Так как 90 ≥ 80, то условие (7) выполняется. Поэтому, в соответствии с операциями на шаге 3, получаем

$x_1 = (\sum_{i=r}^{N} b_i - \sum_{i=1}^{r-1} a_i)/(a_r+b_r) = (100 - 0)/(90 + 20) = 100/110 = 10/11$; $x_2 = x_3 = x_4 = x_5 = 0$.

Возвращаясь к исходной нумерации пунктов, приведённой в левом столбце таблицы 4, окончательно получаем следующее. Участник A получает только 10/11 4-го пункта, участник B – 1/11 4-го пункта и все остальные пункты целиком. Поэтому окончательный делёж принимает вид: x = (0; 0; 0; 0,91; 0). Зная делёж x, можно по формулам (2) подсчитать выигрыши участников. Имеем

$G_A(x) = \sum_{i=1}^{N} a_i x_i =$ 2•0+ 3•0+ 2•0+90•(10/11)+3•0 = 81 $^9/_{11}$;

$G_B(x) = \sum_{i=1}^{N} b_i(1 - x_i)$ = 20•1+20•1+20•1+20•(1/11)+20•1 =.81 $^9/_{11}$.

Как и в примере 5, легко проверить, что данный делёж является справедливым. Заметим, что выигрыш обоих участников значителен. Это происходит из-за большого различия в оценках ■

2.3. Неделимые пункты. Задача справедливого дележа часто возникает при слиянии двух компаний (см. [3]). Речь идёт не столько о финансовых проблемах (которые обычно разрешаются в соответствии с имеющимися у компаний акциями и другими активами), сколько о так называемых «социальных» вопросах: как власть, должности и местопребывание делятся между менеджерами сливающихся компаний. Во многих случаях эти вопросы таковы:

название объединённой компании;

местопребывание штаб-квартиры;

назначение на должности президента и генерального директора;

сокращение служащих с целью избежать дублирования операций после слияния.

Пример 7. Важности, которые каждая из двух сливающихся фирм A и B придаёт указанным выше пунктам, приведены в таблице 5. Заметим, что речь идёт именно о важности, а не о положительном или отрицательном отношении к данному пункту. Например, если по пункту сокращений фирмы указывают 5% и 15%, то это значит, что сокращение 3X сотрудников 1-ой фирмой равносильно сокращению X сотрудников 2-ой фирмой (для неё это более значимый вопрос). Конечно, назначение на должности президента и гендиректора имеет для обеих фирм положительное значение. В таблице 6 приведены те же данные в порядке, предписанном алгоритмом ПП.

Таблица 5

Пункт	A	B
1. Название	10	25
2. Штаб-квартира	20	35
3. Назначение президента	15	20
4. Назначение директора	25	10
5. Сокращения	30	10
Всего	100	100

Таблица 6

Пункт	A	B
1. Сокращения	30	10
2. Назначение директора	25	10
3. Назначение президента	15	20
4. Штаб-квартира	20	35
5. Название	10	25
Всего	100	100

Воспользуемся алгоритмом ПП. Проводя вычисления аналогично вычислениям в примерах 4 и 5, получим делёж $x = (0; 0; {}^5/_7; 1; 1)$. Это значит, что фирма А назначает директора и не проводит сокращения, фирма В определяет название и выбирает место для штаб-квартиры. Наконец, назначение на должность президента делится между фирмами в пропорции 5:2. Выигрыши обеих фирм равны при этом $65\,{}^5/_7$■

Однако указанный в примере 6 делёж, хотя и является формально справедливым, вряд ли реализуем. Действительно, трудно представить должность президента (как и большинство других должностей) разделённой в некоторой пропорции (например, представитель одной фирмы является президентом в течение 5 лет, а другой – в течение 2 лет). Суть дела в том, что не только в данной ситуации, но и во многих других, некоторые (а иногда и все) пункты неделимы. Поэтому метод ПП может оказаться неприменимым – как раз тогда, когда неделимым оказывается тот единственный пункт, который должен делиться в соответствии с методом ПП. Более того, справедливого (в указанном выше смысле) дележа может не существовать. Предположим, что в примере 6 неделимым является только 4-ый пункт (тот, который оценивается участником А в 90%). Действительно, если этот пункт достаётся целиком участнику А, то при любом дележе других пунктов его выигрыш не меньше 90, а выигрыш участника В не превосходит 80. Если же этот пункт достаётся участнику В, то его выигрыш то при любом дележе других пунктов не меньше 20, а выигрыш участника А не превосходит 10. Таким образом, в этой ситуации равноценность и, следовательно, справедливость дележа не может быть обеспечена.

В общем случае, т.е. при любом распределении делимых и неделимых пунктов, необходимые и достаточные условия существования справедливого дележа (в терминах значений важности $a_1, \ldots, a_N$ и $b_1, \ldots, b_N$ всех пунктов) найдены сравнительно недавно (см. литературу в конце части) и здесь не приводятся. Ниже рассматривается случай 5-и пунктов, из которых только один делим, а остальные 4 неделимы. Именно такая ситуация имеет место в примере 7. В следующем примере приводится простой алгоритм, позволяющий найти справедливый делёж, если он есть, или установить его отсутствие.

Пример 8. Рассмотрим задачу справедливого дележа при данных в таблице 7 оценках важности пяти пунктов участниками А и В, из которых только выделенный жирным 2-ой пункт является делимым.

Таблица 7

пп	A	B
1	30	11
2	**12**	**28**
3	16	22
4	25	20
5	17	19

Поскольку 4 пункта (1-й, 3-ий, 4-ый и 5-ый) являются неделимыми, каждый из них целиком достаётся одному из участников. Поэтому распределение неделимых пунктов можно задать двоичным вектором $\sigma = (\sigma_1, \sigma_3, \sigma_4, \sigma_5)$, где

$$\sigma_i = \begin{cases} 1, \text{если } i\text{-ый пункт достаётся участнику А} \\ 0, \text{если } i\text{-ый пункт достаётся участнику В} \end{cases} \quad (i = 1, 3, 4, 5)$$

Выигрыши участников от неделимых пунктов при распределении $\sigma = (\sigma_1, \sigma_3, \sigma_4, \sigma_5)$ даются формулами

$V_A = 30\sigma_1+16\sigma_3+25\sigma_4+17\sigma_5$,
$V_B = 11(1–\sigma_1)+22(1–\sigma_3)+20(1–\sigma_4)+19(1–\sigma_5)$.

1. Вычислим выигрыши участников только от неделимых пунктов при всех распределениях $\sigma = (\sigma_1, \sigma_3, \sigma_4, \sigma_5)$ (их всего 16) и запишем их в следующую таблицу.

σ_1	σ_3	σ_4	σ_5	V_A	V_B	
0	0	0	0	0	72	–
0	0	0	1	17	53	–
0	0	1	0	25	52	–
0	0	1	1	42	33	*
0	1	0	0	16	50	–
0	1	0	1	33	31	*
0	1	1	0	41	30	*
0	1	1	1	58	11	–

σ_1	σ_3	σ_4	σ_5	V_A	V_B	
1	0	0	0	30	61	–
1	0	0	1	47	42	
1	0	1	0	55	41	
1	0	1	1	72	22	–
1	1	0	0	46	39	*
1	1	0	1	63	20	–
1	1	1	0	71	19	–
1	1	1	1	88	0	–

При добавлении какой-либо части 2-го пункта к выигрышу V_A он не может увеличиться больше, чем на 12. Аналогично, выигрыш V_B не может увеличиться больше, чем на 28. Поэтому все строчки таблицы, в которой $V_B – V_A > 12$ и все строчки таблицы, в которой $V_A – V_B > 28$, не соответствуют справедливому дележу (в них разница между выигрышами участников не может быть компенсирована за счёт деления одного делимого 2-го пункта). Поставим в самом правом столбце таблицы знак «–» во всех таких строчках. Если все 16 строчек таблицы содержат минусы, то это означает отсутствие справедливого дележа при данных условиях.

2. Рассмотрим теперь все строчки таблицы, в правой позиции которых нет минусов. Слева от правого столбца стоит пара чисел – выигрыши V_A и V_B при соответствующем распределении неделимых пунктов. В данном случае эти пары таковы: (42,33), (33,31), (41,30), (47,42), (55,41), (46,39). Сравним, например, пару (47,42) с парой (46,39). Если распределение неделимых пунктов даёт выигрыши 47 и 42 (см. таблицу выше), то при любом делении 2-го пункта оба участника получают больше, чем при том же самом делении 2-го пункта и распределении неделимых пунктов, дающих выигрыши 46 и 39. Это рассуждение показывает, что все доминируемые по Парето пары можно далее не рассматривать. Поставим знак * в строчки, соответствующие таким доминируемым парам выигрышей. В результате останутся ничем не помеченные строчки, содержащие недоминируемые по Парето пары (47,42) и (55,41).

3. Каждую из оставшихся пар надо проанализировать следующим образом.

3.1. Пара (47,42) соответствует дележу, при котором участник А получает пункты 1 и 5 (в сумме 47 баллов), а участник В – пункты 3 и 4 (в сумме 42 балла). Осталось разделить делимый 2-ой пункт. Пусть x – доля 2-го пункта, получаемая участником А. Из условия равенства баллов имеем

$47 + 12x = 42 + 28(1–x)$.

Решая уравнение, находим $x = 23/40 = 0,575$; далее $47 + 12*0,575 = 53,9$ и каждый участник получает по 53,9 балла.

3.2. Пара (55,41) соответствует дележу, при котором участник А получает пункты 1 и 4 (в сумме 55 баллов), а участник В – пункты 3 и 5 (в сумме 41 балл). Из условия равенства баллов имеем

$55 + 12x = 41 + 28(1–x)$.

Решая уравнение, находим $x=14/40=0,35$; далее $55+12*0,35=59,2$ и каждый участник получает по 59,2 балла. Поскольку во 2-ом случае выигрыш участников больше, то это решение и соответствует справедливому дележу. В данном случае справедливый делёж x = (1; 0,35; 0;1; 0).

Если же неотмеченным в таблице оказывается только один вариант, то он и есть оптимальный. Но и в этом случае надо решить аналогичное уравнение и чётко указать, кто что получает и чему равен выигрыш участников (напомним, что по определению справедливого дележа выигрыши участников должны совпадать) ■

Пример 9. Продолжение примера 7. Рассмотрим задачу справедливого дележа при данных в таблице 5 оценках важности пяти пунктов участниками А и В. Эти же данные приведены в таблице 8 без названий пунктов. Последняя строчка, соответствующая единственному делимому пункту – сокращениям – выделена жирным.

Таблица 8

Пункт	A	B
1	10	25
2	20	35
3	15	20
4	25	10
5	**30**	**10**
Всего	100	100

Применим метод, подробно описанный в примере 8. Выигрыши участников от неделимых пунктов при распределении $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ даются формулами
$V_A = 10\sigma_1+20\sigma_2+15\sigma_3 +25\sigma_4$,
$V_B = 25(1–\sigma_1)+35(1–\sigma_2)+20(1–\sigma_3)+10(1–\sigma_4)$.

1. Вычислим выигрыши участников только от неделимых пунктов при всех распределениях $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ (их всего 16) и запишем их в следующую таблицу.

σ_1	σ_2	σ_3	σ_4	V_A	V_B	
0	0	0	0	0	90	–
0	0	0	1	25	80	–
0	0	1	0	15	70	–
0	0	1	1	40	60	
0	1	0	0	20	55	–
0	1	0	1	45	45	
0	1	1	0	35	35	
0	1	1	1	60	25	–

σ_1	σ_2	σ_3	σ_4	V_A	V_B	
1	0	0	0	10	65	–
1	0	0	1	35	55	
1	0	1	0	25	45	
1	0	1	1	50	35	–
1	1	0	0	30	30	
1	1	0	1	55	20	–
1	1	1	0	45	10	–
1	1	1	1	70	0	–

При добавлении какой-либо части 5-го пункта к выигрышу V_A он не может увеличиться больше, чем на 30. Аналогично, выигрыш V_B не может увеличиться больше, чем на 10. Поэтому все строчки таблицы, в которой $V_B - V_A > 30$ и все строчки таблицы, в которой $V_A - V_B > 10$, не соответствуют справедливому дележу (в них разница между выигрышами участников не может быть компенсирована за счёт деления одного делимого 5-го пункта). Поставим в самом правом столбце таблицы знак «–» во всех таких строчках..

2. Рассмотрим теперь все строчки таблицы, в правой позиции которых нет минусов. В данном случае эти пары таковы: (40,60), (45,45), (35,35), (35,55), (25,45), (30,30). Из этих 6-и пар недоминирумыми по Парето будут пары (40,60) и (45,45). Все остальные пары доминируемы хотя бы одной из этих двух.

3.1. Пара (40,60) соответствует дележу, при котором участник А получает пункты 3 и 4 (в сумме 40 баллов), а участник В – пункты 1 и 2 (в сумме 60 баллов). Осталось разделить делимый 5-ой пункт. Пусть x – доля 5-го пункта, получаемая участником А. Из условия равенства баллов имеем
$40 + 30x = 60 + 10(1–x)$.
Решая уравнение, находим $x = 0{,}75$; далее $40 + 12*0{,}75 = 62{,}5$ и каждый участник получает по 62,5 балла.

3.2. Пара (45,45) соответствует дележу, при котором участник А получает пункты 2 и 4 (в сумме 45 баллов), а участник В – пункты 1 и 3 (в сумме 45 баллов). Из условия равенства баллов имеем
$45 + 30x = 45 + 10(1–x)$.
Решая уравнение, находим $x = 0{,}25$; далее $45 + 30*0{,}25 = 52{,}5$ и каждый участник получает по 52,5 балла.

Поскольку во 1-ом случае выигрыш участников больше, то это решение и соответствует справедливому дележу. В данном случае справедливый делёж x = (0;; 0; 1; 1; 0,75). Заметим, что выигрыш 62,5 балла меньше выигрыша $65^5/_7$, получаемого методом ПП. Это неудивительно, поскольку решение, в отличие от метода ПП, теперь ищется не на множестве всех дележей, а на вложенном в него множестве дележей, удовлетворяющих дополнительным условиям на деление пунктов 1 – 4 ■

Пример 10. Рассмотрим задачу дележа с исходными данными, приведёнными в таблице 9:
Таблица 9

Пункт	A	B
1	10	30
2	10	20
3	35	18
4	30	20
5	15	12
Всего	100	100

В данном случае предполагается, что пункты 1 и 2 делимы, а пункты 3, 4 и 5 – неделимы. Рассмотрим следующие два дележа: $x = (0;\ 2/3;\ 1;\ 0;\ 1)$, $y = (0;\ 0;\ 1;\ 1;\ 0)$. При дележе xучастник A получает $^2/_3$ пункта 2 и пункты 3 и 5 целиком; участник B – $^1/_3$ пункта 2 и пункты 1 и 4 целиком; оба выигрыша равны $56^2/_3$. При дележеy участник A получает пункты 3 и 4 целиком, участник B – пункты 1, 2 и 5 целиком; выигрыш участника A равен 65, а выигрыш участника B – 62. Оба дележа свободны от зависти, однако делёж x не является Парето-эффективным (при дележе yоба участника получают больше), а делёж y не является равноценным. Поэтому оба дележа не являются справедливыми в смысле введённого определения.

Можно убедиться, что в данном случае справедливого дележа просто не существует. Вопрос о том, какой из двух рассмотренных дележей является «более» справедливым, не решается в рамках математики. Как и в других подобных случаях, суть дела в содержательных аспектах конкретной ситуации ■

3. Пропорциональное представительство

Суть пропорционального представительства состоит в выделении мест в сравнительно малочисленном органе (парламенте, комитете, совете директоров и т.д.) пропорционально гораздо более значительным величинам (числу проголосовавших за ту или иную партию, количеству акций, имеющихся у той или иной группы акционеров и пр.). Рассмотрим иллюстративный пример, демонстрирующий рассматриваемые в данном разделе ситуации.

Пример 11. Наксон (Naxxon), Ароко (Aroco) и Евробайл (Eurobile) образовали консорциум для строительства морской нефтедобывающей платформы у побережья Африки. Компании решили сформировать совет из 9 членов для руководства проектом; число представителей каждой компании в совете должно быть пропорционально числу держателей акций. Наксон имеет 4700 держателей, Ароко – 3700 держателей и Евробайл – 1600 держателей.

Поскольку суммарное число держателей акций равно 10000, а Наксон владеет 4700 акций, то доля его представителей в совете должна составлять 4700 /10000 = 47%, доля Ароко должна составлять 37% и доля Евробайла – 16%. Для обеспечения точной пропорциональности Наксону должно быть выделено 0,47×9 = 4,23 места в совете, Ароко – 0,37×9 = 3,33 места в совете и Евробайлу – 0,16×9 = 1,44 места в совете. Понятно, что все места розданы: 4,23+3,33+1,44=9. Также понятно, что невозможно иметь дробное число представителей в каком бы то ни было органе. Естественно возникает достаточно часто встречающаяся задача – как разделить места пусть не точно пропорционально (это почти всегда невозможно), но в каком-то смысле близко к пропорциональному делению ■

Одним из хорошо известных и достаточно важных примеров пропорционального представительства представляет собой выделение в Палате Представителей Конгресса США числа мест различным штатам пропорционально численности их населения. Поскольку отцы-основатели желали обеспечить каждому гражданину равное представительство в управлении, они поместили требования к числу представителей от штатов в Конгрессе почти в самое начало конституции Соединённых Штатов. В разделе 2 главы 1 говорится:

«Представительство и прямые налоги должны быть поделены между всеми штатами, которые образуют Союз, в соответствии с численностью их населения. ... Эта численность должна быть определена в течение трёх лет после первого заседания Конгресса, и далее должна переопределяться один раз в десять лет, как того требует соответствующий Закон.»

Пропорциональное распределение числа конгрессменов между штатами является не такой простой задачей, как может показаться. Как при делении десяти подарков между тремя маленькими детьми, вопрос состоит в том, как решить, кому именно достанется дополнительное место в Конгрессе. Выступая в 1832 году в Палате Представителей, Даниэль Вебстер сказал:

«Конституция ... не требует идеальной пропорциональности между числом представителей и населением каждого штата, поскольку удовлетворить это требование невозможно, но требует такой близости между соответствующими отношениями, какой только можно достичь.»

В данном разделе излагаются различные подходы к проблеме пропорционального представительства – как те, которые реально использовались в Конгрессе Соединённых Штатов, так и некоторые другие, – и даётся сравнительный анализ их достоинств и недостатков. Обращаем особое внимание на то, что сама по себе проблема пропорционального представительства возникает не только в выборных законодательных органах. Та же проблема возникает и при определении состава правлений крупных компаний, где каждый член представляет акционеров с некоторым суммарным числом акций (как в примере 11); при слиянии компаний; в профсоюзах; в творческих и профессиональных организациях, и т.д. – словом, практически везде, где сравнительно небольшой выборный или назначаемый орган представляет интересы различных групп людей. Как и в других разделах данной главы, речь идёт о правилах взаимодействия различных участников, которыми в данном случае являются большие группы людей, желающие иметь честное и справедливое (с их точки зрения) представительство.

3.1. Метод Гамильтона. Александр Гамильтон (1757 – 1804), один из наиболее активных разработчиков Конституции США, предложил метод, использованный при распределении мест в 1-го состава Конгресса (и ещё в течение 150 лет). Для изложения этого метода рассмотрим следующий иллюстративный пример.

Пример 12. Продолжение примера 11. Представим проведённые в примере 11 вычисления в следующей таблице. Однако ни одна компания не может иметь часть члена совета. Поэтому,

Таблица 10

Компания	**Процент держателей акций**	**Пропорциональное число мест в совете**	**Число мест по методу Гамильтона**
Наксон	47	0,47×9 = 4,23	4
Ароко	37	0,37×9 = 3,33	3
Евробайл	16	0,16×9 = 1,44	2
Всего	100	9,00	9

поскольку Наксону выделено ровно 4,23 члена, он будет иметь 4 или 5. Число 4 называется ***целой частью*** числа 4,23; 0,23 – ***дробной частью*** числа 4,23. Если мы выделим каждой компании целую часть соответствующей ей доли, то (как следует из таблицы 10) Наксон будет иметь четырёх членов, Ароко – трёх и Евробайл – одного. Таким образом, всего имеется 8 членов вместо требуемых 9. Необходимо решить, какой компании отдать оставшееся место. Представляется достаточно разумным отдать это место Евробайлу, так как у него имеется наибольшая дробная часть – 0,44. Таким образом, у Евробайл получает 2 места, Ароко – 3 места и Наксон – 4 места, что показано в правом столбце таблицы 10 ■

Именно так, как в примере 12, работает метод Гамильтона и в общем случае. Прежде чем описать реализующий этот метод алгоритм, введём необходимые понятия ***стандартного делителя*** и ***стандартной квоты***. Обозначим через s численность некоторой представляемой группы, через t– общую численность всех представляемых групп, через n – общее число мест в представляющем эти группы органе. Понятно, что точное число x представителей от данной группы выражается простой формулой

$$x = \frac{s}{t}\times n = s\times\frac{n}{t} = \frac{s}{\frac{t}{n}}. \qquad (9)$$

Величина $\frac{t}{n}$, являющаяся знаменателем в правой части равенства (1), называется стандартным делителем, а число x в левой части равенства (1) называется стандартной квотой, соответствующей данной группе. В словесном виде:

$$\text{стандартная квота} = \frac{\text{численность группы}}{\text{стандартный делитель}}.$$

Применительно к американской конституции вместо общего термина «группа» можно говорить о штате и, соответственно, о населении штата, населении страны и Палате Представителей. В ситуациях типа рассмотренной в примерах 11 и 12 можно говорить о держателях акций из данной фирмы, совете директоров, и т.д. Теперь можно дать описание алгоритма.

Алгоритм метода Гамильтона

1. Найти стандартные квоты для всех представляемых групп.
2. Для каждой стандартной квоты найти целую и дробную часть (напомним, что целой частью любого числа z называется максимальное целое число, не превосходящее данное число z, а дробной частью – разница между числом z и его целой частью).
3. Каждой представляемой группе выделить число мест, равное целой части стандартной квоты данной группы.
4. Занумеровать все группы в порядке убывания дробных частей стандартных квот.
5. Добавить одно место к числу ранее выделенных для группы мест, начиная с 1-ой (в построенном порядке) группы.
6. Добавления на шаге 5 прекращаются, как только суммарная численность выделенных мест окажется равной числу n – числу мест в данном органе, представляющим рассмотренные группы ■

Пример 13. Продолжение примеров 11 и 12. Предположим, что консорциум решил расширить совет до десяти членов. Использование метода Гамильтона для распределения мест в совете, состоящем из десяти членов, проиллюстрировано в таблице 10.

Таблица 11

Компания	Процент держателей акций	Пропорциональное число мест в совете	Число мест по методу Гамильтона
Наксон	47	0,47×10 = 4,7	5
Ароко	37	0,37×10 = 3,7	4
Евробайл	16	0,16×10 = 1,6	1
Всего	100	10,0	10

Заметим, что в данном случае Евробайл потерял одно место в совете, несмотря на то, что все доли держателей акций остались теми же, а число мест даже выросло ■

В 1881 Конгресс сделал удивительное открытие. Численный состав Палаты Представителей был увеличен с 299 человек до 300 (т.е. всего на одно место). При перераспределении мест методом Гамильтона оказалось, что Алабама вместо 8 мест получила 7, при расчётах, базирующихся на той же самой численности населения во всех штатах. Конечно, в этом нет ничего удивительного (тот же эффект имеет место в искусственном примере 13), но в Палате Представителей США это случилось впервые. Более того, оказалось, что этот же эффект (получивший название Алабамского парадокса) повторился через несколько лет при увеличении численности Палаты Представителей с 359 до 360: одно место потерял Арканзас, то же самое случилось и со штатом Мэн.

3.2. Методы Джефферсона и Адамса. Эти два метода (связанные с именами 3-го и 2-го президентов США) используют вместо стандартного делителя $\frac{t}{n}$ (см. формулу (9)) его изменённое значение, называемое ***модифицированным делителем***. При делении численности группы s на модифицированный делитель получаем ***модифицированную квоту*** (вместо стандартной квоты).

Идея метода Джефферсона такова: округлять все квоты (до целых чисел) в ме́ньшую сторону. Если сумма мест окажется меньше, чем заданное суммарное число мест n, то надо уменьшить модифицированный делитель (при этом, в силу формулы (9), модифицированные квоты увеличатся); если сумма мест окажется больше, чем суммарное число мест n, то надо увеличить модифицированный делитель (при этом, в силу формулы (9), модифицированные квоты уменьшатся). Подбираем модифицированный делитель так, чтобы сумма мест в конце концов оказалась равной заданному число мест n (это всегда возможно).

Идея метода Адамса такова: округлять все квоты (до целых чисел) в бо́льшую сторону. Если сумма мест окажется больше, чем заданное суммарное число мест n, то надо увеличить модифицированный делитель (при этом, в силу формулы (9), модифицированные квоты уменьшатся); если сумма мест окажется меньше, чем суммарное число мест n, то надо уменьшить модифицированный делитель (при этом, в силу формулы (9), модифицированные квоты увеличатся). Подбираем модифицированный делитель так, чтобы сумма мест в конце концов оказалась равной заданному число мест n (это всегда возможно).

Проиллюстрируем оба метода на следующем примере.

Пример 14. При указанных ниже данных разделим 16 преподавательских позиций методом Джефферсона. Как уже говорилось, в методе Джефферсона вместо стандартного делителя

Колледж	Образования	Свободных искусств	Бизнеса	Всего
Число студентов	800	1300	1100	3200

$D=\frac{t}{n}$ (см. формулу (9)) используется его изменённое значение D_M, называемое модифицированным делителем. При делении численности s на модифицированный делитель (вместо стандартного делителя) получаем модифицированную квоту x_M (вместо стандартной квоты x).

Метод Джефферсона состоит из следующих шагов.

Шаг 0. Найти стандартный делитель $D = \frac{t}{n}$.

Шаг 1. Выбрать модифицированный делитель D_M ме́ньшим, чем стандартный делитель.

Шаг 2. По формуле $x_M = \frac{s}{D_M}$, полученной из формулы (9) подстановкой модифицированного делителя D_M и модифицированной квоты x_M вместо стандартного делителя и стандартной квоты, находим модифицированную квоту для всех колледжей.

Шаг 3. Округлить все найденные на шаге 2 квоты в ме́ньшую сторону.

Шаг 4. Если сумма округлённых квот окажется меньше, чем заданное суммарное число позиций n, то уменьшить модифицированный делитель и перейти к шагу 2. Если сумма округлённых квот окажется больше, чем заданное суммарное число позиций n, то увеличить модифицированный делитель и перейти к шагу 2. Если сумма округлённых квот окажется равной заданному суммарному числу позиций n, то округлённые квоты и представляют собой искомое распределение позиций.

Шаг 5. Остановка алгоритма.

Последовательные вычисления для данного случая показаны в следующей таблице, которая последовательно заполняется сверху вниз:

Колледж	Образования	Свободных искусств	Бизнеса	Модифицированный делитель
Число студентов	800	1300	1100	
Модифицированная квота	800:190≈4.21	1300:190≈6.84	1100:190≈5.79	190
Округлённая квота	4	6	5	15
Модифицированная квота	800:150 ≈5.33	1300:150≈8.67	1100:150≈7.33	150
Округлённая квота	5	8	7	20
Модифицированная квота	800:180 ≈4.44	1300:180≈7.22	1100:180≈6.11	180
Округлённая квота	4	7	6	17
Модифицированная квота	800:185 ≈4.32	1300:185≈7.03	1100:185≈5.95	185
Округлённая квота	4	7	5	16

Дадим необходимые пояснения. До начала заполнения таблицы выполняется шаг 0, т.е. вычисляется стандартный делитель $D = \frac{t}{n}$. Поскольку в данном случае $t = 3200$, $n = 16$, то $D = 200$.

На шаге 1 положим $D_M = 190 < 200$ и запишем $D_M = 190$ в правую колонку таблицы сразу под заголовком столбца «Модифицированный делитель».

На шаге 2 произведём указанные в алгоритме вычисления и заполним верхнюю строку с именем «Модифицированная квота».
На шаге 3 округлим найденные на шаге 2 числа в сторону уменьшения и запишем их в следующую строку таблицы. Поскольку сумма 15 меньше заданного числа 16, то в соответствии с шагом 4 алгоритма уменьшим D_M, положив $D_M = 150$, и вернёмся к шагу 2.
На шаге 2 произведём указанные в алгоритме вычисления и заполним следующую строку с именем «Модифицированная квота».
На шаге 3 округлим найденные на шаге 2 числа в сторону уменьшения и запишем их в следующую строку таблицы. Поскольку сумма 20 больше заданного числа 16, то в соответствии с шагом 4 алгоритма увеличим D_M, положив $D_M = 180$, и вернёмся к шагу 2.
На шаге 2 произведём указанные в алгоритме вычисления и заполним следующую строку с именем «Модифицированная квота».
На шаге 3 округлим найденные на шаге 2 числа в сторону уменьшения и запишем их в следующую строку таблицы. Поскольку сумма 17 больше заданного числа 16, то в соответствии с шагом 4 алгоритма увеличим D_M, положив $D_M = 185$, и вернёмся к шагу 2.
На шаге 2 произведём указанные в алгоритме вычисления и заполним следующую строку с именем «Модифицированная квота».
На шаге 3 округлим найденные на шаге 2 числа в сторону уменьшения и запишем их в следующую строку таблицы. Поскольку сумма оказалась равной заданному числу 16, то в соответствии с шагом 5 алгоритма прекращаем вычисления. Последняя строка таблицы даёт распределение позиций методом Джефферсона

Конечно, неформальный подбор модифицированного делителя может показаться слишком уж неформальным. На самом же деле теория таких поисковых методов детально разработана. Известны точные оптимальные (по числу шагов) алгоритмы. Однако освоение этих методов оказывается значительно более трудоёмким, чем простой подбор, который всё же гарантированно (и обычно за 2-3 шага) приводит к правильному результату. Заметим также, что при выполнении подбора у нас всегда получается «вилка»: $D_M = 190$ даёт меньше позиций, чем надо, а $D_M = 150$ даёт больше позиций, чем надо. Это даёт гарантию, что «правильное» значение D_M будет больше, чем 150, но меньше, чем 190. Следующая точка 180 уменьшила интервал неопределённости: «правильное» значение D_M будет между 180 и 190. Действительно, очередное значение 185 оказалось «правильным», т.е. приводящим к распределению всех позиций■

Пример 15. Метод Адамса отличается от метода Джефферсона только тем, что округления квот происходят путём увеличения до ближайшего целого, а не уменьшения, как в методе Джефферсона. Проиллюстрируем его использование на тех же исходных данных. Результаты вычислений будем записывать в аналогичную таблицу. Заметим, что стандартный делитель в этом случае является, естественно таким же самым, а в качестве начального значения возьмём $D_M = 220$ (бóльшее, чем стандартный делитель). Последовательные вычисления показаны в следующей таблице:

Колледж	Образования	Свободных искусств	Бизнеса	Модифицированный делитель
Число студентов	800	1300	1100	
Модифицированная квота	800:220≈3.64	1300:220≈5.91	1100:220≈5	220
Округлённая квота	4	6	5	15
Модифицированная квота	800:210 ≈3.81	1300:210≈6.19	1100:210≈5.24	210
Округлённая квота	4	7	6	17
Модифицированная квота	800:215≈3.73	1300:215≈6.05	1100:215≈5.12	215
Округлённая квота	4	7	6	17
Модифицированная квота	800:218≈3.67	1300:218≈5.96	1100:218≈5.05	218
Округлённая квота	4	6	6	16

Обратим внимание, что распределения позиций, найденные методом Джефферсона и методом Адамса, не совпадают. Это не ошибка: речь ведь не идёт о разных методах решения одной и той же формальной задачи, а о разных формализациях самого понятия справедливости ■

3.3. Метод Хантингтона-Хилла. Этот метод состоит в следующем. Обозначим через s_p численность p-ой группы, через k_p – число мест, выделенных этой группе в представительном органе. Отношение s_p к k_p представляет собой среднее число членов группы, приходящееся на одного представителя от данной группы. В идеале все эти числа должны совпадать; в реальности это невозможно, и их следует сделать как можно ближе друг к другу, в чём и состоит задача пропорционального представительства.

Положим $\beta_p = \frac{s_p}{k_p}$ (p = 1, ..., m), где m–число групп. Пусть i и j– номера двух групп. Положим

$$M = \min\{\beta_i, \beta_j\}, \tag{10}$$

$$\alpha_{ij} = \frac{|\beta_i - \beta_j|}{M} \quad (i, j = 1, ..., m;\ i \neq j). \tag{11}$$

Величина α_{ij} называется ***относительной несправедливостью*** для групп i и j. Распределение Хантингтона-Хилла определяется как такое распределение (т.е. набор чисел k_1, ..., k_m, в сумме равных n, где n–общее число мест), для которого величина $\max_{i,j} \alpha_{ij}$ принимает минимально возможное значение. Таким образом, распределение Хантингтона-Хилла минимизирует максимальную относительную несправедливость. Именно такой способ распределения мест между штатами в Палате Представителей принят в США с 1941 года, когда президент Франклин Делано Рузвельт подписал соответствующий закон.

На первый взгляд может показаться, что нахождение распределения Хантингтона-Хилла является вычислительно сложной задачей, требующей анализа всех возможных распределений и вычисления для каждого из них относительной несправедливости. Однако есть достаточно простой алгоритм, основанный на последовательном выделении мест для различных групп. Дадим его описание.

Алгоритм Хантингтона-Хилла.

1. Выделить по одному месту каждой группе.
2. Для каждой группы подсчитать ***индекс Хантингтона-Хилла***:

$$h = \frac{s^2}{k(k+1)},$$

где s – численность группы (например, население штата), k – число уже выделенных данной группе мест.

3. Предоставить очередное место группе с максимальным индексом.
4. Если все места в представительном органе уже заполнены, то алгоритм прекращает работу. В противном случае переходим на шаг 2 ■

Проиллюстрируем работу алгоритма Хантингтона-Хилла на данных из примера 11.

Пример 16. Напомним, что речь идёт о распределении 9 мест в совете пропорционально числу имеющихся у этих трёх компаний акций. Запишем исходные данные в первые три строки следующей таблицы 12:

Таблица 12

Компания	Наксон	Ароко	Євробайл
Процент акций	47	37	16
Начальное распределение 3-ёх мест	1	1	1
Индекс Хантингтона-Хилла	47^2/ (1*2) =1104,5	37^2/ (1*2) = 684,5	16^2/ (1*2) = 128
Распределение 4-ёх мест	2	1	1
Индекс Хантингтона-Хилла	47^2/ (2*3) ≈ 368	37^2/ (1*2) = 684,5	16^2/ (1*2) = 128
Распределение 5-и мест	2	2	1
Индекс Хантингтона-Хилла	47^2/ (2*3) ≈ 368	37^2/ (2*3) ≈ 228	16^2/ (1*2) = 128
Распределение 6-и мест	3	2	1

Индекс Хантингтона-Хилла	$47^2/(3*4) \approx 184$	$37^2/(2*3) \approx 228$	$16^2/(1*2) = 128$
Распределение 7-и мест	3	3	1
Индекс Хантингтона-Хилла	$47^2/(3*4) \approx 184$	$37^2/(3*4) \approx 114$	$16^2/(1*2) = 128$
Распределение 8-и мест	4	3	1
Индекс Хантингтона-Хилла	$47^2/(4*5) \approx 110$	$37^2/(3*4) \approx 114$	$16^2/(1*2) = 128$
Распределение 9-и мест	4	3	2

Таким образом, при делении методом Хантингтона-Хилла получаем распределение (4,3,2). При делении методом Гамильтона получили такое же распределение (см. пример 12). Такое же распределение получим методом Адамса. А метод Джефферсона даёт отличное от этого распределение (5,3,1). Заметим также, что в данном алгоритме расчёт индекса на каждом шаге делается только для одной группы – именно, той, которой добавлено место на предыдущем шаге ■

4. Задания

Задание 1. По набору предпочтений участников построить устойчивое паросочетание. См. пример 3 для образца ■

Варианты для задания 1

01 $P(m_1) = w_2, w_3, w_1;$ $P(w_1) = m_2, m_1, m_3;$ $P(m_2) = w_2, w_1, w_3;$ $P(w_2) = m_3, m_2, m_1;$ $P(m_3) = w_1, w_3, w_2;$ $P(w_3) = m_3, m_1, m_2.$	02 $P(m_1) = w_2, w_3, w_1;$ $P(w_1) = m_2, m_1, m_3;$ $P(m_2) = w_1, w_3, w_2;$ $P(w_2) = m_3, m_2, m_1;$ $P(m_3) = w_2, w_1, w_3;$ $P(w_3) = m_3, m_1, m_2.$
03 $P(m_1) = w_2, w_1, w_3;$ $P(w_1) = m_2, m_1, m_3;$ $P(m_2) = w_2, w_3, w_1;$ $P(w_2) = m_3, m_2, m_1;$ $P(m_3) = w_1, w_3, w_2;$ $P(w_3) = m_3, m_1, m_2.$	04 $P(m_1) = w_2, w_1, w_3;$ $P(w_1) = m_2, m_1, m_3;$ $P(m_2) = w_1, w_3, w_2;$ $P(w_2) = m_3, m_2, m_1;$ $P(m_3) = w_2, w_3, w_1;$ $P(w_3) = m_3, m_1, m_2.$
05 $P(m_1) = w_1, w_3, w_2;$ $P(w_1) = m_2, m_1, m_3;$ $P(m_2) = w_2, w_3, w_1;$ $P(w_2) = m_3, m_2, m_1;$ $P(m_3) = w_2, w_1, w_3;$ $P(w_3) = m_3, m_1, m_2.$	06 $P(m_1) = w_1, w_3, w_2;$ $P(w_1) = m_2, m_1, m_3;$ $P(m_2) = w_2, w_1, w_3;$ $P(w_2) = m_3, m_2, m_1;$ $P(m_3) = w_2, w_3, w_1;$ $P(w_3) = m_3, m_1, m_2.$
07 $P(m_1) = w_2, w_3, w_1;$ $P(w_1) = m_2, m_3, m_1;$ $P(m_2) = w_2, w_1, w_3;$ $P(w_2) = m_3, m_1, m_2;$ $P(m_3) = w_1, w_3, w_2;$ $P(w_3) = m_3, m_2, m_1.$	08 $P(m_1) = w_2, w_3, w_1;$ $P(w_1) = m_2, m_3, m_1;$ $P(m_2) = w_1, w_3, w_2;$ $P(w_2) = m_3, m_1, m_2;$ $P(m_3) = w_2, w_1, w_3;$ $P(w_3) = m_3, m_2, m_1.$
09 $P(m_1) = w_2, w_1, w_3;$ $P(w_1) = m_2, m_3, m_1;$ $P(m_2) = w_2, w_3, w_1;$ $P(w_2) = m_3, m_1, m_2;$ $P(m_3) = w_1, w_3, w_2;$ $P(w_3) = m_3, m_2, m_1.$	10 $P(m_1) = w_2, w_1, w_3;$ $P(w_1) = m_2, m_3, m_1;$ $P(m_2) = w_1, w_3, w_2;$ $P(w_2) = m_3, m_1, m_2;$ $P(m_3) = w_2, w_3, w_1;$ $P(w_3) = m_3, m_2, m_1.$
11 $P(m_1) = w_1, w_3, w_2;$ $P(w_1) = m_2, m_3, m_1;$ $P(m_2) = w_2, w_3, w_1;$ $P(w_2) = m_3, m_1, m_2;$ $P(m_3) = w_2, w_1, w_3;$ $P(w_3) = m_3, m_2, m_1.$	12 $P(m_1) = w_1, w_3, w_2;$ $P(w_1) = m_2, m_3, m_1;$ $P(m_2) = w_2, w_1, w_3;$ $P(w_2) = m_3, m_1, m_2;$ $P(m_3) = w_2, w_3, w_1;$ $P(w_3) = m_3, m_2, m_1.$
13 $P(m_1) = w_2, w_1, w_3;$ $P(w_1) = m_3, m_2, m_1;$ $P(m_2) = w_2, w_3, w_1;$ $P(w_2) = m_2, m_3, m_1;$ $P(m_3) = w_1, w_3, w_2;$ $P(w_3) = m_2, m_3, m_1.$	14 $P(m_1) = w_2, w_1, w_3;$ $P(w_1) = m_3, m_2, m_1;$ $P(m_2) = w_1, w_3, w_2;$ $P(w_2) = m_2, m_3, m_1;$ $P(m_3) = w_2, w_3, w_1;$ $P(w_3) = m_2, m_3, m_1.$
15 $P(m_1) = w_1, w_3, w_2;$ $P(w_1) = m_3, m_2, m_1;$ $P(m_2) = w_2, w_3, w_1;$ $P(w_2) = m_2, m_3, m_1;$ $P(m_3) = w_2, w_1, w_3;$ $P(w_3) = m_2, m_3, m_1.$	16 $P(m_1) = w_2, w_3, w_1;$ $P(w_1) = m_2, m_1, m_3;$ $P(m_2) = w_2, w_1, w_3;$ $P(w_2) = m_3, m_2, m_1;$ $P(m_3) = w_1, w_3, w_2;$ $P(w_3) = m_3, m_1, m_2.$
17 $P(m_1) = w_2, w_3, w_1;$ $P(w_1) = m_2, m_1, m_3;$ $P(m_2) = w_1, w_3, w_2;$ $P(w_2) = m_3, m_2, m_1;$ $P(m_3) = w_2, w_1, w_3;$ $P(w_3) = m_3, m_1, m_2.$	18 $P(m_1) = w_2, w_1, w_3;$ $P(w_1) = m_2, m_1, m_3;$ $P(m_2) = w_2, w_3, w_1;$ $P(w_2) = m_3, m_2, m_1;$ $P(m_3) = w_1, w_3, w_2;$ $P(w_3) = m_3, m_1, m_2.$

19 $P(m_1) = w_2, w_1, w_3;$ $P(w_1) = m_2, m_1, m_3;$ $P(m_2) = w_1, w_3, w_2;$ $P(w_2) = m_3, m_2, m_1;$ $P(m_3) = w_2, w_3, w_1;$ $P(w_3) = m_3, m_1, m_2.$	20 $P(m_1) = w_1, w_3, w_2;$ $P(w_1) = m_2, m_1, m_3;$ $P(m_2) = w_2, w_3, w_1;$ $P(w_2) = m_3, m_2, m_1;$ $P(m_3) = w_2, w_1, w_3;$ $P(w_3) = m_3, m_1, m_2.$
21 $P(m_1) = w_1, w_3, w_2;$ $P(w_1) = m_2, m_1, m_3;$ $P(m_2) = w_2, w_1, w_3;$ $P(w_2) = m_3, m_2, m_1;$ $P(m_3) = w_2, w_3, w_1;$ $P(w_3) = m_3, m_1, m_2.$	22 $P(m_1) = w_2, w_3, w_1;$ $P(w_1) = m_2, m_3, m_1;$ $P(m_2) = w_2, w_1, w_3;$ $P(w_2) = m_3, m_1, m_2;$ $P(m_3) = w_1, w_3, w_2;$ $P(w_3) = m_3, m_2, m_1.$
23 $P(m_1) = w_2, w_3, w_1;$ $P(w_1) = m_2, m_3, m_1;$ $P(m_2) = w_1, w_3, w_2;$ $P(w_2) = m_3, m_1, m_2;$ $P(m_3) = w_2, w_1, w_3;$ $P(w_3) = m_3, m_2, m_1.$	24 $P(m_1) = w_2, w_1, w_3;$ $P(w_1) = m_2, m_3, m_1;$ $P(m_2) = w_2, w_3, w_1;$ $P(w_2) = m_3, m_1, m_2;$ $P(m_3) = w_1, w_3, w_2;$ $P(w_3) = m_3, m_2, m_1.$
25 $P(m_1) = w_2, w_1, w_3;$ $P(w_1) = m_2, m_3, m_1;$ $P(m_2) = w_1, w_3, w_2;$ $P(w_2) = m_3, m_1, m_2;$ $P(m_3) = w_2, w_3, w_1;$ $P(w_3) = m_3, m_2, m_1.$	26 $P(m_1) = w_1, w_3, w_2;$ $P(w_1) = m_2, m_3, m_1;$ $P(m_2) = w_2, w_3, w_1;$ $P(w_2) = m_3, m_1, m_2;$ $P(m_3) = w_2, w_1, w_3;$ $P(w_3) = m_3, m_2, m_1.$
27 $P(m_1) = w_1, w_3, w_2;$ $P(w_1) = m_2, m_3, m_1;$ $P(m_2) = w_2, w_1, w_3;$ $P(w_2) = m_3, m_1, m_2;$ $P(m_3) = w_2, w_3, w_1;$ $P(w_3) = m_3, m_2, m_1.$	28 $P(m_1) = w_2, w_1, w_3;$ $P(w_1) = m_3, m_2, m_1;$ $P(m_2) = w_2, w_3, w_1;$ $P(w_2) = m_1, m_3, m_2;$ $P(m_3) = w_1, w_3, w_2;$ $P(w_3) = m_2, m_3, m_1.$
29 $P(m_1) = w_2, w_1, w_3;$ $P(w_1) = m_3, m_2, m_1;$ $P(m_2) = w_1, w_3, w_2;$ $P(w_2) = m_1, m_3, m_2;$ $P(m_3) = w_2, w_3, w_1;$ $P(w_3) = m_2, m_3, m_1.$	30 $P(m_1) = w_1, w_3, w_2;$ $P(w_1) = m_3, m_2, m_1;$ $P(m_2) = w_2, w_3, w_1;$ $P(w_2) = m_1, m_3, m_2;$ $P(m_3) = w_2, w_1, w_3;$ $P(w_3) = m_2, m_3, m_1.$
31 $P(m_1) = w_1, w_3, w_2;$ $P(w_1) = m_3, m_2, m_1;$ $P(m_2) = w_2, w_1, w_3;$ $P(w_2) = m_1, m_3, m_2;$ $P(m_3) = w_2, w_3, w_1;$ $P(w_3) = m_2, m_3, m_1.$	32 $P(m_1) = w_2, w_3, w_1;$ $P(w_1) = m_2, m_1, m_3;$ $P(m_2) = w_2, w_1, w_3;$ $P(w_2) = m_3, m_2, m_1;$ $P(m_3) = w_1, w_3, w_2;$ $P(w_3) = m_3, m_1, m_2.$

Задание 2. Найти (считая все пункты делимыми) справедливый делёж и соответствующие ему выигрыши алгоритмом ПП. В качестве образца использовать примеры 5 и 6 ■

Варианты для задания 2

пп	*A*	*B*
1	15	20
2	**15**	**21**
3	25	19
4	18	30
5	27	10

01

пп	*A*	*B*
1	**19**	**20**
2	17	19
3	19	37
4	25	15
5	20	9

02

пп	*A*	*B*
1	15	21
2	25	30
3	27	18
4	19	15
5	**14**	**16**

03

пп	*A*	*B*
1	**19**	**18**
2	10	27
3	30	25
4	15	24
5	26	6

04

пп	*A*	*B*
1	10	27
2	18	15
3	19	25
4	30	20
5	**23**	**13**

05

пп	*A*	*B*
1	19	20
2	**17**	**19**
3	19	37
4	25	15
5	20	9

06

пп	*A*	*B*
1	**15**	**20**
2	15	21
3	25	19
4	18	30
5	27	10

07

пп	*A*	*B*
1	**15**	**34**
2	20	12
3	21	19
4	15	25
5	29	10

08

пп	*A*	*B*
1	12	10
2	28	18
3	14	17
4	8	22
5	**38**	**33**

09

пп	*A*	*B*
1	8	23
2	20	16
3	**17**	**14**
4	32	12
5	23	35

10

пп	*A*	*B*
1	**26**	**12**
2	11	28
3	27	10
4	13	21
5	23	29

11

пп	*A*	*B*
1	30	11
2	12	28
3	16	22
4	25	20
5	**17**	**19**

12

пп	*A*	*B*
1	12	10
2	28	18
3	14	17
4	**8**	**22**
5	38	33

13

пп	*A*	*B*
1	15	34
2	**20**	**12**
3	21	19
4	15	25
5	29	10

14

пп	*A*	*B*
1	27	12
2	14	25
3	**10**	**27**
4	12	14
5	37	14

15

пп	*A*	*B*
1	**16**	**24**
2	8	26
3	30	13
4	22	15
5	24	22

16

пп	*A*	*B*
1	16	8
2	18	24
3	**14**	**25**
4	22	18
5	30	25

17

пп	*A*	*B*
1	**15**	**20**
2	15	21
3	25	19
4	18	30
5	27	10

18

пп	*A*	*B*
1	25	19
2	27	10
3	15	20
4	18	30
5	**15**	**21**

19

пп	*A*	*B*
1	16	24
2	**8**	**26**
3	30	13
4	22	15
5	24	22

20

пп	*A*	*B*
1	27	12
2	**14**	**25**
3	10	27
4	12	14
5	37	14

21

пп	*A*	*B*
1	19	18
2	**10**	**27**
3	30	25
4	15	24
5	26	6

22

пп	*A*	*B*
1	15	21
2	25	30
3	27	18
4	**19**	**15**
5	14	16

23

пп	*A*	*B*
1	10	27
2	18	15
3	19	25
4	**30**	**20**
5	23	13

24

пп	*A*	*B*
1	19	18
2	10	27
3	**30**	**25**
4	15	24
5	26	6

25

пп	*A*	*B*
1	10	27
2	18	15
3	**19**	**25**
4	30	20
5	23	13

26

пп	*A*	*B*
1	19	20
2	17	19
3	**19**	**37**
4	25	15
5	20	9

27

пп	*A*	*B*
1	15	20
2	15	21
3	**25**	**19**
4	18	30
5	27	10

28

пп	*A*	*B*
1	26	12
2	11	28
3	**27**	**10**
4	13	21
5	23	29

29

пп	*A*	*B*
1	8	23
2	**20**	**16**
3	17	14
4	32	12
5	23	35

30

пп	*A*	*B*
1	30	11
2	12	28
3	16	22
4	**25**	**20**
5	17	19

31

пп	*A*	*B*
1	15	20
2	**15**	**21**
3	25	19
4	18	30
5	27	10

32

пп	*A*	*B*
1	25	19
2	27	10
3	**15**	**20**
4	18	30
5	15	21

33

пп	*A*	*B*
1	16	24
2	8	26
3	**30**	**13**
4	22	15
5	24	22

34

пп	*A*	*B*
1	**27**	**12**
2	14	25
3	10	27
4	12	14
5	37	14

35

пп	*A*	*B*
1	15	20
2	15	21
3	**25**	**19**
4	18	30
5	27	10

36

пп	*A*	*B*
1	16	8
2	**18**	**24**
3	14	25
4	22	18
5	30	25

37

пп	*A*	*B*
1	**25**	**19**
2	27	10
3	15	20
4	18	30
5	15	21

38

пп	*A*	*B*
1	26	12
2	**11**	**28**
3	27	10
4	13	21
5	23	29

39

пп	*A*	*B*
1	30	11
2	12	28
3	**16**	**22**
4	25	20
5	17	19

40

пп	*A*	*B*
1	12	10
2	28	18
3	**14**	**17**
4	8	22
5	38	33

41

пп	*A*	*B*
1	15	34
2	20	12
3	**21**	**19**
4	15	25
5	29	10

42

пп	*A*	*B*
1	**16**	**8**
2	18	24
3	14	25
4	22	18
5	30	25

43

пп	*A*	*B*
1	15	20
2	15	21
3	25	19
4	**18**	**30**
5	27	10

44

пп	*A*	*B*
1	25	19
2	27	10
3	15	20
4	18	30
5	**15**	**21**

45

пп	*A*	*B*
1	16	24
2	8	26
3	30	13
4	**22**	**15**
5	24	22

46

пп	*A*	*B*
1	12	10
2	**28**	**18**
3	14	17
4	8	22
5	38	33

47

пп	*A*	*B*
1	**8**	**23**
2	20	16
3	17	14
4	32	12
5	23	35

48

пп	*A*	*B*
1	26	12
2	11	28
3	27	10
4	**13**	**21**
5	23	29

49

Задание 3. Считая все пункты неделимыми, кроме одного, выделенного жирным шрифтом, найти справедливые дележи и соответствующие им выигрыши алгоритмом, описанным и продемонстрированным в примерах 8 и 9. Сравнить результаты задания 3 с результатами задания 2 с тем же номером варианта■

Варианты для задания 3 те же, что и для задания 2.

Задание 4. Разделить пропорционально числу студентов заданное число преподавательских позиций по трём факультетам методом Гамильтона. См. алгоритм метода Гамильтона и примеры 12, 13 для образца ■

Варианты для задания 4:

№ пп	Колледж	Образования	Свободных искусств	Бизнеса	Число позиций
01	Число студентов	940	1470	1600	13
02	Число студентов	940	1470	1600	14
03	Число студентов	940	1470	1600	15
04	Число студентов	940	1470	1600	16
05	Число студентов	940	1470	1600	17
06	Число студентов	940	1470	1600	18
07	Число студентов	2750	6040	3350	13
08	Число студентов	2750	6040	3350	14
09	Число студентов	2750	6040	3350	15
10	Число студентов	2750	6040	3350	16
11	Число студентов	2750	6040	3350	17
12	Число студентов	2750	6040	3350	18
13	Число студентов	3750	4040	5500	13
14	Число студентов	3750	4040	5500	14
15	Число студентов	3750	4040	5500	15

16	Число студентов	3750	4040	5500	16
17	Число студентов	3750	4040	5500	17
18	Число студентов	3750	4040	5500	18
19	Число студентов	4550	3100	1900	13
20	Число студентов	4550	3100	1900	14
21	Число студентов	4550	3100	1900	15
22	Число студентов	4550	3100	1900	16
23	Число студентов	4550	3100	1900	17
24	Число студентов	4550	3100	1900	18
25	Число студентов	3000	2200	1800	13
26	Число студентов	3000	2200	1800	14
27	Число студентов	3000	2200	1800	15
28	Число студентов	3000	2200	1800	16
29	Число студентов	3000	2200	1800	17
30	Число студентов	3000	2200	1800	18

Задание 5. Разделить пропорционально числу студентов заданное число преподавательских позиций по трём факультетам методом Джефферсона. См. пример 14 для образца ■

Варианты для задания 5 те же, что и для задания 4.

Задание 6. Разделить пропорционально числу студентов заданное число преподавательских позиций по трём факультетам методом Адамса. См. пример 15 для образца ■

Варианты для задания 6 те же, что и для заданий 4 и 5.

Задание 7. Разделить пропорционально числу акций заданное число мест в совете методами Гамильтона, Джефферсона, Адамса и Хантингтона-Хилла. Ответы представить в небольшой таблице. См. для образца примеры 12, 13 (метод Гамильтона), 14 (метод Джефферсона), 15 (метод Адамса) и 16 (метод Хантингтона-Хилла) ■

Варианты для задания 7:

№ пп	Компания	Наксон	Ароко	Евробайл	Число мест в совете
01	Процент акций	20	50	30	8
02	Процент акций	20	50	30	9
03	Процент акций	20	50	30	11
04	Процент акций	40	20	40	8
05	Процент акций	40	20	40	9
06	Процент акций	40	20	40	11
07	Процент акций	50	40	10	8
08	Процент акций	50	40	10	9
09	Процент акций	55	35	10	10
10	Процент акций	10	60	30	8
11	Процент акций	10	60	30	9
12	Процент акций	10	60	30	10
13	Процент акций	35	35	30	8
14	Процент акций	35	35	30	9
15	Процент акций	35	35	30	10
16	Процент акций	30	40	30	8
17	Процент акций	30	40	30	9
18	Процент акций	30	40	30	11
19	Процент акций	25	35	40	8
20	Процент акций	25	35	40	9
21	Процент акций	25	35	40	10
22	Процент акций	50	25	25	8

23	Процент акций	50	25	25	9
24	Процент акций	50	25	25	10
25	Процент акций	15	45	40	8
26	Процент акций	15	45	40	9
27	Процент акций	15	45	40	10
28	Процент акций	20	25	55	8
29	Процент акций	20	25	55	9
30	Процент акций	20	25	55	10

5. Предметный указатель

Литература к части 3

1. Алескеров Ф.Т., Хабина Э.Л., Шварц Д.А.. Бинарные отношения, графы и коллективные решения. − 2-е изд., перераб. и доп. − М.: ФИЗМАТЛИТ, 2012.− 344 с. – ISBN 978-5-9221-1363-2.
2. Воробьёв Н.Н. Теория игр для экономистов-кибернетиков. – М.: Наука. Гл. ред. физ.-мат. лит., 1985. – 272 с.
3. Коэн Г. Обо всём можно договориться. − М.: АСТ: АСТ МОСКВА, 2010. – 284 с. – ISBN 978-5-403-02812-7.
4. Робертс Ф.С. Дискретные математические модели с приложениями к социальным, биологическим и экологическим задачам. − М.: Наука. Гл. ред. физ.-мат. лит., 1986. − 496 с.
5. Pirnot T.L. Mathematics All Around. – 2nd ed. – Pearson Education, 2004. – 906 p. – ISBN 0-201-79511-6.

Часть 4. ПРИНЯТИЕ РЕШЕНИЙ

Процессы принятия решений лежат в основе любой целенаправленной человеческой деятельности. Оптимальные (эффективные) решения способствуют достижению поставленных целей при минимальных затратах финансовых, трудовых, материальных и сырьевых ресурсов. Методы поиска оптимальных решений для непрерывных моделей рассматриваются в разделах классической математики, связанных с изучением экстремумов функций, а для дискретных моделей – в разработанных значительно позднее методах решения дискретных оптимизационных задач. В обоих случаях решение представляет собой математический объект, основным свойством которого является именно то, что он доставляет экстремум одной заданной функции.

В то же время в реальности мы часто встречаемся с двумя другими ситуациями:

1. Отсутствие критериев оптимизации, когда варианты или альтернативы возможных решений представляют собой цельные нерасчленимые объекты, некоторые из которых можно только попарно сравнивать с точки зрения «качества» (зачастую формально не определённого).

2. Наличие нескольких критериев оптимизации, отражающих различные важные свойства и аспекты альтернатив, которые желательно одновременно улучшить.

В обоих случаях классические оптимизационные модели не работают, поэтому в последние десятилетия большое внимание уделяется новым подходам и методам решения указанных и близких к ним задач. В случае отсутствия критериев оптимизации для нахождения лучших альтернатив используются общие понятия, модели, методы и алгоритмы теории бинарных отношений, формализующей понятие оптимизации на основе попарных сравнений. Элементы этой теории излагаются в главе 14 «Бинарные отношения». В главе 15 «Бинарные отношения в критериальном пространстве» рассмотрены начальные идеи многокритериальной оптимизации, в рамках которой альтернативы представляются точками в критериальном пространстве, а их координаты соответствуют различным свойствам альтернатив. Центральным здесь является понятие множества Парето, содержащего оптимальные с точки зрения нескольких критериев альтернативы из исходного множества. Излагается алгоритм выделения множества Парето. Приводится метод Подиновского, который позволяет за счёт дополнительной качественной информации о важности критериев существенно сокращать число оптимальных альтернатив по сравнению с их числом в множестве Парето.

В главе 16 «Коллективное принятие решений» рассмотрены аспекты принятия коллективных или групповых решений, которые связаны с согласованием различных точек зрения, поиском компромисса и т.д. Понятия и терминология в этой области в большой степени опирается на материал двух предыдущих глав. Излагаются основные методы построения групповых решений.

Глава 17 «Функции выбора» посвящена появившемуся в 70-ые годы XX-ого века новому понятию. Функции выбора, в отличие от бинарных отношений, формализуют понятие оптимальности, не определяемое одними только результатами попарных сравнений альтернатив, но опирающееся на более широкие взаимозависимости между ними. В главе наряду с общим понятием функции выбора излагаются так называемые механизмы выбора – правила определения лучших альтернатив, основанные на понятиях бинарного отношения и критериального пространства, но существенно отличающиеся от рассмотренных в главах 14 и 15 правил выделения альтернатив, оптимальных по заданному бинарному отношению. В частности, рассмотрены турнирное правило и метод идеальной точки.

Как и ранее, изложение относится к конечным множествам и не касается более сложных вопросов, связанных с континуальными случаями, что определяется уже самим названием настоящего пособия.

Глава 14. Бинарные отношения

1. Понятие бинарного отношения
2. Формальное описание и свойства бинарных отношений
3. Понятие *R*-оптимальности
4. Задания
5. Предметный указатель

Во многих случаях, когда необходимо выбрать одну или несколько лучших альтернатив из числа имеющихся, единственной доступной информацией являются результаты попарных сравнений некоторых из них. Часто даже такой неполной информации хватает для обоснованного выбора лучших альтернатив. Понятие бинарного отношения, рассмотренное в настоящей главе, позволяет не только формально представлять операции попарного сравнения альтернатив, но и разработать процедуры выделения (выбора) лучших из них, основываясь на результатах их попарных сравнений.

1. Понятие бинарного отношения

Как и в других ситуациях, начнём с примеров. Рассмотрим высказывания, которые выражают отношения между некоторыми объектами:

«Иван – брат Петра»; «Татьяна старше Александра»; «Киев южнее Москвы»; «Железо тяжелее воды»; «Слово «ночь» и слово «день» содержат одинаковое число букв».

Эти пять предложений выражают отношения разного типа. Однако можно заметить сходство в характере отношений, утверждаемых первым и пятым предложениями. Они говорят о том, что некие два объекта принадлежат <u>общему классу</u>: сыновей общих родителей, слов с фиксированным числом букв. Второе, третье и четвертое отношения имеют то общее, что выражают некоторый <u>порядок</u> объектов в системе.

В дальнейшем эта разница между отношениями того и другого типа будет чётко определена. Первый и пятый пример − это отношения эквивалентности, определяющие разбиения множества объектов на классы подобных друг другу. Остальные три примера – это отношения порядка, устанавливающие относительное расположение объектов в системе.

Важно обратить внимание на тот факт, что во всех пяти примерах четко выделяются названия объектов (Иван, Киев и т. д.) и названия отношений (брат, старше, южнее и др.). Если вместо названия данного объекта подставить в предложение название другого объекта, то возможны следующие ситуации: 1) отношение опять будет выполнено; 2) отношение перестанет выполняться; 3) отношение потеряет смысл. Так, если в четвертое предложение вместо слова «железо» подставить слово «свинец», то высказывание останется истинным. Если в третье предложение вместо слова «Москва» подставить «Ашхабад», то оно перестанет быть истин-ным. Если же в третье предложение вместо слова «Москва» подставить «железо», то высказы-вание потеряет смысл (поскольку высказывание в разделе 1-1 было определено как предложе-ние, которое естественно считать истинным или ложным, то фраза «Киев южнее железа» прос-то не является высказыванием). В отличие от первых четырех, в пятое предложение можно под-ставить любые слова, поскольку для любого слова имеет смысл говорить о числе букв. Здесь сама форма суждения ограничивает класс объектов – объектами отношения могут быть только слова.

Итак, говорить об отношении можно только тогда, когда задано множество объектов, на которых это отношение определено. Отношение может быть определено не только для пар объектов, но и для троек, четверок и т. д. Например, отношение «составлять экипаж лодки-восьмёрки» выполняется для некоторых групп из восьми человек. Это отношение следует отличать от отношения, «входить в экипаж одной и той же лодки-восьмёрки», определенного для пар людей. Пример трехместных (или тернарных) отношений дают алгебраические операции. Отношение «образовывать произведение» имеет смысл для троек чисел $\langle x,y,z\rangle$ и выполняется в том случае, когда $x \cdot y = z$.

Мы будем рассматривать бинарные отношения, т.е. отношения, которые могут выполняться (или не выполняться) для двух объектов из одного и того же множества. Поэтому в дальнейшем будем говорить об отношениях, имея в виду только бинарные отношения.

2. Формальное описание и свойства бинарных отношений

Бинарным отношением R ***на множестве*** Ω называется подмножество R множества $\Omega\times\Omega$ (определение прямого произведение см. в разделе 3-2). Если пара $\langle x, y\rangle$ входит в R, т.е. $\langle x, y\rangle\in R$, то пишут xRy, что читается так: «x находится в отношении R с y». Во многих случаях xRy интерпретируется как «x лучше y», «x доминирует y» и т.д.

Отношение называется ***пустым*** (обозначается знаком пустого множества $\varnothing$), если оно не выполняется ни для одной пары элементов Ω, т.е. соответствующее множество пар пусто. Отношение называется ***полным*** (обозначается U), если оно выполняется для всех пар элементов Ω. Отношение называется ***диагональным*** (обозначается E), если оно выполняется для всех пар элементов Ω, состоящих из совпадающих элементов: $xEy \rightleftarrows x= y$ (напомним, что знак $\rightleftarrows$ обозначает булеву функцию «эквивалентность»; здесь и далее она означает «тогда и только тогда»). Отношение называется ***антидиагональным*** (обозначается $\bar{E}$), если оно выполняется для всех пар элементов Ω, состоящих из несовпадающих элементов: $x\bar{E}y \rightleftarrows x\neq y$.

Поскольку все отношения на Ω – подмножества одного и того же множества $\Omega\times\Omega$, то можно обычным образом определить теоретико-множественные операции с отношениями: $R_1\cup R_2$ (объединение), $R_1\cap R_2$ (пересечение), $\bar{R}$ (дополнение до полного отношения U). Вспоминая определение и свойства графиков (см. раздел 3-4), легко видеть, что бинарное отношение – как любое множество пар – является графиком. Поэтому все операции над графиками переносятся на бинарные отношения. Повторим здесь их вкратце.

Обратным к отношению R называется отношение R^{-1}, определяемое условием: $xR^{-1}y \rightleftarrows yRx$. Произведением отношений R_1 и R_2 называется отношение, обозначаемое $R_1\bullet R_2$, определяемое следующим образом: $x(R_1\bullet R_2)y$, если существует $z\in\Omega$, для которого xR_1z и zR_2y. Если R_1 – отношение «быть братом», R_2 – отношение «быть родителем», то произведение $R_1\bullet R_2$ есть отношение «быть братом одного из родителей», т.е. «быть дядей». Ассоциативный закон, означающий, что $(A\bullet B)\bullet C = A\bullet(B\bullet C)$, позволяет отказаться от расстановки скобок в произведениях и писать просто $A\bullet B\bullet C$ и т.д; для совпадающих множителей можно писать R^n.

Новой операцией является переход к двойственному отношению. Отношение $R^d=\overline{R^{-1}} = (\bar{R})^{-1}$ называется ***двойственным*** к отношению R.

Пример 1. Зададим следующее бинарное отношение R на множестве $\Omega = \{a, b, c, d\}$: aRb, bRc, cRd, dRa, aRa, cRc, bRd, dRb. Построим отношение R^d, двойственное к R.

<u>Шаг 1</u>. Построим отношение $\bar{R}$, дополнительное к R. По определению дополнения в него входят все те и только те пары, которые не входят в R. Поэтому надо просто рассмотреть все пары элементов из Ω (включая пары с совпадающими элементами) и удалить из списка все те, которые образуют отношение R (они приведены в списке). В нашем случае $\Omega = \{a, b, c, d\}$, поэтому множество всех пар таково:

$\Omega\times\Omega =\{\langle a,a\rangle,\langle a,b\rangle,\langle a,c\rangle,\langle a,d\rangle,\langle b,a\rangle,\langle b,b\rangle,\langle b,c\rangle,\langle b,d\rangle,\langle c,a\rangle,\langle c,b\rangle,\langle c,c\rangle,\langle c,d\rangle,\langle d,a\rangle,\langle d,b\rangle,\langle d,c\rangle,\langle d,d\rangle\}$

(всего 16 пар).

Исходное отношение R состоит из следующих пар:

$R = \{\langle a,b\rangle,\langle b,c\rangle,\langle c,d\rangle,\langle d,a\rangle,\langle a,a\rangle,\langle c,c\rangle,\langle b,d\rangle,\langle d,b\rangle\}$

(всего 8 пар). Дополнение $\bar{R}$ к R состоит из всех пар, вошедших в полный список для $\Omega\times\Omega$ и <u>не</u> вошедших в список для R. Составим новый список для $\bar{R}$, просматривая по порядку все пары из полного списка, и добавляя в новый список те из них, которые не входят в R. Начинаем с пары $\langle a,a\rangle$. Она входит в список для R (на 5-м месте) и поэтому не входит в $\bar{R}$. Следующая пара $\langle a,b\rangle$ также входит в список для R (на 1-м месте) и поэтому не входит в $\bar{R}$. Следующая пара $\langle a,c\rangle$ из полного списка не входит в R и поэтому включается в $\bar{R}$. Следующая пара $\langle a,d\rangle$ из полного списка также не входят в R и поэтому включается в $\bar{R}$. Продолжая этот процесс, получаем

$\bar{R} = \{\langle a,c\rangle,\langle a,d\rangle,\langle b,a\rangle,\langle b,b\rangle,\langle c,a\rangle,\langle c,b\rangle,\langle d,c\rangle,\langle d,d\rangle\}$.

<u>Шаг 2</u>. Построим отношение $(\bar{R})^{-1}$, обратное к $\bar{R}$. По определению обратного отношения, надо взять все пары, входящие в исходное отношение, и поменять в них порядок на обратный (вместо пары $\langle x,y\rangle$ пара $\langle y,x\rangle$; все пары вида $\langle x,x\rangle$ являются обратными к самим себе и поэтому входят в обратное отношение, если только входят в исходное отношение). В нашем случае исходным является отношение $\bar{R}$, построенное на шаге 1. Меняя порядок во всех парах из $\bar{R}$, получаем:

$(\bar{R})^{-1}$={⟨c,a⟩ ,⟨d,a⟩,⟨a,b⟩,⟨b,b⟩,⟨a,c⟩,⟨b,c⟩,⟨c,d⟩,⟨d,d⟩}.
Шаг 3. Отношение $(\bar{R})^{-1}$, построенное на шаге 2, является требуемым отношением R^d■

2.1. Свойства бинарных отношений. Перейдём к описанию свойств бинарных отношений. Сначала опишем свойства, относящиеся к отдельным элементам Ω, затем – к их парам, тройкам и, наконец, к произвольным подмножествам Ω. Если высказывание xRy истинно и отношение R интерпретируется как «лучше», то говорят, что элемент x ***доминирует*** элемент y или элемент y ***доминируем*** элементом x.

Отношение R называется ***рефлексивным***, если для любого $x\in\Omega$ верно xRx, и ***антирефлексивным***, если для любого $x\in\Omega$ неверно xRx (т.е. верно $x\bar{R}x$). Те же условия можно записать короче, пользуясь введёнными выше обозначениями: $E\subseteq R$ и $R\subseteq\bar{E}$.

Отношение называется ***симметричным***, если $R\subseteq R^{-1}$. Это значит, что из xRy следует, что yRx. Отношение называется ***асимметричным***, если $R\cap R^{-1} = \varnothing$. Это значит, что из двух выражений xRy и yRx по меньшей мере одно неверно. Отношение «быть братом» не является ни симметричным, ни асимметричным. Действительно, если Пётр – брат Фёдора, то Фёдор – брат Петра; но если Игорь – брат Ольги, то неверно, что Ольга – брат Игоря. Отношение называется ***антисимметричным***, если $R\cap R^{-1}\subseteq E$. Это значит, что xRy и yRx вместе выполняются только тогда, когда $x = y$, или эквивалентно: из xRyи yRx следует, что$x = y$.

Отношение называется ***транзитивным***, если $R^2\subseteq R$, или эквивалентно: из xRyи yRz следует, что xRz. Отношение R называется ***отрицательно транзитивным***, если его дополнение$\bar{R}$транзитивно. Отношение R называется ***сильно транзитивным***, если оно одновременно транзитивно и отрицательно транзитивно.

Отношение называется ***ациклическим***, если для любого k = 1, 2, ... $R^k\cap R^{-1} = \varnothing$, или эквивалентно: из x_1Rx_2, x_2Rx_3, …, $x_{k-1}Rx_k$ следует, что $x_k\bar{R}x_1$.

Ацикличность и транзитивность отношений особенно важны при выборе альтернатив, так как эти свойства выражают некоторые естественные взаимосвязи между объектами. Действительно, если x в каком-либо смысле лучше, чем y, а y в этом же смысле лучше, чем z, то естественно считать, что в этом смысле x лучше, чем z (транзитивность), и, во всяком случае, z не лучше x (ацикличность).

Некоторые свойства отношений оказываются связанными. Приведём простое

Утверждение 1. Если отношение R не антирефрексивно, то оно не асимметричное и не ациклическое.

Действительно, если хотя бы для одного элемента x выполнено xRx, то по определению обратного отношения выполнено $xR^{-1}x$, откуда следует, что пара $\langle x,x\rangle\in R\cap R^{-1}$, т.е. $R\cap R^{-1}\neq\varnothing$. По определению это значит, что отношение R не асимметрично. Далее, пара xRx представляет собой цикл длины 1 и, значит, отношение R не ациклическое ■

Пример 2. Пусть бинарное отношение R на множестве Ω = {a, b, c, d, e} таково: aRb, bRb, aRc, bRe, eRe, eRd, cRd, dRa. Обладает ли это отношение свойствами:

рефлексивности;

асимметричности;

ацикличности?

1. Поскольку не выполнено aRa, то данное отношение рефлексивным не является.
2. Поскольку имеет место bRb, то отношение не антирефлексивно. В силу утверждения 1 отношение R не асимметрично и не ациклическое ■

Пример 3. Пусть бинарное отношение R на множестве Ω = {a, b, c, d, e} таково: aRb, bRb, aRc, bRa, eRe, eRd, cRd, dRc. Обладает ли это отношение свойствами:

антирефлексивности;

асимметричности;

транзитивности?

1. Поскольку имеет место bRb, то отношение R не является антирефлексивным.
2. Поскольку отношение R не является антирефлексивным, то в силу утверждения 1 отношение R не асимметрично.
3. Поскольку верно aRc и cRd,то из транзитивности R должно следовать aRd. Поскольку aRd в заданном списке отсутствует, то отсюда следует, что данное отношение не транзитивно ■

Введём следующее важное понятие. Элементы xи y из Ω называются ***несравнимыми*** по отношению R, если одновременно верны оба высказывания: $x\bar{R}y$и $y\bar{R}x$. По отношению R определяется ***отношение несравнимости*** I_R: $xI_Ry \rightleftarrows x$ и y несравнимы по отношению R. По самому определению ясно, что отношение несравнимости симметрично. Ясно также, что $I_R = \overline{R \cup R^{-1}}$. Естественно, что все свойства I_R определяются свойствами исходного отношения R.

Пример 4. Построим отношение I_R для отношения R из примера 1. Исходное отношение R состоит из пар $\langle a,b\rangle,\langle b,c\rangle,\langle c,d\rangle,\langle d,a\rangle,\langle a,a\rangle,\langle c,c\rangle,\langle b,d\rangle,\langle d,b\rangle$. Отношение R^{-1} состоит из тех же пар в обратном порядке: $R^{-1}= \{\langle b,a\rangle,\langle c,b\rangle,\langle d,c\rangle,\langle a,d\rangle,\langle a,a\rangle,\langle c,c\rangle,\langle d,b\rangle,\langle b,d\rangle\}$. Далее, $R\cup R^{-1}= \{\langle a,a\rangle,\langle a,b\rangle,\langle a,d\rangle,\langle b,a\rangle,\langle b,c\rangle,\langle b,d\rangle,\langle c,b\rangle,\langle c,c\rangle,\langle c,d\rangle,\langle d,a\rangle,\langle d,b\rangle,\langle d,c\rangle\}$. Отсюда $I_R = \overline{R \cup R^{-1}} = \{\langle a,c\rangle,\langle b,b\rangle,\langle c,a\rangle,\langle d,d\rangle\}$. Таким образом, в данном случае I_R не рефлексивно, не антирефлексивно, симметрично (потому что содержит пары $\langle a,c\rangle$ и $\langle c,a\rangle$, а также $\langle b,b\rangle$,и $\langle d,d\rangle$). Далее, I_Rне асимметрично и не антисимметрично; не транзитивно (поскольку содержит пары $\langle a,c\rangle$ и $\langle c,a\rangle$, но не содержит пары $\langle a,a\rangle$); не отрицательно транзитивно, так как его дополнение $R\cup R^{-1}$ не транзитивно (для этого достаточно рассмотреть входящие в него пары $\langle a,b\rangle$ и $\langle b,c\rangle$, поскольку пара $\langle a,c\rangle$ в него не входит), не сильно транзитивно и не ациклическое ■

2.2. Графы бинарных отношений. В случае конечности множества Ω отношения удобно представлять графически. Поставим во взаимно-однозначное соответствие элементам конечного множества Ω различные точки на плоскости: x_1, x_2, …, , x_n. Проведем дугу от x_i к x_j тогда и только тогда, когда выполнено x_iRx_j для соответствующих элементов Ω (при $i = j$ дуга (x_i,x_j) превращается в петлю при вершине x_i). Такой геометрический объект является ориентированным простым графом (см. главу 6). Конкретное расположение вершин (точек) и дуг на плоскости не имеет значения; важно только, что с чем соединено и куда направлена стрелка.

Граф является геометрическим представлением отношения, аналогично тому, как график является геометрическим представлением функции. Геометрический язык полезен, когда граф достаточно прост (либо у него мало вершин, либо он имеет простую структуру). Наоборот, изучать и описывать сложные графы с большим числом вершин часто удобнее в терминах отношений. В дальнейшем будем говорить о ***графе отношения*** R, обозначая его через $G(R)$.

По построению, граф ациклического отношения является ациклическим. В графе рефлексивного отношения имеются петли при всех вершинах, в графе антирефлексивного отношения петель нет. В графе симметричного отношения любые две вершины либо не соединены совсем, либо соединены двумя противоположно направленными дугами. В графе асимметричного отношения любые две вершины либо не соединены совсем, либо соединены одной дугой. При переходе от отношения R к двойственному отношению R^dсоответствующие графы преобразуются следующим образом.

1. Если петли при вершине в графе $G(R)$ нет, то при той же вершине в графе $G(R^d)$ она есть; наоборот, если петля при вершине в графе $G(R)$ есть, то при той же вершине в графе $G(R^d)$ её нет.

2. Если две вершины в графе $G(R)$ не соединены, то те же вершины в графе $G(R^d)$ соединены двумя противоположно направленными дугами; наоборот, если две вершины в графе $G(R)$ соединены двумя противоположно направленными дугами, то те же вершины в графе $G(R^d)$ не соединены.

Пример 5. Рассмотрим бинарное отношение R из примера 1 на множестве $\Omega = \{a, b, c, d\}$: aRb, bRc, cRd, dRa, aRa, cRc, bRd, dRb. Сопоставим элементам a, b, c, d четыре точки A, B, C, D, расположенные в вершинах квадрата. Граф $G(R)$ отношения R показан на рис.1. Граф двойственного отношения $G(R^d)$ из того же примера 1 показан на рис.2■

Естественным образом каждому простому ориентированному графу $G(V,A)$ сопоставляется бинарное отношение $R=R(G)$; множество Ω, на котором оно задано, совпадает с множеством V вершин графа $G(V,A)$; $xRy \leftrightarrow$дуга $(x, y)\in A$. Понятно, что отношение $R(G)$ ациклическое тогда и только тогда, когда граф G является ациклическим; понятно также, что для любого (простого ориентированного) графа Gимеет место равенство $G(R(G)) = G$, т.е определённые в этом разделе соответствия между графами и бинарными отношениями являются взаимно обратными.

2.3. Отношения эквивалентности и порядка. Воспользуемся рассмотренными свойствами для выделения отношений, представляющих интерес для задач выбора лучших альтернатив. Отношение R называется отношением ***эквивалентности*** (***эквивалентностью***), если оно рефлексивно, симметрично и транзитивно. Отношениями эквивалентности являются: отношение «быть на одном курсе» на множестве студентов одного факультета; отношение «иметь одинаковый остаток при делении на 3» на множестве натуральных чисел; отношение подобия на мно-жестве треугольников.

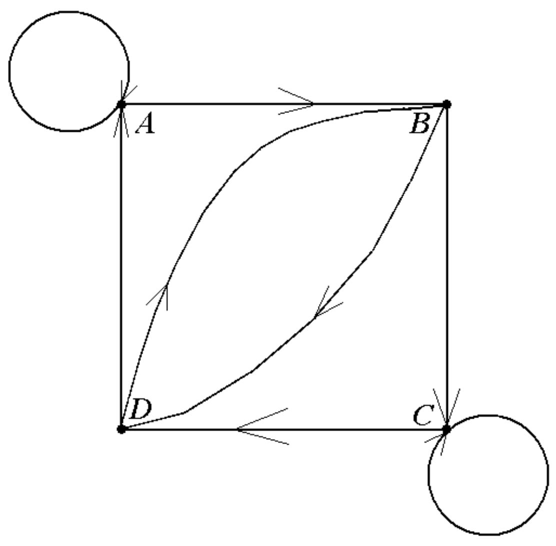


Рис.1. Граф $G(R)$ бинарного отношения R

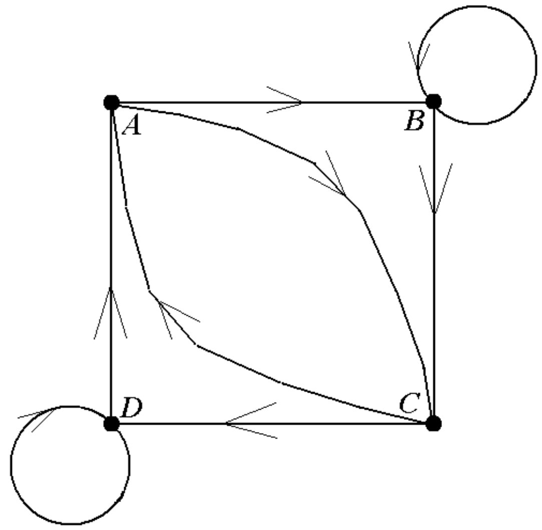


Рис.2. Граф $G(R^d)$ двойственного бинарного отношения R^d

Пусть задано ***разбиение*** множества Ω: $\Omega = \bigcup_{i=1}^{m}\Omega_i$ и $(i\neq j)\rightarrow(\Omega_i\cap\Omega_j = \varnothing)$. Введём на Ω следующее отношение R: xRy тогда и только тогда, когда существует подмножество Ω_i, содержащее одновременно x и y. Имеет место

Утверждение 2. Отношение является отношением эквивалентности на множестве Ω тогда и только тогда, когда оно определяется указанным выше образом по некоторому разбиению этого множества ■

Таким образом, задание эквивалентности на любом множестве Ω равносильно заданию разбиения Ω на классы эквивалентных друг другу элементов.

Введём несколько различных определений, выражающих различные понятия порядка, задаваемого бинарным отношением.

Ациклическое и транзитивное бинарное отношение называется ***частичным порядком***.

Отрицательно транзитивный частичный порядок называется ***слабым порядком***.

Связный слабый порядок называется ***линейным порядком***.

Связным отношением называется бинарное отношение R, такое, что для любых двух элементов $x,y\in\Omega$, таких что $x\neq y$, верно xRy или yRx.

В ряде случаев желательно, чтобы бинарное отношение Rобладало следующими двумя свойствами:

1. Существует разбиение множества Ω:$\Omega = \bigcup_{i=1}^{m}\Omega_i$, такое, что из истинности высказывания xRy следует, что номер подмножества, содержащего x, больше номера подмножества, содержащего y.

2. Отношение несравнимости I_R является отношением эквивалентности.

Для ациклических отношений выполняется свойство 1, но выполнение свойства 2 для них не гарантировано. Однако имеет место

Утверждение 3. Для любого слабого порядка R бинарное отношение несравнимости I_R является отношением эквивалентности■

Утверждения 6-7 и 3 вместе определяют структуру слабого порядка на множестве Ω. Остановимся на этом важном результате подробнее. Итак, пусть R – слабый порядок. Определим подмножества вершин Ω_i следующим образом: Ω_0 состоит из всех элементов $x\in\Omega$, таких, что для любого $y\in\Omega$неверно xRy. Далее, определим Ω_1 как множество всех элементов $x\in\Omega\setminus\Omega_0$ таких, что для любого $y\in\Omega\setminus\Omega_0$ неверно xRy. Продолжая эти операции, получим разбиение Ω. Заметим, что аналогичная процедура построения разбиения для ацикличных ориентированных графов описана в доказательстве утверждения 6-7. Для слабых порядков доказывается, что при этом классы эквивалентности по отношению несравнимости I_R совпадают с построенными множествами Ω_i (i= 1, ..., m).

3. Понятие *R*-оптимальности

Приведенный выше в данной главе материал был связан с формализацией понятия попарного сравнения элементов, которое необходимо для выделения лучшей альтернативы (или нескольких лучших) из всего множества Ω. Чтобы выделить лучшие, необходимо формализовать само понятие лучшего элемента. Воспользуемся для этого аппаратом бинарных отношений. Ниже определяются два понятия «лучшего» элемента, которые восходят к оптимизации числовых функций. В последнем случае элемент считается лучшим, если значение рассматриваемой числовой функции на данном элементе не меньше, чем на всех остальных, или, что эквивалентно – значение рассматриваемой числовой функции на всех элементах не больше, чем на данном. Аналоги этих определений для бинарных отношений уже не оказываются эквивалентными, что связано со значительно бóльшей сложностью рассматриваемых в принятии решений ситуаций.

Введём необходимые понятия. Для каждого $x\in\Omega$определим ***верхний срез*** Rx и ***нижний срез*** xR элемента x по отношению R:

$Rx = \{y\in\Omega | yRx\}$, (1a)

$xR = \{y\in\Omega | xRy\}$. (1b)

Элемент x называется ***максимумом по отношению*** R, если для всех $y\in\Omega$ выполнено xRy, и ***мажорантой по отношению***R, если для всех $y\in\Omega$ выполнено $y\bar{R}x$(т.е. неверно yRx). В терминах срезов элемент x является максимумом, если $xR = \Omega$, и мажорантой, если $Rx = \varnothing$.Содержательно в 1-ом определении речь идёт об элементе, который лучше всех других, а во 2-ом определении – об элементе, который не хуже всех других. Далее множество максимумов по отношению R обозначим через Ω_R, а множество мажорант – через Ω^R. Можно добавить, что максимум доминирует все элементы, а мажоранта никем не доминируема. Заметим, что в силу определения, по крайней мере одно из этих множеств является пустым. Действительно, допустим,что элемент x является максимумом по отношению R. Это значит, что x доминирует все элементы (включая себя), и, следовательно, ни один элемент не может быть недоминируемым. Наоборот, пусть элемент x является мажорантой. Это значит, что он не доминируем никаким элементом y (включая себя). Но тогда не существует элемента, доминирующего все элементы из Ω.

Пример 6. Рассмотрим ещё раз отношение Rна множестве $\Omega = \{a,b,c,d\}$: aRb, bRc, cRd, dRa, aRa, cRc, bRd, dRb из примера 1. Найдём множества Ω_R и Ω^R.

1. Если некоторый элемент x является максимумом, то, по определению, в перечень пар должна входить запись xRy для любого элемента y. Легко проверить, что для элемента a нет записи aRc; для элемента bнет записи bRa; для элемента cнет записи cRa; для элемента d нет записи dRc.Таким образом, множество максимумов пусто, или $\Omega_R = \varnothing$.

2. Если некоторый элемент x является мажорантой, то, по определению, в перечень пар не должна входить ни одна запись, в которой xстоит справа от буквы R, поскольку он доминируем тем элементом, который в этой же записи стоит слева от буквы R. Легко проверить, что таких элементов в данном случае нет. Таким образом, множество мажорант также пусто: $\Omega^R =\varnothing$■

Пример 7. Для отношения Rна множестве $\Omega = \{a,b,c,d\}$:aRb, bRb, cRd, dRa, aRa, cRa, bRd, dRb (почти не отличающегося от отношения из предыдущего примера 6) множество мажорант $\Omega^R = \{c\}$. Действительно, в данном перечне элемент c нигде не стоит справа от буквы R, т.е. он ни-кем не доминируем. Из этого сразу следует, что множество максимумов пусто■

Для графа $G(R)$ отношения R максимум по отношению соответствует вершине, дуги из которой ведут во все вершины графа; мажоранта отношения – это вершина, в которую не входит ни одна дуга. Легко видеть, что вершин обоего вида на графах отношений R и R^d из примера 1 (см. рис.1 и 2) нет. Это значит, что не только для отношенияR (см. пример 6), но и для двойственного к нему отношения R^d максимумов и мажорант нет.

Множества максимумов и мажорант для исходных и двойственных к ним отношений оказываются тесно связанными. Имеет место

Утверждение 4. Для любого бинарного отношения на множестве Ω

$\Omega_R= \Omega^{R^d}$, $\Omega^R = \Omega_{R^d}$■ (2)

В силу утверждения 4, при переходе к двойственному отношению максимумы переходят в мажоранты и наоборот. Поэтому в дальнейшем в качестве оптимальных альтернатив будут рассматриваться мажоранты по отношению R. Множество Ω^R, играющее важную роль в теории выбора, называется также множеством недоминируемых по R альтернатив; входящие в него альтернативы называются ***R*-оптимальными.**

Конечно, есть различные модификации, при которых оба множества – и доминирующих элементов, и недоминируемых элементов – могут быть одновременно непустыми. В частности, можно в обоих определениях вместо «для всех y» писать «для всех $y \neq x$». Однако ни формула (2), ни многие другие важные теоретические результаты при этом не сохранятся. В то же время некоторые полезные способы выбора лучших элементов, основанные на заданном бинарном отношении, но не сводящиеся к выбору максимумов и мажорант (или их аналогов), рассмотрены в разделе 16-3.2.

3.1. Алгоритм построения множества мажорант Ω^R. Поскольку задача выделения R-оптимальных альтернатив имеет определённое практическое и теоретическое значение, остановимся на алгоритме её решения подробней. Как и во всех других рассматриваемых в пособии алгоритмах, надо указать, в каком виде представляются исходные данные. Поскольку при освоении алгоритмов очень полезно выполнить их – для задач небольшой размерности – «руками и глазами», то и представление данных заметно отличается от того, который используется в реальном программировании. Напомним, что для выполнения алгоритма Флёри (поиска эйлерова цикла) предлагалось просто использовать изображение графа.

В данном случае предполагается, что бинарное отношение задано перечнем записей вида xRy, как это делалось в примерах 1 – 3 и 5 – 7. Алгоритм оказывается достаточно простым. Дадим его описание.

Алгоритм 1. Построение множества мажорант Ω^R.

1. Положим $D = \varnothing$.
2. Просматриваем слева направо все записи вида xRy. Для каждой записи
   2.1. Проверяем, входит ли правый элемент y в множество D. Если да, то переходим к следующей записи.
   2.2. Добавляем y к множеству D и переходим к следующей записи.
3. Положим $\Omega^R = \Omega \setminus D$■

Пример 8. Рассмотрим отношения Rна множестве $\Omega = \{a,b,c,d\}$:*aRb, bRb, cRd, dRa, aRa, cRa, bRd, dRb*, взятое из примера 7. Применим к нему алгоритм 1.

1. Положим $D = \varnothing$.
2. Просматриваем все записи вида xRy. Получаем:
   1-ая запись aRb;$b \notin D$.Положим $D = D \cup \{b\} = \{b\}$;
   2-ая запись bRb; $b \in D$;
   3-ья запись cRd; $d \notin D$. Положим $D = D \cup \{d\} = \{b,d\}$;
   4-ая запись dRa; $a \notin D$. Положим $D = D \cup \{a\} = \{a,b,d\}$;
   5-ая запись dRa;$a \in D$;
   6-ая запись aRa;$a \in D$;
   7-ая запись bRd; $d \in D$;
   8-ая запись dRb; $b \in D$.
3. Положим $\Omega^R = \Omega \setminus D = \{a,b,c,d\} \setminus \{a,b,d\} = \{c\}$■

4. Задания

Задание 1. Построить отношение, двойственное к данному отношению. См. пример 1 для образца.

Варианты для задания1:

01. $\Omega = \{a, b, c, d, e, f\}$; *aRa, aRd, aRe, aRf, bRa, cRc, bRc, fRc, bRe, eRa, eRc, eRf, fRa, dRf.*
02. $\Omega = \{a, b, c, d\}$; *bRa, cRb, dRc, dRa, bRb, cRa, bRd, dRd.*
03. $\Omega = \{a, b, c, d, e, f\}$; *aRa, aRd, aRe, aRf, bRa, cRc, bRc, fRc, bRe, eRa, eRc, eRf, fRa, dRf.*
04. $\Omega = \{a, b, c, d\}$; *bRa, cRb, dRc, dRa, bRb, cRa, bRd, dRd.*
05. $\Omega = \{a, b, c, d, e, f\}$; *aRb, bRd, cRf, dRe, eRb, aRc, bRf, fRc, cRd, cRc, eRd, fRa.*
06. $\Omega = \{a, b, c, d, e\}$; *aRb, bRc, aRd, bRa, eRd, cRd, dRc.*
07. $\Omega = \{a, b, c, d, e, f\}$; *aRb, bRd, cRf, dRe, eRb, aRc, bRf, fRc, cRd, cRc, eRd, fRa.*
08. $\Omega = \{a, b, c, d, e\}$; *aRb, bRb, aRc, bRe, eRe, eRd, cRd, dRa.*
09. $\Omega = \{a, b, c, d, e, f\}$; *aRb, bRd, cRf, dRe, eRb, aRc, bRf, fRc, cRd, cRc, eRd, fRa.*
10. $\Omega = \{a, b, c, d, e\}$; *aRb, bRb, aRc, bRa, eRe, eRd, cRd, dRc.*

12. $\Omega = \{a, b, c, d, e\}$; *aRb*, *bRb*, *aRc*, *bRa*, *eRe*, *eRd*, *cRd*, *dRc*.
13. $\Omega = \{a, b, c, d, e\}$; *aRb*, *bRb*, *aRc*, *bRe*, *eRe*, *eRd*, *cRd*, *dRa*.
14. $\Omega = \{a, b, c, d, e, f\}$; *aRb*, *bRd*, *cRf*, *dRe*, *eRb*, *aRc*, *bRf*, *fRc*, *cRd*, *cRc*, *eRd*, *fRa*■

Задание 2. Варианты задания (см. примеры 2 и 3 для образца):

01. Пусть бинарное отношение R на множестве $\Omega = \{a, b, c, d, e\}$ таково: *aRb*, *bRb*, *aRc*, *bRa*, *eRe*, *eRd*, *cRd*, *dRc*.Обладает ли это отношение свойствами:
 антирефлексивности;
 асимметричности;
 транзитивности?

02. Пусть бинарное отношение R на множестве $\Omega = \{a, b, c, d, e\}$ таково: *aRb*, *bRb*, *aRe*, *bRa*, *eRe*, *eRd*, *cRd*, *dRc*. Обладает ли это отношение свойствами:
 рефлексивности;
 антисимметричности;
 ацикличности?

03. Пусть бинарное отношение R на множестве $\Omega = \{a, b, c, d, e\}$ таково: *aRb*, *bRb*, *aRc*, *bRe*, *eRe*, *eRd*, *cRd*, *dRa*. Обладает ли это отношение свойствами:
 антирефлексивности;
 симметричности;
 транзитивности?

04. Пусть бинарное отношение R на множестве $\Omega = \{a, b, c, d, e\}$ таково: *aRb*, *bRb*, *aRc*, *bRe*, *eRe*, *eRd*, *cRd*, *dRa*. Обладает ли это отношение свойствами:
 рефлексивности;
 асимметричности;
 ацикличности?

05. Пусть бинарное отношение R на множестве $\Omega = \{a, b, c, d, e\}$ таково: *aRb*, *bRc*, *aRd*, *bRa*, *eRd*, *cRd*, *dRc*. Обладает ли это отношение свойствами:
 антирефлексивности;
 асимметричности;
 транзитивности?

06. Пусть бинарное отношение R на множестве $\Omega = \{a, b, c, d, e, f\}$ таково: *aRb*, *bRd*, *cRf*, *dRe*, *eRb*, *aRc*, *bRf*, *fRc*, *cRd*, *cRc*, *eRd*, *fRa*. Обладает ли это отношение свойствами:
 асимметричности;
 транзитивности;
 ацикличности?

07. Пусть бинарное отношение R на множестве $\Omega = \{a, b, c, d, e, f\}$ таково: *aRb*, *bRd*, *cRf*, *dRe*, *eRb*, *aRc*, *bRf*, *fRc*, *cRd*, *cRc*, *eRd*, *fRa*. Обладает ли это отношение свойствами:
 симметричности;
 антисимметричности;
 транзитивности;

08. Пусть бинарное отношение R на множестве $\Omega = \{a, b, c, d, e, f\}$таково: *aRb*, *bRd*, *cRf*, *dRe*, *eRb*, *aRc*, *bRf*, *fRc*, *cRd*, *cRc*, *eRd*, *fRa*. Обладает ли это отношение свойствами:
 асимметричности;
 антисимметричности;
 ацикличности?

09. Пусть бинарное отношение R на множестве $\Omega = \{a, b, c, d, e, f\}$ таково: *aRa*, *aRd*, *aRe*, *aRf*, *bRa*, *cRc*, *bRc*, *fRc*, *bRe*, *eRa*, *eRc*, *eRf*, *fRa*, *dRf*. Обладает ли это отношение свойствами:
 рефлексивности;
 антисимметричности;
 ацикличности?

10. Пусть бинарное отношение R на множестве $\Omega = \{a, b, c, d, e, f\}$ таково: *aRa*, *aRd*, *aRe*, *aRf*, *bRa*, *cRc*, *bRc*, *fRc*, *bRe*, *eRa*, *eRc*, *eRf*, *fRa*, *dRf*. Обладает ли это отношение свойствами:
 антирефлексивности;
 асимметричности;
 транзитивности?

11. Пусть бинарное отношение R на множестве $\Omega = \{a, b, c, d\}$ таково: bRa, cRb, dRc, dRa, bRb, cRa, bRd, dRd. Обладает ли это отношение свойствами:

антирефлексивности;

асимметричности;

транзитивности?

12. Пусть бинарное отношение R на множестве $\Omega = \{a, b, c, d\}$ таково: bRa, cRb, dRc, dRa, bRb, cRa, bRd, dRd. Обладает ли это отношение свойствами:

рефлексивности;

антисимметричности;

ацикличности? ■

Задание 3. Найти отношение несравнимости I_R для данного отношения R. См. пример 4 для образца.

Варианты те же, что в задании 1 ■

Задание 4. Данное бинарное отношение и двойственное к нему изобразить графами. См. пример 5 для образца.

Варианты те же, что в задании 1 ■

Задание 5. Для данного орграфа G найти бинарное отношение $R(G)$ (если орграф не является простым, то заменить его простым, как в примере 6-25). Отношение задать перечнем записей вида xRy. См. абзац сразу после примера 5■

Варианты заданий те же, что в задании 6-3.

Задание 6. Для данного бинарного отношения R найти алгоритмом 1 множество мажорант Ω^R. См. пример 8 для образца.

Варианты бинарных отношений к заданию 6:

01. $\Omega = \{a, b, c, d, e, f\}$; aRa, aRd, aRc, aRf, bRa, cRc, bRc, fRc, eRa, eRc, eRf, fRa, dRf.
02. $\Omega = \{a, b, c, d\}$; bRa, cRb, dRc, dRa, bRb, cRa, bRc, dRb.
03. $\Omega = \{a, b, c, d, e, f\}$; aRa, aRd, aRe, aRf, bRa, cRc, bRc, fRc, bRe, eRa, eRc, eRf, fRa, dRf.
04. $\Omega = \{a, b, c, d\}$; bRa, cRb, dRc, dRa, bRb, cRa, bRd, dRd.
05. $\Omega = \{a, b, c, d, e, f\}$; aRb, bRd, cRf, dRe, eRb, aRc, bRf, fRc, cRd, cRc, eRd..
06. $\Omega = \{a, b, c, d, e\}$; aRb, bRc, aRd, bRa, eRd, cRd, dRc.
07. $\Omega = \{a, b, c, d, e, f\}$; aRb, bRd, cRf, eRb, aRc, bRf, fRc, cRd, cRc, eRd, fRa.
08. $\Omega = \{a, b, c, d, e\}$; aRb, bRb, aRc, bRe, eRe, eRd, cRd, dRc.
09. $\Omega = \{a, b, c, d, e, f\}$; aRb, bRd, cRf, dRd, eRb, aRc, bRf, fRc, cRd, cRc, eRd.
10. $\Omega = \{a, b, c, d, e\}$; aRb, bRb, aRc, eRe, eRd, cRd, dRb.
12. $\Omega = \{a, b, c, d, e\}$; aRa, bRc, aRc, bRe, eRe, eRd, cRd, dRc.
13. $\Omega = \{a, b, c, d, e\}$; aRb, bRb, aRc, bRe, eRe, eRd, cRd, dRa.
14. $\Omega = \{a, b, c, d, e, f\}$; aRb, bRd, cRf, dRe, eRb, aRc, bRf, fRc, cRd, cRc, eRd, fRf.
15. $\Omega = \{a, b, c, d, e\}$; aRb, bRb, bRa, eRe, eRd, cRd.
16. $\Omega = \{a, b, c, d, e\}$; aRb, bRb, aRe, eRe, eRd, cRd, dRc.
17. $\Omega = \{a, b, c, d, e\}$; aRc, bRe, eRe, eRd, cRd, dRa.
18. $\Omega = \{a, b, c, d, e\}$; aRb, bRb, aRc, bRe, eRe, dRa.
19. $\Omega = \{a, b, c, d, e\}$; aRb, bRc, aRd, bRa, eRd, cRd, dRc.
20. $\Omega = \{a, b, c, d, e, f\}$; bRd,cRf,aRc, bRf, fRc, cRd, cRc, eRd, fRa.
21. $\Omega = \{a, b, c, d, e, f\}$; aRb, bRd, cRf, dRe, eRb, bRf, fRb, cRd, eRd, fRa.
22. $\Omega = \{a, b, c, d, e, f\}$; aRb, bRd, cRf,dRe, eRb,aRc, bRf, fRc, cRc, fRa.
23. $\Omega = \{a, b, c, d, e, f\}$; aRd, aRe, aRf, cRc, bRc, fRc, bRe, eRc, eRf, fRf, dRf.
24. $\Omega = \{a, b, c, d, e, f\}$; aRa, aRd, aRe, aRf, bRa, cRc, bRc, fRc, bRe, eRa, eRc, eRf, fRa, dRf.
25. $\Omega = \{a, b, c, d\}$; bRa, cRb, dRc, dRa, bRb, cRa, dRc.
26. $\Omega = \{a, b, c, d\}$; bRa, cRb, dRc, dRa, bRb, cRa, bRd, dRd■

5. Предметный указатель

Глава 15. Бинарные отношения в критериальном пространстве

В современной науке о принятии решений считается, что во многих случаях варианты решений характеризуются различными показателями их привлекательности для лица, принимающего решения (ЛПР). Эти показатели называют признаками, факторами, атрибутами или критериями. Мы принимаем для последующего изложения термин «критерий». Использование критериев, по сути дела, знаменует собой переход от описания альтернатив как целостных, нерасчленимых для нас объектов к их описанию как кортежей одной и той же длины, компонентами которых являются количественные или качественные оценки этих альтернатив по рассматриваемым в той или иной ситуации критериям. Такой переход – от альтернатив к их векторным (говорят также – многокритериальным) оценкам является принципиальным шагом, так как он позволяет не только применять разнообразные формальные модели и методы в процессе принятия решений, но и достаточно точно описывать предпочтения ЛПР. Как и в других главах, основное внимание уделяется формальным моделям и алгоритмам решения стандартных задач.

1. Понятие критерия

Будем называть ***критериями оценки альтернатив*** (или просто критериями) показатели их привлекательности (или непривлекательности) для участников процесса выбора. В профессиональной деятельности выбор критериев часто определяется многолетней практикой, опытом. В подавляющем большинстве случаев имеется достаточно много критериев оценок вариантов решений. Эти критерии могут быть ***независимыми*** или ***зависимыми***. Зависимыми называются те критерии, при которых оценка альтернативы по одному из них определяет (однозначно либо с большой степенью вероятности) оценку по другому критерию. Так, мы можем ожидать, что высококачественная элитная квартира является, как правило, дорогой. Зависимость между критериями приводит к появлению целостных образов альтернатив, которые имеют для каждого из участников процесса выбора определенное смысловое содержание.

На сложность задач принятия решений влияет также количество критериев. При небольшом числе критериев (два-три) задача сравнения двух альтернатив достаточно проста и прозрачна, оценки по критериям могут быть непосредственно сопоставлены и выработан компромисс. При большом числе критериев задача становится почти необозримой. К счастью, при большом количестве критериев они обычно могут быть объединены в группы, имеющие конкретное смысловое значение и название. Основанием для естественной группировки критериев является возможность выделить плюсы и минусы альтернатив, их достоинства и недостатки (например, стоимость и эффективность). Такие группы, как правило, независимы. Выявление структуры на множестве критериев делает процесс принятия решений значительно более осмысленным и эффективным.

1.1. Оценки по критериям. Использование критериев для оценки альтернатив требует определения градаций качества: лучших, худших и промежуточных оценок. Иначе говоря, предполагается, что заданы шкалы оценок по критериям.

В принятии решений принято различать шкалы непрерывных и дискретных оценок, шкалы количественных и качественных оценок. Так, для критерия «стоимость» может быть использована непрерывная количественная шкала оценок (в денежных единицах). Для критерия «наличие дачи» может быть качественная двоичная шкала: есть или нет. Кроме категорий «качественные – количественные», «непрерывные – дискретные» в принятии решений различают следующие типы шкал.

1. ***Шкала порядка*** – оценки упорядочены по возрастанию или убыванию качества. Примером может служить шкала обычных оценок знаний студентов: «отлично, хорошо, удовлетворительно, неудовлетворительно». В некоторых странах для той же цели используется шкала «A, B, C, D, F», в которой число градаций на единицу больше.

2. ***Шкала равных интервалов*** – интервальная шкала. Для этой шкалы имеются равные расстояния по изменению качества между оценками. Таковой, например, является шкала температуры. Для интервальной шкалы характерно, что начало отсчета выбирается произвольно, так же как и шаг (расстояние между оценками) шкалы. Вспомните шкалу Цельсия и шкалу Фаренгейта.

3. ***Шкала пропорциональных оценок*** – идеальная шкала. Примером является шкала оценок по критерию стоимости, отсчёт в которой начинается с установленного значения (например, с нулевой стоимости), а градация фиксирована (например, в долларах).

В принятии решений чаще всего используются порядковые шкалы и шкалы пропорциональных оценок.

2. Отношение и множество Парето

Напомним, что m-мерное пространство – это множество всех точек с m координатами (мы «живём» в 3-х мерном пространстве). Многомерное пространство бывает удобным для описания объектов, характеризующихся несколькими параметрами или критериями. При этом оно обычно называется ***критериальным пространством***.

Пусть Ω – множество n точек в многомерном пространстве:

$\Omega = \{x^1, x^2, \ldots, x^n\}$.

Определим ***отношение Парето*** (обозначается через P) на множестве Ω следующим образом:

$xPy \leftrightarrow (x_i \geq y_i,\ i = 1, \ldots, m)$ и $(\exists j\colon x_j > y_j)$. (1)

В силу этого определения $y\bar{P}x$ (т.е. y не «лучше» x по отношению Парето) означает, что хотя бы одна координата x больше, чем та же координата y. ***Множеством Парето*** на Ω называется множество всех мажорант по отношению Парето (см. определение мажоранты по бинарному отношению в разделе 14-3); оно обозначается через Ω^P. Отношение и множество Парето фактически уже использовалось в пособии: в разделе 11-3 при удалении доминируемых строк и столбцов в платёжной матрице и в разделе 13-2.3 при удалении из дальнейшего рассмотрения дележей с ме́ньшими доходами от неделимых пунктов.

Множество Парето названо по имени выдающегося учёного Вильфредо Парето (1848 – 1923), впервые обратившего внимание на альтернативы, не уступающие друг другу по критериальным оценкам, т. е. на альтернативы, из которых ни одна не доминирует другую. Напомним, что такие альтернативы находятся между собой в отношении несравнимости (см. определение в разделе 14-2.1).

По самому определению мажоранты, любые две альтернативы, входящие в мажоранту Ω^P, несравнимы по отношению P. Для выбора лучших из них необходимо использовать дополнительную информацию – естественно, от ЛПР и / или экспертов. Некоторые возможности этого рассмотрены ниже в разделе 4. Но даже само выделение множества Парето оказывается весьма полезным. Понятно, что альтернативы, не входящие в множество Парето, можно просто не рассматривать. Для каждой из них есть не уступающая ей по всем критериям (и хотя бы по одному превосходящая) альтернатива из множества Парето Ω^P. Доля элементов из множества Парето по сравнению с общим числом элементов из Ω обычно невелика (и тем меньше, чем больше число элементов в исходном множестве Ω). Подробнее эти вопросы в пособии не рассматриаются.

Алгоритм построения мажоранты по любому бинарному отношению R описан в разделе 14-3.1. Там предполагалось, что само бинарное отношение R задано в виде полного перечня записей вида xRy. Однако для задач в критериальном пространстве естественно считать, что все элементы множества Ω являются точками в критериальном пространстве E^m. Поэтому алгоритм должен и найти все пары векторных оценок $\langle x,y \rangle$, для которых верно xPy, и удалить все доминируемые альтернативы.

Заданное множество m-мерных точек $\Omega = \{x^1, x^2, \ldots, x^n\}$ представляет собой вход алгоритма. Выходом является множество Парето Ω^P.

Алгоритм 1. Выделение множества Парето.

1. Инициализация. Положим $L = \Omega$, $D = \varnothing$, $x^b = x^1$.
2. Точка x^b сравнивается последовательно со всеми элементами списка L, кроме элемента x^1.

При сравнении элемента x^b с элементом y возможны три случая:

а) $x^b P y$; в этом случае элемент y удаляется из списка L; если в списке остался только один элемент, то переходим к шагу 4; в противном случае элемент x^b сравнивается со следующим элементом списка L;

б) $y P x^b$; в этом случае элемент x^b удаляется из списка; если в списке остался только один элемент, то переходим к шагу 4; в противном случае x^b=y; элемент x^b сравнивается со следующим (после y) элементом списка L;

в) $x^b \bar{P} y$ и $y \bar{P} x^b$; далее элемент x^b сравнивается со следующим элементом списка L.

3. При достижении конца списка положим D= $D\cup\{x^b\}$, L =$L\setminus\{x^b\}$, x^b = 1-ому элементу списка L. Возвращаемся к шагу 2.

4. Положим Ω^P=$D\cup z$, где z – единственный элемент списка L. Алгоритм 1 останавливается■

Пример 1. Найдём множество Парето Ω^P для исходного множества альтернатив Ω= {(3,7,0), (4,5,1), (2,3,0), (3,7,3), (2,6,3), (4,5,2)}. Используем алгоритм 1.

1. Инициализация. Положим L = Ω= {(3,7,0), (4,5,1), (2,3,0), (3,7,3), (2,6,3), (4,5,2)}, D =$\varnothing$, x^b = x^1 = (3,7,0).

2. Сравниваем x^b=(3,7,0) с y=(4,5,1). Имеет место случай в). Переходим к следующему в списке элементу y = (2,3,0).

2. Сравниваем x^b= (3,7,0) с y = (2,3,0). Поскольку (3,7,0)P(2,3,0), то имеет место случай а). Элемент (2,3,0) удаляется из списка, и получаем L ={(3,7,0), (4,5,1), (3,7,3), (2,6,3), (4,5,2)}. Переходим к следующему (после удалённого) элементу y = (3,7,3).

2. Сравниваем x^b= (3,7,0) с y= (3,7,3). Поскольку (3,7,3)P(3,7,0), то имеет место случай б). Удаляем из списка x^b= (3,7,0) и получаем L ={(4,5,1), (3,7,3), (2,6,3), (4,5,2)}. Положим x^b= (3,7,3) и переходим к следующему элементу y= (2,6,3).

2. Сравниваем x^b= (3,7,3) с y= (2,6,3). Поскольку (3,7,3)P(2,6,3), то имеет место случай а). Элемент (2,6,3) удаляется из списка и получаем L ={(4,5,1), (3,7,3), (4,5,2)}. Переходим к следующему (после удалённого) элементу y = (4,5,2)

2. Поскольку x^b= (3,7,3) и y= (4,5,2) несравнимы, имеет место случай в). Поскольку достигнут конец списка, переходим к шагу 3.

3. Положим D= $\varnothing\cup\{x^b\}$ = {(3,7,3)}. Удаляя (3,7,3) из списка L, получим L ={(4,5,1), (4,5,2)}. Положим x^b = (4,5,1) и перейдём к шагу 2.

2. Сравниваем x^b= (4,5,1) с y=(4,5,2). Поскольку (4,5,2)P(4,5,1), то имеет место случай б). Удаляем из списка x^b= (4,5,1) и получаем x^b= (4,5,2), L ={(4,5,2)}. Поскольку список L состоит из одного элемента(4,5,2), переходим на шаг 4.

4. Положим D= $D\cup$\{(4,5,2)\}= {(3,7,3), (4,5,2)}■

Во многих случаях надо не только выделить множество Парето из исходного множества, но и выделить множество Парето из оставшегося (после удаления 1-го множества Парето) множества, и т.д. Эти последовательно выделяемые множества называются ***слоями*** или ***уровнями Парето***. Продемонстрируем весь процесс на том же примере.

Пример 2. Для исходного множества Ω= {(3,7,0), (4,5,1), (2,3,0), (3,7,3), (2,6,3), (4,5,2)} множество Парето Ω^P= {(3,7,3), (4,5,2)} (см. пример 1). Далее, Ω_1= $\Omega\setminus\Omega^P$= {(3,7,0), (4,5,1), (2,3,0), (2,6,3)}. Выделяя тем же алгоритмом 1 из множества Ω_1 его паретовскую часть, находим Ω_1^P = $\{(3,7,0), (4,5,1), (2,3,0), (2,6,3)\}^P$ = {(3,7,0), (4,5,1), (2,6,3)}. Поскольку Ω_2= $\Omega_1\setminus\Omega_1^P$ = {(2,3,0)}, то окончательно получаем, что всего имеется три паретовских слоя: {(3,7,3), (4,5,2)}; {(3,7,0), (4,5,1), (2,6,3)} и {(2,3,0)}■

3. Метод Подиновского

Выделение множества Парето, хотя и позволяет в значительной степени сократить число альтернатив, которые являются «кандидатами» для выбора лучших из них, всё же целесообразно рассматривать лишь как первый этап такого выбора. Дело в том, что при наличии нескольких критериев для выбора лучших вариантов необходимо в той или иной форме использовать предпочтения лица, принимающего решения (ЛПР). Выявление предпочтений ЛПР и разработка основанных на них алгоритмов выбора является центральной и наиболее трудной проблемой в принятии решений. Метод, предложенный более 30 лет назад современным российским учёным В.В. Подиновским (и с тех пор многократно усовершенствованный и модифицированный им), является одним из наиболее эффективных в этой важной области. Здесь даётся его сжатое и неполное изложение.

Как и выше в настоящей главе, речь идёт об объектах (учениках, фирмах, автомобилях, бизнес-планах и т.д.), которые оцениваются по нескольким критериям. Принципиальным в излагаемом ниже подходе является ***однородность критериев***. Это означает, что у всех критериев имеется единая (общая) шкала. Более того, должно выполняться следующее условие однородности: каждая градация шкалы должна отражать одинаковый уровень предпочтений для каждого из критериев. Это условие выполняется, в частности, в том случае, когда градации являются вербальными (словесными). При этом они имеют смысл, одинаковый для всех критериев, например, «превосходно», «отлично», …, «отвратительно». Обычно градации нумеруются в порядке возрастания предпочтительности. Нужно подчеркнуть, что номера градаций, которые часто называют баллами, отражают лишь их упорядоченность по предпочтению. Поэтому с числами – номерами градаций нельзя производить арифметические операции для получения каких-либо иных оценок предпочтений. Например, бессмысленно говорить, будто при отличной оценке успеваемость (знания, умения, навыки и компетенции) в 2,5 раза выше, чем при неудовлетворительной. По этой причине рассматриваемые шкалы являются порядковыми (см.раздел 1.1).

Итак, далее предполагается (и специально не оговаривается), что все критерии однородны, т.е. они имеют единую порядковую шкалу. В качестве основного примера будем рассматривать оценки семи учениц по четырём предметам, имея в виду, что все рассуждения справедливы для любых объектов, оцениваемых по однородным критериям.

Таблица 1

Школьница	Алгебра	Геометрия	Литература	Английский
1. Арсенкова Александра	5	4	4	5
2. Баранникова Елена	3	4	4	3
3. Кулакова Екатерина	3	3	4	3
4. Михайлова Елена	3	3	4	4
5. Морозова Мария	5	4	3	5
6. Перепечаева Арина	3	5	4	4
7. Полищук Наталья	3	5	3	3

3.1. Понятие качественной важности. Под ***качественной важностью*** критериев будем понимать качественные оценки их относительной важности. Суждение «критерий K_i важнее критерия K_j» будем обозначать «словом» $i \succ j$, а суждение «критерии K_i и K_j равно важны» − «словом» $i \approx j$.Для обозначения качественной информация о важности будет использоваться буква греческого алфавита Θ (тэта). Она формируется накопленными (полученными от ЛПР, экспертов и/или других источников) сведениями о том, что некоторые критерии одинаковы по важности и что одни критерии важнее других, т.е. сообщениями вида $i \approx j$и$i \succ j$. Например, если стало известно, что первый критерий важнее второго, а второй и третий критерии равно важны, то $\Theta = \{1 \succ 2, 2 \approx 3\}$.

Обратимся теперь к центральному вопросу точного определения понятий равенства и превосходства критериев в важности. Предположим, что школьницы изучали всего два учебных предмета, так что каждая школьница получила всего две итоговые оценки. Если для характеристики общей успеваемости достаточно (и так нередко делается в жизни) перечислить полученные ими оценки в произвольном порядке, не указывая при этом, какая оценка относится к тому или иному предмету (т.е. сказать, например, что у этой школьницы пятёрка и тройка, а у той − пятёрка и четвёрка), то это и означает, что каждый из предметов имеет одинаковую важность. Если же это не так, причем считается, что из любых двух школьниц, имеющих одинаковые пары оценок, лучше учится та, у которой более высокая оценка по определенному предмету, то этот предмет важнее другого. Если предметов более двух, то сравнивать относительную важность двух выбранных предметов можно путем сопоставления общей успеваемости таких школьниц, которые имеют одинаковые пары оценок по этим предметам, но по всем остальным предметам оценки у них должны быть одинаковыми.

Идеи разобранного примера лежат в основе следующих определений. В них под x^{ij} понимается векторная оценка, полученная из векторной оценки x перестановкой её компонентx_i иx_j. Например, если $x = (5,4,3,4)$, то $x^{14} = (4,4,3,5)$ и $x^{23} = (5,3,4,4)$.

Введём два основных определения. Критерии K_i и K_j ***равно важны***, или ***одинаково важны***, когда любые две векторные оценки x и $x^{i\,j}$ одинаковы по предпочтению. Критерий K_i

важнее критерия K_j, когда всякая векторная оценка x, в которой $x_i > x_j$, предпочтительнее, чем x^{ij}. Согласно первому из этих определений, сообщение $i \approx j$ связывает векторные оценки x и y, такие, что $y = x^{ij}$, отношением ***безразличия*** $I^{i\approx j}$. Согласно второму определению, сообщение $i \succ j$ связывает векторные оценки x и y, такие, что $x_i > x_j, y = x^{ij}$, отношением ***предпочтения*** $P^{i>j}$. Например, верно $(5,4,3,4)P^{1>2}(4,5,3,4)$, поскольку вторая векторная оценка получается из первой перестановкой её первых двух компонент, причем именно в первой векторной оценке на первом месте (как значение более важного критерия) стоит 5 − бóльшее из двух чисел 4 и 5. Иными словами, указанная перестановка приводит к ухудшению первой векторной оценки. Верно также $(5,4,3,4)I^{2\approx 3}(5,3,4,4)$. Действительно, здесь вторая векторная оценка получена из первой перестановкой второй и третьей компонент, а второй и третий критерии равно важны. Поэтому такая перестановка не приводит к изменению предпочтений. Но вот $(5,4,3,4)I^{2\approx 3}(5,3,5,4)$ неверно, поскольку 2-ая векторная оценка <u>не получена</u> из 1-ой перестановкой 2-ой и 3-ьей координат.

Заметим <u>очень важное</u> обстоятельство. Указанные два правила можно применять для сравнения двух векторных оценок только тогда, когда одна из них получена перестановкой двух координат из другой. Если одна оценка получена перестановкой нескольких координат, то надо построить цепочку векторов, соединяющую две данные векторные оценки, так, чтобы правила можно было бы применять к каждой паре соседних оценок. Если же одна векторная оценка не может быть получена никакой перестановкой из другой (например, (5,4,3,3) и (4,5,3,4)), то иногда можно применять правило Парето P: одна оценка не меньше другой по всем координатам и при этом хотя бы по одной координате строго больше. Такие пары можно «встраивать» в цепочки. Примеры будут даны ниже.

3.2. Использование качественной информации о важности критериев для сравнения пары векторных оценок. Накопленную качественную информацию о важности Θ нужно проверять на непротиворечивость, поскольку в неё могут вкрасться ошибки. Противоречивость проявляется в том, что при помощи сообщений из Θ можно составить цикл, приводящий к заключению, что некоторый критерий важнее самого себя. Тогда соответствующие сообщения надо проверить и скорректировать. Так, если Θ содержит сообщения $1 \succ 2, 2 \succ 3,\ 3 \succ 1$, то отсюда будет формально вытекать, например, что $1 \succ 1$. Следовательно, по крайней мере, одно из трёх суждений в Θ ошибочно. Если при проверке выяснится, что неверно суждение $3 \succ 1$, то его нужно заменить на $1 \succ 3$, или $1 \approx 3$, или просто удалить из Θ. Известны формальные методы нахождения таких циклов, но здесь они не рассматриваются. Далее предполагается, что рассматриваемые наборы сообщений из Θ не содержат противоречий.

Пример 3. Построение цепочки оценок. Пусть в двухкритериальной задаче известно, что первый критерий важнее второго, т.е. $\Theta = \{1 \succ 2\}$. Понятно, что для векторных оценок (5,3) и (2,4) неверно $(5,3)P^{1>2}(2,4)$ просто потому, что 2-ая оценка не получена из 1-ой перестановкой координат. Неверно также, что $(5,3)P(2,4)$, поскольку 2-ая координата у первой оценки меньше, чем у второй: 3 < 4. Однако можно составить цепочку из двух звеньев: верных соотношений

$(5,3)P^{1>2}(3,5)$ и $(3,5)P(2,4)$.

Это и означает, что оценка (5,3) превосходит (с учётом имеющейся информации Θ о сравнительной важности критериев) оценку (2,4). Далее используем для этого факта запись $xP^{\Theta}y$■

Однако подобные цепочки можно построить не всегда. Например, при той же информации $\Theta = \{1 \succ 2\}$ невозможно связать цепочкой оценки (5,3) и (4,5). Действительно, по правилу 1 можно утверждать, что $(5,3)P^{1>2}(3,5)$. Далее, можно утверждать, что $(4,5)P(3,5)$. Но цепочка не выстраивается, поскольку здесь имеются только превосходства «в разные стороны»: обе оценки превосходят одну и ту же оценку (3,5), что не несёт никакой информации о сравнении исходных оценок. Легко сформулировать общие утверждения.

1. Если имеются две векторные оценки, такие, что сумма координат у первой не больше, чем сумма координат у второй, то отношение Парето P не встраивается ни в какую цепочку, соединяющую первую оценку со второй.

2. Если имеются две векторные оценки, такие, что сумма координат у первой меньше, чем сумма координат у второй, то первая оценка не может превосходить вторую.

Первое утверждение очевидно в силу того, что у любой оценки, превосходящей по Парето другую, сумма координат обязательно больше. Второе утверждение следует из того, что правила 1 и 2 применяются только к парам оценок, полученным одна из другой перестановкой двух координат. Значит, при произвольной длине цепочек из таких оценок сумма координат остаётся постоянной. А применение правила Парето может только уменьшить сумму координат. Заметим, что эти утверждения ничего не говорят о сравнимости произвольных оценок с совпадающими суммами координат.

Пример 4. Более сложный случай. Рассмотрим 5-ую и 6-ую оценки из таблицы 1: (5,4,3,5) и (3,5,4,4). Предположим, что имеется следующая информация о важности критериев: Θ = $\{1\approx 2,\ 2\succ 3, 3\approx 4\}$. Попробуем сравнить указанные оценки. Имеем
$(5,4,3,5)P^{2>3}(5,3,4,5)I^{1\approx 2}(3,5,4,5)I^{3\approx 4}(3,5,5,4)P(3,5,4,4)$.
При 1-ом сравнении использовано превосходство в важности между 2-ым и 3-ьим критериями, при 2-ом сравнении – равноценность 1-го и 2-го критериев, при 3-ьем сравнении – равноценность 3-го и 4-го критериев, при 4-ом сравнении – превосходство по Парето между соответствующими оценками ■

Подчеркнем двоякую роль построенной в примере 4 цепочки и подобных ей. С одной стороны, такая цепочка от x к y показывает, что верно $xP^{\Theta}y$. С другой стороны, она позволяет наглядно и ясно объяснить, причем в терминах собранных сведений о предпочтениях, почему x предпочтительнее, чем y. Ив этом ещё одно существенное преимущество рассматриваемых методов перед теми, которые опираются на построение обобщенных критериев. Поэтому подобного рода цепочки будем называть ***объясняющими***. Понятно, что в роли объясняющей желательно иметь цепочку возможно меньшей длины.

Конечно, есть формальные методы, находящие цепочки минимальной длины или устанавливающие отсутствие требуемых объясняющих цепочек. Алгоритм одного из них будет изложен в следующем разделе.

Рассмотрим теперь 2-ую и 7-ую оценки из таблицы 1: (3,4,4,3) и (3,5,3,3). Для этих двух оценок не существует связывающих их объясняющих цепочек. Поскольку суммы координат у них одинаковые, то отношение Парето не может быть встроено в такую цепочку (утверждение 1). Поскольку вторая оценка не получена перестановкой координат из первой (в неё входит 5, которой нет в первой оценке), то применением правил 1 и 2 в любом порядке и количестве нельзя получить требуемого сравнения. Заметим, что этот вывод остаётся в силе для любой информации о важности критериев Θ.

Однако можно привести пример, когда информация Θ существенно влияет на сравнение одной и той же пары векторных оценок. Рассмотрим снова 5-ую и 6-ую оценки из таблицы 1: (5,4,3,5) и (3,5,4,4). Предположим, что имеется следующая информация о важности критериев: $\Theta = \{1\approx 2,\ 3\approx 4\}$, т.е. теперь нет информации о сравнении критериев 1 и 2 с критериями 3 и 4. Поэтому у нас нет возможности менять местами любую из двух первых координат с любой из двух последних. Это значит, что на двух последних местах в любой оценке, входящей в цепочку с началом (5,4,3,5), может стоять только 3,5 или 5,3. А на двух последних местах в заключительной оценке (3,5,4,4) стоят 4,4. Поскольку и 3,5, и 5,3 несравнимы по Парето с 4,4, то и сами оценки несравнимы.

В заключение этого раздела рассмотрим крайние случаи.

1. Если имеется информация о равноважности всех критериев, то вопрос о сравнении двух векторных оценок x и y решается следующим образом. Все координаты оценки x упорядочиваются в порядке невозрастания, т.е. каждая следующая координата не превосходит предыдущую (равна или меньше). Полученный вектор обозначим через $x\downarrow$. Аналогично определяем вектор $y\downarrow$. Оценка x лучше оценки $y(xP^{\Theta}y)$ тогда и только тогда, когда $x\downarrow Py\downarrow$.
2. Информация о сравнительной важности критериев отсутствует ($\Theta=\varnothing$). Тогда $P^{\varnothing} = P$ (т.е. одна оценка лучше другой тогда и только тогда, когда она доминирует её по Парето).

3.3. Алгоритм сравнения двух векторных оценок по отношениям Подиновского. В настоящем разделе приводится алгоритм построения объясняющей цепочки для двух любых векторных оценок x, y и любой информации Θ рассматриваемого типа. В результате работы алгоритма либо строится требуемая цепочка, либо устанавливается её отсутствие. Рассматриваются два случая:
А) $S(x) = S(y)$ и $x\downarrow = y\downarrow$;
Б) $S(x) > S(y)$.
В остальных случаях, как было указано выше, ответы очевидны и не требуют специальных алгоритмов.

Рассмотрим случаи А) и Б) по отдельности.

А). Если $S(x) = S(y)$ и$x\downarrow = y\downarrow$, то надо тщательно и терпеливо выписать всё, что можно получить из вектора x с помощью цепочек допустимых перестановок. Для этого предлагается специальный алгоритм 2А.

Алгоритм 2А. Алгоритм состоит в последовательном записывании информации в строчки по определённым правилам.

Шаг 0. В верхнюю (0-ую) строчку напишем исходный вектор x.

Шаг i (i>0). В i-ую строчку последовательно записываются все векторы, которые можно получить из всех векторов, записанных в $(i-1)$-ую строчку, одной допустимой перестановкой координат. (Напомним, что допустимая перестановка координат – это перестановка двух координат s и t, если они указаны в Θ как равноценные или перестановка двух координат с номерами s и t, если указано, что sважнее t, и при этом s-ая координата больше, чем t-ая). При вписывании во все строчки, кроме 0-ой, вписывается слева направо следующие наборы: вектор из предыдущей $(i-1)$-ой строчки, символ операции (превосходство $P^{s>t}$ или равноценность $I^{s\approx t}$), новый вектор (полученный из записанного слева от знака операции перестановкой координат).

При вписывании проверяются три условия:

а) если получившийся вектор уже ранее вписывался на одном из шагов с 0-го до i-го, то ничего не вписывается;

б) если получившийся вектор совпадёт с заданным векторомy, то алгоритм останавливается.

в) если все векторы из $(i-1)$-ой строчки проверены и никаких записей в i-ую строчку не сделано, алгоритм останавливается ■

Работа алгоритма **2А** будет ниже продемонстрирована на примерах.

Если алгоритм **2А** останавливается в соответствии с условием б), то объясняющая цепочка строится от конца к началу, переходя от правых частей записей к левым и поднимаясь к предыдущим строкам. Причём, если в эту цепочку не входит ни одного отношения вида $P^{i>j}$, то это значит, что векторы x и y равноценны по отношению P^{Θ}. Если в эту цепочку входит хотя бы одно отношение вида $P^{i>j}$, то это значит, что x превосходит y по отношению P^{Θ} и при этом y не превосходит x по отношению P^{Θ}. Построенная объясняющая цепочка является кратчайшей.

Заметим, что в случае остановки алгоритма по условию в) никакого вывода о превосходстве y над x или об отсутствии такого превосходства сделать нельзя. Надо применить описанный выше алгоритм к векторам y и xв указанном порядке (т.е. начинать построения с y).

Пример 5. Пусть $\Theta = \{3\succ 2,\ 1\approx 3\}$. Сравним два вектора: x= (5,3,4,5) и y = (4,5,3,5).Поскольку $S(x) = S(y)$ и$x\downarrow = y\downarrow$, то необходимо применить алгоритм **2А**.

В данном случае имеем (по строчкам)

1. (5,3,4,5)
2. (5,3,4,5)$I^{1\approx3}$(4,3,5,5), (5,3,4,5)$P^{3>2}$(5,4,3,5).
3. (4,3,5,5)$P^{3>2}$(4,5,3,5).

Алгоритм останавливается, так как получившийся вектор (4,5,3,5) = y. Вся цепочка такова:

x = (5,3,4,5)$I^{1\approx3}$(4,3,5,5)$P^{3>2}$(4,5,3,5) = y.

Так как в эту цепочку входит отношение превосходства $P^{3>2}$, то в соответствии с описанием алгоритма x превосходит y по отношению P^{Θ}, и при этом y не превосходит x■

Пример 6. Пусть снова $\Theta = \{3\succ 2,\ 1\approx 3\}$.Сравним тот же вектор x = (5,3,4,5) с другим векторомy = (3,5,5,4). Поскольку $S(x) = S(y)$ и $x\downarrow = y\downarrow$, то необходимо применить алгоритм **2А**. В данном случае имеем (по строчкам)

1. (5,3,4,5)
2. (5,3,4,5) $I^{1\approx3}$ (4,3,5,5), (5,3,4,5) $P^{3>2}$ (5,4,3,5).
3. (4,3,5,5) $P^{3>2}$ (4,5,3,5), (5,4,3,5) $I^{1\approx3}$ (3,4,5,5).
4. (4,5,3,5) $I^{1\approx3}$ (3,5,4,5) (применение к (3,4,5,5) отношения $P^{3>2}$ приводит к уже найденному (3,5,4,5)).

Понятно, что новых векторов из (3,5,4,5) не получим: $P^{3>2}$ неприменимо, так как на 3-ем месте стоит 4, на 2-ом – 5 и 4 < 5. А применение $I^{1\approx3}$ приводит к тому, что уже было. В соответствии с описанием алгоритм останавливается. При этом установлено, что вектор (5,3,4,5) не превосходит вектор (3,5,5,4)по отношению P^{Θ}■

Пример 7. Сравним те же два вектора в обратном порядке, т.е. начнём с (3,5,5,4). В соответствии с алгоритмом имеем

1. (3,5,5,4).
2. (3,5,5,4) $I^{1\approx3}$ (5,5,3,4).
Больше ни одного нового вектора получить нельзя, так как $P^{3>2}$ неприменимо, а $I^{1\approx3}$ приводит к тому, что уже было. Поэтому в данном случае вектор (3,5,5,4) не превосходит вектор (5,3,4,5) по отношению P^{Θ}. Вспоминая (см. пример 6), что вектор (5,3,4,5) также не превосходит вектор (3,5,5,4), то можно со всей определённостью сказать, что они несравнимы по этому отношению ■

Пример 8. Пусть теперь Θ= {3≻2, 1≈3, 4≻1}. Рассмотрим те же самые векторы x = (5,3,4,5) и y = (3,5,5,4). Попробуем сравнить их, используя бóльшую информацию Θ. В соответствии с тем же алгоритмом **2А** получаем
1. (5,3,4,5)
2. (5,3,4,5) $I^{1\approx3}$(4,3,5,5), (5,3,4,5)$P^{3>2}$ (5,4,3,5).
3. (4,3,5,5) $P^{3>2}$(4,5,3,5), (4,3,5,5) $P^{4>1}$(5,3,5,4), (5,4,3,5) $I^{1\approx3}$ (3,4,5,5).
4. (4,5,3,5) $P^{4>1}$(5,5,3,4), (4,5,3,5) $I^{1\approx3}$(3,5,4,5), (3,4,5,5) $P^{4>1}$(5,4,5,3).
5. (5,5,3,4) $I^{1\approx3}$(3,5,5,4),
Вектор (3,5,5,4) появился в 5-ой строчке. Это означает, что найдена объясняющая цепочкаx = (5,3,4,5) $I^{1\approx3}$ (4,3,5,5) $P^{3>2}$ (4,5,3,5) $P^{4>1}$ (5,5,3,4) $I^{1\approx3}$(3,5,5,4) = y.
Таким образом, на основе информации Θ = {3≻2, 1≈3, 4≻1} о сравнительной важности критериев можно утверждать, что (5,3,4,5)P^{Θ}(3,5,5,4), т.е. набор оценок (5,3,4,5) является более предпочтительным, чем набор (3,5,5,4). Заметим, что просто «глазами» найти такую цепочку было бы затруднительным ■

Б. Перейдём теперь к случаю Б: $S(x)$ >$S(y)$. В этом случае возможны только два ответа: $xP^{\Theta}y$ или x и y несравнимы. Для выяснения ответа используется алгоритм **2Б**, в большой степени напоминающий алгоритм **2А**.

Алгоритм 2Б.
Шаг 0. В верхнюю (0-ую) строчку напишем исходный вектор x.
Шаг i (i>0). В i-ую строчку последовательно записываются все векторы, которые можно получить из всех векторов, записанных в ($i-1$)-ую строчку, одной допустимой перестановкой координат. (Напомним, что допустимая перестановка координат – это перестановка двух координат sи t, если они указаны в Θ как равноценные или перестановка двух координат с номерами sи t, если указано, что sважнее t, и при этом s-ая координата больше, чем t-ая). При вписывании во все строчки, кроме 0-ой, вписывается слева направо следующие наборы: вектор из предыдущей ($i-1$)-ой строчки, символ операции ($P^{s>t}$ или равноценности $I^{s\approx t}$), новый вектор (полученный из записанного слева от знака операции перестановкой координат).

При вписывании проверяются три условия:

а) если получившийся вектор уже ранее вписывался на одном из шагов с 0-го до i-го, то ничего не вписывается;

б) если получившийся вектор превосходит по отношению Парето заданный вектор y, то алгоритм останавливается.

в) если все векторы из ($i-1$)-ой строчки проверены и никаких записей в i-ую строчку не сделано, алгоритм останавливается■

Работа алгоритма **2Б** будет ниже продемонстрирована на примерах.

Если алгоритм **2Б** останавливается в соответствии с условием б), то объясняющая цепочка строится от конца к началу, переходя от правых частей записей к левым и поднимаясь к предыдущим строкам. В отличие от алгоритма 1, в этом случае всегда x превосходит y по отношению P^{Θ} и при этом y никогда не превосходит x по отношению P^{Θ}. Построенная объясняющая цепочка является кратчайшей.

Заметим, что в случае остановки алгоритма **2Б** по условию в), в отличие от алгоритма **2А**, можно с уверенностью сделать вывод о несравнимости векторов x и y: в силу условия $S(x)$ >$S(y)$ вектор y не может превосходить вектор x.

Пример 9. Пусть Θ = {3≻2, 1≈3}. Сравним два вектора: x = (5,3,4,5) иy = (4,5,3,4). Поскольку $S(x)$>$S(y)$, то необходимо применить алгоритм **2Б**.
В данном случае имеем (по строчкам)
1. (5,3,4,5)
2. (5,3,4,5)$I^{1\approx3}$(4,3,5,5), (5,3,4,5)$P^{3>2}$(5,4,3,5).
3. (4,3,5,5)$P^{3>2}$(4,5,3,5).

Алгоритм останавливается, так как получившийся вектор (4,5,3,5)P(4,5,3,4) = y. Вся цепочка такова:
x = (5,3,4,5) $I^{1\approx 3}$ (4,3,5,5) $P^{3>2}$ (4,5,3,5) P(4,5,3,4) = y.
В соответствии с описанием алгоритма **2Б** x превосходит yпо отношению P^{Θ}, и при этом yне превосходит x■

Пример 10. Пусть снова Θ = {3≻2, 1≈3}. Сравним тот же вектор x = (5,3,4,5) с другим вектором y = (3,5,5,3). Поскольку $S(x)>S(y)$, то необходимо применить алгоритм **2Б**.
В данном случае имеем (по строчкам)
1. (5,3,4,5)
2. (5,3,4,5) $I^{1\approx 3}$ (4,3,5,5), (5,3,4,5) $P^{3>2}$ (5,4,3,5).
3. (4,3,5,5) $P^{3>2}$ (4,5,3,5), (5,4,3,5) $I^{1\approx 3}$ (3,4,5,5).
4. (4,5,3,5) $I^{1\approx 3}$ (3,5,4,5).

Понятно, что новых векторов из (3,5,4,5) не получим: $P^{3>2}$ неприменимо, так как на 3-ем месте стоит 4, на 2-ом – 5 и 4 < 5. А применение $I^{1\approx 3}$ приводит к тому, что уже было.
В соответствии с описанием алгоритм **2Б** останавливается. При этом установлено, что вектор (5,3,4,5) <u>не превосходит</u> вектор (3,5,5,3) по отношению P^{Θ}■

Пример 11. Пусть теперь Θ= {3≻2, 1≈3, 4≻1}. Рассмотрим те же самые векторы x = (5,3,4,5) и y = (3,5,5,3). Попробуем сравнить их, используя бóльшую информацию Ω. В соответствии с тем же алгоритмом **2Б** получаем
1. (5,3,4,5)
2. (5,3,4,5) $I^{1\approx 3}$(4,3,5,5), (5,3,4,5)$P^{3>2}$ (5,4,3,5).
3. (4,3,5,5) $P^{3>2}$(4,5,3,5), (4,3,5,5) $P^{4>1}$(5,3,5,4), (5,4,3,5) $I^{1\approx 3}$ (3,4,5,5).
4. (4,5,3,5) $P^{4>1}$(5,5,3,4), (4,5,3,5) $I^{1\approx 3}$(3,5,4,5), (3,4,5,5) $P^{4>1}$(5,4,5,3).
5. (5,5,3,4) $I^{1\approx 3}$(3,5,5,4),

Алгоритм останавливается, так как появившийся в 5-ой строчке вектор (3,5,5,4) превосходит по Парето (3,5,5,3) = y. Вся цепочка такова:
x = (5,3,4,5) $I^{1\approx 3}$ (4,3,5,5) $P^{3>2}$ (4,5,3,5) $P^{4>1}$ (5,5,3,4) $I^{1\approx 3}$(3,5,5,4) P(3,5,5,3) = y.
В соответствии с описанием алгоритма 2 x превосходит y по отношению P^{Θ}, и при этом y не превосходит x■

4. Задания

Задание 1. Найти алгоритмом 1 множество Парето для заданного множества точек. См. примеры 1 и 2 для образца.

Варианты задания 1:
01. Ω = {(3,7,5), (4,5,1), (2,3,0), (5,6,4), (1,4,3), (8,1,2)}
02. Ω = {(3,7,5), (4,5,-1), (2,3,0), (5,6,-4), (1,4,3), (8,1,2)}
03. Ω = {(3,7,5), (4,5,1), (2,3,0), (5,6,-4), (-1,4,3), (8,1,2)}
04. Ω = {(3,7,5), (4,5,-1), (2,-3,0), (5,6,4), (1,4,3), (8,1,2)}
05. Ω = {(3,7,5), (4,5,1), (2,3,0), (5,6,4), (1,4,-3), (8,1,2)}
06. Ω = {(3,7,5), (4,5,-1), (2,3,0), (-5,6,4), (1,4,3), (8,1,2)}
07. Ω = {(3,7,5), (4,5,1), (2,3,0), (5,6,-4), (-1,4,3), (8,1,2)}
08. Ω = {(3,7,5), (4,5,-1), (2,3,0), (5,6,4), (1,4,3), (8,1,-2)}
09. Ω = {(3,7,5), (4,5,-1), (2,3,0), (5,6,-4), (1,4,3), (8,1,2)}
10. Ω = {(-3,7,5), (4,5,1), (2,3,0), (5,6,4), (1,4,3), (8,1,2)}
11. Ω = {(3,-7,5), (4,5,-1), (2,3,0), (5,6,-4), (1,4,3), (8,1,2)}
12. Ω = {(3,7,-5), (4,5,1), (2,3,0), (5,6,-4), (-1,4,3), (8,1,2)}
13. Ω = {(3,7,5), (-4,5,-1), (2,-3,0), (5,6,4), (1,4,3), (8,1,2)}
14. Ω = {(3,7,5), (4,-5,1), (2,3,0), (5,6,4), (1,4,-3), (8,1,2)}
15. Ω = {(3,7,5), (4,5,1), (2,3,0), (-5,6,4), (1,4,3), (8,1,2)}
16. Ω = {(3,7,5), (4,5,1), (-2,3,0), (5,6,-4), (-1,4,3), (8,1,2)}
17. Ω = {(3,7,5), (4,5,-1), (2,-3,0), (5,6,4), (1,4,3), (8,1,-2)}
18. Ω = {(3,7,5), (4,5,-1), (2,3,-0), (5,6,-4), (1,4,3), (8,1,2)}
19. Ω = {(3,7,5), (4,5,1), (2,3,0), (-5,6,4), (1,4,3), (8,1,2)}
20. Ω = {(3,7,5), (4,5,-1), (2,3,0), (5,-6,-4), (1,4,3), (8,1,2)}
21. Ω = {(3,7,5), (4,5,1), (2,3,0), (5,6,4), (-1,4,3), (8,1,2)}

22. Ω = {(3,7,5), (4,5,-1), (2,-3,0), (5,6,4), (-1,4,3), (8,1,2)}
23. Ω = {(3,7,5), (4,5,1), (2,3,0), (5,6,4), (1,-4,-3), (8,1,2)}
24. Ω = {(3,7,5), (4,5,-1), (2,3,0), (-5,6,4), (1,4,-3), (8,1,2)}
25. Ω = {(3,7,5), (4,5,1), (2,3,0), (5,6,-4), (-1,4,3), (-8,1,2)}
26. Ω = {(3,7,5), (4,5,-1), (2,3,0), (5,6,4), (1,4,3), (8,-1,-2)}
27. Ω = {(3,7,5), (4,5,-1), (2,3,0), (5,6,-4), (1,4,3), (8,1,-2)}
28. Ω = {(3,7,5), (4,5,1), (2,3,0), (5,6,4), (1,4,3), (8,-1,2)}
29. Ω ={(3,7,5), (4,5,-1), (2,3,0), (5,6,-4), (1,4,3), (-8,1,2)}
30. Ω = {(3,7,5), (4,5,1), (2,3,0), (5,6,-4), (-1,4,-3), (8,1,2)}
31. Ω = {(3,7,5), (4,5,-1), (2,-3,0), (5,6,4), (1,-4,3), (8,1,2)}
32. Ω = {(3,7,5), (4,5,1), (2,3,0), (5,6,4), (-1,4,-3), (8,1,2)}
33. Ω = {(3,7,5), (4,5,-1), (2,3,0), (-5,6,-4), (1,4,3), (8,1,2)}
34. Ω = {(3,7,5), (4,5,1), (2,3,0), (5,-6,-4), (-1,4,3), (8,1,2)}
35. Ω = {(3,7,5), (4,5,-1), (2,3,0), (-5,6,4), (1,4,3), (8,1,-2)}
36. Ω = {(3,7,5), (4,5,-1), (2,-3,0), (5,6,-4), (1,4,3), (8,1,2)}
37. Ω = {(3,7,5), (4,5,1), (-2,3,0), (5,6,4), (1,4,3), (8,1,2)}
38. Ω ={(3,7,5), (4,5,1), (2,3,0), (5,6,-4), (1,4,3), (8,1,2)}
39. Ω = {(3,7,5), (4,-5,1), (2,3,0), (5,6,-4), (-1,4,3), (8,1,2)}
40. Ω = {(3,7,5), (-4,5,-1), (2,-3,0), (5,6,4), (1,4,3), (8,1,2)}
41. Ω = {(3,7,-5), (4,5,1), (2,3,0), (5,6,4), (1,4,-3), (8,1,2)}
42. Ω = {(3,-7,5), (4,5,-1), (2,3,0), (-5,6,4), (1,4,3), (8,1,2)}
43. Ω = {(-3,7,5), (4,5,1), (2,3,0), (5,6,-4), (-1,4,3), (8,1,2)}
44. Ω = {(-3,7,5), (-4,5,-1), (2,3,0), (5,6,4), (1,4,3), (8,1,-2)}
45. Ω = {(3,7,5), (-4,5,-1), (-2,3,0), (5,6,-4), (1,4,3), (8,1,2)}
46. Ω = {(3,7,5), (4,5,-1), (-2,3,0), (-5,6,4), (1,4,3), (8,1,-2)}
47. Ω = {(3,7,5), (4,5,-1), (2,3,0), (-5,6,-4), (1,4,3), (8,1,2)}
48. Ω = {(3,7,5), (4,5,-1), (2,3,0), (5,6,4), (-1,4,3), (8,1,-2)}
49. Ω = {(3,7,5), (4,5,-1), (2,3,0), (5,6,-4), (-1,4,3), (-8,1,2)}
50. Ω = {(-3,7,5), (4,5,-1), (2,3,0), (5,6,4), (1,4,3), (-8,1,-2)}■

Задание 2. Для данных по оценкам 7-и учениц найти и нарисовать их паретовские уровни со всеми дугами, соединяющими только соседние уровни. См. пример 2 для образца.

Варианты задания 2:

Школьник	Алгебра	Геометрия	Литература	Английский
1. Арсенкова Александра	5	4	4	5
2. Баранникова Елена	3	4	4	3
3. Кулакова Екатерина	3	3	4	3
4. Михайлова Елена	3	3	4	4
5. Морозова Мария	5	4	3	5
6. Перепечаева Арина	4	3	5	4
7. Полищук Наталья	3	5	3	3

01

Школьник	Алгебра	Геометрия	Литература	Английский
1. Арсенкова Александра	3	4	4	5
2. Баранникова Елена	5	4	4	3
3. Кулакова Екатерина	5	3	4	3
4. Михайлова Елена	5	3	4	4
5. Морозова Мария	3	4	3	5
6. Перепечаева Арина	4	3	5	4
7. Полищук Наталья	5	5	3	3

02

1Школьник	Алгебра	Геометрия	Литература	Английский
1. Арсенкова Александра	5	4	4	5
2. Баранникова Елена	3	4	4	3
3. Кулакова Екатерина	3	5	4	3

4. Михайлова Елена	3	5	4	4
5. Морозова Мария	5	4	3	5
6. Перепечаева Арина	4	5	5	4
7. Полищук Наталья	3	3	3	3

03

Школьник	Алгебра	Геометрия	Литература	Английский
1. Арсенкова Александра	5	4	4	5
2. Баранникова Елена	3	4	4	3
3. Кулакова Екатерина	3	3	4	3
4. Михайлова Елена	3	3	4	4
5. Морозова Мария	5	4	5	5
6. Перепечаева Арина	4	3	3	4
7. Полищук Наталья	3	5	5	3

04

Школьник	Алгебра	Геометрия	Литература	Английский
1. Арсенкова Александра	5	4	4	3
2. Баранникова Елена	3	4	4	5
3. Кулакова Екатерина	3	3	4	5
4. Михайлова Елена	3	3	4	4
5. Морозова Мария	5	4	3	3
6. Перепечаева Арина	4	3	5	4
7. Полищук Наталья	3	5	3	5

05

Школьник	Алгебра	Геометрия	Литература	Английский
1. Арсенкова Александра	5	3	3	5
2. Баранникова Елена	4	3	3	4
3. Кулакова Екатерина	4	4	3	4
4. Михайлова Елена	4	4	3	3
5. Морозова Мария	5	3	4	5
6. Перепечаева Арина	3	4	5	3
7. Полищук Наталья	4	5	4	4

06

Школьник	Алгебра	Геометрия	Литература	Английский
1. Арсенкова Александра	4	5	5	4
2. Баранникова Елена	3	5	5	3
3. Кулакова Екатерина	3	3	5	3
4. Михайлова Елена	3	3	5	5
5. Морозова Мария	4	5	3	4
6. Перепечаева Арина	5	3	4	5
7. Полищук Наталья	3	4	3	3

07

Школьница	Алгебра	Геометрия	Литература	Английский
1. Арсенкова Александра	5	4	4	5
2. Баранникова Елена	3	4	4	3
3. Кулакова Екатерина	3	3	4	3
4. Михайлова Елена	3	3	4	4
5. Морозова Мария	5	4	3	5
6. Перепечаева Арина	3	5	4	4
7. Полищук Наталья	3	5	3	3

08■

Задание 3. При Θ= {3 ≻2, 1≈3, 4≻3}сравнить все пары оценок из данной таблицы. См. примеры 5 – 11 для образца.

Варианты те же, что в задании 2 ■

5. Предметный указатель

Глава 16. Коллективные решения

1. Модель коллективного решения
2. Правила на основе ранжировок
3. Правила на основе численности коалиций
4. Задания
5. Предметный указатель

1. Модель коллективного решения

В основе дальнейшего изложения лежит понятие ***ранжировки***. Ранжировка множества представляет собой упорядоченный набор его элементов (кортеж). Считается, что первый в указанном порядке элемент получает ранг 1, следующий за ним – ранг 2, и т.д., вплоть до последнего элемента, который получает ранг m(m – это число элементов, или мощность исходного множества). Записываться ранжировки множества $\Omega = \{x_1, \ldots, x_m\}$ будут в виде

r: $x_{i_1} \succ x_{i_2} \succ \ldots \succ x_{i_m}$. (1)

Запись (1) означает, что на 1-ом месте в ранжировке rстоит элемент x_{i_1}, за ним – элемент x_{i_2}, и так далее, вплоть до последнего элемента x_{i_m}. Легко понять, что формально любая ранжировка r задаёт отношение линейного порядка P на том же множестве Ω. Именно, xPy тогда и только тогда, когда в ранжировке r элемент xвстречается раньше, чем элемент y(т.е. x записан левее, чем y). Обратно, каждое отношение линейного порядка P на множестве Ω так же естественно определяет некоторую ранжировку r множества Ω.

Описанные ранжировки иногда называются ***строгими***, чтобы отличать их от более общих ранжировок, при которых разные объекты могут получать одинаковые ранги. Таким образом, в общей, или нестрогой ранжировке все элементы исходного множества Ω разбиваются на несколько групп: Ω_1, Ω_2, …, Ω_d, так что при этом все элементы из Ω_1 получают ранг 1, из Ω_2 – ранг 2, и т.д., вплоть до Ω_d, все элементы которого получают ранг d. Нестрогие ранжировки определяют слабый порядок P(см. раздел 14-2.3) на множестве Ω. Именно, xPy тогда и только тогда, когда индекс множества Ω_i, в которое входит элемент x, меньше индекса множества Ω_j, в которое входит элемент y. Обратно, каждое отношение слабого порядка P на множестве Ω так же естественно определяет некоторую ранжировку r множества Ω. Для нестрогих ранжировок будем пользоваться обозначением

$\Omega_1 \succ \Omega_2 \succ \ldots \succ \Omega_d$. (2)

Определим рассматриваемое здесь понятие ***коллективного решения***. Предполагается, что члены группы, состоящей из n участников ($n > 2$), выражают своё мнение относительно альтернатив из множества $\Omega = \{x_1, \ldots, x_m\}$ ($m \geq 2$), в виде строгих ранжировок r_i этого множества Ω (i = 1, …, n). В ранжировке r_i лучшая (с точки зрения i-го участника) альтернатива получает ранг 1, и т.д., вплоть до худшей альтернативы, которая получает ранг m. Под ***коллективным решением*** будет пониматься некоторая (уже не обязательно строгая) ранжировка, которая построена, исходя из ранжировок, данных всеми участниками. Можно сказать, что она агрегирует их индивидуальные предпочтения и выражает коллективное мнение по поводу рассматриваемых альтернатив из Ω. Различные способы построения такого коллективного решения и излагаются в настоящей главе.

Главным признаком, по которому разделяются представленные в настоящей главе методы, является вид используемой информация о ранжировках участников. В разделе 2 рассматриваются коллективные решения, построенные непосредственно по этим ранжировкам. В разделе 3 излагаются методы, использующие не сами исходные ранжировки, а определяемые ими численности некоторых групп участников. Остановимся на этом подробнее.

Любое множество участников назовём ***коалицией***. Пусть a и b– любые две альтернативы. Обозначим через $V(a, b)$ коалицию участников, считающих, что альтернатива a предпочтительней, чем альтернатива b, т.е. a имеет ме́ньший ранг, чем b, в ранжировках участников из $V(a, b)$. В разделе 3 излагаются методы, использующие при построении коллективного решения найденные по исходным ранжировкам числа $|V(a, b)|$ для всех пар альтернатив. Их выделение объясняется не только важностью и распространённостью этих методов для построения коллективного решения, но и тем, что их аналоги и модификации можно применять и для упорядочивания альтернатив на основе одного бинарного отношения безотносительно к проблематике коллективных решений. Среди методов этой группы можно выделить методы, основанные только на информации о том, для каких пар альтернатив числа $|V(a, b)|$ больше, чем $n/2$. Эти широко распространённые методы называются методами, основанными на ***мажоритарном отношении***.

Разумеется, проблематика коллективных решений гораздо шире вышеуказанной постановки. Можно сказать, что здесь рассматривается простейшее понятие коллективного решения. Однако и более сложные модели включают те же базовые элементы: виды индивидуальных предпочтений, вид коллективного предпочтения и само правило его построения.

2. Правила на основе ранжировок

2.1. Правило Бордá. Это правило, названное в честь французского математика Жана-Шарля де Бордá, который его впервые предложил, состоит в том, что суммируются ранги каждой альтернативы, затем альтернативы с наименьшей суммой рангов объявляются самыми предпочтительными, а далее предпочтения выстраиваются в порядке возрастания суммы рангов.

Пример 1. Пусть три участника ранжируют три альтернативы x, y, z следующим образом:

$$r_1: x \succ y \succ z;$$
$$r_2: x \succ z \succ y;$$
$$r_3: y \succ x \succ z.$$

Альтернатива x получила ранг 1 у 1-го и 2-го участников и ранг 2 у 3-его участника. Сумма рангов $S(x) = 1+1+2 = 4$. Альтернатива y получила ранг 2 у 1-го участника, ранг 3 у 2-го участника и ранг 1 у 3-го участника, т.е. $S(y) = 2+3+1 = 6$. Наконец, альтернатива z получила ранг 3 у 1-го и 3-го участников и ранг 2 у 2-го участника, т.е. $S(z) = 3+2+3 = 8$. Таким образом, располагая альтернативы в порядке возрастания суммарных рангов, приходим к строгой ранжировке $r{:}x \succ y \succ z$■

Обратим внимание на то, что в примере 1 коллективное решение, построенное по правилу Бордá, представляет собой строгую ранжировку, т.е., по сути дела, линейный порядок. Но так бывает не всегда. Рассмотрим следующий

Пример 2. Пусть три участника ранжируют четыре альтернативы x, y, z, w, следующим образом:

$$r_1: w \succ x \succ y \succ z;$$
$$r_2: w \succ z \succ x \succ y;$$
$$r_3: w \succ y \succ z \succ x.$$

Подсчитаем суммарные ранги каждой альтернативы. Для альтернативы $wS(w) = 3$. Для альтернативы x имеем $S(x) = 2+3+4 = 9$. Аналогично получаем $S(y) = 3+4+2 = 9$ и $S(z) = 3+2+3 = 9$. Таким образом, в данном случае имеем нестрогую ранжировку r: $\{w\} \succ \{x, y, z\}$ ■

Пример 3. В случае циклически сдвинутых ранжировок

$$r_1: x \succ y \succ z;$$
$$r_2: y \succ z \succ x;$$
$$r_3: z \succ x \succ y$$

получаем$S(x) = S(y) = S(z) = 6$. Поэтому единственная группа альтернатив, получивших ранг 1, совпадает с исходным множеством $\{x, y, z\}$. Нестрогая ранжировка $\{x, y, z\}$ и ранжировкой, по сути дела, не является – все альтернативы равноправны ■

Желание избежать подобных ситуаций инициировало разработку нескольких правил построения агрегирующих ранжировок, которые рассматриваются в разделах 2.2 – 2.4

2.2. Паретовское правило. Пусть, как и ранее, n участников ранжируют m альтернатив, образующих множество Ω. Каждой альтернативе x сопоставляется вектор $v(x) = (v_1(x), \ldots, v_n(x))$, где $v_i(x)$ – ранг, который альтернатива x получает у i-го участника (i= 1, …, n). Определим бинарное отношение R на множестве альтернатив Ω следующим образом: xRy тогда и только тогда, когда вектор $v(x)$ превосходит по Парето вектор $v(y)$ (в данном случае «лучше» значит «меньше»). Легко видеть, что отношение R на множестве Ω будет ациклическим и транзитивным, т.е. по определению из раздела 14-2.3 оно является частичным порядком. Положим Ω_1 равным множеству Ω^R недоминируемых по отношению R альтернатив, $\Omega_2 = (\Omega \backslash \Omega_1)^R$, и т.д.

Присвоим всем элементам множества Ω_1 ранг 1, множества Ω_2 – ранг 2, и т.д. Полученную нестрогую ранжировку можно назвать ***паретовской ранжировкой*** (вообще говоря, она может быть и строгой).

Пример 4. 3 участника ранжировали 5 альтернатив следующим образом:

r_1: $x_1 \succ x_3 \succ x_4 \succ x_2 \succ x_5$;

r_2: $x_2 \succ x_5 \succ x_1 \succ x_3 \succ x_4$;

r_3: $x_2 \succ x_1 \succ x_4 \succ x_5 \succ x_3$.

Напишем для этих альтернатив векторы $v(x)$. По построению, $v(x_1) = (1,3,2)$, $v(x_2) = (4,1,1)$, $v(x_3) = (2,4,5)$, $v(x_4) = (3,5,3)$, $v(x_5) = (5,2,4)$. Эти данные означают, что векторы $v(x_3)$ и $v(x_4)$ доминируются вектором $v(x_1)$, вектор $v(x_5)$ доминируется вектором $v(x_2)$, а векторы $v(x_1)$ и $v(x_2)$ никем не доминируются. Поэтому $\Omega_1 = \{x_1, x_2\}$. Оставшиеся три вектора $v(x_3) = (2,4,5)$, $v(x_4) = (3,5,3)$, $v(x_5) = (5,2,4)$ несравнимы по отношению Парето. Поэтому паретовская ранжировка в данном случае такова: $\{x_1, x_2\} \succ \{x_3, x_4, x_5\}$.

Для сравнения рассмотрим ранжировку, получаемую правилом Бордá. В данном случае суммы рангов таковы: $S(x_1) = 6$, $S(x_2) = 6$, $S(x_3) = 11$, $S(x_4) = 11$, $S(x_5) = 11$, т.е. альтернативы образуют те же самые группы. Таким образом, коллективные решения, полученные правилом Бордá и паретовским правилом, в данном случае совпадают ■

Пример 5. Пусть теперь три участника ранжировали 5 альтернатив следующим образом:

r_1: $x_1 \succ x_2 \succ x_3 \succ x_4 \succ x_5$;

r_2: $x_1 \succ x_2 \succ x_3 \succ x_4 \succ x_5$;

r_3: $x_2 \succ x_3 \succ x_4 \succ x_5 \succ x_1$.

Паретовское правило даёт следующую нестрогую ранжировку: $\{x_1, x_2\} \succ \{x_3\} \succ \{x_4\} \succ \{x_5\}$. В то же время правило Бордá даёт следующие суммы рангов: $S(x_1) = 7$, $S(x_2) = 5$, $S(x_3) = 8$, $S(x_4) = 11$, $S(x_5) = 14$. Получаем строгую ранжировку: $x_2 \succ x_1 \succ x_3 \succ x_4 \succ x_5$, не совпадающую с той нестрогой ранжировкой, которую даёт паретовское правило■

2.3. Правила с удалением альтернатив (передачей голосов). При описании ряда методов построения коллективного решения возникает необходимость в пересчёте ранжировок при удалении одной альтернативы из некоторого множества. Суть дела поясним на примере.

Пример 6. Пересчёт ранжировок. Пусть 5 участников дали следующие ранжировки для пяти альтернатив, образующих исходное множество Ω:

r_1: $x_5 \succ x_3 \succ x_2 \succ x_4 \succ x_1$;

r_2: $x_1 \succ x_5 \succ x_3 \succ x_4 \succ x_2$;

r_3: $x_4 \succ x_1 \succ x_2 \succ x_3 \succ x_5$;

r_4: $x_5 \succ x_1 \succ x_3 \succ x_4 \succ x_2$;

r_5: $x_5 \succ x_3 \succ x_1 \succ x_4 \succ x_2$.

Предположим, что по каким-либо причинам альтернатива x_1 исключается из рассмотрения. По оставшимся альтернативам мнения участников остаются прежними. Это означает, что новые ранжировки (на множестве $\Omega \setminus \{x_1\}$) имеют вид

r_1: $x_5 \succ x_3 \succ x_2 \succ x_4$;

r_2: $x_5 \succ x_3 \succ x_4 \succ x_2$;

r_3: $x_4 \succ x_2 \succ x_3 \succ x_5$;

r_4: $x_5 \succ x_3 \succ x_4 \succ x_2$;

r_5: $x_5 \succ x_3 \succ x_4 \succ x_2$.

Для исходных ранжировок суммы рангов были такими: $S(x_1) = 13$, $S(x_2) = 21$, $S(x_3) = 14$, $S(x_4) = 17$, $S(x_5) = 10$. Для новых ранжировок (после удаления x_1) получаем $S(x_2) = 17$, $S(x_3) = 11$, $S(x_4) = 14$, $S(x_5) = 8$. Коллективное решение для исходных альтернатив по правилу Бордá имеет вид: $x_5 \succ x_1 \succ x_3 \succ x_4 \succ x_2$. После удаления x_1 по правилу Бордá получаем коллективное решение: $x_5 \succ x_3 \succ x_4 \succ x_2$, т.е. в данном случае <u>коллективное решение после удаления одной альтернативы относительно оставшихся альтернатив не изменилось</u> ■

Пример 7. Другой пересчёт ранжировок. Рассмотрим ранжировки

r_1: $x_1 \succ x_2 \succ x_3 \succ x_4 \succ x_5$;

r_2: $x_1 \succ x_2 \succ x_3 \succ x_4 \succ x_5$;

r_3: $x_1 \succ x_2 \succ x_3 \succ x_4 \succ x_5$;

r_4: $x_2 \succ x_3 \succ x_4 \succ x_5 \succ x_1$.

Для них $S(x_1) = 8$, $S(x_2) = 7$, $S(x_3) = 11$, $S(x_4) = 15$, $S(x_5) = 19$. Таким образом, коллективное решение по правилу Бордá – строгая ранжировка $x_2 \succ x_1 \succ x_3 \succ x_4 \succ x_5$. После удаления x_3 получаем ранжировки

$$r_1:\ \ x_1 \succ x_2 \succ x_4 \succ x_5;$$
$$r_2:\ \ x_1 \succ x_2 \succ x_4 \succ x_5;$$
$$r_3:\ \ x_1 \succ x_2 \succ x_4 \succ x_5;$$
$$r_4:\ \ x_2 \succ x_4 \succ x_5 \succ x_1,$$

откуда $S(x_1) = 7$, $S(x_2) = 7$, $S(x_4) = 11$, $S(x_5) = 12$, и коллективное решение по правилу Бордá – нестрогая ранжировка $\{x_1, x_2\} \succ \{x_4\} \succ \{x_5\}$. Таким образом, удаление альтернативы может менять коллективное решение по правилу Бордá для оставшихся■

Нетрудно привести примеры, когда удаление альтернативы меняет коллективное решение для оставшихся и по паретовскому правилу. Можно сделать вывод, что удаление альтернативы – достаточно «сильное» средство, что и объясняет смысл его использования.

Другой особенностью излагаемых методов является необходимость выбора удаляемой альтернативы. Если все условия удаления оказываются совпадающими, то удаляется альтернатива с мéньшим номером. Это соглашение не лучше и не хуже всякого другого, например, случайного выбора.

2.3.1. Правило Хара. Первоначально ищется альтернатива, получившая более 50% первых мест (т.е. рангов 1) в ранжировках участников. Если такая альтернатива существует, то она и объявляется коллективным решением. Точнее. Коллективным решением объявляется нестрогая ранжировка $\Omega_1 \succ \Omega_2$, в которой множество Ω_1 состоит из одной указанной альтернативы, а множество Ω_2 – из всех остальных альтернатив. Если же такой альтернативы нет, то удаляется альтернатива x, получившая наименьшее число первых мест в ранжировках участников. (Именно здесь применяется упомянутое правило о выборе альтернативы с наименьшим номером.) После этого пересчитываются ранги, и процедура повторяется – до тех пор, пока не найдётся альтернатива, получившая более 50% первых мест. Заметим, что она обязательно найдётся, поскольку на последнем шаге останется единственная альтернатива, имеющая 100% первых мест. Коллективным решением после остановки процедуры на любом шаге будет нестрогая ранжировка, определённая так же, как выше: Ω_1 состоит из выбранной альтернативы, Ω_2 – из всех остальных альтернатив ■

Пример 8. Рассмотрим ранжировки, данные тремя участниками 4-ём альтернативам:

$$r_1:\ \ x_1 \succ x_3 \succ x_4 \succ x_2;$$
$$r_2:\ \ x_2 \succ x_4 \succ x_1 \succ x_3;$$
$$r_3:\ \ x_3 \succ x_1 \succ x_2 \succ x_4.$$

Ни одна альтернатива не получила более 50% первых мест, и три из них получили поровну первых мест – по одному. Применяем соглашение об удалении и удаляем 1-ую альтернативу. Пересчитывая ранги, получаем

$$r_1:\ \ x_3 \succ x_4 \succ x_2;$$
$$r_2:\ \ x_2 \succ x_4 \succ x_3;$$
$$r_3:\ \ x_3 \succ x_2 \succ x_4.$$

Теперь альтернатива x_3 имеет два первых места из трёх (более 50%), поэтому коллективное решение таково: $\{x_3\} \succ \{x_1, x_2, x_4\}$ ■

2.3.2. Правило Нансона. Прежде чем переходить к её описанию, введём несколько простых понятий. Обозначим через r_j^i ранг, который j-ая альтернатива получает у i-го участника ($i = 1, \dots, n$; $j = 1, \dots, m$). Суммарный ранг $S(x_j)$, полученный альтернативой x_j у всех участников, выражается формулой

$$S(x_j) = \sum_{i=1}^{n} r_j^i. \tag{3}$$

Введём понятие средней оценки или ***среднего ранга***. Средний ранr определяется формулой

$$\bar{r} = \left(\sum_{j=1}^{m} S(x_j)\right)/m. \tag{4}$$

Нетрудно видеть, что средний ранг всегда равен $\frac{n(m+1)}{2}$. Он не зависит от конкретных рангов r_j^i, определяемых участниками.

Правило Нансона состоит в следующем. Подсчитываются суммарные ранги $S(x_j)$ ($j = 1, \dots, m$) по формуле (3). Далее из списка альтернатив исключаются все альтернативы, у которых суммарные ранги больше среднего ранга $\bar{r}$. После чего ранги пересчитываются и процедура повторяется с ме́ньшем числом альтернатив. Процедура прекращается, когда все суммарные ранги для всех оставшихся элементов не больше соответствующего среднего. Эти оставшиеся (не исключённые) альтернативы объявляются лучшими. Обозначая их множество через Ω_1, а множество $\Omega\backslash\Omega_1$ через Ω_2, получаем коллективное решение в виде нестрогой ранжировки $\Omega_1 \succ \Omega_2$.

Пример 9. Ранжировки 4-ёх участников на множестве $\Omega = \{x_1, x_2, x_3, x_4, x_5\}$ таковы:

$$r_1:\ x_5 \succ x_4 \succ x_3 \succ x_1 \succ x_2;$$
$$r_2:\ x_1 \succ x_5 \succ x_3 \succ x_4 \succ x_2;$$
$$r_3:\ x_4 \succ x_1 \succ x_2 \succ x_5 \succ x_3;$$
$$r_4:\ x_1 \succ x_5 \succ x_3 \succ x_4 \succ x_2.$$

Выберем лучшие альтернативы по правилу Нансона. В данном случае имеем

$$S(x_1) = 8,\ S(x_2) = 18,\ S(x_3) = 14,\ S(x_4) = 11,\ S(x_5) = 9.$$

Средний ранг $\bar{r}$ при $m = 5$ и $n = 4$ равен 12. Удалим альтернативы x_2 и x_3, у которых суммарные ранги больше среднего ранга 12. Останутся альтернативы x_1, x_4, x_5 со следующими ранжировками, полученными из исходных после удаления x_2 и x_3:

$$r_1:\ x_5 \succ x_4 \succ x_1;$$
$$r_2:\ x_1 \succ x_5 \succ x_4;$$
$$r_3:\ x_4 \succ x_1 \succ x_5;$$
$$r_4:\ x_1 \succ x_5 \succ x_4.$$

По ним находим новые суммарные ранги $S(x_1) = 7$, $S(x_4) = 9$, $S(x_5) = 8$. Средний ранг при $m = 3$ и $n = 4$ равен 8. Удаляем альтернативу x_4, у которой суммарный ранг 9 больше среднего ранга 8. Останутся x_1 и x_5 со следующими ранжировками:

$$r_1:\ x_5 \succ x_1;$$
$$r_2:\ x_1 \succ x_5;$$
$$r_3:\ x_1 \succ x_5;$$
$$r_4:\ x_1 \succ x_5.$$

По ним находим новые суммарные ранги:$S(x_1) = 2+1+1+1 = 5$, $S(x_5) = 1+2+2+2 = 7$. Средний ранг при $m = 2$ и $n = 4$ равен 6. Удаляем альтернативу x_5, у которой суммарный ранг 7 больше среднего ранга 6. Останется только альтернатива x_1, которая и принимается в качестве лучшей. Коллективное решение таково: $\{x_1\} \succ \{x_1, x_2, x_3, x_4\}$ ■

2.4. Правило порогового агрегирования. Рассмотрим более сложный метод, в определённом смысле противоположный методам, основанным на суммарных рангах, когда «плохие» оценки альтернативы у одних участников компенсируются «хорошими» оценками у других. Суть метода такова. Как и в разделе 2.2 при описании паретовского правила, каждой альтернативе x сопоставляется вектор $v(x) = (v_1(x), \ldots, v_n(x))$, где $v_i(x)$ – ранг, который альтернатива x получает у i-го участника ($i = 1, \ldots, n$). Определим бинарное отношение R на множестве альтернатив Ω следующим образом: xRy тогда и только тогда, когда число низших рангов m в векторе $v(x)$ меньше числа низших рангов m в векторе $v(y)$; если же они равны, то в векторах $v(x)$ и $v(y)$ подсчитывается число рангов m–1, и т.д. Если же все эти числа в векторах $v(x)$ и $v(y)$ совпадают (это может быть только в случае, когда вектор x может быть получен из вектора y перестановкой компонент), то x и y объявляются несравнимыми по отношению R.

Пример 10. Ранжировки 4-ёх участников на множестве $\Omega = \{x_1, x_2, x_3, x_4, x_5\}$ таковы:

$$r_1:\ x_1 \succ x_4 \succ x_3 \succ x_5 \succ x_2;$$
$$r_2:\ x_1 \succ x_5 \succ x_3 \succ x_4 \succ x_2;$$
$$r_3:\ x_4 \succ x_3 \succ x_2 \succ x_5 \succ x_1;$$
$$r_4:\ x_1 \succ x_5 \succ x_3 \succ x_4 \succ x_2.$$

Сравним альтернативы из Ω по отношению R. Имеем:

$$v(x_1) = (1,1,5,1),$$
$$v(x_2) = (5,5,3,5),$$
$$v(x_3) = (3,3,2,3),$$
$$v(x_4) = (2,4,1,4),$$
$$v(x_5) = (4,2,4,2).$$

Рассмотрим все десять пар альтернатив из Ω. По определению отношения R:

x_1Rx_2 (один низший ранг 5 против трёх), x_3Rx_1 (ноль низших рангов 5 против одного), x_4Rx_1 (ноль низших рангов 5 против одного), x_5Rx_1 (ноль низших рангов 5 против одного), x_3Rx_2 (ноль низших рангов 5 против трёх), x_4Rx_2 (ноль низших рангов 5 против трёх), x_5Rx_2 (ноль низших рангов 5 против трёх), x_3Rx_4 (ноль низших рангов 4 против двух), x_3Rx_5 (ноль низших рангов 4 против двух), x_4Rx_5 (по два низших ранга 4 у каждой альтернативы; по нулю следующих рангов 3 у каждой альтернативы; один следующий ранг 2 против двух). Окончательно имеем:

x_1Rx_2, x_3Rx_1, x_4Rx_1, x_5Rx_1, x_3Rx_2, x_4Rx_2, x_5Rx_2, x_3Rx_4, x_3Rx_5, x_4Rx_5 ■ (5)

Утверждение 1. Отношение R является ациклическим и транзитивным, т.е. оно является частичным порядком (см. раздел 14-2.3) ■

Положим Ω_1 равным множеству Ω^R недоминируемых по отношению R альтернатив, $\Omega_2 = (\Omega\backslash\Omega_1)^R$, и т.д. Присвоим всем элементам множества Ω_1 ранг 1, множества Ω_2 – ранг 2, и т.д. Полученная ранжировка является результатом применения правила порогового агрегирования.

Пример 11. По данным примера 10 построим итоговую ранжировку. Имеем (см. (5)):

$$x_1Rx_2,\ x_3Rx_1,\ x_4Rx_1,\ x_5Rx_1,\ x_3Rx_2,\ x_4Rx_2,\ x_5Rx_2,\ x_3Rx_4,\ x_3Rx_5,\ x_4Rx_5.$$

Единственной недоминируемой альтернативой является x_3. Единственной альтернативой, доминируемой только x_3, является x_4. Далее идут x_5 (доминируемая только x_3 иx_4), x_1 и x_2. Общая ранжировка имеет вид:

$x_3 \succ x_4 \succ x_5 \succ x_1 \succ x_2$. (6)

Сравним найденную ранжировку с ранжировкой, построенной по тем же самым данным правилом Бордá. Имеем

$$v(x_1) = (1,1,5,1),$$
$$v(x_2) = (5,5,3,5),$$
$$v(x_3) = (3,3,2,3),$$
$$v(x_4) = (2,4,1,4),$$
$$v(x_5) = (4,2,4,2),$$

откуда$S(x_1) = 8$, $S(x_2) = 18$, $S(x_3) = 11$, $S(x_4) = 11$, $S(x_5) = 12$. Сама (нестрогая) ранжировка такова

$$\{x_1\} \succ \{x_3, x_4\} \succ \{x_5\} \succ \{x_2\},$$

что существенно отличается от строгой ранжировки (6), полученной правилом порогового агрегирования ■

3. Правила на основе численности коалиций $V(a, b)$

Как и ранее, предполагается, что члены группы, состоящей из n участников ($n> 2$), выражают своё мнение относительно альтернатив из множества $\Omega = \{x_1, \dots, x_m\}$ ($m \geq 2$), в виде строгих ранжировок r_i этого множества Ω ($i = 1, \dots, n$). В ранжировке r_i лучшая (с точки зрения i-го участника) альтернатива получает ранг 1, и т.д., вплоть до худшей альтернативы, которая получает ранг m. Напомним понятия из раздела 1. Пусть a и b – любые две альтернативы. Через $V(a, b)$ обозначена коалиция участников, считающих, что альтернатива aпредпочтительней, чем альтернатива b, т.е. a имеет мéньший ранг, чем b, в ранжировках участников из $V(a, b)$. В настоящем разделе излагаются методы, в которых используются только численности коалиций $|V(a, b)|$ для всех пар альтернатив.

3.1. Турнирное правило. Из многих методов, которые при построении коллективного решения используют не сами исходные ранжировки, а подсчитанные по ним числа $|V(a, b)|$ для всех пар альтернатив, в разделе 3.1 рассмотрены правила, использующие турнирную матрицу. Два варианта такой матрицы − S^+ и S^+ − определяются следующим образом.

$S^+ = (a_{ij})$, где$a_{ij} = |V(x_i, x_j)|$ $(i, j = 1, \dots, m;\ i \neq j)$, $a_{ii} = \infty$ $(i = 1, \dots, m)$; (7a)

$S^+ = (a_{ij})$, где$a_{ij} = |V(x_i, x_j)|$ $(i, j = 1, \dots, m;\ i \neq j)$, $a_{ii} = -\infty$ $(i = 1, \dots, m)$. (7b)

Таким образом, элемент a_{ij} в обеих матрицах равен числу участников, в ранжировках которых ранг альтернативы i меньше, чем ранг альтернативы j, т.е. по мнению которых альтернатива i предпочтительней, чем альтернатива j.

1. Коллективное решение определяется максимизацией по i выражения $\min_{j=1,\dots,m} a_{ij}$, зависящего только от$i$, в котором числа a_{ij} берутся из матрицы S^+.

2. Коллективное решение определяется минимизацией поj выражения $\max_{i=1,\dots,m} a_{ij}$, зависящего только от$i$, в котором числа a_{ij} берутся из матрицы S^+.

Рассмотрим эти решения по отдельности.

1. **Максиминная процедура.** При фиксированном $i$$a_{ij}$ равно числу участников, которые считают, что альтернатива i предпочтительней альтернатива j. Минимизация по j приводит к та-кой альтернативе j, которую минимальное (по сравнению с другими альтернативами) число участников считает хуже, чем i. Другими словами, для сравнения с альтернативой i это – «худший» случай.

Естественно считать альтернативу лучшей, если для неё худший случай всё же лучше, чем худший случай для остальных. Заметим, что ровно та же аргументация используется в теории матричных игр при определении оптимальной стратегии строчного игрока A (см. начало раздела 11-2).

Пример 12a. Рассмотрим ранжировки для 4-ёх альтернатив из примера 8:

$$r_1:\ x_1 \succ x_3 \succ x_4 \succ x_2;$$
$$r_2:\ x_2 \succ x_4 \succ x_1 \succ x_3;$$
$$r_3:\ x_3 \succ x_1 \succ x_2 \succ x_4.$$

В данном случае имеется 4 альтернативы. Они образуют 6 неупорядоченных пар $\{x_i, x_j\}$, для каждой из которых нетрудно определить числа a_{ij} и a_{ji}. Начнём с пары $\{x_1, x_2\}$. Два из трёх участников считают, что альтернатива x_1 лучше, чем альтернатива x_2. Поэтому $a_{12} = 2$ и, следовательно, $a_{21} = 1$. Аналогично, находим $a_{13} = 2$, $a_{31} = 1$, $a_{14} = 2$, $a_{41} = 1$, $a_{23} = 1$, $a_{32} = 2$, $a_{24} = 2$, $a_{42} = 1$, $a_{34} = 2$, $a_{43} = 1$.

Составим матрицу S^+. Слева запишем саму матрицу, а справа – минимизируемое выражение.

	x_1 x_2 x_3 x_4	$\min\limits_{j=1,\ldots,m} a_{ij}$
x_1 x_2 x_3 x_4	$\begin{pmatrix} \infty & 2 & 2 & 2 \\ 1 & \infty & 1 & 2 \\ 1 & 2 & \infty & 2 \\ 1 & 1 & 1 & \infty \end{pmatrix}$	2 1 1 1

Максимум 2 достигается при $i = 1$. Следующее по величине значение 1 получается при $i = 2, 3, 4$. Значит, за коллективное решение принимается нестрогая ранжировка $\{1\} \succ \{2, 3, 4\}$. Заметим, что правило Бордá в данном случае даёт строгую ранжировку $1 \succ 3 \succ 2 \succ 4$. Можно сказать, что эти результаты, хотя и не совпадают, не слишком противоречат друг другу ■

2. Минимаксная процедура. При фиксированном $j a_{ij}$ равно числу участников, которые считают, что альтернатива i предпочтительней альтернатива j. Максимизация по i приводит к такой альтернативе i, которую максимальное (по сравнению с другими альтернативами) число участников считает лучше, чем j. Другими словами, для сравнения с фиксированной альтернативой j это – «худший» случай для j. Естественно считать альтернативу лучшей, если для неё худший случай всё же лучше, чем худший случай для остальных. Ровно та же аргументация используется в теории матричных игр при определении оптимальной стратегии столбцового игрока B (см. начало раздела 11-2).

Пример 12b. Рассмотрим ту же ранжировку, что и в примере 12a для 4-ёх альтернатив:

$$r_1:\ x_1 \succ x_3 \succ x_4 \succ x_2;$$
$$r_2:\ x_2 \succ x_4 \succ x_1 \succ x_3;$$
$$r_3:\ x_3 \succ x_1 \succ x_2 \succ x_4.$$

Числа a_{ij}, конечно, остаются теми же самыми. Матрица S^- отличается от S^+ только минусами перед ∞. Однако её окаймление несколько отличается. В соответствии с процедурой ищутся не минимумы по строкам, а максимумы по столбцам. Это отражено в следующей таблице.

	x_1 x_2 x_3 x_4
x_1 x_2 x_3 x_4	$\begin{pmatrix} -\infty & 2 & 2 & 2 \\ 1 & -\infty & 1 & 2 \\ 1 & 2 & -\infty & 2 \\ 1 & 1 & 1 & -\infty \end{pmatrix}$
$\max\limits_{i=1,\ldots,m} a_{ij}$	1 2 2 2

Минимум достигается при $j= 1$. Следующее по величине значение 2 получается при $j = 2, 3, 4$. Значит, за коллективное решение принимается нестрогая ранжировка $\{1\} \succ \{2, 3, 4\}$. Заметим, что минимаксная процедура приводит к тому же коллективному решению, что и максиминная (см. пример 12a) ■

Совпадение ранжировок, даваемых обеими процедурами, не является случайным.

Утверждение 2. Ранжировки, полученные максиминной и минимаксной процедурой, совпадают ■

3.2. Правила на основе мажоритарного отношения. Пусть, как и ранее, n участников задают строгие ранжировки на множестве альтернатив $\Omega = \{x_1, \ldots, x_m\}$. Определим бинарное отношение M на множестве Ω следующим образом. Пусть (a, b) – произвольная упорядоченная пара альтернатив. Будем считать, что $aMb \rightleftarrows |V(a, b)| > n/2$, т.е. альтернатива a предпочтительнее альтернативыb с точки зрения более чем половины участников. Построенное отношение M называется ***мажоритарным***.

Пример 13а.Пусть 4 участника на множестве из 5-и альтернатив дали следующие ранжировки:

$$r_1:\ x_5 \succ x_1 \succ x_4 \succ x_3 \succ x_2;$$
$$r_2:\ x_1 \succ x_5 \succ x_3 \succ x_4 \succ x_2;$$
$$r_3:\ x_4 \succ x_1 \succ x_2 \succ x_5 \succ x_3;$$
$$r_4:\ x_5 \succ x_1 \succ x_3 \succ x_4 \succ x_2.$$

Построим по этим ранжировкам мажоритарное отношение M. Рассмотрим все 10 неупорядоченных пар альтернатив. Начнём с x_1 и x_2. По мнению всех 4-ёх участников, ранг x_1 меньше ранга x_2. Поэтому в соответствии с определением мажоритарного отношения получаем x_1Mx_2. Также по мнению всех участников, ранг x_1 меньше ранга x_3, поэтому x_1Mx_3. Далее, по мнению 3-ёх из 4-ёх участников, ранг x_1 меньше ранга x_4, поэтому x_1Mx_4. Наконец, по мнению 2-ух из 4-ёх участников, ранг x_1 меньше ранга x_5, а по мнению двух других участников, наоборот, ранг x_5 меньше ранга x_1. Таким образом, здесь ровно половина участников поддерживает каждую точку зрения и, в соответствии с определением, x_1с x_5, как и x_5с x_1, не находятся в отношении M, что записывается как $x_1\overline{M}x_5$ и $x_5\overline{M}x_1$. Аналогично устанавливаем, что x_5Mx_2, x_5Mx_3, x_5Mx_4, а также x_3Mx_2 и x_4Mx_2. Никаких других пар в отношение M не входит.

Таким образом, мажоритарное отношение M в данном случае задаётся перечнем:

$$x_1Mx_2,\ x_1Mx_3,\ x_1Mx_4,\ x_5Mx_2,\ x_5Mx_3,\ x_5Mx_4,\ x_3Mx_2,\ x_4Mx_2\ \blacksquare$$

Мажоритарное отношение M является антирефлексивным (ни одна альтернатива не превосходит саму себя) и асимметричным ($aMb \rightarrow b\overline{M}a$). Вообще говоря, мажоритарное отношение M не является ни ациклическим, ни транзитивным. По этой причине для построения коллективного решения в виде ранжировки невозможно использовать стандартную процедуру последовательного выделения недоминируемых множеств (как в разделах 2.2 и 2.4). Для выполнения такой процедуры требуется, чтобы отношение было ациклическим.

Именно поэтому было предложено несколько различных правил построения нестрогой ранжировки по мажоритарному отношению. Некоторые из них приводятся ниже. Обычно такие правила хорошо иллюстрируются соответствующими действиями над графами отношений (см. раздел 14-2.2). Граф $G(M)$ мажоритарного отношения называется ***мажоритарным графом***.

Пример 13b. Построим мажоритарный граф по мажоритарному отношению, приведённому в примере **13а.** В соответствии с описанием в разделе 14-2.2 построения графа по бинарному отношению, в данном случае граф содержит 5 вершин и 8 дуг, соответствующих записям в перечне пар, входящих в отношение. Мажоритарный граф показан на рис.1.■

Альтернатива, недоминируемая по мажоритарному отношению M, называется ***победителем Кондорсе***. В случае нечётного числа участников может быть не более одного победителя Кондорсе. При любом числе участников множество победителей Кондорсе может быть пустым. В примере 13а имеется два победителя Кондорсе – альтернативы x_1 и x_5.

Легко понять, что победителям Кондорсе в мажоритарном отношении M соответствуют вершины мажоритарного графа $G(M)$, в которые не входит ни одна дуга этого графа.

3.2.1. Правила выбора минимального доминирующего и недоминируемого множества. Множество Q называется ***доминирующим***, если каждая альтернатива в Q доминирует каждую альтернативу вне Q в смысле мажоритарного отношения M, т.е. $(\forall x \in Q)\ (\forall y \in \Omega \setminus Q)\ xMy$. Доминирующее множество Q называется ***минимальным***, если никакое его собственное подмножество не является доминирующим множеством. Если для мажоритарного отношения M минимальное доминирующее множество единственно, то коллективным решением объявляется именно оно. В противном случае коллективный выбор представляет собой объединение этих множеств. Нестрогая ранжировка, как и ранее в подобных ситуациях, состоит из двух подмножеств Ω: выбранного множества Ω_1 и его дополнения Ω_2.

Множество Q называется ***недоминируемым***, если никакая альтернатива вне Q не доминирует какую бы то ни было альтернативу из Q по мажоритарному отношению M , т.е. $(\forall x \in \Omega \backslash Q)$ $(\forall y \in Q)$ $x\overline{M}y$. Недоминируемое *множество* Q называется ***минимальным***, если никакое его собственное подмножество не является недоминируемым множеством.

Если для мажоритарного отношения M минимальное доминирующее множество единственно, то коллективным решением объявляется именно оно. В противном случае коллективный выбор представляет собой объединение этих множеств. Нестрогая ранжировка состоит из двух подмножеств Ω: выбранного множества Ω_1 и его дополнения Ω_2.

Пример 14. Рассмотрим мажоритарное отношение и соответствующий ему мажоритарный граф из примера 13. Рис.1 ясно демонстрирует, что доминирующие множества в данном случае отсутствуют. В то же время оба множества – состоящее из одной альтернативы $\{x_1\}$ и состоящее из одной альтернативы $\{x_5\}$ – являются минимальными недоминируемыми множествами. В таком случае в соответствии с правилом множеством Ω_1 лучших элементов объявляется множество $\{x_1, x_5\}$ ■

3.2.2. Правило выбора слабоустойчивого множества. ***Слабоустойчивым*** называется не-пустое множество $B \subseteq \Omega$, удовлетворяющее следующему условию: если для $x \in B$ существует $y \in \Omega \backslash B$, такой, что yMx, то существует $z \in B$, такой, что zMy. Определение проиллюстрировано на рис.2. Множество B называется ***минимальным слабоустойчивым множеством***, если оно не имеет собственных слабоустойчивых подмножеств.

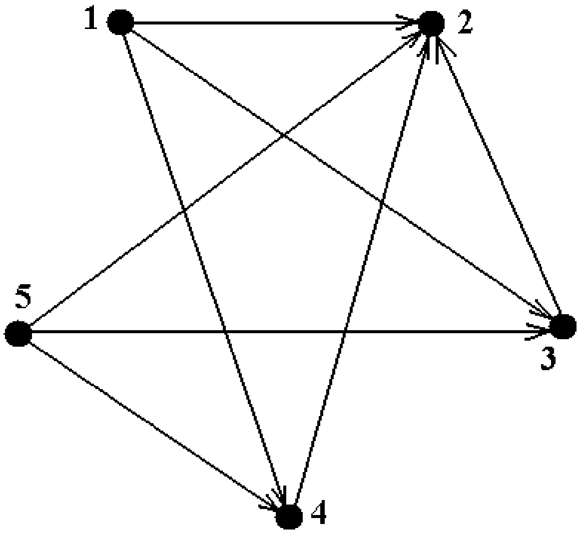


Рис. 1. Пример мажоритарного графа

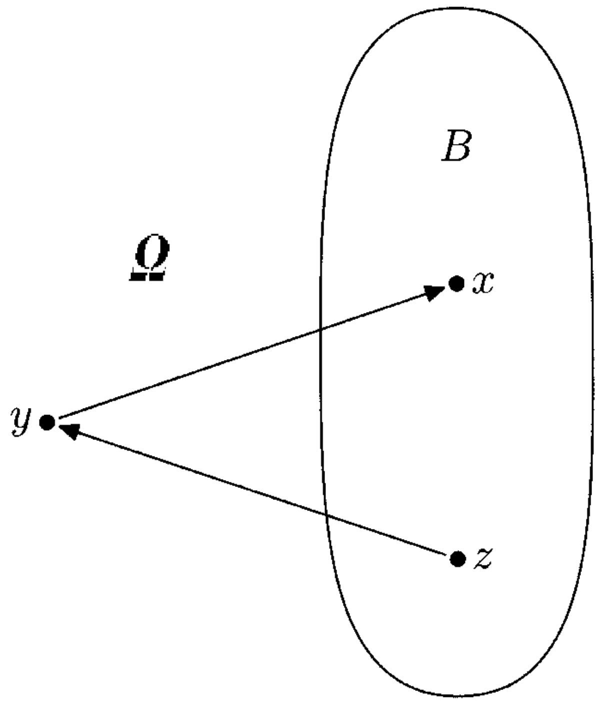


Рис.2. Иллюстрация слабой устойчивости

Правило выбора слабоустойчивого множества на мажоритарном отношении M состоит в выборе минимального слабоустойчивого множества. Однако минимальных слабоустойчивых множеств может быть несколько. Тогда в качестве окончательного выбора берётся их объединение. Заметим, что слабоустойчивые множества существуют для любого бинарного отношения (в частности, для любого мажоритарного отношения), поскольку множество Ω всех альтернатив слабоустойчиво в силу «ложности посылки».

Пример 15. Рассмотрим мажоритарное отношение и соответствующий ему мажоритарный граф из примера 13. Рис.1 ясно демонстрирует, что оба множества альтернатив – $\{x_1\}$ и $\{x_5\}$ – являются минимальными слабоустойчивыми множествами. Дело в том, что альтернативы y, такой, что yMx_1, просто не существует, и то же самое верно и для x_5. При этом других минимальных слабоустойчивых множеств в данном случае нет. Поэтому в соответствии с правилом множеством Ω_1 лучших элементов объявляется множество $\{x_1, x_5\}$ ■

Результат примера 15 не случаен. Имеет место

Утверждение 3. Одноэлементное множество альтернатив является минимальным слабо-устойчивым множеством в мажоритарном отношении тогда и только тогда, когда оно состоит из победителя Кондорсе ■

3.2.3. Первое правило Коупленда. Напомним понятия верхнего и нижнего среза альтернативы x по отношению R, определённых в разделе 14-3 формулами (14-1):

$$Rx = \{y\in\Omega|yRx\}, xR = \{y\in\Omega|xRy\}.$$

Определим на множестве альтернатив $\Omega = \{x_1, x_2, \ldots, x_m\}$ числовую функцию $K(x)$ формулой $K(x) = |xR| - |Rx|$.(8)

Число альтернатив, которые, по мнению большинства участников, доминируемы альтернативой x, равно $|xR|$. Число альтернатив, которые, по мнению большинства участников, доминируют альтернативу x, равно $|Rx|$.Поскольку для предпочтительности альтернативы x желательно, чтобы первое из этих чисел было как можно больше, а второе – как можно меньше, то функция $K(x)$ является разумной мерой оптимальности альтернатив.

Первое правило Коупленда определяет (как и все остальные рассматриваемые в настоящем разделе правила) нестрогую ранжировку по мажоритарному отношению M. Это делается следующим образом. По мажоритарному отношению Mпо формуле (8) определяется функция $K(x)$. Далее, всем альтернативам, максимизирующим функцию $K(x)$, присваивается ранг 1, альтернативам, на которых $K(x)$ принимает следующее по величине значение – ранг 2, и т.д., вплоть до минимального значения $K(x)$.

Пример 16. Рассмотрим мажоритарное отношение и соответствующий ему мажоритарный граф из примера 13. Функция $K(x)$ по определению равна числу дуг, выходящих из вершины x, минус число дуг, входящих в вершину x. В данном случае непосредственно на рис.1 видно, что $K(1) = 3$ (три дуги выходят и ни одна не входит); $K(5) = 3$; $K(3) = -1$; $K(4) = -1$; $K(2) = -4$. Поэтому получаем такую ранжировку: $\{x_1, x_5\} \succ \{x_3, x_4\} \succ \{x_2\}$, которая и является искомым коллективным решением ■

В настоящей главе рассмотрено несколько правил построения коллективного решения. «Пропуская» через эти правила различные наборы индивидуальных ранжировок, можно заметить, что в самых «лучших» случаях (когда все ранжировки практически совпадают) и в самых «худших» (когда они отличаются только циклическими сдвигами, и никакого превосходства у одних альтернатив над другими нет), все правила дают одно и то же (линейный порядок в лучших случаях и одну группу в худшем). А в более интересных и менее очевидных промежуточных случаях результаты могут отличаться. Можно повторить один из законов О' Мёрфи: есть правила для принятия решений, но нет правила для выбора этих правил. Говоря более серьёзно, выбор метода построения коллективного решения зависит от ситуации, в которой его предполагается применять, и от квалификации людей, оценивающих как альтернативы, так и саму ситуацию.

4. Задания

В настоящей главе все задания состоят в применении к заданному набору ранжировок одного из рассмотренных методов принятия коллективного решения. Поэтому данный раздел начинается со списка таких наборов.

Варианты для заданий главы 16:

01 $r_1: x_5\succ x_1\succ x_4\succ x_3\succ x_2;$ $r_2: x_1\succ x_5\succ x_3\succ x_4\succ x_2;$ $r_3: x_4\succ x_1\succ x_2\succ x_5\succ x_3;$ $r_4: x_5\succ x_4\succ x_1\succ x_3\succ x_2.$	02 $r_1: x_5\succ x_1\succ x_4\succ x_2\succ x_3;$ $r_2: x_1\succ x_5\succ x_3\succ x_4\succ x_2;$ $r_3: x_4\succ x_1\succ x_2\succ x_5\succ x_3;$ $r_4: x_5\succ x_1\succ x_3\succ x_4\succ x_2.$	03 $r_1: x_5\succ x_1\succ x_3\succ x_4\succ x_2;$ $r_2: x_1\succ x_5\succ x_3\succ x_4\succ x_2;$ $r_3: x_4\succ x_1\succ x_5\succ x_2\succ x_3;$ $r_4: x_5\succ x_1\succ x_3\succ x_4\succ x_2.$
04 $r_1: x_5\succ x_1\succ x_3\succ x_2\succ x_4;$ $r_2: x_1\succ x_5\succ x_3\succ x_4\succ x_2;$ $r_3: x_4\succ x_1\succ x_2\succ x_3\succ x_5;$ $r_4: x_5\succ x_1\succ x_3\succ x_4\succ x_2.$	05 $r_1: x_5\succ x_1\succ x_2\succ x_3\succ x_4;$ $r_2: x_1\succ x_5\succ x_2\succ x_4\succ x_3;$ $r_3: x_4\succ x_1\succ x_2\succ x_5\succ x_3;$ $r_4: x_5\succ x_1\succ x_3\succ x_4\succ x_2.$	06 $r_1: x_5\succ x_1\succ x_2\succ x_4\succ x_3;$ $r_2: x_1\succ x_5\succ x_2\succ x_3\succ x_4;$ $r_3: x_4\succ x_1\succ x_2\succ x_5\succ x_3;$ $r_4: x_5\succ x_1\succ x_3\succ x_4\succ x_2.$
07 $r_1: x_5\succ x_3\succ x_1\succ x_2\succ x_4;$ $r_2: x_1\succ x_5\succ x_3\succ x_4\succ x_2;$ $r_3: x_4\succ x_1\succ x_2\succ x_5\succ x_3;$ $r_4: x_5\succ x_1\succ x_3\succ x_4\succ x_2.$	08 $r_1: x_5\succ x_3\succ x_1\succ x_4\succ x_2;$ $r_2: x_1\succ x_5\succ x_3\succ x_4\succ x_2;$ $r_3: x_4\succ x_1\succ x_2\succ x_5\succ x_3;$ $r_4: x_5\succ x_1\succ x_3\succ x_4\succ x_2.$	09 $r_1: x_5\succ x_3\succ x_2\succ x_1\succ x_4;$ $r_2: x_1\succ x_5\succ x_3\succ x_4\succ x_2;$ $r_3: x_4\succ x_1\succ x_5\succ x_2\succ x_3;$ $r_4: x_5\succ x_1\succ x_3\succ x_4\succ x_2.$

10 $r_1: x_5 \succ x_3 \succ x_2 \succ x_4 \succ x_1;$ $r_2: x_1 \succ x_5 \succ x_3 \succ x_4 \succ x_2;$ $r_3: x_4 \succ x_1 \succ x_2 \succ x_3 \succ x_5;$ $r_4: x_5 \succ x_1 \succ x_3 \succ x_4 \succ x_2.$	11 $r_1: x_5 \succ x_3 \succ x_4 \succ x_1 \succ x_2;$ $r_2: x_1 \succ x_5 \succ x_3 \succ x_4 \succ x_2;$ $r_3: x_4 \succ x_1 \succ x_2 \succ x_5 \succ x_3;$ $r_4: x_5 \succ x_1 \succ x_3 \succ x_4 \succ x_2.$	12 $r_1: x_5 \succ x_3 \succ x_4 \succ x_1 \succ x_2;$ $r_2: x_1 \succ x_5 \succ x_3 \succ x_4 \succ x_2;$ $r_3: x_1 \succ x_4 \succ x_2 \succ x_5 \succ x_3;$ $r_4: x_5 \succ x_1 \succ x_3 \succ x_4 \succ x_2.$
13 $r_1: x_5 \succ x_1 \succ x_3 \succ x_4 \succ x_2;$ $r_2: x_1 \succ x_5 \succ x_3 \succ x_4 \succ x_2;$ $r_3: x_4 \succ x_1 \succ x_5 \succ x_2 \succ x_3;$ $r_4: x_5 \succ x_1 \succ x_3 \succ x_4 \succ x_2.$	14 $r_1: x_5 \succ x_1 \succ x_3 \succ x_2 \succ x_4;$ $r_2: x_1 \succ x_5 \succ x_3 \succ x_4 \succ x_2;$ $r_3: x_4 \succ x_1 \succ x_2 \succ x_3 \succ x_5;$ $r_4: x_5 \succ x_1 \succ x_3 \succ x_4 \succ x_2.$	15 $r_1: x_5 \succ x_1 \succ x_2 \succ x_3 \succ x_4;$ $r_2: x_1 \succ x_5 \succ x_2 \succ x_4 \succ x_3;$ $r_3: x_4 \succ x_1 \succ x_2 \succ x_5 \succ x_3;$ $r_4: x_5 \succ x_1 \succ x_3 \succ x_4 \succ x_2.$
16 $r_1: x_5 \succ x_1 \succ x_2 \succ x_4 \succ x_3;$ $r_2: x_1 \succ x_5 \succ x_2 \succ x_3 \succ x_4;$ $r_3: x_4 \succ x_1 \succ x_2 \succ x_5 \succ x_3;$ $r_4: x_5 \succ x_1 \succ x_3 \succ x_4 \succ x_2.$	17 $r_1: x_5 \succ x_3 \succ x_1 \succ x_2 \succ x_4;$ $r_2: x_1 \succ x_5 \succ x_3 \succ x_4 \succ x_2;$ $r_3: x_4 \succ x_1 \succ x_2 \succ x_5 \succ x_3;$ $r_4: x_5 \succ x_1 \succ x_3 \succ x_4 \succ x_2.$	18 $r_1: x_5 \succ x_3 \succ x_1 \succ x_4 \succ x_2;$ $r_2: x_1 \succ x_5 \succ x_3 \succ x_4 \succ x_2;$ $r_3: x_4 \succ x_1 \succ x_2 \succ x_5 \succ x_3;$ $r_4: x_5 \succ x_1 \succ x_3 \succ x_4 \succ x_2.$
19 $r_1: x_5 \succ x_3 \succ x_2 \succ x_1 \succ x_4;$ $r_2: x_1 \succ x_5 \succ x_3 \succ x_4 \succ x_2;$ $r_3: x_4 \succ x_1 \succ x_5 \succ x_2 \succ x_3;$ $r_4: x_5 \succ x_1 \succ x_3 \succ x_4 \succ x_2.$	20 $r_1: x_5 \succ x_3 \succ x_2 \succ x_4 \succ x_1;$ $r_2: x_1 \succ x_5 \succ x_3 \succ x_4 \succ x_2;$ $r_3: x_4 \succ x_1 \succ x_2 \succ x_3 \succ x_5;$ $r_4: x_5 \succ x_1 \succ x_3 \succ x_4 \succ x_2.$	21 $r_1: x_5 \succ x_3 \succ x_4 \succ x_1 \succ x_2;$ $r_2: x_1 \succ x_5 \succ x_3 \succ x_4 \succ x_2;$ $r_3: x_4 \succ x_1 \succ x_2 \succ x_5 \succ x_3;$ $r_4: x_5 \succ x_1 \succ x_3 \succ x_4 \succ x_2.$
22 $r_1: x_5 \succ x_1 \succ x_4 \succ x_3 \succ x_2;$ $r_2: x_1 \succ x_5 \succ x_3 \succ x_4 \succ x_2;$ $r_3: x_4 \succ x_1 \succ x_2 \succ x_5 \succ x_3;$ $r_4: x_5 \succ x_4 \succ x_1 \succ x_3 \succ x_2.$	23 $r_1: x_5 \succ x_1 \succ x_4 \succ x_2 \succ x_3;$ $r_2: x_1 \succ x_5 \succ x_3 \succ x_4 \succ x_2;$ $r_3: x_4 \succ x_1 \succ x_2 \succ x_5 \succ x_3;$ $r_4: x_5 \succ x_1 \succ x_3 \succ x_4 \succ x_2.$	24 $r_1: x_1 \succ x_3 \succ x_5 \succ x_4 \succ x_2;$ $r_2: x_3 \succ x_5 \succ x_1 \succ x_2 \succ x_4;$ $r_3: x_2 \succ x_4 \succ x_1 \succ x_5 \succ x_3.$
25 $r_1: x_1 \succ x_2 \succ x_3 \succ x_5 \succ x_4;$ $r_2: x_3 \succ x_5 \succ x_1 \succ x_2 \succ x_4;$ $r_3: x_2 \succ x_4 \succ x_1 \succ x_5 \succ x_3.$	26 $r_1: x_1 \succ x_2 \succ x_4 \succ x_3 \succ x_5;$ $r_2: x_2 \succ x_5 \succ x_1 \succ x_3 \succ x_4;$ $r_3: x_2 \succ x_1 \succ x_4 \succ x_5 \succ x_3.$	27 $r_1: x_1 \succ x_2 \succ x_4 \succ x_5 \succ x_3;$ $r_2: x_3 \succ x_4 \succ x_1 \succ x_2 \succ x_5;$ $r_3: x_2 \succ x_1 \succ x_4 \succ x_5 \succ x_3.$
28 $r_1: x_1 \succ x_2 \succ x_5 \succ x_3 \succ x_4;$ $r_2: x_3 \succ x_5 \succ x_1 \succ x_2 \succ x_4;$ $r_3: x_2 \succ x_1 \succ x_4 \succ x_5 \succ x_3.$	29 $r_1: x_1 \succ x_2 \succ x_5 \succ x_4 \succ x_3;$ $r_2: x_3 \succ x_5 \succ x_1 \succ x_2 \succ x_4;$ $r_3: x_2 \succ x_1 \succ x_4 \succ x_5 \succ x_3.$	30 $r_1: x_1 \succ x_3 \succ x_2 \succ x_4 \succ x_5;$ $r_2: x_3 \succ x_5 \succ x_1 \succ x_4 \succ x_2;$ $r_3: x_2 \succ x_1 \succ x_4 \succ x_5 \succ x_3.$
31 $r_1: x_1 \succ x_3 \succ x_2 \succ x_5 \succ x_4;$ $r_2: x_3 \succ x_5 \succ x_1 \succ x_2 \succ x_4;$ $r_3: x_2 \succ x_1 \succ x_4 \succ x_5 \succ x_3.$	32 $r_1: x_1 \succ x_3 \succ x_4 \succ x_2 \succ x_5;$ $r_2: x_2 \succ x_5 \succ x_1 \succ x_3 \succ x_4;$ $r_3: x_2 \succ x_1 \succ x_4 \succ x_5 \succ x_3.$	33 $r_1: x_1 \succ x_3 \succ x_4 \succ x_5 \succ x_2;$ $r_2: x_3 \succ x_5 \succ x_1 \succ x_2 \succ x_4;$ $r_3: x_2 \succ x_1 \succ x_4 \succ x_5 \succ x_3.$
34 $r_1: x_1 \succ x_2 \succ x_3 \succ x_4 \succ x_5;$ $r_2: x_3 \succ x_5 \succ x_1 \succ x_2 \succ x_4;$ $r_3: x_2 \succ x_1 \succ x_4 \succ x_5 \succ x_3.$	35 $r_1: x_1 \succ x_3 \succ x_2 \succ x_4;$ $r_2: x_2 \succ x_4 \succ x_1 \succ x_3;$ $r_3: x_3 \succ x_1 \succ x_2 \succ x_4.$	36 $r_1: x_1 \succ x_3 \succ x_2 \succ x_4;$ $r_2: x_2 \succ x_1 \succ x_4 \succ x_3;$ $r_3: x_2 \succ x_1 \succ x_3 \succ x_4.$
37 $r_1: x_1 \succ x_2 \succ x_3 \succ x_4;$ $r_2: x_2 \succ x_4 \succ x_1 \succ x_3;$ $r_3: x_1 \succ x_3 \succ x_2 \succ x_4.$	38 $r_1: x_1 \succ x_3 \succ x_4 \succ x_2;$ $r_2: x_2 \succ x_4 \succ x_1 \succ x_3;$ $r_3: x_3 \succ x_1 \succ x_2 \succ x_4.$	39 $r_1: x_2 \succ x_1 \succ x_3 \succ x_4;$ $r_2: x_2 \succ x_4 \succ x_1 \succ x_3;$ $r_3: x_1 \succ x_3 \succ x_2 \succ x_4.$
40 $r_1: x_2 \succ x_3 \succ x_5 \succ x_4;$ $r_2: x_2 \succ x_4 \succ x_1 \succ x_3;$ $r_3: x_3 \succ x_1 \succ x_2 \succ x_4.$	41 $r_1: x_1 \succ x_2 \succ x_4 \succ x_3;$ $r_2: x_2 \succ x_4 \succ x_3 \succ x_1;$ $r_3: x_1 \succ x_3 \succ x_2 \succ x_4.$	42 $r_1: x_1 \succ x_4 \succ x_2 \succ x_3;$ $r_2: x_2 \succ x_1 \succ x_4 \succ x_3;$ $r_3: x_3 \succ x_1 \succ x_2 \succ x_4.$
43 $r_1: x_2 \succ x_1 \succ x_4 \succ x_3;$ $r_2: x_1 \succ x_4 \succ x_2 \succ x_3;$ $r_3: x_1 \succ x_3 \succ x_2 \succ x_4.$	44 $r_1: x_2 \succ x_4 \succ x_3 \succ x_5;$ $r_2: x_2 \succ x_4 \succ x_1 \succ x_3;$ $r_3: x_1 \succ x_3 \succ x_2 \succ x_4.$	45 $r_1: x_1 \succ x_3 \succ x_2 \succ x_4;$ $r_2: x_2 \succ x_4 \succ x_1 \succ x_3;$ $r_3: x_3 \succ x_1 \succ x_2 \succ x_4.$

Задание 1. Найти коллективное решение, используя правило Бордá ■

Задание 2. Найти коллективное решение для данных из задания 1, используя паретовское правило ■

Задание 3. Найти коллективное решение, используя правило Хара ■

Задание 4. Найти коллективное решение, используя правило Нансона■

Задание 5. Найти коллективное решение, используя правило порогового агрегирования■

Задание 6. Найти коллективные решения минимаксной процедурой ■

Задание 7. Построить мажоритарные отношения и соответствующие им мажоритарные графы. Найти победителей Кондорсе. Если таковых нет, специально отметить это ■

Задание 8. Найти коллективное решение по правилу выбора минимального доминирующего множества ■

Задание 9. Найти коллективное решение по правилу выбора минимального недоминируемого множества ■

Задание 10. Найти коллективное решение по правилу выбора слабоустойчивого множества■

Задание 11. Найти коллективное решение по первому правилу Коупленда ■

Задание 12. Составить таблицу с ответами заданий 1 – 11. В последней графе указать число различных ответов ■

5. Предметный указатель

Глава 17. Функции выбора

1. Понятие функции выбора
2. Функции выбора и механизмы выбора
3. Метатеория функций выбора
4. Логические формы функций выбора
5. Задания
6. Предметный указатель

Понятие бинарного отношения, рассмотренное в главах 14 и 15, позволяет разработать процедуры выделения (выбора) лучших альтернатив, основываясь на результатах их попарных сравнений. Однако во многих практически важных случаях сделать обоснованный выбор, используя только результаты попарных сравнений альтернатив, оказывается затруднительным. Дело в том, что зачастую решающую роль для выбора лучших альтернатив играют не только попарные, но и более сложные «групповые» сравнения (подобно тому, как в регрессионном анализе необходимо учитывать множественные регрессии, а в теории игр – возможные действия коалиций участников). Для анализа и реализации возникающих при таком взгляде более широких возможностей и неожиданных эффектов уже достаточно давно (более 40 лет назад) были введены функции выбора, кратко рассмотренные в настоящей главе.

1. Понятие функции выбора

В практической ситуации выбора при некотором множестве альтернатив Ω лицо, принимающее решение (ЛПР), выбирая альтернативу, руководствуется своим личным представлением о лучших альтернативах. У разных ЛПР в одной и той же ситуации (при одном Ω) представление о лучших альтернативах может различаться, а, следовательно, они могут выбирать разные альтернативы. При этом каждый из них может привести вполне рациональное обоснование сделанного выбора. Даже при выборе одних и тех альтернатив разными лицами обоснования могут различаться. Таким образом, по известному выбору в конкретной ситуации вряд ли можно сказать что-либо определенное о тех причинах, которые побудили сделать именно данный выбор, а не другой, т. е. восстановить логику выбора данного ЛПР.

Рассмотрим теперь несколько взаимосвязанных ситуаций выбора, в которых разные множества альтернатив X являются подмножествами одного и того же «универсального» (в данном контексте) множества альтернатив Ω. Обозначим через $C(X)$ множество альтернатив, выделенных ЛПР, из множества X, и установим связи между множествами $C(X)$ при разных множествахX. Заметим, что выбор при этом осуществляется одним и тем же ЛПР. Далее везде термин ***выбор из X*** будет использоваться для обозначения множества $C(X)$.

Пример 1. Пусть Ω – множество всех групп ВУЗа, X – произвольное подмножество Ω (множество групп 3-го курса, множество групп экономического факультета и т.д.). Пусть $C(X)$ – лучшая группа из множества групп X. Отвлекаясь от того, кто и по каким признакам (критериям) выбирает лучшую группу, естественно считать, что лучшая группа вуза будет лучшей группой своего курса, своего факультета и т.д. Формально это записывается в общем виде следующим образом:

если $X'\subseteq X$ и $x\in C(X)\cap X'$, то $x\in C(X')$. (1)

Формула (1) означает, что всякий элемент x, выбранный из множества X(т.е. принадлежащий $C(X)$), будет выбран также из любого содержащего элемент x подмножества X' множества X (т.е. элемент x должен принадлежать $C(X')$). Конкурс на лучшую группу, в котором лучшая группа ВУЗа оказывается не лучшей группой своего факультета, вряд ли можно считать объективным, независимо от того, по каким критериям подводятся его итоги ■

Итак, не всякий выбор в конкретной ситуации может быть признан логически обоснованным при известных выборах в других ситуациях, связанных с данной, так как множества $C(X)$ оказываются зависимыми при разных X, например, в смысле (1). Для формализации взаимной зависимости выборов $C(X)$ при взаимосвязанных ситуациях пользуются понятием функции выбора. ***Функцией выбора C*** называется соответствие, сопоставляющее каждому $X\subseteq\Omega$ некоторое его подмножество $C(X)\subseteq X$. В терминологии раздела 3-5 можно сказать, что это функция типа $2^{\Omega}\rightarrow 2^{\Omega}$(напомним, что через 2^A обозначен булеан конечного множества A, т.е. множество всех подмножеств множества A). От произвольной функции данного типа функция выбора отличается условием $C(X)\subseteq X$ для любого множества $X\subseteq\Omega$. При этом множество $C(X)$ интерпретируется как подмножество лучших (в том или ином смысле) элементов множества X. Можно сказать, что функция выбора описывает «выбор в целом», а не только отдельный акт выбора.

Отметим, что в общем определении функции выбора никаких априорных ограничений на множество $C(X)$ не накладывается. В частности, не исключается возможность пустого выбора, т.е. $C(X) = \varnothing$. Обычно пустой выбор называют ***«отказом от выбора»*** или ***альтернативой «статус-кво»***. Примером, когда возникает отказ от выбора, может быть ситуация, в которой студент не берет ни одной из предлагаемых ему библиотекой книг для подготовки к экзаменам, или ситуация, в которой покупатель уходит из магазина, не купив в нем ничего.

2. Функции выбора и механизмы выбора

В общем случае функция выбора на множестве альтернатив Ωявляется абстрактным объектом. Для её явного задания надо определить значения $C(X)$ для всех подмножеств X множества Ω. Как и для булевых функций от n переменных, мощность множества отправления (см. раздел 3-5) функции выбора (т.е. число всех подмножеств Ω) равна 2^n, где n – число элементов в самом множестве Ω. Поэтому явное задание произвольной функции выбора, как и явное задание произвольной булевой функции, мало обозримо и практически бесполезно.

Для булевых функций принципиальным был переход от самих функций к реализующим их формулам. Формула не только даёт значительное более обозримое и ясное представление реализуемой ей функции. Она даёт возможность быстрого подсчёта значения функции на любом конкретном наборе значений переменных, т.е. по сути дела определяет алгоритм вычисления функции. Выигрыш здесь определяется тем, что в подавляющем большинстве случаев интерес представляют значения функции лишь на небольшом числе наборов значений переменных.

Естественно, что при исследовании функций выбора также целесообразно перейти от общего формального понятия к конкретным функциям, определяемым конкретными алгоритмами определения множеств предпочтительных альтернатив $C(X)$, а не просто их абстрактным заданием (например, в виде списков входящих в них элементов). Такие алгоритмы получили название «***механизмов выбора***», подчёркивающим их конструктивную направленность.

Далее в разделе рассматривается нескольких наиболее известных механизмов выбора.

2.1. ***Парнодоминантный механизм выбора*** Сопоставим произвольному бинарному отношению R на множестве Ωдве различные функции выбора на Ω. Положим для любого $X\subseteq\Omega$

$$C^R(X) = \{x\in X|\ (\forall y\in X)y\bar{R}x\}, \qquad (2a)$$

$$C_R(X) = \{x\in X|\ (\forall y\in X)xRy\}. \qquad (2b)$$

Функции выбора $C^R(X)$ и $C_R(X)$, называются функциями ***блокировки***и ***предпочтения***, порождёнными бинарным отношением R.

Сравнивая формулы (2a), (2b) с определением мажорант и максимумов из раздела 14-3, видим, что $C^R(X)$ состоит из всех мажорант X^R, а $C_R(X)$ состоит из всех максимумов X_R по отношению R на множестве X. Смысл названий «***блокировка»***и «***предпочтение»*** состоит в следующем: в случае (2a), как только для какого-нибудь y выполняется yRx, то xне выбирается, т.е. более предпочтительный элемент y «блокирует» x; в случае (2b) элемент x выбирается из X, как только он предпочтительнее всех элементов из того же предъявления X.

Несмотря на значительное различие в определениях блокировки и предпочтения, они сводятся друг к другу. В силу утверждения 14-4 блокировка по отношению R совпадает с предпочтением по двойственному к нему отношению R^d и наоборот. Поэтому из двух функций выбора, порожденных бинарным отношением R, достаточно рассматривать только одну. Всюду в дальнейшем, если не оговорено противное, будем сопоставлять бинарному отношению R функцию блокировки C^R, определяемую формулой (2a), и называть ее ***функцией выбора, порожденной отношением R***. Такие функции называются также ***нормальными***.

Пример 2. Произвольная функция выбора C не обязательно совпадает с функцией вида C^R для некоторого бинарного отношения R. Рассмотрим следующую функцию выбора на $\Omega = \{x, y\}$:

$C(x) = x$, $C(y) = \varnothing$, $C(x, y) = \{x, y\}$. (3)

Выясним, существует ли бинарное отношение R на Ω такое, что $C = C^R$. Допустим, что такое R существует. Из формулы (2а) и из того, что $C^R(x) = x$, сразу следует, что верно $x\bar{R}x$, т.е. неверно xRx. Аналогично, из $C^R(y) = \varnothing$ следует, что верно yRy. Но тогда неверно $y\bar{R}y$, и, значит, $y \notin C^R(\Omega)$, что противоречит (3); значит, предположение о существовании бинарного отношения R на Ω, такого, что $C = C^R$, неверно. Другими словами, функция выбора (5) не является нормальной■

Пример 3. Рассмотрим функцию выбора на $\Omega = \{x, y, z\}$, у которой

$C(x, y) = x$, $C(x, y, z\} = y$ (4)

(остальные значения могут быть произвольными).

Выясним, существует ли бинарное отношение R на Ω такое, что $C = C^R$. Допустим, что такое R существует. Из того, что $y \notin C^R(x, y)$, следует, что выполняется по крайней мере одно из двух условий: yRy или xRy. В обоих случаях элемент y не может принадлежать множеству $\{x, y, z\}^R$, так как его выбор блокируется либо элементом x (в случае xRy), либо самим элементомy (в случае yRy). Но это противоречит условию $C(x, y, z\} = y$, что и доказывает отсутствие требуемого бинарного отношения R.

Выбор (4) означает следующее. Из пары элементов $\{x, y\}$ предпочтительным является элемент x; однако «в присутствии» элемента z из той же пары $\{x, y\}$ предпочтительным теперь является элемент y. В таких случаях говорят, что выбор является ***контекстно-зависимым***■

Пример 4. Приведем еще один пример, который показывает, что бинарное отношение и «выбор в целом» на двухэлементном множестве являются существенно различными понятиями, хотя с первого взгляда могут показаться тождественными. Рассмотрим всевозможные функции выбора C на $\Omega = \{x, y\}$ и определим для каждой из них отношения R_1 и R_2 такие, что $C = C^{R_1}$ и $C = C_{R_2}$, если это возможно (Примеры 2 и 3 показывают, что это возможно не всегда). Отношения R_1 и R_2 будем задавать графами, которые представлены в двух последних столбцах таблицы 1. Соответствующие этим графам функции выбора $C = C^{R_1} = C_{R_2}$ представлены во 2-м – 4-м столбцах. Например, строки 5 и 6 таблицы 1 показывают, что уже для двухэлементных множеств не всякая функция выбора C может быть порождена бинарным отношением ■

Итак, каждому бинарному отношению R на Ω соответствует некоторая порожденная им парно-доминантная функция выбора C^R; разным R могут соответствовать одинаковые C^R; не все функции выбора парно-доминантны.

2.2. Турнирный механизм выбора. Пусть задана матрица $T = (t_{ij})$ $(i, j = 1, \ldots, n)$ и для всех $i t_{ii} = 0$, а при всех $i \neq j t_{ij} \geq 0$ и $t_{ij} + t_{ji} = m$, где $m > 0$ – заданное целое положительное число. Матрица подобного рода может трактоваться как таблица ***m-кругового турнира***, в котором участвует nигроков, а игры происходят без ничьих. При такой трактовке t_{ij}означает число кругов турнира, в которых i-ый игрок выиграл у j-го игрока. В случае однокругового турнира ($m = 1$) матрица T заполнена нулями и единицами, имея на главной диагонали только нули.

В задаче выбора, использующей турнирную матрицу, вариантами служат игроки, обозначаемые далее через $x_1, \ldots, x_n$; эти же символы используются в турнирной матрице для наименования строк и столбцов. Нули на главной диагонали матрицы объясняются тем, что игрок x_i «сам с собой» не играет.

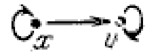
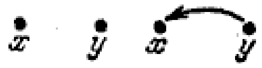
Если матрица T задана, то по ней может быть построен ориентированный взвешенный граф $G(T)$, дуга (x_i, x_j) которого направлена от вершины x_iк x_j, если $t_{ij} > 0$, и тогда этой дуге приписывается вес t_{ij}. Такой граф очевидно антирефлексивен и заполнен: между каждой парой вершин имеется хотя бы одна дуга. В случае $m = 1$ граф $G(T)$ асимметричен, и «вес» дугам может не приписываться, так как у всех дуг этот вес равен 1. Общий список игроков составляет множество A всех альтернатив; любое подмножество игроков из этого списка, т. е. турнир любого состава – предъявление X.

Теперь введём правила выбора, которые отражают идею выбора лучших игроков из множестваX. Рассмотрим два таких правила: ***правило «сумма очков»*** $\pi_{со}$ и ***максиминное правило*** «гарантированного результата» π_{mm}.

Обозначим через $S_X(x_i)$ сумму элементов t_{ij} строки x_i подматрицы T_X. Тогда правило π_{co} «сумма очков» определяется так:

π_{co}: $y \in C(X) \rightleftarrows ((y \in X) \wedge (\forall x \in X)\ [S_X(y) \geq S_X(x)])$. (5)

Таблица 1. Бинарные отношения и функции выбора на двухэлементном множестве

№	$C^i(x)$	$C^i(y)$	$C^i(x, y)$	R_1	R_2
1	x	y	$\{x, y\}$		
2	x	y	x		
3	x	y	y		
4	x	y	$\varnothing$		
5	x	$\varnothing$	$\{x, y\}$	не существует	не существует
6	$\varnothing$	y	$\{x, y\}$	не существует	не существует
7	x	$\varnothing$	x		
8	x	$\varnothing$	y	не существует	не существует
9	$\varnothing$	y	y		
10	$\varnothing$	y	x	не существует	не существует
11	x	$\varnothing$	$\varnothing$		
12	$\varnothing$	y	$\varnothing$		
13	$\varnothing$	$\varnothing$	$\{x, y\}$	не существует	не существует
14	$\varnothing$	$\varnothing$	x	не существует	не существует
15	$\varnothing$	$\varnothing$	y	не существует	не существует
16	$\varnothing$	$\varnothing$	$\varnothing$		
					

Согласно этому правилу, из X выбираются те игроки, которые в множестве X имеют наибольшее число побед. Конечно, такой выбор (при любой матрице T_X) не пуст, но не обязательно одноэлементен, так как максимальное число побед могут иметь сразу несколько игроков.

Обозначим через $\rho_X(x_i)$ минимальный элемент t_{ij} в строке x_i подматрицы T_X среди элементов, расположенных вне главной диагонали (для которых $i \neq j$). Тогда максиминное правило выбора победителя определяется так:

π_{mm}: $y \in C(X) \rightleftarrows ((y \in X) \wedge (\forall x \in X)\, [\rho_X(y) \geq \rho_X(x)])$. (6)

Это правило π_{mm} «объявляет» победителями игроков, у которых наименьшее число выигрышей максимально, т.е. использует логику гарантированного результата.

Пример 5. В таблице 2 приведён пример турнирной матрицы для пятикругового турнира, в котором участвуют четыре игрока x_1, x_2, x_3, x_4 (т.е. $\Omega = \{x_1, x_2, x_3, x_4\}$). В правых столбцах выписаны значения $S_\Omega(x)$ и $\rho_\Omega(x)$. По правилу π_{co} победитель – игрок x_4, а по правилу π_{mm}– игроки x_2 и x_4. Для подтаблицы, соответствующей турниру T_X, составленному из игроков x_1, x_2, x_3, победителем по обоим правилам является игрок x_1 (см. таблицу 3). Заметим, что для множеств $X = \{x_1, x_2, x_3\}$ и $\Omega = \{x_1, x_2, x_3, x_4\}$ условие (Н) выполняется для правила π_{co} «сумма очков» и не выполняется для правила π_{mm} «максимин». Действительно, по правилу π_{mm} элемент x_2 выбирается из $\Omega = \{x_1, x_2, x_3, x_4\}$ и не выбирается из его подмножества $X = \{x_1, x_2, x_3\}$, содержащего x_2.

Таблица 2. Турнир 4×4

	x_1	x_2	x_3	x_4	$S_\Omega(x)$	$\rho_\Omega(x)$
x_1	0	3	4	0	7	0
x_2	2	0	2	3	7	2
x_3	1	3	0	3	7	1
x_4	5	2	2	0	9	2

Таблица 3. Подтурнир 3×3

	x_1	x_2	x_3	$S_X(x)$	$\rho_X(x)$
x_1	0	3	4	7	3
x_2	2	0	2	4	2
x_3	1	3	0	4	1■

Заметим, что при проверке сформулированных выше условий (Н) и (О) для отрицательного вывода достаточно невыполнения условия только для одной пары множеств X_1 и X_2, таких, что $X_1 \subseteq X_2$. А для положительного вывода требуется проверка всех таких пар. Обратим внимание на то, что правило π_{co} суммы очков при однокруговом турнире совпадает с 1-ым правилом Коупленда из раздела 16-3.

2.3. Совокупно-экстремальный механизм выбора. Рассмотрим следующую конструкцию. Пусть на множестве Ω задано несколько обычных числовых функций, имеющих смысл критериев: g_1, g_2, …, g_m. Из предъявляемого для выбора множества $X \subseteq \Omega$ сначала отбираются элементы, максимальные по 1-му критерию g_1, затем – максимальные по 2-му критерию g_2, и т.д., вплоть до последнего критерия g_m, а затем берётся объединение полученных выборов. Функции выбора, построенные указанным образом, называются ***совокупно-экстремальными***.

2.4. Паретовский механизм выбора. Пусть опять на множестве Ω задано несколько обычных числовых функций, имеющих смысл критериев:g_1, g_2, …, g_m. На предъявляемом для выбора множестве $X \subseteq \Omega$ определим выбор $C(X)$ следующим образом: $x \in C(X)$ тогда и только тогда, когда для любого другого элемента $y \in X$ неверно, что

$g_i(y) \geq g_i(x)$ $(i = 1, 2, \ldots, n)$ и $\exists j$: $g_j(y) > g_j(x)$.

Функции выбора, построенные указанным образом, называются **паретовскими**.

3. Метатеория функций выбора

Функции выбора удобно классифицировать по тем условиям, которые обычно рассматриваются при их изучении. Приведем основные – часто упоминаемые, наиболее исследованные и используемые – классы функций выбора.

Условие наследования (Н):

$$\text{если } X' \subseteq X, \text{ то } C(X') \supseteq C(X) \cap X'.$$

Условие независимости от отвергнутых альтернатив (О):

$$\text{если } C(X) \subseteq X' \subseteq X, \text{ то } C(X') = C(X).$$

Условие согласия (С):

$$C(X_1)\cap C(X_2) \subseteq C(X_1\cup X_2).$$

Условие независимости от пути (П):

$$C(X_1\cup X_2) = C(C(X_1) \cup C(X_2))$$

Условие (Н) совпадает с формулой (1), смысл которой уже обсуждался. Просто повторим: если рассмотреть выбор из произвольного множества и выбор из некоторого его подмножества, то все альтернативы, которые были выбраны из исходного множества и вошли в рассматриваемое подмножество, будут выбраны также из этого подмножества.

Смысл условия (О) состоит в том, что если рассмотреть произвольное подмножество X', содержащее все альтернативы, выбранные из X, то выбор из X' будет совпадать с выбором из исходного подмножества; в частности, $C(C(X)) = C(X)$. Если проведён конкурс, в котором проект x не включен в число лучших, то в конкурсе, в котором участвуют все те же проекты, что и в первом конкурсе, за исключением x, состав победителей останется прежним.

Смысл условия (С) состоит в том, что альтернативы, которые были выбраны как из X_1, так и из X_2 (если таковые есть), будут также выбраны из их объединения $X_1\cup X_2$.

Смысл условия (П) таков. Если это условие нарушено, то можно найти в Ω два подмножества X_1 и X_2, предъявив которые порознь, осуществив выбор из них и используя в качестве следующего предъявления объединение выбранных так альтернатив, можно получить иной результат, чем при «разовом» предъявлении исходного множества $X_1\cup X_2$. Условие (П) исключает такую возможность.

Каждое из сформулированных условий (Н, О, С, П) определяет некоторый класс функций выбора, удовлетворяющих данному условию. Сохраним за классами названия соответствующих условий. Запись $C\in$Н будет означать, что рассматривается функция, удовлетворяющая условию наследования. Функции выбора, для которых выбор их любого непустого множества непуст, назовём ***непустыми***. Некоторые другие классы функций выбора будут определены далее.

3.1. Связи между классами. В этом разделе вкратце приводятся классические результаты о связях введённых основных классов функций выбора между собой и с некоторыми классами, определяемыми рассмотренными выше механизмами выбора.

Утверждение 1. Н∩О = П ■

Это утверждение означает, что функция выбора удовлетворяет условиям (Н) и (О) тогда и только тогда, когда она удовлетворяет условию (П). Другую характеризацию этого же класса П даёт

Утверждение 2. Непустая функция выбора принадлежит классу П тогда и только тогда, когда она является совокупно-экстремальной ■

Утверждение 3. Функция выбора является парно-доминантной тогда и только тогда, когда она удовлетворяет условиям (Н) и (С) ■

Утверждение 4. Непустая функция выбора является паретовской тогда и только тогда, когда она удовлетворяет условиям (Н), (О) и (С) ■

4. Логические формы функций выбора

Пусть $\Omega = \{x_1, \ldots, x_n\}$ – множество альтернатив, на котором задана функция выбора C. Логической формой выбора (ЛВФ(C)) на Ω называется следующий упорядоченный набор n булевых функций от(n–1) переменных:

$$\langle f_1(\beta_2(X), \ldots, \beta_n(X)), \ldots, f_i(\beta_1(X), \ldots, \beta_{i-1}(X), \beta_{i+1}(X), \ldots, \beta_n(X)), \ldots, f_n(\beta_1(X), \ldots, \beta_{n-1}(X)) \rangle,$$

где

$$\beta_i(X) = \begin{cases} 1, \text{если } x_i \in C(X) \\ 0, \text{если } x_i \notin C(X) \end{cases}; \quad (7)$$

и

$$\beta_i(X)\wedge f_i(\beta(X)) = 1 \rightleftarrows x_i\in C(X); \quad (8)$$

ЛВФ(C) = $\langle f_1, f_2 \ldots, f_n\rangle$.

Нетрудно заметить, что в силу (8) ЛВФ(C) по функции выбора C определяется однозначно. Наоборот, упорядоченный набор из n булевых функций от (n–1) переменных однозначно определяет некоторую функцию выбора C по формулам (8) и (7). Таким образом, задание функции выбора эквивалентно заданию её ЛВФ. Это позволяет во многих случаях свести исследования свойств функций выбора, их классификацию, взаимосвязи и декомпозиции к анализу хорошо известных свойств логических функций. Приведём только одно простое

Утверждение 5. Функция выбора CсЛВФ(C) = $\langle f_1, f_2\ldots, f_n\rangle$ является а) наследственной (Н); б) независимой от отвергнутых альтернатив (О); в) удовлетворяющей условию согласия (С) тогда и только тогда, когда:

а) для любых двух наборов $\beta^1 = \langle\beta_1^1, \ldots, \beta_{n-1}^1\rangle$ и $\beta^2 = \langle\beta_1^2, \ldots, \beta_{n-1}^2\rangle$ имеет место импликация $(\beta^1 \leq \beta^2) \rightarrow (f_i(\beta^1) \geq f_i(\beta^2))(i = 1, \ldots, n)$;

б) для любых двух наборов $\beta^1 = \langle\beta_1^1, \ldots, \beta_{n-1}^1\rangle$ и $\beta^2 = \langle\beta_1^2, \ldots, \beta_{n-1}^2\rangle$ имеет место импликация $\beta_i^2 f_i(\beta^2) \leq \beta_i^1 \leq \beta_i^2$ $(i = 1, \ldots, n) \rightarrow \beta_i^2 f_i(\beta^2) = \beta_i^1 f_i(\beta^1)$ $(i = 1, \ldots, n)$;

в) $f_i(\beta^1) \wedge f_i(\beta^2) \leq f_i(\beta^1 \vee \beta^2)$ $(i = 1, \ldots, n)$ ■

Знак $\leq$ между векторами означает покомпонентные неравенства, а знак $\vee$ – покомпонентные дизъюнкции.

Аналогичным образом в терминах условий на компоненты ЛВФ формулируются и многие другие условия на функции выбора. Приведём примеры, иллюстрирующие введённые понятия.

Пример 6. Пусть

$\Omega = \{a, b, c\}$, $C(\Omega) = \{a,c\}$, $C(a,b) = \{a\}$, $C(a,c) = \varnothing$, $C(b,c) = \{b\}$, $C(a) = \varnothing$, $C(b) = b$, $C(c) = \{c\}$. Выпишем логическую форму данной функции выбора. В соответствии с общей конструкцией нужно определить три функции от двух переменных: $f_1(\beta_2,\beta_3)$, $f_2(\beta_1,\beta_3)$, $f_3(\beta_1,\beta_2)$. Функция $f_1(\beta_2,\beta_3)$ определяется из условий:

$f_1(\beta_2,\beta_3) = 1$ элемент a выбирается из множества X, и

$$\beta_2 = \begin{cases} 1, \text{если } b \in X \\ 0, \text{если } b \notin X \end{cases}$$

$$\beta_3 = \begin{cases} 1, \text{если } c \in X \\ 0, \text{если c} \notin X \end{cases}$$

Аналогично определяются функции $f_2(\beta_1,\beta_3)$ и $f_3(\beta_1,\beta_2)$.

В данном случае по условию a выбирается из $X = \{a, b, c\}$. Это значит, что $f_1(1,1) = 1$. Далее, a выбирается из $X = \{a, b\}$. Поскольку $b \in X$, $c \notin X$, то $\beta_2 = 1$, $\beta_3 = 0$ и $f_1(1,0) = 1$. Далее, a не выбирается из $X = \{a, c\}$. Поскольку $b \notin X$, $c \in X$, то $\beta_2 = 0$, $\beta_3 = 1$ и $f_1(0,1) = 0$. Наконец, a не выбирается из $X = \{a\}$. Поскольку $b \notin X$, $c \notin X$, то $\beta_2 = 0$, $\beta_3 = 0$ и $f_1(0,0) = 0$. Таким образом, функция $f_1(\beta_2,\beta_3)$ определена на всех 4-х наборах:

β_2	β_3	$f_1(\beta_2,\beta_3)$
0	0	0
0	1	0
1	0	1
1	1	1

Аналогичные рассуждения показывают, что функции $f_2(\beta_1,\beta_3)$ и $f_3(\beta_1,\beta_2)$ задаются таблицами:

β_1	β_3	$f_2(\beta_1,\beta_3)$
0	0	1
0	1	1
1	0	0
1	1	0

β_2	β_3	$f_3(\beta_1,\beta_2)$
0	0	1
0	1	0
1	0	0
1	1	1

Вспоминая таблицу истинности 5-5 для всех функций от двух переменных, получаем, что

$f_1(\beta_2,\beta_3) = \beta_2, f_2(\beta_1,\beta_3) = \neg\beta_1, f_3(\beta_1,\beta_2) = \beta_1 \rightleftarrows \beta_2$■

Пример 7. Проверим, удовлетворяет ли функция выбора из примера 6условию наследственности. В силу утверждения 5 необходимым и достаточным условием наследственности является антимонотонность всех трёх функций $f_1(\beta_2,\beta_3)$, $f_2(\beta_1,\beta_3)$, $f_3(\beta_1,\beta_2)$, образующих логическую форму данной функции выбора.

Антимонотонность f означает, что $(\beta^1 \le \beta^2) \to (f(\beta^1) \ge f(\beta^2))$ (неравенство между наборами β^1 и β^2 является покомпонентным, а между значениями функции на двух наборах – числовым). Рассмотрим функцию $f_1(\beta_2,\beta_3)$ на двух наборах: $\beta^1 = (0,0)$ и $\beta^2 = (1,1)$. Так как $\beta^1 \le \beta^2$, то по условию антимонотонности должно быть $f(\beta^1) \ge f(\beta^2)$. В данном же случае $f(\beta^1) = 0$, $f(\beta^2) = 1$, что противоречит требуемому неравенству. Значит, данная функция выбора не является наследственной ■

Пример 8. По данному семейству 3-ёх булевых функций от 2-ух переменных построим функцию выбора C, для которой данное семейство образует логическую форму. Пусть

$$f_1(\beta_2,\beta_3) = \beta_2\vee\beta_3,\ f_2(\beta_1,\beta_3) = \neg\beta_1\wedge\beta_3,\ f_3(\beta_1,\beta_2) = \beta_1 \rightleftarrows \beta_2.$$

Начнём с одноэлементных множеств. Определим $C(a)$. В данном случае $X = \{a\}$, поэтому $\beta_2(X) = 0$, $\beta_3(X) = 0$. Так как $f_1(\beta_2,\beta_3) = \beta_2\vee\beta_3 = 0 \vee 0 = 0$, то $C(a) = \varnothing$. Далее, для $C(b)$ имеем $\beta_1(X) = 0$, $\beta_3(X) = 0$. Так как $f_2(\beta_1,\beta_3) = \neg\beta_1\wedge\beta_3 = 0$, то $C(b) = \varnothing$. Далее, для $C(c)$ имеем $\beta_1(X) = 0$, $\beta_2(X) = 0$. Так как $f_3(\beta_1,\beta_2) = \beta_1 \rightleftarrows \beta_2 = 0 \rightleftarrows 0 = 1$, то $C(c) = \{c\}$.

Перейдём к двухэлементным множествам. Определим $C(a,b)$.В данном случае $X = \{a,b\}$, поэтому $\beta_1(X) = 1$, $\beta_2(X) = 1$, $\beta_3(X) = 0$. Так как $f_1(\beta_2,\beta_3) = \beta_2\vee\beta_3 = 1\vee 0 = 1$, то $a \in C(a,b)$. Так как $f_2(\beta_1,\beta_3) = \neg\beta_1\wedge\beta_3 = 0$, то $b \notin C(a,b)$. Поэтому $C(a,b) = \{a\}$. Определим $C(a,c)$. В данном случае $X = \{a,c\}$,$\beta_1(X) = 1$, $\beta_2(X) = 0$, $\beta_3(X) = 1$, откуда $f_1(\beta_2,\beta_3) = \beta_2\vee\beta_3 = 1$, $f_3(\beta_1,\beta_2) = \beta_1 \rightleftarrows \beta_2 = 0$. Поэтому $C(a,c) = \{a\}$. Определим $C(b,c)$. В данном случае $X = \{b,c\}$,$\beta_1(X) = 0$, $\beta_2(X) = 1$, $\beta_3(X) = 1$, откуда $f_2(\beta_1,\beta_3) = \neg\beta_1\wedge\beta_3 = 1$, $f_3(\beta_1,\beta_2) = \beta_1 \rightleftarrows \beta_2 = 0$. Поэтому $C(b,c) = \{b\}$.

Наконец, для единственного трёхэлементного множества $\{a,b,c\}$ имеем $X = \{a,b,c\}$,$\beta_1(X) = 1$, $\beta_2(X) = 1$, $\beta_3(X) = 1$, откуда $f_1(\beta_2,\beta_3) = \beta_2\vee\beta_3 = 1$, $f_2(\beta_1,\beta_3)\neg\beta_1\wedge\beta_3 = 0$, $f_3(\beta_1,\beta_2) = \beta_1 \rightleftarrows \beta_2 = 1$. Поэтому $C\{a,b,c\} = \{a,c\}$. Объединяя рассмотренные случаи, получаем функцию выбора $C(X)$, определённую на всех непустых подмножествах множества $\{a,b,c\}$:

$$C(a) = \varnothing,\ C(b) = \varnothing,\ C(c) = \{c\},\ C(a,b) = \{a\},\ C(a,c) = \{a\},\ C(b,c) = \{b\},\ C\{a,b,c\} = \{a,c\}$$ ■

5. Задания

Задание 1. По данной турнирной матрице пятикругового турнира определить по правилам π_{co}: и π_{mm} победителей всего турнира и победителей подтурниров, составленных из игроков $\{x_1, x_2, x_3\}$ и $\{x_2, x_3, x_4\}$. См. пример 5 для образца.

Варианты турнирных матриц:

	x_1	x_2	x_3	x_4
x_1	0	2	4	1
x_2	3	0	3	5
x_3	1	2	0	2
x_4	4	0	3	0

01

	x_1	x_2	x_3	x_4
x_1	0	1	2	3
x_2	4	0	4	0
x_3	3	1	0	1
x_4	2	5	4	0

02

	x_1	x_2	x_3	x_4
x_1	0	3	1	4
x_2	2	0	2	0
x_3	4	3	0	3
x_4	1	5	2	0

03

	x_1	x_2	x_3	x_4
x_1	0	1	4	4
x_2	4	0	3	2
x_3	1	2	0	1
x_4	1	3	4	0

04

	x_1	x_2	x_3	x_4
x_1	0	2	4	3
x_2	3	0	1	5
x_3	1	4	0	2
x_4	2	0	3	0

05

	x_1	x_2	x_3	x_4
x_1	0	3	1	2
x_2	2	0	4	5
x_3	4	1	0	3
x_4	3	0	2	0

06

	x_1	x_2	x_3	x_4
x_1	0	2	4	3
x_2	3	0	1	5
x_3	1	4	0	2
x_4	0	0	3	0

07

	x_1	x_2	x_3	x_4
x_1	0	4	3	1
x_2	1	0	2	5
x_3	2	3	0	3
x_4	4	0	2	0

08■

Задание 2. Выписать ЛВФ для заданной функции выбора на $\Omega = \{a, b, c\}$. См. для образца пример 6.

Варианты функций выбора для задания 2:

$C(\Omega) = \{a,c\}$, $C(a,b) = \varnothing$, $C(a,c) = \{a\}$, $C(b,c) = \{c\}$, $C(a) = \{a\}$, $C(b) = \varnothing$, $C(c) = \{c\}$.
$C(\Omega) = \{a,b\}$, $C(a,b) = \{a\}$, $C(a,c) = \{c\}$, $C(b,c) = \{b,c\}$, $C(a) = \varnothing$, $C(b) = \varnothing$, $C(c) = \{c\}$.
$C(\Omega) = \{b,c\}$, $C(a,b) = \{b\}$, $C(a,c) = \{a,c\}$, $C(b,c) = \varnothing$, $C(a) = \{a\}$, $C(b) = \{b\}$, $C(c) = \{c\}$.
$C(\Omega) = \{a,b\}$, $C(a,b) = \{a,b\}$, $C(a,c) = \varnothing$, $C(b,c) = \{b,c\}$, $C(a) = \varnothing$, $C(b) = \{b\}$, $C(c) = \{c\}$.
$C(\Omega) = \{a,b,c\}$, $C(a,b) =\{a\}$, $C(a,c) = \varnothing$, $C(b,c) = \{b\}$, $C(a) = \{a\}$, $C(b) = \varnothing$, $C(c) = \varnothing$.
$C(\Omega) = \varnothing$, $C(a,b) = \{b\}$, $C(a,c) = \{c\}$, $C(b,c) = \{c\}$, $C(a) = \varnothing$, $C(b) = \varnothing$, $C(c) = \varnothing$.
$C(\Omega) = \{a,b\}$, $C(a,b) = \{b\}$, $C(a,c) = \{a,c\}$, $C(b,c) = \{b\}$, $C(a) = \{a\}$, $C(b) = \{b\}$, $C(c) = \varnothing$.
$C(\Omega) = \{b,c\}$, $C(a,b) = \varnothing$, $C(a,c) = \{a,c\}$, $C(b,c) = \varnothing$, $C(a) = \varnothing$, $C(b) = \{b\}$, $C(c) = \varnothing$■

Задание 3. Для функции выбора, заданной ЛФВ, проверить условия (Н) и (О), пользуясь результатом утверждения 5. См. для образца пример 7.

Варианты ЛВФ для задания 3:

01. $f_1(\beta_2,\beta_3) = \beta_2\vee\beta_3$, $f_2(\beta_1,\beta_3) = \beta_1\wedge\beta_3$, $f_3(\beta_1,\beta_2) = \neg\beta_1 \rightleftarrows \beta_2$
02. $f_1(\beta_2,\beta_3) = \beta_2\rightarrow\neg\beta_3$, $f_2(\beta_1,\beta_3) = \beta_1\wedge\beta_3$, $f_3(\beta_1,\beta_2) = \beta_1 \rightleftarrows \beta_2$
03. $f_1(\beta_2,\beta_3) = \neg\beta_2\wedge\neg\beta_3$, $f_2(\beta_1,\beta_3) = \beta_1\wedge\beta_3$, $f_3(\beta_1,\beta_2) = \beta_1 \rightleftarrows \beta_2$
04. $f_1(\beta_2,\beta_3) = \neg\beta_2\wedge\neg\beta_3$, $f_2(\beta_1,\beta_3) = \beta_1\rightarrow\beta_3$, $f_3(\beta_1,\beta_2) = \beta_1 \rightleftarrows \beta_2$
05.$f_1(\beta_2,\beta_3) = \neg\beta_2\wedge\neg\beta_3$, $f_2(\beta_1,\beta_3) = \beta_1\wedge\beta_3$, $f_3(\beta_1,\beta_2) = \beta_1 \rightleftarrows \beta_2$
06. $f_1(\beta_2,\beta_3) = \neg\beta_2\wedge\neg\beta_3$, $f_2(\beta_1,\beta_3) = \beta_1\wedge\beta_3$, $f_3(\beta_1,\beta_2) = \beta_1\oplus\beta_2$
07. $f_1(\beta_2,\beta_3) = \beta_2\wedge\neg\beta_3$, $f_2(\beta_1,\beta_3) = \neg\beta_1\wedge\beta_3$, $f_3(\beta_1,\beta_2) = \beta_1\oplus\beta_2$
08. $f_1(\beta_2,\beta_3) = \beta_2\wedge\beta_3$, $f_2(\beta_1,\beta_3) = \neg\beta_1\wedge\beta_3$, $f_3(\beta_1,\beta_2) = \beta_1\oplus\beta_2$
09. $f_1(\beta_2,\beta_3) = \beta_2\wedge\beta_3$, $f_2(\beta_1,\beta_3) = \neg\beta_1\rightarrow\beta_3$, $f_3(\beta_1,\beta_2) = \beta_1\oplus\beta_2$
10. $f_1(\beta_2,\beta_3) = \neg\beta_2\wedge\neg\beta_3$, $f_2(\beta_1,\beta_3) = \beta_1\rightarrow\beta_3$, $f_3(\beta_1,\beta_2) = \beta_1\vee\beta_2$■

Задание 4. По данному семейству 3-ёх булевых функций от 2-ух переменных построить функцию выбора C, для которой данное семейство образует логическую форму. См для образца пример 8.

Варианты задания 4 совпадают с вариантами задания 3 ■

6. Предметный указатель

согласия
функция выбора
блокировки
наследственная
независимая от отвергнутых альтернатив
непустая
нормальная
паретовская
парно-доминантная
порожденная отношением R
предпочтения
совокупно-экстремальная
m-круговой турнир

Литература к части 4

1. Айзерман М.А., Алескеров Ф.Т. Выбор вариантов: основы теории. – М.: Наука. Гл. ред. физ.-мат. лит., 1990. − 240с. – ISBN5-02-014091-0.

2. Алескеров Ф.Т., Хабина Э.Л., Шварц Д.А.. Бинарные отношения, графы и коллективные решения. − 2-е изд., перераб. и доп. − М.: ФИЗМАТЛИТ, 2012. − 344 с. – ISBN 978-5-9221-1363-2.

3. Ларичев О.И. Теория и методы принятия решений, а также Хроника событий в Волшебных странах: Учебник. Изд. третье, перераб. и доп. – М.: Университетская книга, Логос, 2006. – 392 с. – ISBN5-98704-132-5.

4. Робертс Ф.С. Дискретные математические модели с приложениями к социальным, биологическим и экологическим задачам. − М.: Наука. Гл. ред. физ.-мат. лит., 1986. − 496 с.

5. Теория выбора и принятия решений: Учебное пособие.− М.: Наука. Гл. ред. физ.-мат. лит. 1982. – 328 с.

Содержание

Александр Анатольевич Рубчинский

Дискретные математические модели. Начальные понятия и стандартные задачи

Учебное пособие

Ответственный редактор *Н. Соломадина*
Верстальщик *С. Мартынович*

Сдано в набор 03.05.2016
Подписано к печати 02.05.2016

Формат 60x90/16

Печ. л. 16,81

Тираж 500
Заказ 16-05-03

Издательство «Директ-Медиа»
117342, Москва, ул. Обручева, 34/63, стр. 1
Тел/факс + 7 (495) 334-72-11
E-mail: manager@directmedia.ru
www.biblioclub.ru
www.directmedia.ru