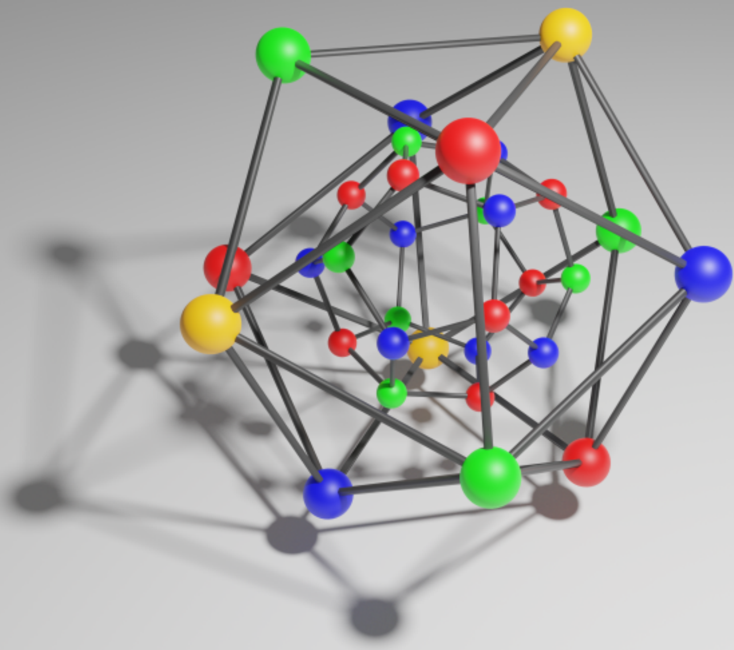
М.И.Дехтярь, С.М.Дудаков, Б.Н.Карлов



ЗАДАЧНИК по дискретной математике

математические основы информатики

Министерство науки и высшего образования
Российской Федерации

Тверской государственный
университет

$\boxed{\text{М. И. Дехтярь}}$
С. М. Дудаков
Б. Н. Карлов

Задачник по дискретной математике

Учебное пособие

Издание второе, переработанное и дополненное

Тверь — 2021

УДК 510 + 519
ББК 22.12 + 22.176 я73-4
Д 39

Тверской государственный университет
Факультет прикладной математики и кибернетики

Дехтярь М. И., Дудаков С. М., Карлов Б. Н.
Д 39 Задачник по дискретной математике : Учебное пособие. — [2-е изд., перераб. и доп.] — Тверь : Твер. гос. ун-т, 2021. — 368 с.

Учебное пособие адресовано изучающим курс дискретной математики, прежде всего, студентам младших курсов, обучающимся по направлениям укрупненных групп 01.03.00 «Математика и механика», 02.03.00 «Компьютерные и информационные науки», 09.03.00 «Информатика и вычислительная техника».

Настоящий сборник задач является пособием для практических занятий по некоторым разделам дискретной математики и может быть использован преподавателями и студентами для подготовки к семинарским занятиям и контрольным работам.

В книгу вошло более шестисот задач и упражнений, многие из которых состоят из нескольких независимых подзадач. Почти все задачи снабжены ответами, а многие более сложные задачи — указаниями или решениями.

УДК 510 + 519
ББК 22.12 + 22.176 я73-4

Оглавление

Предисловие

Ко второму изданию

Настоящая книга является вторым изданием задачника [4]. В связи выходом с нового, значительно переработанного, издания [3] учебника [2], на котором ранее базировался задачник [4], возникла необходимость переработки задачника и внесения дополнений.

Исправления в основном сводятся к приведению обозначений и определений в соответствие с [3]. Кроме того, исправлены замеченные неточности и опечатки. Исключены задания, которые непосредственно связаны с текстом учебника [2], и вне его контекста теряют смысл.

В связи с перегруппировкой материала в учебнике [3] произошла и соответствующая перегруппировка задач. Например, раздел 12 из предыдущего издания, который был посвящён сразу ориентированным и неориентированным графам, в настоящем разбит на два отдельных раздела. Это, а также добавление новой части, посвящённой кодированиям, привело к изменению нумерации разделов.

В числе дополнений прежде всего следует упомянуть задания на темы, которые в [2] отсутствовали: табличный метод построения многочленов Жегалкина, свойства неориентированных графов, кодирования и ряд других.

Добавлено много задач по уже имевшимся темам. Для ряда задач включены ранее отсутствовавшие решения или указания.

Всего в настоящее издание было добавлено более двухсот новых задач.

Все замечания и предложения следует отправлять авторам по адресам электронной почты:

sergeydudakov@yandex.ru
bnkarlov@gmail.com

К первому изданию

Настоящий сборник задач является пособием для практических занятий по дискретной математике и предназначен в основном для студентов первого курса Тверского государственного университета, которые обучаются по направлениям 010300.62 — «Фундаментальная информатика и информационные технологии», 010400.62 — «Прикладная математика и информатика», 080500.62 — «Бизнес-информатика» и 230700.62 — «Прикладная информатика». Он может быть использован преподавателями для подготовки к практическим занятиям и контрольным работам.

По темам задач и по структуре, а также по терминологии и обозначениям данный задачник является дополнением к учебному пособию [2], которое на протяжении ряда лет используется в ТвГУ в качестве основного учебника по курсу дискретной математики. В частности, в задачник вошли почти все задачи из [2]. Опыт преподавания показал, что этот набор следовало бы существенно расширить. Другие известные нам пособия и задачники[1] по дискретной математике не содержат всех существенных для нашей программы разделов. Например, во многих отсутствуют основы теории конечных автоматов и алгоритмов, нигде нет задач по Хорновским формулам и упорядоченным бинарным диаграммам решений.

Задачник может использоваться и как независимое пособие, поскольку задачам каждого раздела предшествует небольшое теоретическое введение, содержащее основные определения и формулы, используемые при их решении.

[1] Некоторые из них, послужившие источниками ряда задач, указаны в списке литературы.

В книге помещено около 400 задач и упражнений, многие из которых состоят из нескольких независимых подзадач. Условно их можно отнести к следующим категориям.

1) Упражнения на закрепление основного материала, в частности, задачи на выполнение стандартных преобразований и на «прокрутку» тех или иных процедур и алгоритмов.

2) Задачи на доказательство (как правило, не очень сложное) отдельных утверждений об основных изучаемых понятиях.

3) Задачи на установление свойств новых, не определённых в основном тексте понятий. Эти задачи расширяют материал книги [2] и предназначены для самых любознательных читателей.

Большинство задач типа 1) снабжено ответами, а многие задачи типов 2) и 3) сопровождаются также указаниями и решениями.

Задачи в сборнике собраны в 23[2] раздела. Первые три являются введением и позволяют получить первоначальные представления о множествах, отношениях и функциях, приобрести некоторые навыки в проведении доказательств методом математической индукции, научиться применять основные формулы комбинаторики. Разделы 4–9 посвящены булевым функциям. Тематика первых пяти из них достаточно традиционна. Задачи раздела 9 посвящены интересному подклассу булевых формул — хорновским формулам — и эффективным алгоритмам проверки выводимости для них. Задачи разделов 10 и 11 предназначены для студентов специальностей, по которым не предусмотрены отдельные курсы по математической логике. Они в основном должны научить пониманию семантики формул логики предикатов и продемонстрировать её связь с теорией баз данных. Разделы 12–14 включают задачи, связанные с основами теории графов, важным подклассом графов — деревьями, а также несколькими классическими алгоритмами на графах. Разделы 15 и 16 посвящены двум разным способам представления булевых функций с помощью графов: логическим схемам и упорядоченным бинарным диаграммам

[2] В настоящем издании раздел 12 разбит на два, добавлен раздел о кодированиях, поэтому нумерация, начиная с раздела 13, изменена.

решений. Конечные автоматы рассматриваются в разделах 17–19. Учащиеся должны научиться строить конечные автоматы и регулярные выражения для достаточно простых языков, детерминизировать недетерминированные конечные автоматы и использовать лемму о разрастании для доказательства неавтоматности некоторых языков. Последняя часть задачника посвящена началам теории алгоритмов. Основные задачи в разделах 20–22 относятся к реализации различных арифметических функций с помощью одного из трёх алгоритмических языков: структурированных программ, частично рекурсивных определений и машин Тьюринга. В задачах раздела 23 содержится материал о моделировании одних моделей алгоритмов другими и о неразрешимых алгоритмических проблемах.

1. Множества, отношения и функции

Основные множества. Символом $\varnothing$ обозначается пустое множество: для любого x выполнено $x \notin \varnothing$; $\omega = \{0, 1, 2, 3, \ldots\}$ — множество натуральных чисел; $\mathbb{R}$ — множество всех вещественных (действительных) чисел.

Операции над множествами. Объединение множеств:

$$A \cup B = \{x : x \in A \text{ или } x \in B\}.$$

Пересечение множеств:

$$A \cap B = \{x : x \in A \text{ и } x \in B\}.$$

Разность множеств:

$$A \setminus B = \{x : x \in A \text{ и } x \notin B\}.$$

При наличии универсума U дополнение множества A:

$$\bar{A} = U \setminus A.$$

Симметрическая разность (или дизъюнктивная сумма):

$$A \dot{-} B = (A \setminus B) \cup (B \setminus A).$$

Декартово произведение множеств $A_1, \ldots, A_n$:

$$A_1 \times \ldots \times A_n = \{(a_1, \ldots, a_n) : a_1 \in A_1, \ldots, a_n \in A_n\}.$$

Декартова степень: при $n \geqslant 1$

$$A^n = \underbrace{A \times A \times \ldots \times A}_{n \text{ раз}}.$$

Множество A является подмножеством множества B (записывается в виде $A \subseteq B$), если все элементы A являются также элементами B. Множество подмножеств множества A

$$\mathsf{P}(A) = \{x : x \subseteq A\}.$$

Отношения и функции. Бинарное (или двухместное) отношение между элементами множеств A и B — подмножество R их декартова произведения $A \times B$. Область определения R:

$$\operatorname{dom} R = \{x : \text{существует } y \text{ такое, что } (x, y) \in R\}.$$

Область значений R:

$$\operatorname{rng} R = \{y : \text{существует } x \text{ такое, что } (x, y) \in R\}.$$

Обратное отношение для бинарного отношения R:

$$R^{-1} = \{(x, y) : (y, x) \in R\}.$$

Композиция отношений $R_1 \subseteq A \times B$ и $R_2 \subseteq B \times C$:

$$R_1 \circ R_2 = \{(x, z) : \text{существует } y \in B \text{ такое, что выполнено } (x, y) \in R_1 \text{ и } (y, z) \in R_2\}.$$

Бинарное отношение R на множестве A называется отношением эквивалентности, если для него выполнены следующие условия:

1) рефлексивность: $(a, a) \in R$ для любого $a \in A$;
2) симметричность: для любых $a, b \in A$ если выполнено $(a, b) \in R$, то $(b, a) \in R$;
3) транзитивность: для любых $a, b, c \in A$ если $(a, b) \in R$ и $(b, c) \in R$, то и $(a, c) \in R$.

Классы эквивалентности отношения R: для $a \in A$ его класс эквивалентности $[a]_R$ включает все эквивалентные a элементы:

$$[a]_R = \{b \in A : (a, b) \in R\}.$$

Бинарное отношение R на множестве A называется отношением частичного порядка, если для него выполнены следующие условия:

1) антисимметричность: для любых $a, b \in A$, если $(a, b) \in R$ и $(b, a) \in R$, то $a = b$;
2) транзитивность.

Если для $a, b \in A$ выполнено хотя бы одно из $(a, b) \in R$, $(b, a) \in R$ или $a = b$, то a и b называют сравнимыми. Отношением линейного порядка называют такое отношение частичного порядка R, что кроме условий 1) и 2) выполнено

3) линейность: любые элементы A сравнимы.

Бинарное отношение R частичного порядка на множестве A называется

4) нестрогим, если выполнена рефлексивность;
5) строгим, если выполнена антирефлексивность: $(a, a) \notin R$ для любого $a \in A$.

Элемент частично упорядоченного множества называется наибольшим (соответственно, наименьшим), если все другие элементы меньше (соответственно, больше) его. Элемент называется максимальным (соответственно, минимальным), если не существует элемента, который был бы больше (соответственно, меньше) его.

Бинарное отношение f называется функцией (или отображением) из A в B, если $\operatorname{dom} f = A$, $\operatorname{rng} f \subseteq B$ и для всех x, y_1, y_2 из того, что $(x, y_1) \in f$ и $(x, y_2) \in f$, следует, что $y_1 = y_2$. Записывается это в виде: $f : A \to B$. Для функции f вместо $(x, y) \in f$ пишем $f(x) = y$. Если $\operatorname{dom} f = A \subseteq A'$, то f называется частичной функцией из A' в B. Функция f называется разнозначной (или обратимой, инъективной, 1-1-функцией), если для любых x_1, x_2, y из того, что $f(x_1) = y$ и $f(x_2) = y$ следует, что $x_1 = x_2$. Функция $f : A \to B$ называется сюръективной, если $\operatorname{rng} f = B$. Функция $f : A \to B$ называется взаимно однозначной (или биективной), если она разнозначна и сюръективна. Взаимно однозначная функция $f : A \to A$ называется перестановкой множества A. Образ множества $C \subseteq A$ при отображении $f : A \to B$ — это множество

$$f[C] = \{f(x) : x \in C\}.$$

Прообраз множества $D \subseteq B$ при отображении $f : A \to B$ — это множество

$$f^{-1}[D] = \{x \in A : f(x) \in D\}.$$

С помощью $|A|$ обозначается мощность множества A. Если множество конечно, то мощность множества — это просто количество элементов в нём, натуральное число. Множества A и B называются равномощными (обозначается $|A| = |B|$), если существует взаимно однозначная функция $f : A \to B$. Множество A имеет мощность меньшую или равную B (обозначается $|A| \leqslant |B|$), если существует разнозначная функция $f : A \to B$ или, что эквивалентно, существует сюръективная функция $g : B \to A$. Множество называется счётным, если оно имеет мощность меньшую или равную мощности множества натуральных чисел ω, то есть $|A| \leqslant |\omega|$. Последнее означает, что все элементы множества можно пронумеровать натуральными числами, возможно, с повторениями.

Алфавит — произвольное конечное множество $\Sigma = \{a_1, \ldots, a_m\}$, элементы которого называются символами (или буквами). Слово в алфавите Σ — конечная последовательность символов этого алфавита: $w = a_{i_1} \ldots a_{i_n}$, $a_{i_j} \in \Sigma$ при $j = 1, \ldots, n$. Число n тогда называется длиной слова и обозначается $|w|$. Пустое слово длины ноль не содержит ни одного символа и обозначается ε. Множество всех слов алфавита Σ обозначается с помощью Σ^*.

Конкатенацией слов $u = a_{i_1} \ldots a_{i_n}$ и $v = a_{k_1} \ldots a_{k_\ell}$, называется слово $w = u\&v = a_{i_1} \ldots a_{i_n} a_{k_1} \ldots a_{k_\ell}$ длины $n+\ell$. Обычно знак конкатенации & опускают и пишут uv. Слово u при этом называют префиксом, а v — суффиксом слова w.

Для представления конкатенаций слова с собой же используют «степенную» форму записи:

$$w^n = \underbrace{w \,\&\, w \,\&\, \cdots \,\&\, w}_{n \text{ раз}}.$$

Задачи

1. Найти все подмножества следующих множеств: $\varnothing$, $\{\varnothing\}$, $\{1, 2, 3\}$, $\{a, \{1, 2\}, \varnothing\}$. ⊛

2. Дано множество $A = \{0, \{0, 1, 2\}, \{3\}, 4, \{\{5\}\}, 6\}$. Определить, какие из следующих множеств $B = \{0, 4\}$, $C = \{6, \{3\}, 0\}$, $D = \{0, 3\}$, $E = \{\{0, 1, 2\}, \{3\}\}$, $F = \{0, \{5\}\}$, $G = \{\{3\}, 2, \{\{5\}\}, 6\}$ не являются подмножествами A? ⊛

3. Даны множества: $A = \{a, b, \{\varnothing\}, \{a, c, d\}\}$, $B = \{a, c, e, \{a\}, \{b\}\}$ и $C = \{a, b, c, d, \{e\}, \varnothing\}$. Найти множество $D = (A \cup B) \setminus C$. Какова его мощность? ⊛

4. Даны множества: $A = \{a, b, c, \{\varnothing\}, \{a\}\}$, $B = \{a, e, \{a\}, \{b\}, \varnothing\}$ и $C = \{a, b, d, \{e\}, \{\varnothing\}\}$. Найти множество $D = (A \setminus B) \cap C$. Какова его мощность? ⊛

5. Пусть $A = \{0, 1\}$, $B = \{a, b, c\}$. Найти множества $A \times B$ и $B \times A$. ⊛

6. Определить, для каких множеств A и B выполняется равенство $A \times B = B \times A$? ⊛

7. Даны четыре множества: $A = \{0, 1, 2\}$, $B = \{2, 3\}$, $C = \{a, b, c\}$ и $D = \{a, c, e\}$. Определить, чему равны следующие множества:

(а) $F_1 = (A \setminus B) \times (C \cap D)$;

(б) $F_2 = (A \cap B) \times (C \cap D)$;

(в) $F_3 = (B \setminus A) \times (C \setminus D)$. ⊛

8. Доказать следующие включения:

(а) $A \cap B \subseteq A \subseteq A \cup B$;

(б) $A \setminus B \subseteq A$. ⊛

9. Доказать следующие тождества для любых множеств A, B, C:

(а) $A \cup A = A \cap A = A$;

(б) $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$;

(в) $(A \cup B) \cap A = (A \cap B) \cup A = A$;

(г) $A \setminus (B \cup C) = (A \setminus B) \cap (A \setminus C)$;

(д) $A \setminus (B \cap C) = (A \setminus B) \cup (A \setminus C)$;

(е) $(A \setminus B) \cap C = (A \cap C) \setminus B$;

(ж) $A \setminus (B \setminus C) = (A \setminus B) \cup (A \cap C)$;

(з) $(A \setminus B) \setminus C = A \setminus (B \cup C)$;

(и) $A \cup \varnothing = \varnothing \cup A = A$;

(к) $A \cap \varnothing = \varnothing \cap A = \varnothing$;

(л) $A \dot{-} \varnothing = \varnothing \dot{-} A = A$;

(м) $A \dot{-} A = \varnothing$. ⊛

10. Пусть множества A, B, C и их дополнения являются подмножествами универсума U. Доказать следующие тождества:

(а) $\overline{A \cap B} = \bar{A} \cup \bar{B}$;

(б) $\overline{A \cup B} = \bar{A} \cap \bar{B}$;

(в) $A \setminus \bar{B} = A \cap B$;

(г) $\bar{A} \setminus B = \bar{A} \cap \bar{B}$;

(д) $\bar{A} \setminus \bar{B} = B \setminus A$. ⊛

11. Доказать, что

(а) $A \times (B \cup C) = (A \times B) \cup (A \times C)$;

(б) $A \times (B \cap C) = (A \times B) \cap (A \times C)$;

(в) $A \times (B \setminus C) = (A \times B) \setminus (A \times C)$;

(г) если $A \subseteq B$ и $C \subseteq D$, то $(A \times C) = (A \times D) \cap (B \times C)$. ⊛

12. Доказать, что включение $A \subseteq B$ выполнено тогда и только тогда, когда выполнено $\mathsf{P}(A) \subseteq \mathsf{P}(B)$. ⊛

13. Доказать, что

(а) $A \subseteq B \cap C$ тогда и только тогда, когда $A \subseteq B$ и $A \subseteq C$;

(б) $A \subseteq B \setminus C$ тогда и только тогда, когда $A \subseteq B$ и $A \cap C = \varnothing$. ⊛

14. Доказать следующие равенства и включения:

(а) $\mathsf{P}(A \cap B) = \mathsf{P}(A) \cap \mathsf{P}(B)$;

(б) $\mathsf{P}(A \cup B) \supseteq \mathsf{P}(A) \cup \mathsf{P}(B)$;

(в) $\mathsf{P}(A \setminus B) \subseteq (\mathsf{P}(A) \setminus \mathsf{P}(B)) \cup \{\varnothing\}$.

Привести примеры, когда указанные включения будут строгими. ⊛

15. Для каждого из следующих отношений найти $\operatorname{dom} R$, $\operatorname{rng} R$, R^{-1}, $R \circ R$, $R \circ R^{-1}$:

(а) $R = \{(x, y) : x, y \in \omega$ и x делит $y\}$, x делит y, если существует такое z, что $xz = y$;

(б) $R = \{(x, y) : x, y \in \omega$ и $x + y \leqslant 10\}$;

(в) $R = \{(x, y) : x, y \in \omega$ и $y = 3x + 1\}$;

(г) $R = \{(x, x^2) : x \in \omega$ и $x \leqslant 10\}$;

(д) $R = \{(a, b), (b, c), (b, d), (c, d), (d, b)\}$. ⊛

16. Для каждого из следующих бинарных отношений на множестве $A = \{a, b, c\}$ определить, является ли оно рефлексивным, симметричным, антисимметричным, транзитивным:

(а) $R_1 = \{(a, a), (a, b), (b, a), (b, b), (c, c)\}$;

(б) $R_2 = \{(a, a), (a, c), (c, b), (a, b), (b, b), (c, c)\}$;

(в) $R_3 = \{(a, a), (a, c), (c, b), (a, b)\}$. ⊛

17. Построить множество наименьшей мощности, на котором существует нетранзитивное отношение. ⊛

18. На множестве всех непустых отрезков числовой прямой заданы три отношения:

(а) $P = \{([a, b], [c, d]) : c < a < b < d\}$,

(б) $Q = \{([a, b], [c, d]) : a < c < b < d\}$ и

(в) $R = \{([a, b], [c, d]) : b < c\}$.

Определить, какие из них являются отношениями частичного порядка. ⊛

19. Пусть бинарные отношения P и Q на множестве A являются рефлексивными. Определить, какие из следующих отношений также являются рефлексивными:

(а) $P \cap Q$; (б) $P \cup Q$; (в) $P \circ Q$; (г) P^{-1}. ⊛

20. Пусть бинарные отношения P и Q на множестве A являются симметричными. Определить, какие из следующих отношений также являются симметричными:

(а) $P \cap Q$; (б) $P \cup Q$; (в) $P \circ Q$; (г) P^{-1}. ⊛

21. Пусть бинарные отношения P и Q на множестве A являются антисимметричными. Определить, какие из следующих отношений также являются антисимметричными:

(а) $P \cap Q$; (б) $P \cup Q$; (в) $P \circ Q$; (г) P^{-1}. ⊛

22. Пусть бинарные отношения P и Q на множестве A являются транзитивными. Определить, какие из следующих отношений также являются транзитивными:

(а) $P \cap Q$; (б) $P \cup Q$; (в) $P \circ Q$; (г) P^{-1}. ⊛

23. Пусть множество $S = \{(i, j) : 1 \leqslant i, j \leqslant 8\}$ задаёт клетки шахматной доски. Описать следующие бинарные отношения на S:

(а) $L = \{(a, b)$: ладья за один ход может перейти с клетки a на клетку $b\}$;

(б) $K = \{(a, b)$: конь за один ход может перейти с клетки a на клетку $b\}$.

Будут ли эти отношения эквивалентностями? Описать отношение $L \circ L$. ⊛

24. Пусть $A = \{1, 2, 3, 4, 5\}$. Определить, какие из следующих отношений на A являются отношениями эквивалентности. Для тех из них, которые являются отношениями эквивалентности, найти их классы эквивалентности.

(а) $\{(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (1, 4), (4, 1)\}$;

(б) $\{(1, 2), (2, 1), (3, 3), (4, 5), (5, 4), (5, 5)\}$;

(в) $\{(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (1, 4), (4, 1), (4, 2), (2, 4)\}$;

(г) $\{(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (1, 2), (2, 1), (2, 4), (4, 2), (3, 5), (5, 3), (1, 4), (4, 1)\}$;

(д) $\{(a, b) : a \in A,\ b \in A$ и $a + b$ делится на $3\}$;

(е) $\{(a, b) : a \in A,\ b \in A$ и $a - b$ делится на $2\}$;

(ж) $\{(a, b) : a \in A,\ b \in A$ и $a + 3$ делится на $b\}$. $\circledast$

25. Пусть Π — множество всех прямых на евклидовой плоскости. Определить, будут ли следующие отношения на Π отношениями эквивалентности:

(а) параллельность прямых (будем считать, что прямая параллельна себе самой);

(б) перпендикулярность прямых. $\circledast$

26. Для каждого из следующих бинарных отношений над множеством положительных натуральных чисел определить, является ли оно рефлексивным, антирефлексивным, симметричным, антисимметричным, транзитивным:

(а) $D_1 = \{(x, y) : x$ делится на $y\}$;

(б) $D_2 = \{(x, y) : x$ делится на y или y делится на $x\}$;

(в) $D_3 = \{(x, y) : x$ не делится на $y\}$. $\circledast$

27. Для каждого из следующих трёх отношений R_i, $i = 1, 2, 3$, определённых на совокупности всех непустых подмножеств действительных (вещественных) чисел, определить, являются ли они рефлексивными, симметричными, антисимметричными, транзитивными, отношениями частичного порядка:

(а) $R_1 = \{(A, B)$: для любого $\varepsilon > 0$ существуют $a \in A$ и $b \in B$ такие, что $|a - b| \leqslant \varepsilon\}$;

(б) $R_2 = \{(A, B)$: для любых $a \in A$ и $\varepsilon > 0$ существует $b \in B$ такое, что $|a - b| \leqslant \varepsilon\}$;

(в) $R_3 = \{(A, B)$: для любых $a \in A$, $b \in B$ и $\varepsilon > 0$ существуют $a' \in A$ и $b' \in B$ такие, что $|a - b'| \leqslant \varepsilon$ и $|a' - b| \leqslant \varepsilon\}$. ⊛

28. Пусть Π — множество многоугольников на плоскости. Будут ли следующие отношения отношениями эквивалентности на Π:

(а) x и y возможно совместить;

(б) x и y подобны;

(в) x и y имеют одинаковый угол;

(г) x и y пересекаются;

(д) x и y имеют одинаковую площадь;

(е) x и y имеют общую вершину;

(ж) x и y равносоставлены. ⊛

29. Пусть W — множество слов алфавита Σ. Будут ли следующие отношения отношениями эквивалентности на W:

(а) x и y состоят из одних и тех же символов без учёта количества;

(б) x и y состоят из одних и тех же символов с учётом количества;

(в) xy имеет чётную длину;

(г) x и y имеют одинаковую длину;

(д) x и y имеют одинаковую длину и отличаются не более чем в одной позиции;

(е) x и y имеют хотя бы одну общую букву;

(ж) x и y начинаются одной и той же буквой. ⊛

30. Пусть $A = \{a_1, \ldots, a_m\}$ — произвольный конечный алфавит, то есть множество символов. Обозначим через A^n множество слов длины n в алфавите A (это обозначение согласовано с тем же обозначением декартовой степени A, так как степень A^n состоит из всех последовательностей элементов A длины n).

(а) Определим следующее отношение R_1 на словах из A^n. Пусть $v = a_{i_1} a_{i_2} \ldots a_{i_n}$, $w = a_{j_1} a_{j_2} \ldots a_{j_n}$. Тогда $(v, w) \in R_1$ тогда и только тогда, когда $i_k \leqslant j_k$ для всех k от 1 до n и $i_k < j_k$ для некоторого такого k, то есть номер каждой буквы слова v не больше номера той же буквы в слове w и хотя бы у одной из букв он меньше. Определить, является ли это отношение R_1 отношением частичного (линейного) порядка.

(б) Определим следующее отношение R_2 на словах из A^*. Пусть $v = a_{i_1}a_{i_2}\ldots a_{i_n}$, $w = a_{j_1}a_{j_2}\ldots a_{j_r}$. Тогда $(v, w) \in R_2$ тогда и только тогда, когда существует такое k в интервале от 1 до n, что $i_l = j_l$ при $l < k$ и $i_k < j_k$ или $n < r$ и первые n символов w совпадают со словом v. Определить, является ли это отношение R_2 отношением частичного (линейного) порядка. ⊛

Замечание 1. Определённое в пункте (а) отношение R_1 называется *отношением покоординатного порядка*, а отношение R_2 из пункта (б) — *отношением лексикографического порядка*. В соответствии с лексикографическим порядком упорядочены, например, слова в словарях и энциклопедиях.

31. Определить, какие из следующих утверждений выполняются для каждого частично упорядоченного множества:

(а) существует в точности один наибольший элемент;

(б) существует не более одного наибольшего элемента;

(в) существует в точности один максимальный элемент;

(г) существует не более одного максимального элемента;

(д) каждый наибольший элемент является максимальным;

(е) каждый максимальный элемент является наибольшим;

(ж) все наибольшие элементы попарно сравнимы;

(з) все максимальные элементы попарно сравнимы;

(и) различные максимальные элементы попарно несравнимы;

(к) среди максимальных элементов есть наибольший;

(л) если максимальный элемент только один, то он — наибольший. ⊛

32. Для каждой из следующих функций найти её область значений и определить, какие из них являются разнозначными, сюръективными, взаимно однозначными функциями.

(а) $f : \mathbb{R} \to \mathbb{R}$, $f(x) = 3x + 1$;

(б) $f : \mathbb{R} \to \mathbb{R}$, $f(x) = x^2 + 1$;

(в) $f : \mathbb{R} \to \mathbb{R}$, $f(x) = x^3 - 1$;

(г) $f : \mathbb{R} \to \mathbb{R}$, $f(x) = e^x$;

(д) $f : \mathbb{R} \to \mathbb{R}$, $f(x) = \sqrt{3x^2 + 1}$;

(е) $f : [-{}^{\pi}/2, {}^{\pi}/2] \to \mathbb{R}$, $f(x) = \sin x$;

(ж) $f : [0, \pi] \to \mathbb{R}$, $f(x) = \sin x$;

(з) $f : \mathbb{R} \to [-1, 1]$, $f(x) = \sin x$;

(и) $f : \mathbb{R} \to \mathbb{R}$, $f(x) = x^2 \sin x$. ⊛

33. Даны две функции $g : A \to B$ и $f : B \to C$. Пусть $g \circ f : A \to C$ — композиция этих функций, то есть $(g \circ f)(x) = f(g(x))$. Определить, какие из следующих утверждений верны.

(а) Если g является разнозначной функцией, то $g \circ f$ также является разнозначной.

(б) Если g и f являются сюръективными функциями, то $g \circ f$ также является сюръективной.

(в) Если g и f являются взаимно однозначными функциями, то $g \circ f$ также является взаимно однозначной.

(г) Если $g \circ f$ является разнозначной функцией, то f также является разнозначной.

(д) Если $g \circ f$ является разнозначной функцией, то g также является разнозначной.

(е) Если $g \circ f$ является сюръективной функцией, то f также является сюръективной. ⊛

34. Доказать следующие включения и равенства для образов и прообразов:

(а) $f[A \cap B] \subseteq f[A] \cap f[B]$;

(б) $f[A \cup B] = f[A] \cup f[B]$;

(в) $f[A \setminus B] \supseteq f[A] \setminus f[B]$;

(г) $f^{-1}[A \cap B] = f^{-1}[A] \cap f^{-1}[B]$;

(д) $f^{-1}[A \cup B] = f^{-1}[A] \cup f^{-1}[B]$;

(е) $f^{-1}[A \setminus B] = f^{-1}[A] \setminus f^{-1}[B]$.

Привести примеры, когда эти включения будут строгими. ⊛

35. Доказать, что функция f на множестве A является частичным порядком на A в том и только том случае, когда $f(f(x)) = f(x)$ для всех $x \in A$. ⊛

36. Пусть $f : A \to A$ — разнозначная функция, $A_0 = A$, $A_{i+1} = f[A_i]$ для $i \in \omega$, $B = \bigcap\limits_{i \in \omega} A_i$, а g — ограничение функции f на множество B. Доказать, что функция g является взаимно однозначной функцией на множестве B. ⊛

37. Доказать, что если множества A и B конечны, то для мощностей выполнены следующие равенства

(а) $|A \times B| = |A| \cdot |B|$;

(б) $|A \cup B| = |A| + |B| - |A \cap B|$;

(в) $|\{0,1\}^n| = 2^n$;

(г) $|\mathsf{P}(A)| = 2^{|A|}$. ⊛

38. Найти, сколько существует k-местных отношений на множестве мощности n. ⊛

39. Определить, сколько бинарных отношений существует на множестве мощности n:

(а) всего;

(б) рефлексивных;

(в) антирефлексивных;

(г) симметричных;

(д) антисимметричных;

(е) линейных порядков. ⊛

40. Доказать, что следующие множества счётны:

(а) множество всех точек евклидовой плоскости с целочисленными координатами;

(б) множество всех векторов размера $n > 0$ с неотрицательными целочисленными координатами, то есть множество

$$\omega^n = \{(k_1, k_2, \ldots, k_n) : k_i \in \omega, i = 1, 2, \ldots, n\};$$

(в) множество всех векторов с неотрицательными целочисленными координатами, то есть множество $\bigcup\limits_{n=1}^{\infty} \omega^n$;

(г) множество всех слов в алфавите с m символами;

(д) множество всех рациональных чисел;

(е) множество всех многочленов с целыми коэффициентами;

(ж) множество всех корней таких многочленов. $\circledast$

2. Метод математической индукции

Пусть $U(n)$ — это некоторое утверждение, однозначно зависящее от целочисленного параметра n. Доказательство утверждения «для всех натуральных n, начиная с n_0, выполняется $U(n)$» по индукции в простейшем случае включает два этапа:

1) Базис индукции состоит в доказательстве утверждения $U(n_0)$ для некоторого начального значения n_0.
2) Шаг индукции состоит в предположении справедливости $U(n)$ при $n = k \geqslant n_0$ и доказательстве из этого предположения справедливости утверждения $U(k + 1)$.

В более сложных ситуациях базис может содержать не один, а несколько случаев, например, $n = 0$ и $n = 1$ для чисел Фибоначчи.

Индукционный шаг может тоже принимать разные формы, например: если утверждение $U(k)$ верно для всех k меньших n, то $U(n)$ верно для n.

Другой вариант — обратная индукция, когда утверждение доказывается от бóльших значений n к меньшим. В этом случае базис рассматривается для наибольшего возможного значения n, а индукционный шаг заключается в переходе от n к $n - 1$.

В любом случае главным условием применения метода индукции является то, что из базисных значений n с помощью многократных применений индукционного шага можно прийти к любому другому.

Задачи

41. Используя индукцию, доказать, что (а) $2n+1 \leqslant 2^n$ при $n \geqslant 3$;

(б) $n^2 \leqslant 2^n$ при $n \geqslant 4$. ⊛

42. Используя индукцию, доказать, что при натуральных n

(а) $7^n - 1$ делится на 6;

(б) $11^n - 6$ делится на 5;

(в) $67^n - 23^n$ делится на 4;

(г) $3^n + 7^n - 2$ делится на 8;

(д) $3^n + n^2$ делится на 4 при нечётных n. ⊛

43. Доказать, что $1^4+2^4+\ldots+n^4 = \dfrac{1}{30}n(n+1)(2n+1)(3n^2+3n-1)$. ⊛

44. Доказать, что $1\cdot 2 + 2\cdot 3 + \ldots + n(n+1) = \dfrac{n(n+1)(n+2)}{3}$. ⊛

45. Доказать, что $(1+2+3+\cdots+n)^2 = 1^3+2^3+3^3+\cdots+n^3$. ⊛

46. Доказать, что ⊛

$$1\cdot n + 2\cdot(n-1) + 3\cdot(n-2) + \cdots + n\cdot 1 = \frac{n(n+1)(n+2)}{6}.$$

47. Доказать, что $1+x+x^2+x^3+\ldots+x^n = \dfrac{x^{n+1}-1}{x-1}$ при $x \neq 1$. ⊛

48. Доказать формулу Муавра для натуральных степеней комплексного числа: если $z = r(\cos\varphi + i\sin\varphi)$, то $z^n = r^n(\cos n\varphi + i\sin n\varphi)$. Здесь i — мнимая единица, $i^2 = -1$, r и φ — действительные числа, $r \geqslant 0$. ⊛

49. Доказать, что n различных прямых на плоскости разбивают её на области, которые можно закрасить белой и чёрной красками так, что смежные области будут закрашены разными красками. ⊛

50. Доказать, что n различных прямых на плоскости, никакие две из которых не параллельны и никакие три из которых не пересекаются в одной точке, разбивают плоскость на $(n^2+n+2)/2$ областей. ⊛

51. Пусть в выпуклом n-угольнике, $n \geqslant 4$, никакие три диагонали не пересекаются в одной точке. Доказать, что в нём имеется $D(n) = \dfrac{(n-3)(n-2)(n-1)n}{24}$ точек пересечения диагоналей. ⊛

52. Доказать, что если стоимость некоторого письма больше или равна 6 рублей, то его можно точно оплатить, используя двух- и семирублёвые марки. ⊛

53. Найти ошибку в следующем доказательстве «по индукции» утверждения: для всех $n \geqslant 1$ справедливо неравенство $3^n > 3(n+1)+1$.

Пусть для некоторого $k \geqslant 1$ неравенство справедливо, то есть $3^k > 3(k+1)+1$, обозначим это неравенство $(*)$. Докажем, что оно верно и для $n = k+1$, то есть $3^{k+1} > 3(k+2)+1$. Для этого заметим, что для любого $k \geqslant 1$ верно неравенство $2 \cdot 3^k > 3$. Прибавив его левую и правую часть к соответствующим частям неравенства $(*)$, получим $3^k + 2 \cdot 3^k > 3(k+1)+1+3$ или $3^{k+1} > 3(k+2)+1$, что и требовалось.

Установить, при каких n справедливо неравенство $3^n > 3(n+1)+1$ на самом деле. ⊛

54. Найти ошибку в доказательстве «по индукции» следующего утверждения. Пусть n окружностей на плоскости расположены так, что любые две из них имеют ровно две общие точки, а для любой третьей окружности O одна из этих точек лежит внутри O, а другая — снаружи O. Тогда плоскость делится этими окружностями на $N(n) = 1 + n(n-1)$ частей.

Базис: если $n = 0$, то имеется только одна часть. Шаг: пусть для n окружностей утверждение верно, $N(n) = 1 + n(n-1)$. Тогда $(n+1)$-я окружность имеет $2n$ точек пересечения с предыдущими (с каждой по две). Следовательно, она проходит через $2n$ областей и делит каждую из них на две. Поэтому количество областей возрастает на $2n$ и мы получаем $N(n+1) = N(n) + 2n = 1 + n(n-1) + 2n = 1 + (n+1)n$.

Установить настоящее значение $N(n)$. ⊛

55. Пусть $n \geqslant 1$ сфер в пространстве расположены так, что любые три из них имеют ровно две общие точки, а для любой четвёртой сферы S одна из этих точек лежит внутри S, а другая — снаружи S. Доказать, что пространство делится этими сферами на $N(n) = 2n + \dfrac{n(n-1)(n-2)}{3}$ кусков. ⊛

56. Предположим, что некоторая колония бактерий $\mathcal{C}$ состоит из конечного количества особей, для каждой из которых указано натуральное число — максимальное количество делений, которое бактерия может пережить, то есть

$$\mathcal{C} = \{(b,t) : b \text{ — идентификатор бактерии,} \\ t \text{ — максимальное количество делений}\}.$$

На каждом шаге в жизни колонии происходит одно событие: выбирается какая-нибудь одна бактерия (b,t) и, если $t = 0$, то она удаляется из колонии («умирает»), а если $t > 0$, то эта бактерия заменяется на некоторое конечное множество бактерий («делится»), при этом все её потомки способны к делению менее t раз.

Доказать, что любая такая колония $\mathcal{C}$ в конце концов станет пустой («вымрет»).

Указание. Использовать индукцию по максимальному количеству делений T: $T = \max\{t : (b,t) \in \mathcal{C}\}$, а внутри — индукцию по количеству таких бактерий. ⊛

57. Числа Фибоначчи определяются следующими рекуррентными соотношениями: $F_0 = 0$, $F_1 = 1$, $F_{n+2} = F_{n+1} + F_n$. Доказать, что

(а) $F_n = \dfrac{1}{\sqrt{5}}\left[\left(\dfrac{1+\sqrt{5}}{2}\right)^n - \left(\dfrac{1-\sqrt{5}}{2}\right)^n\right]$;

(б) $F_1 + F_2 + F_3 + \dots + F_n = F_{n+2} - 1$;

(в) $n \cdot F_1 + (n-1) \cdot F_2 + (n-2) \cdot F_3 + \dots + 1 \cdot F_n = F_{n+4} - n - 3$ при $n \geqslant 1$;

(г) $F_{n-1}F_{n+1} = F_n^2 + (-1)^n$ для $n \geqslant 1$;

(д) числа F_n и F_{n+1} взаимно просты при $n \geqslant 1$;

(е) число F_n чётное тогда и только тогда, когда n делится на 3;

(ж) $F_{n+m} = F_{n-1}F_m + F_nF_{m+1}$ при $n \geqslant 1$. ⊛

58. Доказать, что при $n \geqslant 1$ ⊛

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}.$$

59. Доказать, что существует F_{n+2} последовательностей нулей и единиц длины n, в которых нет двух единиц подряд. ⊛

60. Задача о Ханойских башнях. Имеется три стержня. На один из них надето несколько дисков, диаметры которых убывают снизу вверх (получается детская пирамидка). Разрешается перекладывать верхний диск с любого стержня на любой другой, но при этом запрещено класть диск большего диаметра на диск меньшего диаметра. Требуется перенести всю пирамидку на другой стержень.

(а) Доказать, что n дисков можно переместить за $2^n - 1$ операций.

(б) Доказать, что n дисков нельзя переместить меньше чем за $2^n - 1$ операций.

(в) Легенда гласит, что в начале времён в храме города Бенарес бог Брахма создал три стержня и надел на один из них 64 диска. Монахи непрерывно переносят диски со стержня на стержень по вышеописанным правилам. Когда они перенесут все диски на другой стержень, наступит конец света. Используя результаты предыдущих пунктов, определить, следует ли ожидать конца света в ближайшем будущем. ⊛

61. Корректность алгоритма Евклида нахождения наибольшего общего делителя натуральных чисел основана на таком утверждении: НОД$\{x, y\}$ = НОД$\{x-y, y\}$ при $y \leqslant x$. Используя это же утверждение, доказать, что существуют целые a и b такие, что НОД$\{x, y\} = ax + by$. *Указание.* Использовать индукцию по $x + y$ для фиксированного $d =$ НОД$\{x, y\}$. ⊛

3. Элементы комбинаторики

Пусть X — множество «мест» (ящиков), Y — множество «объектов», $|X| = n$, $|Y| = m$.

Размещение с повторениями m объектов по n местам — это функция из X в Y, которая говорит для каждого места, какой объект на нём находится. Количество таких функций равно $F_m^n = m^n$. В частности, количество всех слов длины n в m-буквенном алфавите (размещение m возможных букв на n позиций в слове) равно m^n. Количество всех подмножеств множества X равно 2^n (на каждом месте размещаем 1, если это место входит в подмножество, или 0 в противном случае).

Размещение без повторений m объектов по n местам — это разнозначная функция из X в Y. Количество таких функций равно

$$A_m^n = m(m-1)\dots(m-n+1).$$

Если $m \geqslant n$, то $A_m^n = \dfrac{m!}{(m-n)!}$.

Перестановку n объектов можно рассматривать как размещение без повторений n объектов на n мест, количество всех перестановок n объектов равно $A_n^n = n!$.

Сочетание из m объектов по n — это n-элементное подмножество Y. Количество сочетаний из m объектов по n равно

$$C_m^n = \frac{A_m^n}{n!} = \frac{m!}{n!(m-n)!}.$$

Бином Ньютона:

$$(x+y)^m = \sum_{k=0}^{m} C_m^k x^k y^{m-k}.$$

П р и н ц и п в к л ю ч е н и я и и с к л ю ч е н и я. Если имеется m объектов, каждый из которых может обладать (или не обладать) свойствами $p_1, p_2, \ldots, p_n$, то

$$m' = m - \sum_{i=1}^{n} N\{p_i\} + \sum_{1 \leqslant i<j \leqslant n} N\{p_i, p_j\} - \sum_{1 \leqslant i<k<j \leqslant n} N\{p_i, p_k, p_j\} + \ldots,$$

где $N\{p_i, \ldots, p_j\}$ — количество объектов, которые обладают свойствами $p_i, \ldots, p_j$ (не обязательно только ими), а m' — количество объектов, не обладающих ни одним из свойств $p_1, p_2, \ldots, p_n$.

Задачи

62. Пусть X — множество из трёх элементов. Найти количество

(а) двухместных функций, которые можно определить на X;

(б) трёхместных функций, которые можно определить на X;

(в) трёхместных отношений, которые можно определить на X. ⊛

63. Пусть $A = \{a, b, c, d, e, f, g, h\}$. Сколько существует

(а) трёхэлементных подмножеств множества A?

(б) пятиэлементных подмножеств множества A, содержащих b?

(в) пятиэлементных подмножеств множества A, не содержащих b?

(г) пятиэлементных подмножеств множества A, содержащих c, но не содержащих f и h?

(д) подмножеств множества A, содержащих хотя бы три элемента?

(е) подмножеств множества A, содержащих не более шести элементов? ⊛

64. Сколько существует строго возрастающих последовательностей целых чисел, которые

(а) начинаются 1 и заканчиваются 7?

(б) начинаются 4 и заканчиваются 11?

(в) начинаются k и заканчиваются $k + m$? ⊛

65. Предположим, что имеется n попарно непересекающихся множеств A_1, A_2, …, A_n и $|A_i| = k_i$, $i = 1, \ldots, n$. Доказать, что количество множеств, содержащих не более чем по одному элементу из каждого A_i, равно $(k_1 + 1)(k_2 + 1) \cdots (k_n + 1)$. ⊛

66. Пусть $n = p_1^{k_1} p_2^{k_2} \cdots p_r^{k_r}$ — разложение n на простые сомножители. Используя предыдущую задачу, доказать, что количество различных делителей n равно $t(n) = \prod\limits_{i=1}^{r}(k_i + 1)$. Доказать, что n является квадратом тогда и только тогда, когда $t(n)$ нечётно. ⊛

67. Число e является суммой следующего ряда: $e = \sum\limits_{i=0}^{\infty} {}^1/_{i!}$. Пусть e_n — сумма первых n членов этого ряда: $e_n = \sum\limits_{i<n} {}^1/_{i!}$. Доказать, что A_n — общее количество размещений без повторений из n предметов по любому количеству мест m, равняется $e_n n!$. Сделать вывод, что при бóльших n оно приближённо равно $en!$. ⊛

68. Доказать тождества:

(а) $C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$;

(б) $nC_{n-1}^{k-1} = kC_n^k$;

(в) $C_n^k C_{n-k}^{m-k} = C_m^k C_n^m$. ⊛

69. Доказать с помощью математической индукции формулу бинома Ньютона. ⊛

70. Доказать, что

$$C_n^0 < C_n^1 < \ldots < C_n^{\lfloor n/2 \rfloor - 1} < C_n^{\lfloor n/2 \rfloor} = \\ = C_n^{\lceil n/2 \rceil} > C_n^{\lceil n/2 \rceil + 1} > \ldots > C_n^{n-1} > C_n^n,$$

где $\lfloor x \rfloor$ и $\lceil x \rceil$ означают округление числа x до ближайшего целого вниз и вверх соответственно. ⊛

71. Сколько имеется различных путей на плоскости из точки $(0, 0)$ в точку (x, y), проходящих через точки с целочисленными координатами, в которых каждый отрезок единичной длины идёт либо слева направо, либо снизу вверх? ⊛

72. (а) В разложении $(x + 2y)^7$ найти коэффициент при $x^3 y^4$.

(б) В разложении $(5x-y)^6$ найти коэффициент при x^3y^3. ⊛

73. В разложении $(1+x^2+x^5)^{15}$ найти коэффициент при x^{20}. ⊛

74. Доказать, что количество разбиений n-элементного множества на k подмножеств, первое из которых содержит n_1 элементов, второе — n_2 элементов, …, k-е — n_k элементов, равно $\dfrac{n!}{n_1!n_2!\dots n_k!}$. ⊛

75. Определить, сколько различных слов можно составить из букв слова АБРАКАДАБРА, если каждую букву нужно использовать в точности один раз. ⊛

76. Доказать равенство ⊛

$$(x_1+x_2+\dots+x_k)^n=\sum_{n_1+n_2+\dots+n_k=n}\frac{n!}{n_1!n_2!\dots n_k!}x_1^{n_1}x_2^{n_2}\dots x_k^{n_k}.$$

77. Найти количество раздач для каждой из следующих карточных игр.

(а) При игре в бридж колоду из 52 карт раздают четырём игрокам — каждому по 13 карт.

(б) При игре в преферанс колоду из 32 карт раздают трём игрокам — каждому по 10 карт, а оставшиеся 2 карты оставляют в прикупе. Порядок карт в прикупе имеет значение.

(в) При игре в «дурака» колоду из 36 карт раздают четырём игрокам — каждому по 6 карт, а оставшиеся 12 карт оставляют в прикупе. Порядок карт в прикупе имеет значение.

(г) Пусть в некоторой карточной игре колоду из N карт раздают k игрокам — каждому по r карт, а оставшиеся $m=N-kr$ карт оставляют в прикупе. Вывести формулы, определяющие, каким количеством способов можно произвести такую раздачу в случаях, когда (i) порядок карт в прикупе несущественен, (ii) порядок карт в прикупе имеет значение. ⊛

78. Преподаватель рассчитывает читать один и тот же курс в течение 20 лет. Чтобы не наскучить студентам, он решил рассказывать им каждый год три анекдота, причём этот набор из трёх анекдотов не должен повторяться. Каково минимальное количество анекдотов, которые он должен приготовить? ⊛

79. Известно, что из всех пятиэлементных подмножеств множества $\{1, 2, 3, \ldots, n\}$, $n \geqslant 7$, четвёртая часть содержит элемент 7. Определить, чему равно число n. ⊛

80. На острове живёт племя туземцев, у которых набор зубов во рту состоит максимально из 30 зубов. При этом на острове нет двух жителей с одинаковыми наборами зубов (наборы одинаковы, если каждый зуб у обоих либо одновременно присутствует, либо одновременно отсутствует). Может ли на этом острове быть больше жителей чем в

(а) Торжке? (в) Москве? (д) всём мире? ⊛
(б) Твери? (г) России?

81. Доказать тождество Коши: ⊛

$$C_{n+m}^{k} = \sum_{i=0}^{i=k} C_n^i C_m^{k-i}.$$

82. Доказать, что число C_n^k нечётно тогда и только тогда, когда $\rho(k) \subseteq \rho(n)$, где значение функции $\rho(x)$ — это множество двоичных разрядов, которые содержат единицы в двоичной записи числа x. Иначе говоря, если в двоичной записи k на каком-то разряде стоит единица, то и в двоичной записи n на этом разряде должна стоять единица.
Указание. Использовать индукцию по n, представить $n = 2^u + m$ для $1 \leqslant m \leqslant 2^u$ и применить тождество Коши. ⊛

83. Мультимножеством называется неупорядоченный набор элементов, каждый из которых может встречаться в нём сколько угодно раз. Два мультимножества равны, если каждый элемент встречается в них одно и тоже количество раз. Например, мультимножество $\{1, 2, 1, 3\}$ равно мультимножеству $\{3, 1, 1, 2\}$ и не равно мультимножеству $\{1, 2, 2, 3\}$.

Доказать, что количество способов, которыми можно породить k-элементное мультимножество, имея n попарно различных элементов и используя каждый из них сколько угодно раз, равно C_{n+k-1}^{k}. ⊛

84. Определить, сколько натуральных решений имеет уравнение $x+y+z+u=10$? ⊛

85. Сколько натуральных решений имеет уравнение $x_1+\dots+x_n=k$, если $x_i \geqslant m$ для всех $i=1,\dots,n$? ⊛

86. Сколько натуральных решений имеет уравнение $x_1+\dots+x_n=k$, если $k \geqslant (n-1)m$ и $x_i \leqslant m$ для всех $i=1,\dots,n$? ⊛

87. В парламенте некой страны имеется 401 место. Сколько имеется вариантов распределения мест в парламенте между тремя партиями, при которых ни одна из них не имеет абсолютного большинства (более половины мест)? ⊛

88. В кондитерском магазине продаются четыре сорта пирожных: заварные, песочные, «картошка» и бисквитные. Сколькими способами можно купить

(а) 6 пирожных?

(б) 7 пирожных?

(в) 7 пирожных, если должно быть куплено хотя бы одно пирожное каждого сорта? ⊛

89. Назовём два исхода первенства России по футболу *совпадающими в главном*, если в этих исходах совпадают обладатели золотых, серебряных и бронзовых медалей, а также две команды, покидающие премьер-лигу (то есть занявшие два последних места). Найти количество *различных в главном* исходов (напомним, что в первенстве участвуют 16 команд). ⊛

90. Сколько существует способов расположить n шаров, из них — m чёрных, остальные — белые, в один ряд, чтобы никакие два чёрных шара не оказались рядом? ⊛

91. Доказать тождество $\sum\limits_{2m\leqslant n+1} C_{n-m+1}^{m} = F_{n+2}$. ⊛

92. На книжной полке стоят n книг. Сколькими способами можно взять m книг, так чтобы не брать соседние книги и книги, стоящие на краях полки? ⊛

93. За круглым столом короля Артура сидят 12 рыцарей. Каждый из них враждует со своими соседями. Нужно выбрать 5 рыцарей, чтобы освободить принцессу. Сколькими способами это можно сделать так, чтобы среди выбранных рыцарей не оказалось врагов?

Решить эту задачу в случае, когда из n рыцарей за столом нужно выбрать k рыцарей. ⊛

94. Доказать принцип включения и исключения в теоретико-множественной форме: если $A_1, \dots, A_n$ — это конечные множества, то ⊛

$$\left|\bigcup_{i=1}^{n} A_i\right| = -\sum_{\substack{I\subseteq\{1,\dots,n\}\\ I\neq\varnothing}} (-1)^{|I|} \left|\bigcap_{i\in I} A_i\right|.$$

95. Сколько в первой сотне положительных натуральных чисел не делится ни на одно из чисел 2, 3, 5? А в первой тысяче чисел? ⊛

96. Определить, сколько целочисленных решений имеет следующая система: ⊛

$$\begin{cases} x + y + z = 11 \\ 0 \leqslant x \leqslant 4 \\ 0 \leqslant y \leqslant 3 \\ 0 \leqslant z \leqslant 7 \end{cases}$$

97. Найти количество перестановок из n элементов, при которых ни один элемент не остаётся в первоначальном положении. ⊛

98. Сколько существует перестановок букв $\{a, c, f, m, p, r, t, x\}$, если

(а) нет никаких ограничений;

(б) между a и c должны стоять две или три буквы;

(в) буквы a и c не должны быть разделены двумя или тремя буквами;

(г) первые четыре буквы должны быть выбраны из a, c, f и r;

(д) буквы a, c, f и r должны стоять рядом? ⊛

99. Две команды играют в волейбол до трёх побед. Сколько существует различных вариантов изменения счёта по партиям? ⊛

100. В олимпиаде по программированию участвуют n студентов. Три человека, занявшие призовые места, получают дипломы первой, второй и третьей степени, ещё пять человек получают одинаковые дипломы лауреатов. Сколько существует различных вариантов награждения? ⊛

101. n девушек водят хоровод. Сколькими различными способами они могут встать в круг? ⊛

102. Пять юношей и пять девушек купили 10 билетов в кино в одном ряду. Сколькими способами они могут занять места, если

(а) все юноши сядут рядом;

(б) два юноши сядут по краям;

(в) никакие два юноши не будут сидеть рядом;

(г) ни юноши, ни девушки не будут сидеть все вместе;

(д) один юноша и одна девушка всегда будут сидеть рядом;

(е) один юноша и одна девушка откажутся сесть рядом? ⊛

103. Доказать, что количество слов длины n в алфавите из k букв, в которых каждая буква встречается хоть один раз, равно ⊛

$$U^*(n,k) = \sum_{i=0}^{k-1} (-1)^i C_k^i (k-i)^n.$$

104. У заведующего лабораторией имеется пять сотрудников. Сколькими способами он может распределить среди них 8 различных заданий так, чтобы у каждого было хоть одно задание? ⊛

4. Логика высказываний и булевы функции

Пусть $\mathbb{B} = \{0, 1\}$. Единичный n-мерный куб — это $\mathbb{B}^n$, то есть множество всех двоичных последовательностей (векторов) длины n.

n-местная булева функция — это n-местная функция на множестве $\mathbb{B}$: $f : \mathbb{B}^n \to \mathbb{B}$. С помощью $\mathcal{P}_n$ обозначается множество всех n-местных булевых функций. При $n = 0$ множество $\mathcal{P}_0$ состоит из двух нульместных функций-констант: 0 и 1.

Каждую булеву функцию можно задать таблицей, указав значение этой функции для каждого набора аргументов, или в виде множества вершин куба $\mathbb{B}^n$, на которых значение функции равно 1. Наборы аргументов в такой таблице, как правило, располагают в лексикографическом порядке.

Пусть $\mathcal{B}$ — некоторое (конечное или бесконечное) множество булевых функций, $\boldsymbol{V} = \{x_1, x_2, \ldots\}$ — счётное множество пропозициональных переменных, то есть переменных, принимающих значения из $\mathbb{B}$. Определим по индукции множество префиксных булевых формул над $\mathcal{B}$ с переменными из $\boldsymbol{V}$. Одновременно будем определять числовую характеристику $\mathrm{depth}(\Phi)$ формулы Φ, называемую её глубиной.

- Базис индукции. Каждая переменная $x_i \in \boldsymbol{V}$ и каждая константа $c \in \mathcal{B}$ является формулой, глубина которой равна 0, то есть $\mathrm{depth}(x_i) = \mathrm{depth}(c) = 0$.
- Шаг индукции. Допустим, что $f^{(m)} \in \mathcal{B}$, $\Phi_1, \ldots, \Phi_m$ — формулы, $\max\{\mathrm{depth}(\Phi_i) : i = 1, \ldots, m\} = k$. Тогда строка $\Phi = f(\Phi_1, \ldots, \Phi_m)$ является формулой, её глубина $\mathrm{depth}(\Phi)$ равна $k + 1$.

Для формул над множеством функций

$$\mathcal{B}_e = \{0, 1, \neg, \wedge, \vee, \to, \oplus, \leftrightarrow, \uparrow, \downarrow\},$$

x_1	0	1	x_1	$\neg x_1$
0	0	1	0	1
1	0	1	1	0

Рис. 1: Булевы функции от одной переменной

x_1	x_2	$\wedge$	$\vee$	$\to$	$\oplus$	$\leftrightarrow$	$\uparrow$	$\downarrow$
0	0	0	0	1	0	1	1	1
0	1	0	1	1	1	0	1	0
1	0	0	1	0	1	0	1	0
1	1	1	1	1	0	1	0	0

Рис. 2: Основные булевы функции от двух переменных

представленных в таблицах на рис. 1 и 2, будем использовать инфиксную запись, в которой имя двухместной функции (логической связки) помещается между первым и вторым аргументами.

При инфиксной записи считается, что булевы функции имеют следующий приоритет в порядке убывания: $\neg, \wedge, \vee, \oplus, \to$. При этом операции $\neg$ и $\to$ являются правоассоциативными (скобки расставляются справа налево), а $\wedge, \vee, \oplus$ — левоассоциативными (скобки расставляются слева направо).

Формулой логики высказываний называется инфиксная формула над множеством булевых функций $\{\neg, \wedge, \vee, \to\}$. Функция, отображающая множество переменных $\boldsymbol{V}$ в множество $\mathbb{B}$, называется интерпретацией. Значением формулы Φ с пропозициональными переменными $x_1, \ldots, x_n$ в интерпретации I называется значение соответствующей булевой функции на аргументах $I(x_1)$, ..., $I(x_n)$. Если значение формулы равно 1, то говорят, что формула в интерпретации истинна, иначе — ложна.

Формула называется выполнимой, если при некоторых значениях входящих в неё переменных её значение равно 1. Формула называется тождественно истинной, если при любых значениях входящих в неё переменных её значение равно 1, тождественно ложной, если при любых значениях входящих в неё переменных её значение равно 0. Длина формулы — это количество входящих в неё символов функций и переменных.

Задачи

105. Доказать, что количество k-мерных граней в n-мерном кубе равно $F(n, k) = C_n^k \cdot 2^{n-k}$. Найти общее количество $F(n)$ граней n-мерного куба. ⊛

106. Пусть $\bar{a} = (a_1, a_2, \ldots, a_n)$ и $\bar{b} = (b_1, b_2, \ldots, b_n)$ — два набора (две точки) n-мерного единичного куба $\mathbb{B}^n$. Р а с с т о я н и е м Х э м м и н г а между $\bar{a}$ и $\bar{b}$ называется число $H(\bar{a}, \bar{b}) = \sum\limits_{i=1}^{n} |a_i - b_i|$. Доказать, что это расстояние удовлетворяет следующим свойствам:

(а) неотрицательность: $H(\bar{a}, \bar{b}) \geqslant 0$, при этом $H(\bar{a}, \bar{b}) = 0$ тогда и только тогда, когда $\bar{a} = \bar{b}$;

(б) симметричность: $H(\bar{a}, \bar{b}) = H(\bar{b}, \bar{a})$;

(в) неравенство треугольника: $H(\bar{a}, \bar{b}) \leqslant H(\bar{a}, \bar{c}) + H(\bar{c}, \bar{b})$. ⊛

107. Пусть $\bar{a}, \bar{b} \in \mathbb{B}^n$ — точки n-мерного единичного куба, $H(\bar{a}, \bar{b}) = k$. Доказать, что тогда

(а) самый короткий путь по рёбрам $\mathbb{B}^n$, соединяющий $\bar{a}$ и $\bar{b}$, содержит k рёбер;

(б) всего существует $k!$ различных таких путей. ⊛

108. Пусть в последовательности векторов

$$\bar{a}^{(0)} = (0, \ldots, 0), \bar{a}^{(1)}, \ldots, \bar{a}^{(n)} = (1, \ldots, 1) \in \mathbb{B}^n$$

каждый следующий элемент получен из предыдущего заменой одного из нулей на единицу. Доказать, что система уравнений $H(\bar{x}, \bar{a}^{(i)}) = k_i$, $i = 0, \ldots, n$, $k_i \in \omega$, имеет не более одного решения в $\mathbb{B}^n$. ⊛

109. Найти вершины единичного куба $\mathbb{B}^5$, которые отстоят от вершин $(0,0,0,0,0)$, $(1,0,0,0,0)$, $(1,1,0,0,0)$, $(1,1,1,0,0)$, $(1,1,1,1,0)$, $(1,1,1,1,1)$ на указанные расстояния соответственно:

(а) $2,1,2,3,2,3$; (в) $4,3,2,3,4,5$; (д) $2,3,2,3,2,3$;

(б) $3,4,5,4,3,2$; (г) $3,2,1,2,3,2$; (е) $2,3,1,3,2,3$. ⊛

110. Какие подмножества вершин $\mathbb{B}^4$ соответствуют следующим булевым функциям:

(а) $f_1(x_1, x_2, x_3, x_4) = 1 \Leftrightarrow x_1 = 0$;

(б) $f_2(x_1, x_2, x_3, x_4) = 1 \Leftrightarrow x_4 = 1$;

(в) $f_3(x_1, x_2, x_3, x_4) = 1 \Leftrightarrow x_1 + x_2 \geqslant x_3 + x_4$;

(г) $f_4(x_1, x_2, x_3, x_4) = 1 \Leftrightarrow x_1x_2 = 0$ или $x_3x_4 = 1$? ⊛

111. Построить таблицы значений для следующих булевых функций:

(а) $f_1(x_1, x_2, x_3) = 1 \Leftrightarrow x_1 + x_3 \geqslant x_2$;

(б) $f_2(x_1, x_2, x_3) = 1 \Leftrightarrow$ сумма $x_1 + x_2 + x_3$ чётна;

(в) $f_3(x_1, x_2, x_3) = 0 \Leftrightarrow (x_1 = x_2$ или $x_1 = x_3)$;

(г) $f_4(x_1, x_2, x_3) = \begin{cases} x_2, & \text{если } x_1 = 1; \\ x_3, & \text{иначе.} \end{cases}$ ⊛

112. Назовём *триггером* булеву функцию T с несколькими аргументами, для которой выполнено $T(q, 0, \ldots, 0) = q$ для любого $q \in \mathbb{B}$. Выписать всевозможные

(а) RS-триггеры $T^{(3)}$, для которых всегда выполнены равенства $T(q, 1, 0) = 0$ и $T(q, 0, 1) = 1$;

(б) D-триггеры $T^{(3)}$, для которых всегда выполнены равенства $T(q, d, 0) = q$ и $T(q, d, 1) = d$. ⊛

113. Для каждой из следующих формул определить её глубину и построить таблицу задаваемой ею функции:

(а) $\Psi_1 = ((x_1 \to \neg x_3) \vee (x_2 \oplus x_3))$;

(б) $\Psi_2 = (\neg(x_1 \uparrow x_2) \leftrightarrow (\neg x_1 \wedge x_2))$;

(в) $\Psi_3 = ((x_2 \oplus \neg x_3) \wedge ((x_1 \vee x_2) \to (x_1 \leftrightarrow \neg x_3)))$;

(г) $\Psi_4 = ((\neg x_1 \wedge x_2) \downarrow (x_1 \to \neg x_3))$;

(д) $\Psi_5 = (((x_1 \oplus x_2) \oplus x_3) \uparrow (\neg x_2 \downarrow x_3))$. ⊛

114. Определить, какие из следующих формул логики высказываний являются тождественно истинными:

(а) $\Psi_1 = ((x \to \neg y) \to ((\neg y \to z) \to (x \to z)))$;

(б) $\Psi_2 = ((x \to \neg y) \to ((x \to z) \to (\neg y \to z)))$;

(в) $\Psi_3 = ((x \to \neg y) \to ((x \to (\neg y \to z)) \to (x \to z)))$;

Цифра	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0

Продолжение				
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Рис. 3: Таблица из задачи 115.

(г) $\Psi_4 = ((\neg x \to y) \to ((\neg y \to z) \to (x \to z)))$. $\circledast$

115. В представленной на рис. 115 таблице показано кодирование десятичных цифр от 0 до 9 с помощью четырёх пропозициональных переменных $A, B, C, D \in \mathbb{B}$. Не перечисленные в таблице варианты значений A, B, C и D являются ошибочными кодами. Какие из следующих булевых формул задают множество всех ошибочных кодов:

(а) $(\neg A \to (B \wedge C))$;
(б) $((A \wedge B) \vee (C \wedge D))$;
(в) $((A \wedge B) \vee (A \oplus C))$;
(г) $((A \wedge \neg B) \vee (C \wedge D))$;
(д) $((A \wedge B) \vee (A \wedge C))$;
(е) $(A \wedge (B \oplus C))$? $\circledast$

116. Используя таблицу кодов из предыдущей задачи и предполагая, что код не является ошибочным, построить формулу с четырьмя переменными A, B, C и D, которая будет иметь значение 1 тогда и только тогда, когда кодируемое число

(а) равно 6;
(б) делится на 4;
(в) делится на 3;
(г) является простым;
(д) является составным;
(е) является числом Фибоначчи. $\circledast$

117. Переменные (A_1, B_1, C_1, D_1) и (A_2, B_2, C_2, D_2) кодируют числа x_1 и x_2 соответственно (см. задачу 115). Предполагая, что оба кода не являются ошибочными, записать формулу, которая будет иметь значение 1 тогда и только тогда, когда (а) $x_1 = x_2$; (б) $x_1 < x_2$; (в) $x_1 \leqslant x_2$. ⊛

118. Переменные (C_1, D_1) и (C_2, D_2) кодируют числа x_1 и x_2 соответственно из множества $\{0, 1, 2, 3\}$ (см. задачу 115). Записать формулы p_A, p_B, p_C и p_D, значения которых кодируют произведение этих чисел. ⊛

119. Какие из следующих условий можно выразить булевыми формулами от переменных x_1, x_2, x_3, x_4, использующими лишь логические связки $\wedge$ и $\vee$?

(а) Не менее двух переменных из x_1, x_2, x_3, x_4 имеют значение 1.

(б) Не все из переменных из x_1, x_2, x_3, x_4 имеют значение 0.

(в) Нечётное количество переменных из x_1, x_2, x_3, x_4 имеют значение 1. ⊛

120. Назовём наборы $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{B}^n$ и $\beta = (\beta_1, \ldots, \beta_n) \in \mathbb{B}^n$ *соседними*, если они находятся в соседних строках таблицы для функции от n переменных, то есть представляют двоичные записи чисел i_α и i_β, для которых $|i_\alpha - i_\beta| = 1$.

Найти количество функций в $\mathcal{P}_n$, которые на каждой паре соседних наборов принимают

(а) одинаковые значения;

(б) разные значения. ⊛

121. Найти количество n-местных булевых функций, принимающих значение 1 ровно на k наборах. ⊛

122. Найти количество функций в $\mathcal{P}_n$, которые меняют своё значение при изменении любого аргумента. ⊛

123. Найти количество функций в $\mathcal{P}_n$, коммутативных по любой паре аргументов, то есть значение которых не меняется при перестановке произвольных аргументов. ⊛

124. Назовём наборы $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{B}^n$ и $\beta = (\beta_1, \ldots, \beta_n) \in \mathbb{B}^n$ *противоположными*, если $\alpha_i + \beta_i = 1$ для всякого i (иначе говоря, $\alpha_i = 1$ в том и только том случае, когда $\beta_i = 0$).

Найти количество функций в $\mathcal{P}_n$, которые на каждой паре противоположных наборов принимают разные значения. ⊛

125. Пусть $n = 2k$. Назовём набор $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{B}^n$ *парным*, если $\alpha_i = \alpha_{k+i}$ для всех $i = 1, \ldots, k$, то есть $\alpha = \alpha'\alpha'$ для некоторого набора α' размера k. Найти количество функций в $\mathcal{P}_n$, которые на всех парных наборах принимают одинаковое значение. ⊛

126. Определить, сколько всего трёхместных булевых функций можно построить, используя не более одной связки из таблиц 1 и 2 на стр. 36 и при том не более одного раза. ⊛

127. Пусть F — конечное множество булевых функций, ε — положительная, сколь угодно малая константа. Доказать, что при достаточно больши́х n формулами над F с количеством символов менее $2^n/n$ можно представить не более $2^{2^n/n^{1-\varepsilon}}$ булевых функций. Таким образом, почти все булевы функции представляются только «длинными» формулами. ⊛

128. Аргумент x_i булевой функции f называется *фиктивным*, если значение f никогда не меняется при изменении значения x_i. Найти количество n-местных булевых функций, не имеющих фиктивных аргументов, для $n = 2, 3, 4$. ⊛

129. Индукцией по построению формулы Φ доказать неравенство

$$\operatorname{depth}(\Phi)^x_\Psi \leqslant \operatorname{depth}\Phi + \operatorname{depth}\Psi.$$

Привести пример, когда неравенство будет строгим. ⊛

130. Администратор базы данных обнаружил, что одна или несколько из трёх записей его базы A, B и C ошибочна. Он установил, что

(а) если запись B корректна, то A ошибочна;

(б) хотя бы одна запись из пары B, C корректна и хотя бы одна запись из пары A, C корректна;

(в) если A ошибочна, то хотя одна из записей B или C корректна (но не обе вместе).

Описать знания администратора в виде формулы логики высказываний. Может ли он сделать вывод, что запись B ошибочна? Можно ли достоверно утверждать, что ошибочная запись единственна? ⊛

131. Программист Пётр использовал в своей программе на языке C три целочисленные переменные x, y, z. В определённом месте программы он поместил условный оператор:

```
if(x * y >= 0 || x * z >= 0) x = 1; else x = 2;
```

Проанализировав свою программу, Пётр установил, что перед выполнением этого оператора выполнены следующие условия:

(а) если $z < 0$, то $x < 0$ или $y \geqslant 0$;

(б) $x \geqslant 0$ или $y < 0$;

(в) если $y < 0$, то хотя бы одна из переменных x или z отрицательна, но не обе вместе.

Описать знания Петра в виде формулы логики высказываний. Может ли он оптимизировать программу, заменив указанный условный оператор на присваивание `x = 1;` или на присваивание `x = 2;`? Если «да», то на какое? ⊛

132. Комитет состоит из пяти членов. Решения принимаются большинством голосов, однако, если председатель голосует «против», то решение не принимается. Построить формулу логики высказываний, зависящую от пяти пропозициональных переменных x_1, x_2, x_3, x_4, y (x_i означает, что i-й член комитета голосует «за», y означает, что председатель голосует «за»), значение которой равно 1 тогда и только тогда, когда в результате голосования решение принимается. ⊛

133. В каждом из следующих текстов выделить элементарные высказывания, обозначить их пропозициональными переменными и записать формулу логики высказываний, отражающую содержание текста.

(а) Миша решил сделать небольшой ремонт в квартире и поклеить новые обои или покрасить пол. Вместе с обоями нужно будет обновить ламинат, сменить двери и стеклопакет в окне. После покраски пола нужно установить натяжной потолок, новую люстру и мебель. Если обновить ламинат, то тоже придётся покупать новую мебель. Старые обои нельзя оставлять со старой мебелью, а старую мебель — со старой люстрой.

(б) Если будет мороз, то будет солнечно или при пасмурной погоде пойдёт снег, а если мороза не будет, то будет солнечно или при пасмурной погоде пойдёт дождь. В случае дождя или снега будет сыро, а в случае дождя — ещё и грязно. Если будет солнечно, то не будет ни сырости, ни грязи.

(в) Когда Вова дома, то его кошка ест или спит, а если Вовы нет, то она ещё и играет. Если кошка ест, то нужно купить корм. Если кошка играет, то придётся покупать новую мебель или новые обои. За кормом нужно идти в зоомагазин «Всё для кошки», а за мебелью или обоями — в хозяйственный магазин «Всё для дома». В любой из магазинов придётся ехать на такси. Вова не поедет на такси в магазин «Всё для дома». ⊛

134. Детектив Ш. Холмс подозревает в совершении преступления трёх лиц: A, B и C. Он установил, что

(а) если B преступник, то и C является преступником;

(б) кто-то из пары A, C является преступником, но не оба вместе;

(в) если C не преступник, то и A не преступник.

Описать знания Ш. Холмса в виде формулы логики высказываний и построить таблицу её значений. Может ли он сделать вывод, что C является преступником? Можно ли достоверно утверждать, что преступник действовал в одиночку? ⊛

135. Детектив Э. Пуаро подозревает в совершении преступления трёх лиц: A, B и C. Они дали следующие показания:

- A: если B преступник, то C невиновен;
- B: если A виновен, то и C является преступником;
- C: A преступник.

Э. Пуаро установил, что (а) если A сказал правду, то B соврал, и (б) показания C ложны. Описать знания Э. Пуаро в виде формулы логики высказываний и построить таблицу её значений. Может ли он сделать вывод, что B является преступником? Мог ли преступник быть один? ⊛

136. Пусть во всех зоопарках, где есть львы и носороги, нет жирафов; во всех зоопарках, где есть носороги и нет жирафов, есть львы; наконец, во всех зоопарках, где есть львы и жирафы, есть и носороги. Как вы думаете, может ли существовать такой зоопарк, в котором есть львы, но нет ни жирафов, ни носорогов? ⊛

137. Анна, Беатриса и Вера как-то обнаружили, что все они в одинаковых джинсах. Известно, что у Анны есть джинсы с карманами, узкие джинсы и вылинявшие джинсы без карманов; у Беатрисы — джинсы без карманов и вылинявшие узкие джинсы с карманами. И наконец, Вера имеет джинсы-клёши и синие узкие джинсы с карманами. Как выглядят их одинаковые джинсы? ⊛

138. Один студент, плохо знавший английский язык, попал на стажировку в Англию. Ему надо было попасть в Манчестер. Он проголосовал на шоссе и остановил машину, где находились отец, мать и дочь, которые ответили ему на вопрос об их поездке. Каждый произнесённый ими ответ он перевёл двумя разными способами и не мог решить, каков смысл фразы на самом деле. Вот, что они говорили (второй возможный смысл указан в скобках).

Отец: мы отправляемся в Манчестер (мы едем из Ньюкасла).

Мать: мы не отправляемся в Манчестер, а едем из Ньюкасла (мы не остановились в Ливерпуле и едем из Ньюкасла).

Дочь: мы не едем из Ньюкасла (мы остановились в Ливерпуле).

Следует ли студенту садиться в эту машину? ⊛

139. Льюис Кэрролл, автор широко известной книги «Алиса в Стране чудес» (менее известно, что он был математиком), любил задавать следующую задачу из четырёх фраз: «Из двух одно: или злоумышленник уехал в экипаже, или свидетель ошибся. Если злоумышленник имел сообщника, то он уехал в экипаже. У злоумышленника не было ни сообщника, ни ключа; или у него был сообщник и был ключ. У злоумышленника был ключ». Какой вывод отсюда можно сделать? ⊛

140. Профессор экономики сформулировал студентам три правила:

(а) если инвестиции не увеличатся, то возрастут правительственные расходы или возникнет безработица;

(б) если правительственные расходы не возрастут, то налоги снизятся;

(в) если налоги снизятся и инвестиции увеличатся, то не возникнет безработица.

Оказалось, что в данный момент инвестиции не увеличились.

Могут ли студенты сделать вывод о том, что правительственные расходы возрастут? ⊛

5. Эквивалентность формул

Из формулы Φ с л е д у е т формула Ψ (это записывается в виде $\Phi \Rightarrow \Psi$), если формула Ψ истинна во всех интерпретациях, в которых истинна формула Φ.

Две формулы Φ и Ψ называются э к в и в а л е н т н ы м и, если они следуют друг из друга или, что то же самое, они имеют одинаковые значение во всех интерпретациях: $I(\Phi) = I(\Psi)$ для всех I. Обозначается эквивалентность формул с помощью $\Phi \equiv \Psi$.

Запись $(\Phi)^x_\Psi$ означает результат замены в формуле Φ всех вхождений пропозициональной переменной x на формулу Ψ. Если $\Phi \equiv \Psi$, то $(\Theta)^x_\Phi \equiv (\Theta)^x_\Psi$.

Для формул логики высказываний выполняются перечисленные далее основные эквивалентности.

1) Ассоциативность:

$$((\Phi \circ \Psi) \circ \Theta) \equiv (\Phi \circ (\Psi \circ \Theta)),$$

где $\circ \in \{\wedge, \vee, \oplus\}$.

2) Коммутативность:

$$(\Phi \circ \Psi) \equiv (\Psi \circ \Phi),$$

где $\circ \in \{\wedge, \vee, \oplus\}$.

3) Дистрибутивность:

$$((\Phi \vee \Psi) \wedge \Theta) \equiv ((\Phi \wedge \Theta) \vee (\Psi \wedge \Theta));$$
$$((\Phi \wedge \Psi) \vee \Theta) \equiv ((\Phi \vee \Theta) \wedge (\Psi \vee \Theta));$$
$$((\Phi \oplus \Psi) \wedge \Theta) \equiv ((\Phi \wedge \Theta) \oplus (\Psi \wedge \Theta)).$$

4) Двойное отрицание: $\neg\neg\Phi \equiv \Phi$.

5) Законы де Моргана:

$$\neg(\Phi \vee \Psi) \equiv (\neg\Phi \wedge \neg\Psi);$$
$$\neg(\Phi \wedge \Psi) \equiv (\neg\Phi \vee \neg\Psi).$$

6) Идемпотентность:

$$(\Phi \wedge \Phi) \equiv \Phi, \quad (\Phi \vee \Phi) \equiv \Phi.$$

7) 0-1-законы:

$$\begin{array}{rrr} (\Phi \wedge \neg\Phi) \equiv 0; & (\Phi \vee \neg\Phi) \equiv 1; & (\Phi \oplus \Phi) \equiv 0; \\ (\Phi \wedge 0) \equiv 0; & (\Phi \vee 0) \equiv \Phi; & (\Phi \oplus 0) \equiv \Phi; \\ (\Phi \wedge 1) \equiv \Phi; & (\Phi \vee 1) \equiv 1; & (\Phi \oplus 1) \equiv \neg\Phi. \end{array}$$

8) Представление импликации:

$$(\Phi \to \Psi) \equiv (\neg\Phi \vee \Psi), \quad \neg(\Phi \to \Psi) \equiv (\Phi \wedge \neg\Psi).$$

9) Представление сложения по модулю два:

$$(\Phi \oplus \Psi) \equiv ((\Phi \wedge \neg\Psi) \vee (\neg\Phi \wedge \Psi)) \equiv ((\Phi \vee \Psi) \wedge (\neg\Phi \vee \neg\Psi)).$$

10) Законы поглощения:

$$\Phi \vee (\Phi \wedge \Psi) \equiv \Phi; \tag{1}$$
$$(\Phi \wedge \Psi) \vee (\neg\Phi \wedge \Theta) \vee (\Psi \wedge \Theta) \equiv (\Phi \wedge \Psi) \vee (\neg\Phi \wedge \Theta) \tag{2}$$

Задачи

141. Доказать, что из формулы Φ следует формула Ψ тогда и только тогда, когда $\Phi \wedge \Psi \equiv \Phi$, тогда и только тогда, когда $\Phi \vee \Psi \equiv \Psi$. ⊛

142. Проверить все основные эквивалентности, приведённые в начале раздела, непосредственно построив истинностные таблицы для функций, представляемых их левыми и правыми частями.

143. Назовём логическим произведением формулу, имеющую вид $\Phi_1 \wedge \Phi_2 \wedge \ldots \wedge \Phi_n$. Её подформулы $\Phi_i, 1 \leqslant i \leqslant n$, будем называть сомножителями. Аналогично, логической суммой назовём формулу вида $\Phi_1 \vee \Phi_2 \vee \ldots \vee \Phi_n$. Её подформулы Φ_i, $1 \leqslant i \leqslant n$, будем называть слагаемыми.

Показать, что из основных тождеств можно вывести следующие правила преобразования логических произведений и сумм:

(а) если в логическом произведении хотя бы один из сомножителей равен 0, то и всё произведение равно 0;

(б) если в логической сумме хотя бы одно из слагаемых равно 1, то и вся сумма равна 1;

(в) если в логическом произведении $n \geqslant 2$ и есть сомножитель равный 1, то его можно вычеркнуть;

(г) если в логической сумме $n \geqslant 2$ и есть слагаемое равное 0, то его можно вычеркнуть. ⊛

144. Используя основные тождества, доказать эквивалентность следующих пар формул.

(а) $\neg(x \vee \neg y) \wedge (x \to \neg y)$ и $\neg x \wedge y$;

(б) $\neg((x \wedge \neg y) \to (\neg x \vee z))$ и $x \wedge \neg y \wedge \neg z$;

(в) $(x \oplus y) \to (x \wedge \neg y)$ и $(\neg x \wedge \neg y) \vee x$. ⊛

145. Вывести законы поглощения, используя предыдущие эквивалентности. ⊛

146. Доказать следующие эквивалентности:

(а) $\Phi \to 0 \equiv \neg\Phi$;

(б) $1 \to \Phi \equiv \Phi$;

(в) $(\Phi \to \Psi) \to \Psi \equiv \Phi \vee \Psi$;

(г) $\neg\Phi \to \Phi \equiv \Phi$;

(д) $(\Phi \to \Phi) \to \Phi \equiv \Phi$;

(е) $\Phi \leftrightarrow \Psi \equiv (\Phi \to \Psi) \wedge (\Psi \to \Phi)$;

(ж) $\Phi \vee (\neg\Phi \wedge \Psi) \equiv \Phi \vee \Psi$;

(з) $(\Phi \to \Psi) \oplus (\Psi \to \Phi) \equiv \Phi \oplus \Psi$.

(и) $\Phi \to (\Psi \vee \Theta) \equiv (\Phi \wedge \neg\Theta) \to \Psi$;

(к) $(\Phi \wedge \Theta) \to \Psi \equiv \Phi \to (\Psi \vee \neg\Theta)$. ⊛

147. Используя основные тождества, доказать тождественную истинность следующих формул:

(а) $(x \to (y \to x))$;

(б) $((x \wedge y) \to x)$;

(в) $(x \to (x \vee y))$;

(г) $((x \to \neg y) \to \neg(x \wedge y))$;

(д) $(((x \to y) \wedge (y \to z)) \to (x \to z))$;

(е) $(((x \to y) \vee (x \to z)) \to (x \to (y \vee z)))$;

(ж) $((x \to y) \to ((x \to \neg y) \to \neg x))$;

(з) $((x \to (y \to z)) \to ((x \to y) \to (x \to z)))$. ⊛

148. Пусть интерпретация J отличается от интерпретации I только тем, что $J(x) = I(\Theta)$. Индукцией по построению формулы Φ доказать, что $J(\Phi) = I\big((\Phi)^x_\Theta\big)$. ⊛

149. Доказать, что если $\Phi \equiv \Psi$, то $(\Phi)^x_\Theta \equiv (\Psi)^x_\Theta$. ⊛

150. Пусть формулы Φ_0 и Φ_1 получены из формулы Φ заменой переменной x_i на 0 и 1 соответственно. Доказать, что формула Φ эквивалентна любой из следующих:

(а) $(\neg x_i \wedge \Phi_0) \vee (x_i \wedge \Phi_1)$; (б) $(\neg x_i \to \Phi_0) \wedge (x_i \to \Phi_1)$. ⊛

151. Булева функция $f^*(x_1, \dots, x_n)$ называется *двойственной* к функции $f(x_1, \dots, x_n)$, если $f^*(x_1, \dots, x_n) = \neg f(\neg x_1, \dots, \neg x_n)$ для каждого набора значений переменных. Например, конъюнкция двойственна к дизъюнкции и наоборот.

(а) Доказать, что отношение двойственности симметрично.

(б) Пусть $f(y_1, \dots, y_m), g_1(x_1, \dots, x_n), \dots, g_m(x_1, \dots, x_n)$ — булевы функции и

$$h(x_1, \dots, x_n) = f(g_1(x_1, \dots, x_n), \dots, g_m(x_1, \dots, x_n)).$$

Установить следующий принцип двойственности: двойственная функция от суперпозиции функций равна суперпозиции двойственных функций: ⊛

$$h^*(x_1, \dots, x_n) = f^*(g_1^*(x_1, \dots, x_n), \dots, g_m^*(x_1, \dots, x_n)).$$

152. Построить двойственные функции для функций

(а) $\neg$, (б) $\to$, (в) $\oplus$, (г) $\uparrow$. ⊛

153. Построить двойственные функции для функций, заданных следующими формулами:

(а) $(x \to \neg(\neg y \oplus x \oplus z))$; (в) $((x \vee \neg y) \to (\neg x \downarrow y))$. ⊛

(б) $((x \uparrow y) \oplus (\neg x \to z))$;

154. (а) Доказать, что среди 20 формул от переменных x_1, x_2 всегда найдутся две эквивалентные.

(б) Верно ли, что среди 20 формул от переменных x_1, x_2 всегда найдутся три попарно эквивалентные?

(в) Пусть имеется k формул от n переменных $x_1, \dots, x_n$. Требуется выбрать несколько попарно эквивалентных формул. Какое количество таких формул можно гарантированно найти? ⊛

155. Назовём *импликативной* формулу Φ (и определяемую ей функцию), которая получена какой-либо расстановкой скобок в строке $x_1 \to x_2 \to \dots \to x_{n-1} \to x_n$. Доказать, что

(а) каждая импликативная формула принимает значение 1 как минимум на одном наборе;

(б) каждая импликативная формула принимает значение 0 как минимум на одном наборе;

(в) для каждой импликативной формулы с $n \geqslant 2$ переменными и каждого значения σ_k переменной x_k существует возможность расширить σ_k до набора значений $\bar{\sigma}$ всех переменных, на котором формула имеет значение 1;

(г) каждая импликативная функция не имеет фиктивных аргументов;

(д) никакие две разные импликативные формулы не эквивалентны. ⊛

156. Доказать, что функция f может быть получена из некоторой импликативной функции g подстановкой переменных:

$$f(x_1, \dots, x_n) = g(x_{i_1}, \dots, x_{i_m})$$

тогда и только тогда, когда f задаётся формулой вида $x_i \vee \Psi$ для некоторой переменной x_i. ⊛

6. Дизъюнктивные и конъюнктивные нормальные формы

Пусть $\boldsymbol{V} = \{x_1, \ldots, x_n\}$ — это конечное множество пропозициональных переменных. Введём для $x \in V$ и $\sigma \in \mathbb{B}$ такое обозначение:

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

Элементарная конъюнкция — это формула вида

$$x_{i_1}^{\sigma_1} \wedge x_{i_2}^{\sigma_2} \wedge \cdots \wedge x_{i_k}^{\sigma_k}.$$

Элементарная дизъюнкция — это формула вида

$$x_{i_1}^{\sigma_1} \vee x_{i_2}^{\sigma_2} \vee \cdots \vee x_{i_k}^{\sigma_k}.$$

Дизъюнктивная нормальная форма (сокращённо — ДНФ) $\mathcal{D}$ — это дизъюнкция элементарных конъюнкций: $\mathcal{D} = K_1 \vee K_2 \vee \ldots \vee K_r$, где K_j, $j = 1, \ldots, r$, — это элементарные конъюнкции. ДНФ $\mathcal{D}$ называется совершенной, если в каждую элементарную конъюнкцию K_j каждая переменная из $\boldsymbol{V}$ входит в точности один раз. Для каждой булевой функции f не тождественно равной нулю существует задающая её ДНФ:

$$\mathcal{D}_f = \bigvee_{f(\sigma_1, \ldots, \sigma_n) = 1} \bigwedge_{i=1}^{n} x_i^{\sigma_i}. \tag{3}$$

Конъюнктивная нормальная форма (сокращённо — КНФ) $\mathcal{C}$ — это конъюнкция элементарных дизъюнкций: $\mathcal{C} = D_1 \wedge D_2 \wedge \ldots \wedge D_r$, где D_j, $j = 1, \ldots, r$, — это элементарные дизъюнкции. КНФ $\mathcal{C}$ называется совершенной, если в каждую элементарную дизъюнкцию D_j каждая переменная из $\boldsymbol{V}$ входит в точности один раз. Для каждой булевой функции f не тождественно равной единице существует задающая её КНФ:

$$\mathcal{C}_f = \bigwedge_{f(\sigma_1,\ldots,\sigma_n)=0} \bigvee_{i=1}^{n} x_i^{\neg\sigma_i}. \tag{4}$$

Импликантом для формулы Ф (или булевой функции f, которая задаётся формулой Ф) называется выполнимая элементарная конъюнкция, из которой Ф следует. Импликант называется минимальным, если никакая его часть импликантом для Ф не является. Сокращённой ДНФ для формулы Ф называется дизъюнкция всех её минимальных импликантов. Она эквивалентна Ф и единственна с точностью до перестановок элементов конъюнкций и дизъюнкций.

Метод Блейка для построения сокращённой ДНФ $\mathcal{D}$ для формулы Ф заключается в следующем:

1) Построить для Ф совершенную ДНФ $\mathcal{D}_1$.
2) Для получения $\mathcal{D}_2$ применять к $\mathcal{D}_1$, сколько возможно, справа налево закон поглощения (2).
3) Для получения $\mathcal{D}_3$ применять к $\mathcal{D}_2$, сколько возможно, слева направо закон поглощения (1).
4) Для получения $\mathcal{D}$ исключить из $\mathcal{D}_3$ повторы элементарных конъюнкций.

Минимальная ДНФ для функции f — это эквивалентная ей ДНФ, имеющая минимальную длину.

Задачи

157. Доказать эквивалентности при $\sigma, \tau \in \mathbb{B}$, $\sigma \neq \tau$ ⊛

$$1^\sigma \equiv \sigma; \quad 0^\sigma \equiv \neg\sigma; \quad \sigma^\sigma \equiv 1; \quad \sigma^\tau \equiv 0. \tag{5}$$

158. Определить, сколько из n переменных $x_1, \ldots, x_n$ и их отрицаний можно составить (а) неэквивалентных элементарных конъюнкций, содержащих ровно k элементов и не содержащих одну и ту же переменную два или более раз;

(б) всего неэквивалентных элементарных конъюнкций. ⊛

159. Доказать равенства (3) и (4): для каждого набора значений аргументов $\sigma_1, \ldots, \sigma_n$ выполнено ⊛

$$f(\sigma_1, \ldots, \sigma_n) = \mathcal{D}_f(\sigma_1, \ldots, \sigma_n) = \mathcal{C}_f(\sigma_1, \ldots, \sigma_n).$$

160. Предложить процедуры для решения следующих задач:

(а) По произвольной элементарной конъюнкции построить эквивалентную ей совершенную ДНФ с заданным множеством переменных.

(б) По произвольной элементарной дизъюнкции построить эквивалентную ей совершенную КНФ с заданным множеством переменных. ⊛

161. Доказать, что для всех $k \leqslant n$ каждую булеву функцию $f \in \mathcal{P}_n$ можно представить в виде

$$f(x_1, \ldots, x_k, x_{k+1}, \ldots, x_n) = \\ = \bigvee_{(\sigma_1, \ldots, \sigma_k) \in \mathbb{B}^k} x_1^{\sigma_1} \wedge \ldots \wedge x_k^{\sigma_k} \wedge f(\sigma_1, \ldots, \sigma_k, x_{k+1}, \ldots, x_n).$$

Такое представление называется разложением f по $x_1, \ldots, x_k$. При $k = n$ из него получается совершенная ДНФ (3). ⊛

162. Предложить метод одновременного построения эквивалентных ДНФ и КНФ для произвольной формулы Φ логики высказываний, используя индукцию по построению Φ. ⊛

163. Используя основные эквивалентности, построить для следующих формул эквивалентные совершенные ДНФ и КНФ:

(а) $\Phi_1 = ((x \uparrow y) \vee (z \oplus x)) \to \neg y$;

(б) $\Phi_2 = ((\neg x \oplus z) \to (x \vee y)) \wedge (\neg x \to \neg y)$;

(в) $\Phi_3 = (((z \vee x) \oplus y) \vee (x \to \neg z)) \wedge z$. ⊛

164. Преобразовать следующие ДНФ в эквивалентные совершенные ДНФ:

(а) $(x \wedge y) \vee (\neg x \wedge z) \vee (x \wedge z)$;

(б) $(\neg x \wedge y) \vee z \vee (x \wedge \neg z)$;

(в) $x \vee (\neg x \wedge z) \vee (y \wedge z) \vee (\neg x \wedge \neg y)$. ⊛

165. Преобразовать следующие КНФ в эквивалентные совершенные КНФ:

(а) $(x \vee y) \wedge (\neg x \vee z) \wedge (x \vee \neg z)$;

(б) $\neg x \wedge (x \vee y) \wedge z \wedge (\neg x \vee \neg z)$;

(в) $(\neg x \vee z) \wedge (y \vee z) \wedge (\neg x \vee \neg y)$. ⊛

166. Привести пример трёхместной булевой функции f, обладающей следующим свойством: если в совершенной ДНФ для f заменить конъюнкции на дизъюнкции, а дизъюнкции на конъюнкции, то получится совершенная КНФ для f. ⊛

167. С помощью основных эквивалентностей преобразовать следующие ДНФ в КНФ:

(а) $(x \wedge y) \vee (\neg x \wedge y \wedge z) \vee (x \wedge z)$;

(б) $(\neg x \wedge \neg y \wedge z) \vee (\neg x \wedge y \wedge z) \vee (x \wedge y \wedge \neg z)$;

(в) $x \vee (\neg x \wedge z) \vee (\neg x \wedge y \wedge z) \vee (\neg x \wedge \neg y \wedge \neg z)$. ⊛

168. Функция $f(x, y, z)$, задана следующей последовательностью восьми нулей и единиц при лексикографическом упорядочении аргументов: $f = (1011\,1010)$. Определить, какие из следующих элементарных конъюнкций являются импликантами и какие из них — минимальными импликантами для функции f:

(а) $\neg x \wedge y \wedge z$; (в) $x \wedge \neg y$; (д) $\neg y$;

(б) $\neg z$; (г) $\neg x \wedge y$; (е) $x \wedge \neg z$. ⊛

169. Доказать, что минимальная ДНФ состоит из минимальных импликантов. ⊛

170. Доказать, что совершенная, сокращённая и минимальная ДНФ для функции $\mathrm{odd}(x_1, \ldots, x_n) = x_1 \oplus \cdots \oplus x_n$ совпадают и имеют 2^{n-1} элементарных конъюнкций длины n. ⊛

171. Доказать, что сокращённая ДНФ для функции odd из задачи 170 на предыдущей странице состоит из наибольшего количества элементарных дизъюнкций среди функций с n переменными. ⊛

172. Наборы значений трёх аргументов x, y и z булевых функций упорядочены лексикографически. Их значения задаются следующими последовательностями восьми нулей и единиц. Определить для каждой из следующих совершенные ДНФ и КНФ и сокращённую ДНФ:

(а) $f = (1011\,0011)$;
(б) $f = (0011\,1001)$;
(в) $f = (1110\,1011)$;
(г) $f = (0011\,1011)$;
(д) $f = (0001\,0111)$;
(е) $f = (0111\,0101)$. ⊛

173. Найти эквивалентные сокращённые ДНФ и доказать эквивалентность следующих пар формул:

(а) $\Phi = (((\neg x \wedge \neg y) \to \neg z) \wedge (x \to y))$, $\Psi = ((1 \oplus y) \to (\neg x \wedge (1 \oplus z)))$;

(б) $\Phi = (\neg((x_1 \to x_2) \vee \neg(x_2 \to x_1)) \wedge x_3)$, $\Psi = \neg((x_1 \wedge x_3) \to x_2)$;

(в) $\Phi = \neg(\neg x \wedge y \wedge \neg z) \to ((y \oplus 1) \wedge ((x \oplus 1) \to \neg(\neg u \vee z)))$,
$\Psi = (\neg x \vee y) \to ((\neg u \vee y \vee z) \to (\neg(x \vee \neg y) \wedge \neg z))$;

(г) $\Phi = (\neg(x \to (\neg y \to (x \wedge \neg z))) \wedge (z \vee \neg(x \wedge y)))$,
$\Psi = ((x \wedge z) \oplus (x \wedge y \wedge z))$;

(д) $\Phi = (((x \wedge y) \to \neg z) \wedge (\neg x \to \neg y))$, $\Psi = (y \to (x \wedge (z \oplus 1)))$;

(е) $\Phi = (((x \vee y) \to \neg z) \wedge ((x \wedge z) \to y))$, $\Psi = (z \to ((x \oplus 1) \wedge \neg y))$. ⊛

174. Допустим, ДНФ $\mathcal{D}$ не содержит отрицаний и к ней не применимы законы поглощения. Доказать, что $\mathcal{D}$ является сокращённой ДНФ. ⊛

175. Привести пример ДНФ $\mathcal{D}$ и минимального импликанта C для неё таких, что C не является частью какой-то конъюнкции из $\mathcal{D}$. ⊛

7. Многочлены Жегалкина

Значение формулы $x \wedge y$ равно произведению значений x и y, поэтому вместо конъюнкции $\wedge$ можно писать обычное умножение: xy вместо $x \wedge y$.

Многочлен Жегалкина — это формула вида $K_1 \oplus \cdots \oplus K_n$, где каждая K_i, $i = 1, \ldots, n$, является нулём, единицей или произведением какого-то количества переменных.

Многочлен Жегалкина называется приведённым, если он равен 0 либо все его слагаемые не равны 0, не содержат повторяющихся множителей и сами попарно различны (с точностью до перестановки сомножителей).

При построении многочлена Жегалкина для произвольной формулы можно применять следующие эквивалентности:

$$\neg x \equiv (x \oplus 1), \tag{6}$$

$$(x_1 \wedge x_2) \equiv (x_1 x_2), \tag{7}$$

$$(x_1 \vee x_2) \equiv (x_1 x_2 \oplus x_1 \oplus x_2), \tag{8}$$

$$(x_1 \oplus x_2)(x_3 \oplus x_4) \equiv (x_1 x_3 \oplus x_1 x_4 \oplus x_2 x_3 \oplus x_2 x_4). \tag{9}$$

Пусть $K_i = \bigwedge\limits_{\sigma_i^j = 1} x_j$, где $i = \sigma_i^1 \sigma_i^2 \ldots \sigma_i^n$ — двоичное представление натурального числа i, $i = 0, \ldots, 2^n - 1$. Тогда многочлен Жегалкина для n-местной булевой функции f можно записать в виде:

$$f(x_1, \ldots, x_n) = \alpha_0 K_0 \oplus \alpha_1 K_1 \oplus \alpha_2 K_2 \oplus \ldots \oplus \alpha_{2^n - 1} K_{2^n - 1}, \tag{10}$$

где $\alpha_i \in \mathbb{B}$ для $i = 0, \ldots, 2^n - 1$. Метод неопределённых коэффициентов заключается в подстановке в это равенство всевозможных значений переменных и решении получившейся треугольной системы линейных уравнений относительно α_i.

Другой, табличный, способ нахождения коэффициентов многочлена Жегалкина состоит в следующем. Пусть по-прежнему $i = \sigma_i^1 \sigma_i^2 \ldots \sigma_i^n$ —

двоичное представление натурального числа i, $i = 0, \dots, 2^n - 1$. При помощи f_i обозначим значение булевой функции $f(x_1, \dots, x_n)$ на наборе аргументов $(\sigma_i^1, \sigma_i^2, \dots, \sigma_i^n)$. Для функции f определим булеву треугольную матрицу J^f размера $2^n \times 2^n$:

- Нулевая строка содержит значения f_i: $J_{0,i} = f_i$.
- Каждая следующая строка содержит суммы по модулю два чисел сверху от текущего и следующего за ним: $J^f_{i+1,\ell} = J^f_{i,\ell} \oplus J^f_{i,\ell+1}$ для $\ell = 0, \dots, 2^n - i - 1$.

Тогда нулевой столбец этой матрицы даст упорядоченный по i набор коэффициентов α_i для многочлена Жегалкина функции f.

Задачи

176. Доказать эквивалентности (6)–(9). ⊛

177. Используя основные эквивалентности и тождества (6)–(9), найти эквивалентные приведённые многочлены Жегалкина и доказать эквивалентность следующих пар формул:

(а) $\Phi = ((z \wedge (x \to y)) \vee \neg(\neg x \to z))$, $\Psi = (x \to (y \wedge z))$;

(б) $\Phi = \neg(x \to (y \wedge z)) \wedge (\neg y \vee \neg x)$,
$\Psi = (\neg(x \to y) \vee z) \wedge \neg((\neg x \wedge z) \vee (y \wedge z))$;

(в) $\Phi = (((y \wedge z) \to \neg(x \vee z)) \wedge \neg(\neg y \wedge z \wedge x))$, $\Psi = (z \to (\neg y \wedge \neg x))$;

(г) $\Phi = \neg(((x \to y) \vee \neg z) \wedge x)$, $\Psi = (\neg(z \to y) \vee \neg x)$;

(д) $\Phi = (\neg((x \to y) \vee \neg(y \to x)) \wedge z)$, $\Psi = \neg((x \wedge z) \to y)$. ⊛

178. При лексикографическом упорядочении наборов аргументов значения булевой функции $f(x_1, x_2, x_3)$ образуют последовательность (1001 0001). Выписать в явном виде систему уравнений для нахождения многочлена Жегалкина методом неопределённых коэффициентов, найти её решение, построить многочлен Жегалкина для f. ⊛

179. Найти многочлены Жегалкина (методом неопределённых коэффициентов и с помощью матрицы J) для следующих функций $f(x, y, z)$. Считаем, что наборы аргументов функций упорядочены лексикографически и значения на них задаются последовательностью 8 нулей и единиц:

(а) $f = (0010\,1100)$;
(б) $f = (1110\,1100)$;
(в) $f = (1100\,0011)$;
(г) $f = (0110\,1011)$. ⊛

180. Найти методом неопределённых коэффициентов многочлены Жегалкина для следующих функций:

(а) $f(x, y, z) = (x \uparrow y) \to (\neg y \vee z)$;

(б) значения функции $g(x, y, z, u)$ задаются следующей последовательностью нулей и единиц при лексикографическом упорядочении аргументов: $(1010\,1100\,0110\,1101)$;

(в) $h(x, y, z) = (x \wedge \neg y \wedge \neg z) \vee (\neg x \to (y \wedge z \wedge \neg u))$. ⊛

181. Доказать, что для булевой функции $f(x_1, x_2, \ldots, x_n)$ многочлен Жегалкина можно получить, раскрыв скобки в следующей формуле и упростив её: ⊛

$$p_f(x_1, x_2, \ldots, x_n) = \\ = \bigoplus_{f(\sigma_1, \sigma_2, \ldots, \sigma_n) = 1} (x_1 \oplus \sigma_1 \oplus 1)(x_2 \oplus \sigma_2 \oplus 1) \ldots (x_n \oplus \sigma_n \oplus 1).$$

182. Используя предыдущую задачу, найти многочлены Жегалкина для следующих функций. Значения функций $f(x, y, z)$ и $g(x, y, z)$ заданы последовательностями нулей и единиц при лексикографическом упорядочении аргументов:

(а) $f = (0110\,1000)$;

(б) $g = (1010\,1100)$;

(в) $h(x, y, z) = (x \wedge \neg y \wedge z) \to (y \wedge z)$. ⊛

183. Доказать, что многочлен Жегалкина тождественно равен нулю тогда и только тогда, когда все его коэффициенты α_i в представлении (10) равны нулю. ⊛

184. *Длиной* многочлена Жегалкина называется количество его слагаемых. Например, многочлен $p(x_1, x_2) = 1 \oplus x_1 \oplus x_1 x_2$ имеет длину три. Определить, сколько существует различных многочленов Жегалкина от n переменных длины k, которые одновременно обращаются в ноль на наборах $(0, 0, \ldots, 0)$ и $(1, 1, \ldots, 1)$, состоящих из одних нулей и единиц, соответственно. ⊛

185. Пусть приведённый многочлен Жегалкина $p(x_1, \ldots, x_n)$ принимает значение 1 в точности на одном наборе $(\sigma_1, \ldots, \sigma_n)$. Найти его длину. ⊛

186. Найти булеву функцию от n переменных, у которой приведённый многочлен Жегалкина имеет наибольшую длину. ⊛

187. Что произойдёт с булевой функцией, если в многочлене Жегалкина (10) все коэффициенты изменить на противоположные? ⊛

188. Пусть приведённый многочлен Жегалкина $p(x_1, \ldots, x_n)$ принимает значение 1 в точности на двух противоположных наборах: $(\sigma_1, \ldots, \sigma_n)$ и $(\neg\sigma_1, \ldots, \neg\sigma_n)$. Найти его длину. ⊛

189. Доказать, что приведённый многочлен Жегалкина $p(x_1, \ldots, x_n)$ задаёт функцию, которая коммутативна по любой паре аргументов, тогда и только тогда, когда для любого $k = 0, \ldots, n$ многочлен или содержит всевозможные слагаемые с k множителями, или не содержит ни одного из них. ⊛

190. Пусть длина многочлена Жегалкина определена как в задаче 184 на противоположной странице. Индукцией по $n \geqslant 1$ доказать, что для всякого $k = 0, \ldots, 2^n$ существует многочлен Жегалкина $p(x_1, \ldots, x_n)$ длины не большей чем n, который обращается в единицу в точности на k наборах аргументов. ⊛

191. Пусть многочлен Жегалкина задаётся формулой

$$p_n(x_1, \ldots, x_n) = 1 \oplus x_1 \oplus x_1x_2 \oplus x_1x_2x_3 \oplus \ldots \oplus x_1x_2 \ldots x_n.$$

Доказать индукцией по $n \geqslant 1$, что этот многочлен обращается в единицу в точности на $\frac{1}{3}(2^{n+1} + (-1)^n)$ наборах аргументов. ⊛

192. Пусть ненулевой многочлен Жегалкина с n переменными не содержит слагаемых с более чем k переменными, $k \leqslant n$. Используя индукцию по n, доказать, что количество единиц среди значений этого многочлена не меньше 2^{n-k}. ⊛

8. Полные множества функций и теорема Поста

З а м ы к а н и е множества булевых функций F (обозначается с помощью $[F]$) — это множество всех функций, которые можно задать с помощью формул над F. Множество функций F называется з а м к н у т ы м, если $F = [F]$.

Основные замкнутые множества функций:

1) Множество $\mathcal{S}_0$ булевых функций, с о х р а н я ю щ и х н о л ь, то есть таких f, что $f(0, \ldots, 0) = 0$.

2) Множество $\mathcal{S}_1$ булевых функций, с о х р а н я ю щ и х е д и н и ц у, то есть таких f, что $f(1, \ldots, 1) = 1$.

3) Множество $\mathcal{S}$ с а м о д в о й с т в е н н ы х булевых функций, то есть таких f, что $\neg f(x_1, \ldots, x_n) = f(\neg x_1, \ldots, \neg x_n)$.

4) Множество $\mathcal{L}$ л и н е й н ы х булевых функций, то есть таких f, что $f(x_1, \ldots, x_n) = \alpha_0 \oplus \alpha_1 x_1 \oplus \cdots \oplus \alpha_n x_n$ для каких-то $\alpha_0, \alpha_1, \ldots, \alpha_n \in \mathbb{B}$.

5) Множество $\mathcal{M}$ м о н о т о н н ы х булевых функций, то есть таких f, что $f(\sigma_1, \ldots, \sigma_n) \geqslant f(\tau_1, \ldots, \tau_n)$ всегда, когда $\sigma_1 \geqslant \tau_1, \ldots, \sigma_n \geqslant \tau_n$.

Множество булевых функций F называется п о л н ы м, если формулами над ним можно задать любую булеву функцию, то есть $[F] = \mathcal{P}$.

Т е о р е м а П о с т а о п о л н о т е: множество булевых функций F является полным тогда и только тогда, когда оно не включено ни в один из классов $\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}, \mathcal{L}, \mathcal{M}$.

Полное множество булевых функций называется б а з и с о м, если при удалении из него любой функции оно становится неполным.

Задачи

193. Определить, какие из следующих формул задают самодвойственные функции:

(а) $\Phi_1 = x \vee (\neg y \wedge z)$;

(б) $\Phi_2 = (\neg x \wedge y) \vee (z \wedge \neg(x \oplus y))$;

(в) $\Phi_3 = (x \wedge \neg z) \vee (y \wedge \neg z) \vee (x \wedge y)$;

(г) $\Phi_4 = (\neg x \to (y \vee z)) \to (\neg y \wedge z)$. ⊛

194. Определить, какие из следующих формул задают монотонные функции:

(а) $\Phi_1 = (y \to \neg x) \to (y \wedge z)$;

(б) $\Phi_2 = (\neg(x \wedge y) \vee \neg z) \to (\neg x \to (y \wedge z))$;

(в) $\Phi_3 = (\neg x \to (y \wedge \neg z)) \to y$;

(г) $\Phi_4 = \neg z \to (y \wedge \neg x)$. ⊛

195. Доказать полноту множества $\{\downarrow\}$, включающего только стрелку Пирса, непосредственно выразив через неё отрицание, дизъюнкцию и конъюнкцию. ⊛

196. Доказать, что множество $\{\vee, \wedge, \to\}$ не является полным. Можно ли его сделать полным, добавив некоторую константу? ⊛

197. Определить принадлежность каждой из функций, представленных в таблице на рис. 4 на следующей странице, каждому из множеств $\mathcal{S}_0$, $\mathcal{S}_1$, $\mathcal{S}$, $\mathcal{L}$ и $\mathcal{M}$. ⊛

198. Используя результаты задачи 197, определить, какие из троек функций, представленных на рис. 4 на следующей странице, являются полными множествами. Имеются ли среди них полные множества из двух функций? Из одной функции? Какие из них будут базисами? ⊛

199. Какие функции следует удалить из множества $F = \{f, g, h\}$, чтобы оно стало базисом? ⊛

$$f(x,y) = x \vee y; \quad g(x,y) = x \to \neg y; \quad h(x,y) = x \oplus y.$$

200. Найти все базисы, которые можно получить, удаляя функции из множества $\{0, 1, \wedge, \vee, \to\}$. ⊛

x_1	x_2	x_3	f_1	f_2	f_3	f_4	f_5	f_6	f_7
0	0	0	1	1	0	0	0	0	1
0	0	1	0	1	0	1	0	0	1
0	1	0	0	0	1	1	1	0	0
0	1	1	1	0	0	0	1	1	0
1	0	0	1	0	1	1	0	1	1
1	0	1	0	0	0	0	0	0	1
1	1	0	0	1	1	0	1	0	0
1	1	1	0	1	1	1	1	0	0

Рис. 4: Функции из задачи 197.

x_1	x_2	x_3	f	g	h
0	0	0	1	1	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	0	0
1	0	0	0	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	1	0

Рис. 5: Функции f, g, h.

201. Найти все базисы, составленные из функций на рис. 5. Для каждого из них выразить с помощью функций базиса обе константы, отрицание $\neg$ и импликацию $\to$. ⊛

202. Выразить функции 0, 1, $\vee$, $\wedge$, $\leftrightarrow$ с помощью формул, построенных из функций полного множества $\{\neg, \to\}$. ⊛

203. Булевы функции заданы последовательностью их значений при лексикографическом упорядочении аргументов. Найти все базисы, которые можно составить из следующих функций.

(а) $f_1 = (0011\,1100)$; (д) $f_5 = (1110\,1000)$; (и) $f_9 = (0101\,0111)$;
(б) $f_2 = (1010\,0101)$; (е) $f_6 = (0111\,0001)$; (к) $f_{10} = (0001\,0101)$.
(в) $f_3 = (0110\,1001)$; (ж) $f_7 = (1001\,1011)$;
(г) $f_4 = (1001\,0110)$; (з) $f_8 = (1100\,1000)$;

Для каждого базиса показать, как с помощью его элементов построить константы, отрицание и конъюнкцию (или дизъюнкцию). ⊛

204. Определить, как по заданной ДНФ (КНФ) определить, будет ли она сохранять ноль или единицу, не вычисляя её значений. ⊛

205. Доказать, что замыканием множества $\{\wedge, \vee, \to\}$ является $\mathcal{S}_1$. Можно ли то же самое утверждать для множества $\{\wedge, \to\}$? ⊛

206. Доказать, что замыканием каждого из множеств $\{\leftrightarrow, \wedge\}$, $\{\leftrightarrow, \vee\}$, $\{\leftrightarrow, \to\}$ является $\mathcal{S}_1$. ⊛

207. Найти замыкание множества $\{\wedge, \oplus\}$. ⊛

208. Назовём функцию $f(x_1, \ldots, x_n)$ линейной по переменной x_i, если f можно представить в виде $f(x_1, \ldots, x_n) = g(x_1, \ldots, x_n) \oplus \alpha_i x_i$, где $g(x_1, \ldots, x_n)$ — некоторая функция, не зависящая от x_i, $\alpha_i \in \{0, 1\}$. Доказать, что булева функция $f(x_1, \ldots, x_n)$ является линейной тогда и только тогда, когда она линейна по всем переменным. ⊛

209. Доказать, что функция $f(x_1, \ldots, x_n)$ линейна по переменной x_i тогда и только тогда, когда при изменении значения переменной x_i значение функции либо всегда меняется (то есть для всех значений остальных переменных), либо никогда не меняется (то есть f от x_i не зависит). ⊛

210. Доказать, что замыканием множества $\{\leftrightarrow\}$ является $\mathcal{S}_1 \cap \mathcal{L}$. ⊛

211. Доказать, что для монотонных функций $f^{(n)} \in \mathcal{M}$ справедливо представление

$$f(x_1, \ldots, x_n) = (x_i \wedge f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)) \vee \\ \vee f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n),$$

для всех $i = 1, \ldots, n$. Вывести отсюда (индукцией по n), что для всякой монотонной функции, отличной от константы, существуют задающие её ДНФ и КНФ, не содержащие отрицаний переменных. ⊛

212. Доказать, что булева функция $f^{(n)}$ является монотонной тогда и только тогда, когда она монотонна по каждому аргументу: для всех $i = 1, \ldots, n$ и всех $\sigma_1, \ldots, \sigma_n \in \mathbb{B}$ выполнено неравенство ⊛

$$f(\sigma_1, \ldots, \sigma_{i-1}, 0, \sigma_{i+1}, \ldots, \sigma_n) \leqslant f(\sigma_1, \ldots, \sigma_{i-1}, 1, \sigma_{i+1}, \ldots, \sigma_n).$$

213. Доказать, что булева функция $f(x_1, \ldots, x_n)$ отличная от константы является монотонной тогда и только тогда, когда её сокращённая ДНФ не содержит отрицаний. ⊛

214. Построить все монотонные функции вида $f(x, y, z)$. ⊛

215. Определить, что будет замыканием множества $\{\wedge, \vee\}$. ⊛

216. Доказать, что количество монотонных булевых функций вида $f(x_1, \ldots, x_n)$ на один больше количества подмножеств V множества $\mathsf{P}\{x_1, \ldots, x_n\}$, составленных из попарно не сравнимых с помощью $\subseteq$ множеств. ⊛

217. Доказать, что булева функция f является самодвойственной тогда и только тогда, когда $f^* = f$ (см. задачу 151 на стр. 49). ⊛

218. Доказать, что булева функция $f(x_1, \ldots, x_n)$ является самодвойственной тогда и только тогда, когда она удовлетворяет свойству из задачи 166 на стр. 54. ⊛

219. Доказать, что двойственной к линейной (монотонной) функции снова будет линейная (соответственно, монотонная) функция. Определить, как ведёт себя двойственная функция f^*, если f сохраняет ноль или единицу (см. задачу 151 на стр. 49). ⊛

220. Доказать, что n-местная булева функция $f(x_1, \ldots, x_n)$ самодвойственна тогда и только тогда, когда

$$f(x_1, \ldots, x_n) = g(x_1 \oplus x_2, \ldots, x_1 \oplus x_n) \oplus x_1$$

для некоторой $(n-1)$-местной булевой функции g. ⊛

221. Определить количество функций из $\mathcal{P}_n$, принадлежащих каждому из множеств $\mathcal{S}_0$, $\mathcal{S}_1$, $\mathcal{S}$ и $\mathcal{L}$. ⊛

222. Найти количество булевых функций от n переменных, являющихся одновременно самодвойственными и линейными. ⊛

223. Найти количество булевых функций от n переменных, являющихся одновременно монотонными и линейными. ⊛

224. Определить количество функций из $\mathcal{P}_n$, принадлежащих каждому из классов:

(а) $\mathcal{S}_0 \cap \mathcal{S}_1$; (г) $\mathcal{L} \cap \mathcal{S}_0$; (ж) $(\mathcal{S} \setminus \mathcal{S}_0) \cap \mathcal{S}_1$;

(б) $\mathcal{S}_0 \cup \mathcal{S}_1$; (д) $\mathcal{L} \cap \mathcal{S}_1$; (з) $\mathcal{M} \setminus \mathcal{S}_0$;

(в) $\mathcal{S} \cap \mathcal{S}_0$; (е) $\mathcal{S} \cap \mathcal{S}_0 \cap \mathcal{S}_1$; (и) $\mathcal{M} \setminus \mathcal{S}_1$. ⊛

225. Сколько имеется монотонных булевых функций от n переменных, у которых значения зависят лишь от количества единиц в наборе аргументов? ⊛

226. Доказать, что в множестве $\mathcal{P}_n$ имеется не меньше $2^{C_n^{\lfloor n/2 \rfloor}}$ монотонных функций. ⊛

227. Доказать, что если $f(x_1, \ldots, x_n)$ — линейная функция, отличная от константы, то количество единиц в таблице истинности равно 2^{n-1}. Верно ли обратное? ⊛

228. Доказать, что если функция f построена только с помощью дизъюнкции и импликации, то количество единиц в таблице истинности не меньше половины. Верно ли обратное? ⊛

229. Доказать, что в условии задачи 205 на стр. 63 конъюнкцию удалить нельзя. ⊛

230. Найти все трёхместные булевы функции f, которые в одиночку образуют полное множество $\{f\}$. Найти их количество. ⊛

231. Доказать, что если двухместная булева функция сохраняет ноль и единицу, то она является монотонной. Показать, что уже для трёхместных функций это неверно. ⊛

232. Доказать, что если двухместная булева функция самодвойственна, то она линейна. Показать, что уже для трёхместных функций это неверно. ⊛

233. Доказать, что если базис не содержит функций местности более двух, то он содержит не более трёх функций. Привести пример, который показывает, что число три уменьшить нельзя. ⊛

234. Доказать, что если базис не содержит нульместных функций и ни одна из функций не имеет фиктивных аргументов, то он содержит не более трёх функций. Привести пример, который показывает, что число три уменьшить нельзя. ⊛

9. Хорновские формулы и задача получения продукции

Пусть $\boldsymbol{V}$ — это множество пропозициональных переменных. Хорновская формула — это формула вида

$$(a_1 \wedge a_2 \wedge \dots \wedge a_r) \to b,$$

где $a_i, b \in \boldsymbol{V}$. Хорновская формула φ является следствием множества хорновских формул F, если на всяком наборе значений переменных из $\boldsymbol{V}$, на котором истинны все формулы из F, истинна и φ (обозначение: $F \Rightarrow \varphi$).

Технологический процесс t задаётся множеством $L_t \subseteq \boldsymbol{V}$ исходных продуктов этого процесса и его результирующим продуктом $b_t \in \boldsymbol{V}$.

Задача получения продукции: выяснить по заданному набору исходных продуктов $X \subseteq \boldsymbol{V}$ и результирующему продукту $y \in \boldsymbol{V}$, можно ли с помощью технологических процессов из множества F получить выход y по входным продуктам из X.

Для множества технологических процессов F и исходного множества продуктов X замыканием X с помощью F называется множество продуктов

$$\mathrm{cl}(X, F) = \{y \in \boldsymbol{V} : F \Rightarrow \Big(\bigwedge_{x \in X} x\Big) \to y\}.$$

Простейший алгоритм построения замыкания заключается в выборе из F на каждом шаге продукций, у которых вся левая часть уже получена и добавлением правой части к полученным продуктам ([3], рис. 13, алгоритм Замыкание).

Более совершенный алгоритм учитывает для каждой продукции количество ещё не полученных продуктов в левой части и при получении каждого из них уменьшает соответствующий счётчик на единицу. Как только счётчик стал равен нулю, продукт из правой части считается полученным ([3], рис. 15, алгоритм ОптЗам).

Задачи

235. Определить, каким из классов $\mathcal{S}_0$, $\mathcal{S}_1$, $\mathcal{S}$, $\mathcal{L}$, $\mathcal{M}$ принадлежит булева функция $f(x_1, \ldots, x_n, y)$, задаваемая хорновской формулой $(x_1 \wedge \cdots \wedge x_n) \to y$. ⊛

236. Пусть задано множество хорновских формул

$$F = \{(x \wedge y) \to z, (v \wedge z) \to x, \\ (v \wedge z) \to y, (z \wedge v) \to u, (u \wedge x) \to w\}.$$

Какие из следующих формул являются следствиями F?

(а) $(v \wedge z) \to w$; (б) $(x \wedge y) \to w$; (в) $(x \wedge y \wedge z) \to w$. ⊛

237. Доказать, что каждое множество хорновских формул имеет модель, то есть интерпретацию, в которой все они истинны. ⊛

238. Доказать, что если КНФ в каждой элементарной конъюнкции содержит в точности одну переменную без отрицания, то она эквивалентна конъюнкции хорновских формул. ⊛

239. Пусть J_i, $i = 1, \ldots, n$ — интерпретации. Назовём их пересечением интерпретацию $K = J_1 \ldots J_n$ такую, что для любой переменной x выполнено $K(x) = 1$ тогда и только тогда, когда $J_i(x) = 1$ для всех $i = 1, \ldots, n$. Доказать, что если хорновская формула имеет модели J_i, $i = 1, \ldots, n$, то их пересечение $K = J_1 \ldots J_n$ тоже будет её моделью. ⊛

240. Доказать, что формула Φ эквивалентна конъюнкции хорновских формул тогда и только тогда, когда сокращённая ДНФ для ¬Φ содержит в каждой элементарной конъюнкции в точности одну переменную с отрицанием. ⊛

241. Доказать, что формула Φ эквивалентна конъюнкции хорновских формул тогда и только тогда, когда выполнены одновременно два условия:

(а) Φ сохраняет единицу, то есть имеет значение 1 в интерпретации I_1, в которой все переменные имеют значение 1;

(б) если Φ истинна в интерпретациях $J_1, \ldots, J_k$, то Φ истинна в их пересечении (задача 239 на противоположной странице).

Указание. Рассмотреть Ψ — конъюнкцию всех хорновских формул, которые следуют из Φ. ⊛

242. Доказать, что формулы $a \oplus b$ и $a \vee b$ не эквивалентны никаким конъюнкциям хорновских формул. ⊛

243. *Нормальной* назовём формулу вида

$$(a_1^{\sigma_1} \wedge a_2^{\sigma_2} \wedge \cdots \wedge a_r^{\sigma_r}) \to b,$$

где $\sigma_i \in \{0, 1\}$ (см. раздел 6). Доказать, что каждая нормальная формула может быть получена как суперпозиция хорновских. ⊛

244. Доказать, что функция эквивалентна конъюнкции нормальных формул тогда и только тогда, когда она сохраняет единицу. ⊛

245. Назовём *сложным технологическим процессом* такой процесс t, который по набору исходных продуктов L_t одновременно производит некоторое множество продуктов B_t (а не один продукт b_t). Доказать, что сложному технологическому процессу соответствует конъюнкция хорновских формул. ⊛

246. Используя алгоритм Замыкание, вычислить замыкание для набора исходных продуктов $X = \{c, d\}$ и системы технологических процессов F:

$$\begin{array}{rrr} a, b, d \to h; & e, f \to c; & h, d, c \to g; \\ a, c, d, g \to f; & b, k \to a; & d, g, a \to e; \\ d, g \to b; & d, c \to k; & c, d, k \to h. \end{array}$$

Определить, какая цепочка процессов приводит к получению e. ⊛

247. Используя алгоритм ОптЗам, вычислить замыкание для набора исходных атрибутов $X = \{a, f\}$ и следующей системы зависимостей F:

$$\begin{array}{rlrlrl} a, b, c \to h & (1); & e, f \to c & (3); & g, d \to e & (5); \\ a, c, d, g \to h & (2); & f, a \to d & (4); & d, f, a \to g & (6). \end{array}$$

Определить, какая цепочка процессов приводит к получению h. ⊛

248. Пусть $\boldsymbol{V} = \{a, b, c, d, e, f, g, h\}$, $X = \{b, f\}$, а множество F состоит из следующих шести процессов:

$$a, b, c, h \to d \quad (1); \qquad g, b \to e \quad (3); \qquad f, e \to d \quad (5);$$
$$b, c, d \to a \quad (2); \qquad e, f \to c \quad (4); \qquad b, f \to g \quad (6).$$

Используя алгоритм ОптЗам, определить, какая последовательность процессов приводит к получению a. ⊛

10. Логика предикатов

Сигнатурой языка логики предикатов называется пара $\Sigma = (\boldsymbol{P}, \boldsymbol{V})$, в которой $\boldsymbol{P} = \{P_1^{(n_1)}, P_2^{(n_2)}, \ldots, P_k^{(n_k)}, \ldots\}$ — имена предикатов (или предикатные символы), $\boldsymbol{V} = \{x_1, x_2, \ldots, x_m, \ldots\}$ — имена объектов (или предметные переменные).

Атомная формула сигнатуры Σ — это строка одного из двух следующих видов: $P^{(n)}(x_1, \ldots, x_n)$ или $x_1 \approx x_2$ для произвольных $P^{(n)} \in \boldsymbol{P}$, $x_1, x_2, \ldots, x_n \in \boldsymbol{V}$. Формула сигнатуры Σ определяется индуктивно:

1) Каждая атомная формула является формулой. Все вхождения переменных в эту формулу являются свободными.
2) Если Φ и Ψ — формулы, то строки вида $(\Phi \wedge \Psi)$, $(\Phi \vee \Psi)$, $(\Phi \to \Psi)$ и $\neg\Phi$ — тоже формулы. Все вхождения переменных в новые формулы остаются такими же, какими были в исходных.
3) Если Φ — формула, $x \in \boldsymbol{V}$, то строки вида $(\forall x)\Phi$ и $(\exists x)\Phi$ — тоже формулы. Строки $(\forall x)$ и $(\exists x)$ называются кванторами всеобщности и существования соответственно по переменной x. Формула Φ называется областью действия соответствующего квантора. Все свободные вхождения переменной x в формулу Φ становятся связанными кванторами $(\forall x)$ или $(\exists x)$ соответственно. Остальные вхождения переменных остаются такими же, какими были в формуле Φ.

Формула без свободных переменных называется замкнутой. Формула, имеющая вид $Q\Phi$, где Q — последовательность кванторов, а Φ — бескванторная формула, называется предварённой.

С помощью $(\Phi)_y^x$ обозначается результат замены всех свободных вхождений переменной x в формуле Φ на y. При этом требуется, чтобы вновь полученные вхождения переменной y тоже были свободными, иначе результат замены неопределён.

Интерпретация сигнатуры $\Sigma = (\boldsymbol{P}, \boldsymbol{V})$ — это тройка $I = (A, \nu, \sigma)$, в которой

1) A — произвольное непустое множество, предметная область или носитель;
2) ν — функция, которая каждому символу $P^{(n)} \in \boldsymbol{P}$ ставит в соответствие n-местное отношение на множестве A;
3) σ — функция, отображающая $\boldsymbol{V}$ в A.

Значение переменной $x \in \boldsymbol{V}$ в интерпретации I — это $\sigma(x)$. С помощью $(I)^x_a$, где $x \in \boldsymbol{V}$, $a \in A$, обозначается интерпретация, которая отличается от I только тем (если вообще отличается), что значением x является a.

Значение каждой формулы в интерпретации $I = (A, \nu, \sigma)$ равняется 0 или 1.

1) Значение формулы $x \approx y$ равно 1, если $\sigma(x) = \sigma(y)$.
2) Значение $P(x_1, \ldots, x_n)$ равно 1, если $(\sigma(x_1), \ldots, \sigma(x_n)) \in \nu(P)$.
3) Значение формул $(\Phi \wedge \Psi)$, $(\Phi \vee \Psi)$, $(\Phi \to \Psi)$ и $\neg\Phi$ определяется как в логике высказываний.
4) Значение формулы $(\forall x)\Phi$ равно 1, если значение формулы Φ равно 1 в интерпретациях $(I)^x_a$ для всех $a \in A$.
5) Значение формулы $(\exists x)\Phi$ равно 1, если значение формулы Φ равно 1 в интерпретации $(I)^x_a$ хотя бы для одного $a \in A$.

Формула Φ называется истинной в интерпретации, если её значение равно 1, в противном случае формула называется ложной. Формула Φ называется тождественно истинной, если она истинна во всех интерпретациях.

Формула Ψ следует из формулы Φ (записывают в виде $\Phi \Rightarrow \Psi$), если Ψ истинна во всех интерпретациях, в которых истинна Φ. Формулы Φ и Ψ называются эквивалентными (записывают в виде $\Phi \equiv \Psi$), если их значения совпадают во всех интерпретациях.

Некоторые из эквивалентностей логики предикатов:

$$\neg(\forall x)\Phi \equiv (\exists x)\neg\Phi; \qquad \Psi \wedge (\forall x)\Phi \equiv (\forall x)(\Psi \wedge \Phi);$$
$$\neg(\exists x)\Phi \equiv (\forall x)\neg\Phi; \qquad \Psi \wedge (\exists x)\Phi \equiv (\exists x)(\Psi \wedge \Phi);$$
$$(\exists x)\Psi \equiv (\forall x)\Psi \equiv \Psi; \qquad \Psi \vee (\forall x)\Phi \equiv (\forall x)(\Psi \vee \Phi);$$
$$(\exists x)\Phi \equiv (\exists y)(\Phi)^x_y; \qquad \Psi \vee (\exists x)\Phi \equiv (\exists x)(\Psi \vee \Phi).$$
$$(\forall x)\Phi \equiv (\forall y)(\Phi)^x_y;$$

Здесь x — любая переменная, Φ и Ψ — любые формулы, причём Ψ не содержит свободных вхождений x.

Задачи

249. Для каждой из следующих формул определить, какие вхождения переменных в них являются свободными, а какие — связанными (и каким квантором). Здесь Q — трёхместный предикатный символ.

(а) $(\forall x)((\exists z)Q(x,z,y) \to (\forall x)(Q(z,x,y) \to (\exists y)(Q(z,y,x) \vee x \approx z)));$

(б) $(\exists y)((\exists x)(\neg x \approx y \wedge Q(x,x,y) \wedge$
$\wedge(\forall x)(Q(x,z,y) \to (\exists z)(Q(x,y,z) \vee Q(z,z,y))));$

(в) $(\forall x)((\exists z)Q(z,z,x) \wedge (\exists x)Q(y,y,x)) \vee$
$\vee(\exists x)((\exists z)Q(x,x,z) \to (\forall x)Q(x,y,y));$

(г) $(\exists y)Q(x,y,z) \vee (\exists x)(Q(z,x,y) \wedge (\forall y)Q(y,x,y)) \wedge$
$\wedge(\forall z)(\exists x)Q(z,z,x).$ ⊛

250. Для каждой формулы Φ из предыдущей задачи определить, какие из следующих замен возможны: $(\Phi)^x_y$, $(\Phi)^x_z$, $(\Phi)^y_x$, $(\Phi)^y_z$, $(\Phi)^z_x$, $(\Phi)^z_y$. Объяснить, почему. В тех случаях, когда это возможно, найти результат замены. ⊛

251. Пусть R — двухместный предикатный символ. Написать замкнутые формулы, означающие:

(а) R рефлексивно;

(б) R антирефлексивно;

(в) R симметрично;

(г) R антисимметрично;

(д) R транзитивно;

(е) R — отношение эквивалентности;

(ж) R — строгий линейный порядок. ⊛

252. Пусть $\leqslant$ — двухместный предикатный символ, который означает отношение нестрогого частичного порядка. Написать формулы, имеющие одну свободную переменную x, означающие:

(а) x — наибольший элемент;

(б) x — максимальный элемент;

(в) x — единственный максимальный элемент;

(г) x сравним со всеми элементами;

(д) x сравним со всеми максимальными элементами;

(е) x не сравним в точности с одним элементом. ⊛

253. Будем считать, что предметная область — это множество людей, а сигнатура содержит символы $P^{(2)}$, $S^{(2)}$ и $M^{(1)}$, означающие отношения «быть родителем», «состоять в браке с» и «быть мужчиной» соответственно. Написать формулы, означающие:

(а) x является внуком y;

(б) у x есть не менее двух детей;

(в) x и y имеют одного общего ребёнка;

(г) у x есть незамужняя сестра;

(д) x и y — двоюродные братья;

(е) x женат, а из его сыновей женат только один. ⊛

254. Предметная область включает только действительные числа $\mathbb{R}$ и одноместные бесконечно дифференцируемые функции на множестве $\mathbb{R}$. Сигнатура содержит трёхместный предикатный символ V, $V(f, x, y)$ говорит «значение функции f на аргументе x равно y»; и двухместный символ D, где $D(f, g)$ означает «f — производная g». Записать формулы, обозначающие следующее. Допускается строить вспомогательные формулы и использовать построенные ранее:

(а) x — число;

(б) f — функция;

(в) f — функция-константа;

(г) f — функция тождественно равная нулю;

(д) $x = 0$;

(е) f — тождественная функция;

(ж) $x = 1$;

(з) f — экспонента e^x;

(и) $x + y = z$ — для чисел x, y, z;

(к) $xy = z$ — для чисел x, y, z;

(л) x — неотрицательное число;

(м) число x меньше числа y;

(н) функция f принимает все действительные значения;

(о) функция f является взаимно однозначной;

(п) функция f — это синус;

(р) число x равно π;

(с) число x является натуральным. ⊛

255. Элементарной арифметикой натуральных чисел называется следующая интерпретация $\mathcal{N} = (\omega, \nu, \sigma)$: носитель — множество натуральных чисел ω, трёхместные предикатные символы $A(x, y, z)$ и $M(x, y, z)$ означают, что $x + y = z$ и $xy = z$ соответственно, а названия чисел записывают обычным образом в десятичной системе: 0 — нуль, 1 — один и т. д. Также есть переменные x, y, z и другие, значения которых заранее неизвестны.

Записать формулы со свободными переменными из $\{x, y, z\}$, которые истинны в элементарной арифметике натуральных чисел тогда и только тогда, когда

(а) x является чётным числом;

(б) x и y взаимно просты;

(в) z лежит в интервале между x и y;

(г) x является наибольшим общим делителем y и z;

(д) x является наименьшим общим кратным всех чисел из промежутка $[y; z)$. ⊛

256. Пусть сигнатура содержит один двухместный предикатный символ $\subseteq$, который в интерпретации $I = (\mathsf{P}(A), \nu, \sigma)$ означает, как обычно, «быть подмножеством». Записать формулы со свободными переменными из $\{x, y, z\}$, которые истинны в такой интерпретации тогда и только тогда, когда

(а) $x = \varnothing$; (в) $|x| = 1$; (д) $x = y \cup z$;

(б) $x = A$; (г) $x = y \cap z$; (е) $x = y \setminus z$. ⊛

257. Пусть сигнатура содержит два предикатных символа $L^{(3)}$ и $E^{(4)}$. Геометрией Тарского называется интерпретация, носителем которой является множество точек на евклидовой плоскости, $L(x, y, z)$ означает, что точки x, y и z лежат на одной прямой и именно в такой последовательности (y между x и z, не обязательно они различны), а $E(x, y, u, v)$ означает, что длины отрезков xy и uv равны.

Записать формулы со свободными переменными из множества $\{x, y, z, u\}$, которые являются истинными в геометрии Тарского тогда и только тогда, когда (а) x, y и z образуют треугольник;

(б) x является серединой отрезка yz;

(в) луч xy является биссектрисой угла $\angle zxu$;

(г) угол $\angle xyz$ является прямым;

(д) точки x, y, z, u лежат на одной окружности;

(е) треугольник $\triangle xyz$ является остроугольным;

(ж) отрезок xy короче, чем zu;

(з) точка x лежит внутри окружности с центром y и точкой z, лежащей на ней. ⊛

258. Представить каждое из следующих предложений формулой логики предикатов, определив в каждом случае подходящую сигнатуру.

(а) Не все студенты изучают и анализ, и историю.

(б) Только один студент не сдавал экзамен по дискретной математике.

(в) Только один студент сдал все экзамены на 100 баллов.

(г) Максимальные баллы, полученные по дискретной математике, превышают максимальные баллы, полученные по информатике.

(д) Имеется брадобрей, бреющий только тех жителей города, которые не бреются сами.

(е) Есть политики, которые могут обманывать всех людей некоторое время, есть политики, которые могут обманывать некоторых людей всё время, но никто не может обманывать всех людей всё время. ⊛

259. Написать формулу логики предикатов, истинную только в интерпретациях, носитель которых содержит один элемент (два, три, $\ldots$, k элементов). ⊛

260. Написать формулу логики предикатов, истинную только в интерпретациях, в которых $(n+1)$-местное отношение F означает функцию. ⊛

261. Определить, какие из следующих формул истинны в элементарной арифметике (задача 255 на стр. 75), пояснить, что они означают:

(а) $(\forall x)(\exists y)(\exists z)(A(x,y,z) \wedge M(x,y,z))$;

(б) $(\forall x)(\exists y)(A(y,y,x) \vee (\exists z)(A(y,y,z) \wedge A(z,1,x)))$;

(в) $(\forall x)(\forall y)(\forall z)((\exists u)M(x,u,y) \wedge (\exists u)M(y,u,z) \to (\exists u)M(x,u,z))$;

(г) $(\forall x)(\exists y)(M(y,y,x) \vee (\exists z)(M(y,y,z) \wedge M(z,y,x)))$;

(д) $$(\forall x)(M(x,x,x) \vee (\exists y)(M(x,x,y)\wedge \\ \wedge(\exists u)(\exists v)(\exists w)(A(x,u,v) \wedge A(v,w,y)\wedge \\ \wedge(\exists z)(M(u,w,z) \wedge \neg z \approx 0))))).$$ ⊛

262. Определить, какие из следующих формул истинны в геометрии Тарского (задача 257 на стр. 75), пояснить, что они означают:

(а) $(\forall x)(\forall y)(\exists z)(L(x,y,z) \wedge E(x,y,y,z))$;

(б) $(\forall x)(\forall y)(\forall z)(\exists u)(E(u,x,u,y) \wedge E(u,y,u,z))$;

(в) $$(\forall x)(\forall y)(\exists u)(\exists v)(\forall z)(E(z,x,z,y) \to \\ \to L(z,u,v) \vee L(z,v,u) \vee L(u,z,v));$$

(г) $$(\forall x)(\exists y)(\exists z)(\exists u)(L(y,u,z)\wedge \\ \wedge(\forall v)(L(y,v,z) \wedge E(x,u,x,v) \to u \approx v));$$

(д) $$(\forall x)(\forall y)(\forall z)(\neg x \approx y \wedge \neg E(x,y,x,z) \to \\ \to (\forall u)(E(x,z,x,u) \wedge L(x,u,y) \to \\ \to (\forall v)(E(x,u,x,v) \wedge E(y,u,y,v) \to u \approx v))).$$ ⊛

263. Сигнатура и интерпретация взяты из задачи 254 на стр. 74. Определить, какие из следующих формул истинны в этой интерпретации, пояснить, что они означают:

(а) $(\forall x)(\forall y)((\exists z)(D(z,x) \wedge D(z,y)) \to x \approx y)$;

(б) $(\forall x)(\forall y)(\exists z)(\forall u)(\forall v)(\forall w)(V(x,u,v) \wedge V(y,v,w) \to V(z,u,w))$;

(в) $(\forall x)(\exists y)(\forall u)(\forall v)(V(x,u,v) \to V(y,v,u))$;

(г) $$(\forall x)(\forall y)((\exists z)(D(z,x) \wedge D(z,y))\wedge \\ \wedge(\exists u)(\exists v)(V(x,u,v) \wedge V(y,u,v)) \to x \approx y);$$

(д) $(\exists x)(D(x,x) \wedge (\forall y)\neg V(x,y,y))$;

(е) $(\exists x)(\neg D(x,x) \wedge (\exists y)(D(y,x) \wedge D(x,y)))$. ⊛

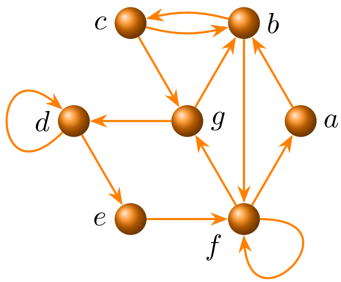


Рис. 6: Граф $\mathfrak{G}$ из задачи 264.

264. Сигнатура Σ состоит из двухместного предикатного символа $E^{(2)}$, предметных переменных $a, \ldots, g$ и других. Интерпретация $\mathfrak{G}$ изображена на рис. 6, стрелка из x в y означает, что $(x, y) \in E$ (такие интерпретации называются графами). Определить, какие из следующих формул истинны в этой интерпретации, пояснить, что они означают:

(а) $(\forall x)(\exists y)(\exists z)(E(x, y) \wedge E(z, x))$;

(б) $(\forall x)(\exists y)(\exists z)(E(x, y) \wedge E(y, z) \wedge E(z, x))$;

(в) $(\forall x)(\exists z)(E(z, z) \wedge (E(x, z) \vee (\exists y)(E(x, y) \wedge E(y, z))))$;

(г) $(\forall x)(\forall y)(E(x, y) \to \neg E(x, x) \vee \neg E(y, y))$;

(д) $(\exists x)(\forall y)(\forall z)(E(x, y) \wedge E(y, z) \to \neg x \approx y \wedge \neg x \approx z)$;

(е) $(\exists x)(\exists y)(E(x, y) \wedge (\forall z)(E(x, z) \to y \approx z))$;

(ж) $(\exists x)(\forall y)(\forall z)(\forall u)(E(g, y) \wedge E(y, z) \wedge E(z, u) \to$
$\to \neg y \approx x \wedge \neg z \approx x \wedge \neg u \approx x)$. $\circledast$

265. Доказать следующие эквивалентности:

(а) $(\exists x)(\Phi \vee \Psi) \equiv (\exists x)\Phi \vee (\exists x)\Psi$;

(б) $(\forall x)(\Phi \wedge \Psi) \equiv (\forall x)\Phi \wedge (\forall x)\Psi$;

(в) $(\exists x)(\exists y)\Phi \equiv (\exists y)(\exists x)\Phi$;

(г) $(\forall x)(\forall y)\Phi \equiv (\forall y)(\forall x)\Phi$. $\circledast$

266. Доказать следующие эквивалентности:

(а) $(\exists x)(x \approx y \wedge \Phi) \equiv (\Phi)^x_y$;

(б) $(\forall x)(x \approx y \to \Phi) \equiv (\Phi)^x_y$;

(в) $\Phi \vee (\exists x)\Phi \equiv (\exists x)\Phi$;

(г) $\Phi \wedge (\forall x)\Phi \equiv (\forall x)\Phi$. ⊛

267. Доказать следования:

(а) $(\exists x)(\Phi \wedge \Psi) \Rightarrow (\exists x)\Phi \wedge (\exists x)\Psi$;

(б) $(\forall x)\Phi \vee (\forall x)\Psi \Rightarrow (\forall x)(\Phi \vee \Psi)$;

(в) $(\forall x)(\Phi \to \Psi) \Rightarrow (\forall x)\Phi \to (\forall x)\Psi$;

(г) $(\exists x)(\forall y)\Phi \Rightarrow (\forall y)(\exists x)\Phi$. ⊛

268. Привести примеры формул без отрицаний элементарной арифметики, которые показывают, что следования из предыдущей задачи в обратную сторону неверны. ⊛

269. Привести к предварённому виду следующие формулы, Q — двухместный предикатный символ:

(а) $\neg(\forall z)\neg(\forall x)(\exists y)(\forall u)(Q(x, y) \wedge Q(z, u))$;

(б) $(\exists x)(\forall y)Q(x, y) \to (\exists x)(\forall y)Q(y, x)$;

(в) $\neg((\forall x)((\forall y)Q(x, y) \to Q(x, z)) \vee ((\forall z)Q(z, x) \wedge \neg(\exists y)Q(y, x)))$. ⊛

270. С помощью преобразований доказать следующие эквивалентности, формула Θ не содержит переменной x свободно:

(а) $(\exists x)(\Phi \to \Psi) \equiv (\forall x)\Phi \to (\exists x)\Psi$;

(б) $(\forall x)(\Theta \to \Phi) \equiv \Theta \to (\forall x)\Phi$;

(в) $(\forall x)(\Phi \to \Theta) \equiv (\exists x)\Phi \to \Theta$. ⊛

271. Доказать, что бескванторная формула логики предикатов Φ, которая не содержит знака равенства $\approx$, тождественно истинна тогда и только тогда, когда она может быть получена из некоторой тождественно истинной булевой формулы Ψ заменой всех вхождений каждой пропозициональной переменной в Ψ на некоторую атомную подформулу из Φ. ⊛

272. С помощью преобразований доказать тождественную истинность следующих формул:

(а) $(\forall x)(\Phi \to \Psi) \to ((\forall x)\Phi \to (\forall x)\Psi)$;

(б) $(\forall x)(\Phi \to \Psi) \to ((\exists x)\Phi \to (\exists x)\Psi)$;

(в) $(\forall x)\Phi \to ((\exists x)\Psi \to (\exists x)(\Phi \wedge \Psi))$;

(г) $((\forall x)\Phi \to (\forall x)\neg\Psi) \to ((\forall x)(\Phi \to \Psi) \to (\exists x)\neg\Phi)$. ⊛

273. Доказать, что следующая формула не может быть истинна ни на какой конечной предметной области, но может быть истинна на бесконечных: ⊛

$$(\exists x)(\forall y)\neg R(y,x) \wedge (\forall x)(\exists y)R(x,y) \wedge \\ \wedge (\forall x)(\forall y)(\forall z)(R(x,y) \wedge R(z,y) \to x \approx z).$$

11. Логика предикатов и базы данных

Схема n-местного отношения R — это набор $(\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n)$ атрибутов, каждый из которых содержит имя и область значений: $\mathcal{A}_i = (A_i, D_i)$. Схема базы данных состоит из перечня имён отношений с указанием схемы каждого из них. Состояние базы данных — это отображение из схемы базы данных в множество конечных отношений, имеющих соответствующие схемы.

С точки зрения логики предикатов схема базы данных — это сигнатура, состояние базы данных — это интерпретация, в которой все отношения конечны.

Реляционная алгебра — это простейшие операции над отношениями, по своим выразительным возможностям реляционная алгебра совпадает с логикой предикатов. Основные действия реляционной алгебры:

- если R и S — два отношения с одной и той же схемой Σ, то их объединение $R \cup S$, пересечение $R \cap S$ и разность $R \setminus S$ — это отношения с той же самой схемой Σ;
- если R и S — два отношения со схемами Σ_R и Σ_S соответственно, то их декартово произведение $R \times S$ — это отношение со схемой (Σ_R, Σ_S). Если Σ_R и Σ_S содержат атрибуты с одинаковыми именами, то, чтобы различать их, к именам добавляют имя отношения. Например, вместо атрибута с именем A в схемах Σ_R и Σ_S будут атрибуты с именами $R.A$ и $S.A$ в схеме (Σ_R, Σ_S);
- если R — отношение со схемой Σ, Θ — булева комбинация равенств и других отношений предметной области, использующая имена атрибутов, то результатом фильтрации $(R : \Theta)$ является отношение R' со схемой Σ, содержащее в точности те наборы R, которые удовлетворяют формуле Θ;

- если R — это отношение со схемой $(A_1, \ldots, A_n)$, и задана схема проекции $X = (A_{i_1}/B_1, \ldots, A_{i_m}/B_m)$, причём имена $B_1, \ldots, B_m$ попарно различны, то проекцией отношения R на X является отношение $S[X]$ со схемой $(B_1, \ldots, B_m)$, которое состоит из всех наборов вида $(a_{i_1}, \ldots, a_{i_m})$, где $(a_1, \ldots, a_n)$ — всевозможные элементы R. Если $B_j = A_{i_j}$ для всех $j = 1, \ldots, m$, то проекцию обозначают просто как $R[A_{i_1}, \ldots, A_{i_m}]$.

Из основных операций реляционной алгебры можно построить некоторые производные, широко используемые на практике:

- пусть R и S — это отношения со схемами $(A_1, \ldots, A_n, B_1, \ldots, B_m)$ и $(B_1, \ldots, B_m, C_1, \ldots, C_k)$ соответственно (общие атрибуты изображены для удобства по краям, но вообще это не требуется). Тогда естественное соединение $R \bowtie S$ отношений R и S имеет схему
$$(A_1, \ldots, A_n, B_1, \ldots, B_m, C_1, \ldots, C_k)$$
и состоит из всевозможных наборов вида
$$(a_1, \ldots, a_n, b_1, \ldots, b_m, c_1, \ldots, c_k),$$
где $(a_1, \ldots, a_n, b_1, \ldots, b_m) \in R$, $(b_1, \ldots, b_m, c_1, \ldots, c_k) \in S$.
- пусть R и S — это отношения со схемами Σ_R и Σ_S соответственно, Θ — булева комбинация равенств и других отношений предметной области, использующая имена атрибутов. Тогда Θ-соединение $R \overset{\Theta}{\bowtie} S$ отношений R и S имеет схему (Σ_R, Σ_S) и состоит из всевозможных наборов вида $(a_1, \ldots, a_n, b_1, \ldots, b_m)$, удовлетворяющих формуле Θ, где $(a_1, \ldots, a_n) \in R$, $(b_1, \ldots, b_m) \in S$. Как и в случае декартова произведения при совпадении имён в Σ_R и Σ_S к ним добавляется имя соответствующего отношения.

Задачи

274. Пусть заданы следующие атрибуты:
$$(A, \{a_1, a_2, a_3\}), \quad (B, \omega), \quad (C, \{c_1, c_2, c_3\}), \quad (D, \omega).$$

Пусть отношение R со схемой (A, B, C) равняется
$$R = \{(a_1, 7, c_1), (a_1, 5, c_2), (a_2, 7, c_1), (a_3, 4, c_3)\},$$

Сотрудники				
Номер Сотрудника	ФИО	Отдел	Должность	Оклад
1	Иванов А.А.	торговый	менеджер	7000
2	Сидоров Н.П.	плановый	экономист	5000
3	Сидорова М.И.	торговый	зав. складом	6000
4	Ольгина Н.А.	плановый	экономист	5500
5	Горев С.В.	плановый	зав. отделом	10000

Комнаты		
НомерСотрудника	Этаж	НомерКомнаты
3	2	17
1	2	17
7	2	18
5	3	7
2	3	27

Оборудование		
Этаж	НомерКомнаты	Название
2	17	компьютер
2	17	принтер
3	7	ксерокс
3	25	принтер

Рис. 7: Состояние базы данных.

а отношение S со схемой (C, D) равняется

$$S = \{(c_1, 2), (c_2, 8), (c_2, 7), (c_1, 4), (c_2, 5)\}.$$

Определить результаты следующих выражений реляционной алгебры:

(а) $(R : A \approx a_1)$;

(б) $R[B] \times S[D]$;

(в) $R \bowtie S$;

(г) $R[A, C] \bowtie R[C]$;

(д) $R \overset{R.C \not\approx S.C \wedge R.B > S.D}{\bowtie} S$. $\circledast$

275. Определить, являются ли операции фильтрации и проекции дистрибутивными относительно объединения, пересечения и разности, то есть выполняются ли равенства

(а) $(R \cup S : \Theta) = (R : \Theta) \cup (S : \Theta)$;

(б) $(R \cap S : \Theta) = (R : \Theta) \cap (S : \Theta)$;

(в) $(R \setminus S : \Theta) = (R : \Theta) \setminus (S : \Theta)$;

(г) $(R \cup S)[X] = R[X] \cup S[X]$;

(д) $(R \cap S)[X] = R[X] \cap S[X]$;

(е) $(R \setminus S)[X] = R[X] \setminus S[X]$. ⊛

276. Для базы данных на рис. 7 на предыдущей странице построить выражение реляционной алгебры, задающее список фамилий сотрудников, в комнатах которых нет никакого оборудования. Построить формулу логики предикатов, определяющую то же отношение. ⊛

277. Пусть есть отношения R и S со схемами $(A_1, \ldots, A_n, B_1, \ldots, B_m)$ и $(B_1, \ldots, B_m)$ соответственно. Ч а с т н ы м R/S от деления отношения R на отношение S называется наибольшее отношение Q с атрибутами $A_1, \ldots, A_n$, каждый набор t которого в соединении с каждым набором $s \in S$ входит в R, то есть $Q \times S \subseteq R$. Построить выражение реляционной алгебры, эквивалентное R/S, и формулу логики предикатов, определяющую это отношение. ⊛

278. Написать SQL-запросы и соответствующие формулы для получения следующей информации из базы данных на рис. 7 на предшествующей странице, вычислить результат:

(а) найти всех сотрудников с окладом больше 5500;

(б) найти все отделы, в которых есть сотрудники с окладом больше 8000;

(в) составить список должностей и получаемых по ним окладов;

(г) составить список сотрудников торгового отдела, получающих зарплату от 6000 до 6500 и работающих не на третьем этаже;

(д) составить список комнат, где все сотрудники получают оклад меньше 7500. ⊛

279. Определить, какие из следующих ограничений целостности выполняются для состояния базы данных на рис. 7 на стр. 83,:

(а) $(\forall n)(\forall x)(\forall y)(\forall z)(\forall v)(\forall x_1)(\forall y_1)(\forall z_1)(\forall v_1)$
$$(\text{Сотрудники}(n, x, y, z, v) \wedge \text{Сотрудники}(n, x_1, y_1, z_1, v_1) \rightarrow$$
$$\rightarrow x \approx x_1 \wedge y \approx y_1 \wedge z \approx z_1 \wedge v \approx v_1);$$

(б) $(\forall ns)(\forall x)(\forall y)(\text{Комнаты}(ns, x, y) \rightarrow$
$$\rightarrow (\exists z)(\exists u)(\exists v)(\exists w)\text{Сотрудники}(ns, z, u, v, w));$$

(в) $(\forall x)(\forall y)(\forall z)(\forall v)(\forall o)(\text{Сотрудники}(x, y, z, v, o) \rightarrow$
$$\rightarrow 1000 < o \wedge o < 25000).$$ ⊛

280. Написать формулы, выражающие следующие ограничения целостности для базы данных на рис. 7 на стр. 83, определить, какие из них выполняются для приведённого её состояния:

(а) в отношении Комнаты набор атрибутов (НомерСотрудника, НомерКомнаты) является ключом;

(б) для каждого человека из отношения Сотрудники в отношении Комнаты определено его место работы;

(в) в отношении Комнаты номера всех комнат на втором этаже больше 10, но меньше 20, а номера всех комнат на третьем этаже больше 20. ⊛

12. Ориентированные графы

(Ориентированный) граф $\mathfrak{G}$ — это пара (V, E), где V — конечное множество вершин (узлов, точек) графа, а $E \subseteq V \times V$ — некоторое множество пар вершин, или бинарное отношение на V. Элементы E называют рёбрами. Ребро вида (u, u) называется петлёй.

Полустепень исхода вершины v — это количество исходящих из v рёбер, а полустепень захода — это количество входящих в v рёбер.

Размеченный граф — это граф $\mathfrak{G} = (V, E)$, снабжённый одной или двумя функциями разметки вида $\ell : V \to M$ и $c : E \to L$, где M и L — множества меток вершин и рёбер, соответственно.

Мультиграф отличается от графа тем, что в нём между двумя вершинами может быть несколько рёбер, метки которых могут отличаться.

Два графа $\mathfrak{G}_1 = (V_1, E_1)$ и $\mathfrak{G}_2 = (V_2, E_2)$ называются изоморфными, если между их вершинами существует взаимно однозначное соответствие $\varphi : V_1 \to V_2$ такое, что для любой пары вершин $u, v \in V_1$ выполнено

$$(u, v) \in E_1 \iff (\varphi(u), \varphi(v)) \in E_2.$$

Путь в графе $\mathfrak{G} = (V, E)$ — это последовательность вершин $(v_1, \ldots, v_n)$, в котором каждая предыдущая вершина соединена со следующей ребром: $(v_i, v_{i+1}) \in E$ для $i = 1, \ldots, n-1$. Этот путь ведёт из начальной вершины v_1 в конечную вершину v_n и имеет длину $n-1$. В этом случае будем говорить, что v_n достижима из v_1. В частности, каждая вершина v достижима сама из себя путём длины ноль: (v).

Путь называется простым, если все вершины на нём, кроме, быть может, первой и последней, различны.

Цикл — это содержащий хотя бы одно ребро путь, в котором начальная вершина совпадает с конечной. Ациклическим называется граф без циклов. Ориентированный граф называется сильно связным, если любая пара вершин в нём соединена путями в обоих направлениях.

Матрица смежности графа $\mathfrak{G} = (V, E)$ с n пронумерованными вершинами $V = \{v_1, \ldots, v_n\}$ — это булева матрица $A_{\mathfrak{G}}$ размера $n \times n$, в которой

$$a_{i,j} = \begin{cases} 1, & \text{если } (v_i, v_j) \in E, \\ 0 & \text{в противном случае.} \end{cases}$$

Список смежности L_v для вершины v включает все смежные с ней вершины, то есть $L_v = (w_1, \ldots, w_k)$, если $(v, w_1), \ldots, (v, w_k)$ — это в точности все рёбра, исходящие из v. Представление графа $\mathfrak{G} = (V, E)$ с вершинами $V = \{v_1, \ldots, v_n\}$ с помощью списков смежности — это слово

$$(v_1(L_{v_1}), v_2(L_{v_2}), \ldots, v_n(L_{v_n})).$$

Граф достижимости $\mathfrak{G}^* = (V, E^*)$ для ориентированного графа $\mathfrak{G} = (V, E)$ имеет то же множество вершин V и множество рёбер

$$E^* = \{(u, v) : \text{ в } \mathfrak{G} \text{ вершина } v \text{ достижима из } u\}.$$

Вершины v и w ориентированного графа $\mathfrak{G} = (V, E)$ называются взаимно достижимыми, если в $\mathfrak{G}$ вершины v и w достижимы друг из друга. Классы эквивалентности по отношению взаимной достижимости называются компонентами сильной связности графа.

Граф сильной достижимости $\mathfrak{G}^*_* = (V, E^*_*)$ для ориентированного графа $\mathfrak{G} = (V, E)$ имеет то же множество вершин V и множество рёбер

$$E^*_* = \{(u, v) : \text{ в } \mathfrak{G} \text{ вершины } v \text{ и } u \text{ взаимно достижимы}\}.$$

Пусть K и K' — компоненты сильной связности графа $\mathfrak{G}$. Компонента K достижима из компоненты K', если $K = K'$ или существуют такие две вершины $u \in K$ и $v \in K'$, что u достижима из v. Компонента K строго достижима из K', если при этом $K \neq K'$. Компонента K называется минимальной, если она не является строго достижимой ни из какой компоненты.

Подмножество вершин $W \subseteq V$ ориентированного графа $\mathfrak{G} = (V, E)$ называется порождающим, если каждую вершину графа можно достичь из некоторой вершины W. Подмножество вершин $W \subseteq V$ называется базой графа, если оно является порождающим, но никакое его собственное подмножество порождающим не является. Подмножество вершин W является базой $\mathfrak{G}$ тогда и только тогда, когда оно содержит по одной вершине из каждой минимальной компоненты сильной связности $\mathfrak{G}$ и не содержит никаких других вершин.

Задачи

281. Для заданного множества вершин $v_1, \ldots, v_n$ найти

(а) общее количество графов;

(б) количество графов без петель;

(в) количество графов, между любыми двумя вершинами которых имеется не более одного ребра;

(г) количество графов без петель, между любыми двумя вершинами которых имеется не более одного ребра;

(д) количество т у р н и р о в. Турнир — это граф без петель, между любыми двумя различными вершинами которого имеется в точности одно ребро. ⊛

282. Доказать, что в ориентированном графе без циклов есть хотя бы один исток и хотя бы один сток. ⊛

283. Пусть $\mathfrak{G} = (V, E)$ — это ориентированный граф без циклов и $|E| > 0$. Какие из следующих утверждений верны?

(а) Любой путь начинается в истоке или заканчивается в стоке.

(б) Любые два пути наибольшей длины имеют хотя бы одну общую вершину.

(в) Из любой вершины достижим некоторый сток.

(г) В $\mathfrak{G}$ есть вершина, полустепени исхода и захода которой равны нулю. ⊛

284. Пусть $\mathfrak{G} = (V, E)$ — это ациклический граф. Доказать, что его вершины можно пронумеровать $V = \{v_1, \ldots, v_n\}$ так, что если выполнено $(v_i, v_j) \in E$, то $i < j$. ⊛

285. Пусть $\mathfrak{G} = (V, E)$ — граф, $V = \{v_1, \ldots, v_n\}$, A — матрица смежности, $W \subseteq V$, а вектор-столбец w содержит нули и единицы, причём $w_i = 1$ означает $v_i \in W$. Определить смысл булевых произведений Aw и $w^T A$. ⊛

286. Граф называется о д н о р о д н ы м, если полустепени захода и исхода всех его вершин равны одному и тому же числу r. Доказать, что для каждого $n > r$ существует однородный граф с n вершинами и полустепенью r. ⊛

287. Доказать, что в турнире из вершины v с наибольшей полустепенью исхода в любую другую ведёт путь длины не более двух. ⊛

288. Индукцией по количеству вершин доказать следующее утверждение: в турнире есть простой путь, который проходит через все вершины. ⊛

289. Чемпионат организован по круговой системе (каждая команда играет по одному матчу с каждой, победитель определяется по количеству выигранных матчей). Доказать, что если победитель чемпионата проиграл команде A, то A проиграла некоторой команде B, которая, в свою очередь, проиграла победителю. ⊛

290. Граф называется полусвязным, если для любой пары вершин существует путь из одной из них в другую. Доказать, что граф является полусвязным тогда и только тогда, когда в нём есть путь, проходящий через все вершины. ⊛

291. Определить, что представляет собой граф достижимости для

(а) графа с n вершинами и пустым множеством рёбер;

(б) графа с n вершинами: $V = \{v_1, \ldots, v_n\}$, рёбра которого образуют цикл: $E = \{(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n), (v_n, v_1)\}$. ⊛

292. Построить представления в виде матрицы смежности и списков смежности для ориентированного графа ⊛

$$\mathfrak{G} = (\{a, b, c, d\}, \{(a, b), (a, c), (a, a), (b, a), (b, b), (c, a), (c, d), (d, b)\}).$$

293. Вычислить матрицу графа достижимости $A_{\mathfrak{G}^*}$ для следующего графа

$$\mathfrak{G} = (\{a, b, c, d, e, f, g\},$$
$$\{(a, b), (b, a), (a, c), (b, d), (e, d), (d, f), (f, c), (c, f), (g, e)\})$$

и построить соответствующий ей граф достижимости. Найти все базы графа $\mathfrak{G}$. ⊛

294. Построить для изображённых на рис. 8 на следующей странице ориентированных графов $\mathfrak{G}_1$ и $\mathfrak{G}_2$ их матрицы смежности и списки смежности. Вычислить матрицы достижимости и построить соответствующие графы достижимости. ⊛

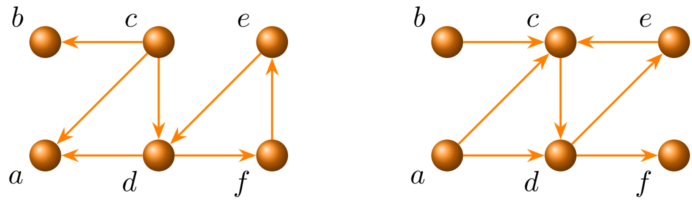


Рис. 8: Графы $\mathfrak{G}_1$ и $\mathfrak{G}_2$.

295. Пусть граф $\mathfrak{G} = (V, E)$ задан своей матрицей смежности:

$$A_{\mathfrak{G}} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Построить граф достижимости $\mathfrak{G}^* = (V, E^*)$ для $\mathfrak{G}$ и определить, сколько в нём новых рёбер, то есть чему равна разность $|E^*| - |E|$. ⊛

296. Доказать, что граф без петель ациклический тогда и только тогда, когда все компоненты сильной связности содержат по одному элементу. ⊛

297. Показать, что для каждого положительного n существует граф без циклов с n вершинами и $\frac{n(n-1)}{2}$ рёбрами. Доказать, что в любом графе, в котором рёбер больше, обязательно есть цикл. ⊛

298. Доказать, что в ориентированном графе без циклов существует единственная база, состоящая из всех истоков. ⊛

299. Дан ориентированный граф $\mathfrak{G} = (V, E)$, где $V = \{a, b, c, d, e, f, g, h\}$, $E = \{(a, d), (a, f), (a, h), (b, e), (c, b), (c, e), (d, g), (e, b), (e, d), (e, h), (f, a), (f, b), (f, c), (f, d), (f, e), (f, g), (f, h), (g, h), (h, d), (h, g)\}$. Построить матрицы смежности, достижимости, взаимной достижимости. Найти компоненты сильной связности. ⊛

300. Дан ориентированный граф $\mathfrak{G} = (V, E)$, где $V = \{a, b, c, d, e, f, g, h\}$, $E = \{(a, c), (a, d), (a, e), (a, f), (b, c), (b, g), (b, h), (c, b), (c, h), (d, a), (d, b), (d, g), (e, c), (e, f), (e, g), (e, h), (f, c), (f, e),$

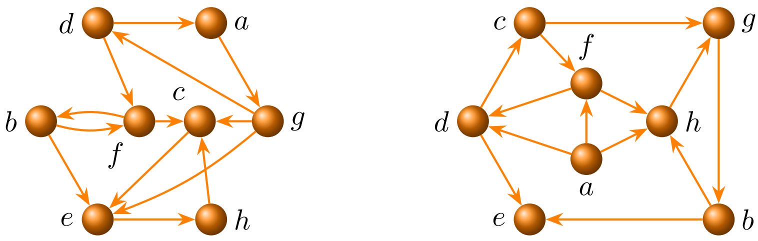


Рис. 9: Графы $\mathfrak{G}_1$ и $\mathfrak{G}_2$.

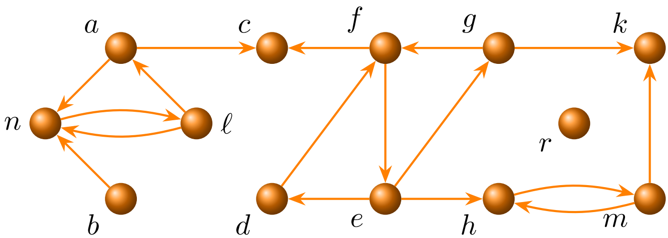
$(f,\ g)$, $(g,\ b)$, $(g,\ h)$, $(h,\ g)\}$. Построить матрицы смежности, достижимости, взаимной достижимости. Найти компоненты сильной связности. ⊛

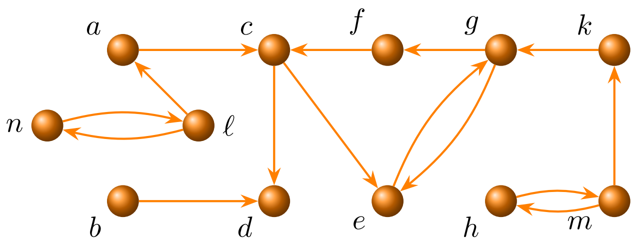
301. Дан ориентированный граф $\mathfrak{G} = (V, E)$, где $V = \{a,\ b,\ c,\ d,\ e,\ f,\ g,\ h\}$, $E = \{(a,\ c),\ (a,\ d),\ (a,\ f),\ (a,\ g),\ (b,\ e),\ (b,\ h),\ (c,\ a),\ (c,\ b),\ (c,\ h),\ (d,\ b),\ (d,\ c),\ (d,\ e),\ (d,\ g),\ (d,\ h),\ (e,\ h),\ (f,\ h),\ (g,\ a),\ (g,\ c),\ (g,\ f),\ (h,\ e)\}$. Построить матрицы смежности, достижимости, взаимной достижимости. Найти компоненты сильной связности. ⊛

302. Для графов $\mathfrak{G}_1$ и $\mathfrak{G}_2$ на рис. 9 определить компоненты сильной связности и отношение строгой достижимости на них. ⊛

303. Определить для каждого из графов $\mathfrak{G}_i$, $i = 1, 2, 3$, изображённых на рис. 10 на следующей странице, компоненты сильной связности, отношение строгой достижимости на них и все базы. ⊛

304. Доказать, что граф является сильно связным тогда и только тогда, когда в нём есть цикл, содержащий все вершины. ⊛





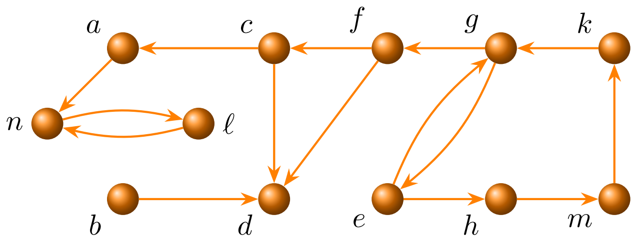


Рис. 10: Графы $\mathfrak{G}_1$, $\mathfrak{G}_2$ и $\mathfrak{G}_3$.

13. Неориентированные графы

Неориентированный граф $\mathfrak{G} = (V, E)$ — это граф, у которого для каждого ребра $(u, v) \in E$ имеется противоположное ребро $(v, u) \in E$, то есть отношение E симметрично. Пара рёбер $\{(v, u), (u, v)\}$ называется в этом случае неориентированным ребром. При разметке неориентированных графов каждое неориентированное ребро получает одну метку.

Матрица смежности неориентированного графа симметрична.

Если $e = (u, v) \in E$, то вершины u и v называются смежными в $\mathfrak{G}$, а ребро e и эти вершины называются инцидентными. Степенью вершины v в неориентированном графе называется количество инцидентных v рёбер, петля у вершины учитывается два раза. Обозначается степень вершины v при помощи $\deg v$. Вершина степени 0 называется изолированной, степени 1 — висячей. Вершина графа называется чётной (соответственно нечётной), если её степень является чётной (соответственно нечётной).

Циклом в неориентированном графе называется путь $(v_1, \ldots, v_n)$, в котором $v_1 = v_n$ и любые два последовательных неориентированных ребра различны.

Неориентированный граф называется связным, если любая пара вершин в нём соединена путём. Для неориентированных графов отношения достижимости и сильной достижимости совпадают.

Компоненты связности в неориентированном графе — это классы эквивалентности по отношению достижимости. Ребро (v, w) неориентированного графа $\mathfrak{G} = (V, E)$ называется мостом $\mathfrak{G}$, если при его удалении из E количество компонент связности графа увеличивается.

Граф называется плоским (или планарным), если его можно изобразить на евклидовой плоскости так, чтобы рёбра попарно не пересекались. Часть плоскости, не содержащая рёбер, но ограниченная со всех сторон (извне или снаружи) рёбрами (если они есть), называется гранью. Вершины и рёбра выпуклого многогранника G образуют плоский граф, граням многогранника соответствуют грани графа.

Если $\mathfrak{G}$ — связный плоский граф, то справедлива формула Эйлера: $e + 2 = v + f$, где e — количество рёбер, v — вершин, f — граней.

Путь (цикл) в графе называется эйлеровым, если он проходит по каждому ребру в точности один раз. Граф в котором есть эйлеров путь, называют полуэйлеровым, граф, в котором есть эйлеров цикл, — эйлеровым.

Связный граф является эйлеровым тогда и только тогда, когда все его вершины чётны. Связный неэйлеров граф $\mathfrak{G}$ является полуэйлеровым тогда и только тогда, когда он содержит в точности две нечётных вершины. При этом эйлеров путь начинается в одной нечётной вершине и заканчивается в другой.

Путь (цикл) в графе $\mathfrak{G}$ называют гамильтоновым, если он проходит по каждой из вершин в точности по одному разу (за исключением первой и последней вершины для цикла, они совпадают по определению). Граф в котором есть гамильтонов цикл, тоже называется гамильтоновым.

Раскраской называют разметку вершин графа такую, что каждые две смежные вершины имеют различные метки. Наименьшее количество цветов, в которое можно раскрасить граф, называется хроматическим числом графа. Граф, который можно раскрасить в два цвета, называется двудольным (или бихроматическим). Граф является двудольным тогда и только тогда, когда в нём нет циклов нечётной длины.

Задачи

305. Найти количество рёбер в n-вершинном неориентированном графе без петель, в котором любая пара различных вершин соединена ребром. $\circledast$

306. Доказать, что сумма степеней всех вершин произвольного неориентированного графа равна удвоенному количеству рёбер. $\circledast$

307. Доказать лемму «о рукопожатиях»: количество нечётных вершин в графе чётно.

Название происходит из следующей интерпретации: при рукопожатиях, которыми обменялись пришедшие на вечеринку гости, количество людей, пожавших руку нечётное количество раз, является чётным. ⊛

308. Может ли в государстве, в котором из каждого города выходит ровно пять дорог, быть ровно 77 дорог между городами? Может ли быть 80 дорог? ⊛

309. Перечислить все неизоморфные неориентированные графы без петель, у которых не более четырёх вершин. ⊛

310. Доказать, что ребро является мостом тогда и только тогда, когда оно не входит ни в какой простой цикл. ⊛

311. Пусть неориентированный граф $\mathfrak{G} = (V, E)$ связен. Доказать, что следующие два условия эквивалентны:

(а) граф $\mathfrak{G}$ не имеет мостов;

(б) существует ориентация рёбер графа $\mathfrak{G}$, при которой в нём будет только одна компонента сильной связности. ⊛

312. Доказать, что неориентированный связный граф с n вершинами

(а) содержит $n - 1$ или более рёбер;

(б) если содержит n или более рёбер, то в графе имеется как минимум один цикл. ⊛

313. Доказать, что во всякой группе V из шести человек есть трое попарно знакомых или трое попарно незнакомых. Переформулировать указанную задачу в терминах графов. ⊛

314. Привести пример, показывающий, что для пяти человек утверждение из предыдущей задачи может не выполняться. ⊛

315. Доказать, что во всяком графе V из девяти вершин без петель есть четверо попарно соединённых или трое попарно не соединённых. *Указание.* Рассмотреть два случая: (а) когда есть вершина степени не больше 4, (б) когда её нет. ⊛

316. Доказать, что во всяком графе V из 18 вершин без петель есть четыре попарно соединённых вершины или четыре попарно не соединённых. ⊛

317. Доказать, что неориентированный граф $\mathfrak{G} = (V, E)$ связен тогда и только тогда, когда для каждого разбиения $V = V_1 \cup V_2$ с непустыми

V_1 и V_2 существует ребро, соединяющее какую-то вершину из V_1 с какой-то вершиной из V_2. ⊛

318. Доказать, что если в неориентированном графе имеется ровно две нечётные вершины, то они связаны путём. ⊛

319. Чему равно количество компонент связности неориентированного графа ⊛

$$\mathfrak{G} = (\{1,2,3,4,5,6,7,8,9\}, \{(1,4),(2,7),(3,9),(7,4),(1,5),(6,7)\})?$$

320. Доказать, что в связном неориентированном графе любые два простых пути максимальной длины имеют общую вершину. ⊛

321. (а) Доказать, что если неориентированный граф $\mathfrak{G} = (V, E)$ не является связным графом, то его дополнение, то есть граф $\bar{\mathfrak{G}} = (V, \bar{E})$, является связным (здесь $\bar{E} = V^2 \setminus E$).

(б) У задачи из (а) имеется следующая популярная интерпретация. В стране Приозерия каждая пара городов соединена в точности одним транспортным маршрутом: или водным, или автобусным. Доказать, что существует вид транспорта, которым можно доехать из любого города страны в любой другой (возможно с пересадками). ⊛

322. Пусть $\mathfrak{G} = (V, E)$ — неориентированный граф. Представлением графа $\mathfrak{G}$ пересечениями называется пара (A, f), где A — произвольное множество, а $f : V \to \mathsf{P}(A)$ — разнозначная функция такая, что для любых различных вершин u и v множества $f(u)$ и $f(v)$ пересекаются тогда и только тогда, когда $(u, v) \in E$. Доказать, что для каждого графа такое представление существует. ⊛

323. Пусть неориентированный граф без петель $\mathfrak{G} = (V, E)$ имеет k компонент связности. Доказать, что тогда ⊛

$$|E| \leqslant \frac{(|V| - k)(|V| - k + 1)}{2}.$$

324. Определить, какое наименьшее количество рёбер необходимо добавить к связному графу, чтобы он мог стать эйлеровым или полуэйлеровым. ⊛

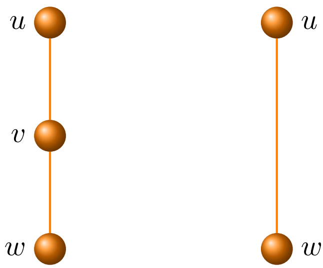


Рис. 11: Исключение вершины v.

325. Для геодезических исследований территорию триангулируют: разбивают на треугольные части. В вершинах треугольников устанавливают специальные знаки — вышки. Определить, сколько всего вышек потребуется для триангуляции, если территория разделена на N треугольников, а на её границе располагается K вышек. ⊛

326. Определить, сколько рёбер и вершин будет иметь многогранник, у которого f граней и все они треугольные. Для каких f такие многогранники могут существовать? ⊛

327. Исключением вершины называется такая операция с графом. Если есть вершина v степени 2 с инцидентными рёбрами (v, u) и (v, w), $u \neq w$, то мы удаляем вершину v и инцидентные рёбра, а вместо них добавляем ребро (u, w), рис. 11. Доказать, что при исключении вершин планарность и эйлеровость графа не меняются, а хроматическое число и гамильтоновость могут измениться. ⊛

328. Для каждого из пяти правильных многогранников изобразить плоский граф, образованный его вершинами и рёбрами. Определить, каким будет этот граф: эйлеровым, гамильтоновым, найти хроматическое число. ⊛

329. Доказать, что не существует многогранника, у которого все грани являются шестиугольными. ⊛

330. Назовём многогранник однородным, если все его вершины имеют одну и ту же степень, а все грани имеют одно и то же количество сторон. Доказать, что граф рёбер любого однородного многогранника совпадает с графом одного из правильных многогранников. ⊛

331. Показать, что наличие в графе эйлерова и гамильтонова циклов друг от друга не зависит. ⊛

332. Определить, является ли следующий граф $\mathfrak{G} = (V, E)$ эйлеровым (полуэйлеровым). Если $\mathfrak{G}$ не является эйлеровым, то удалить из $\mathfrak{G}$ минимальное количество рёбер, чтобы он им стал. Построить в исходном или в получившемся графе эйлеров цикл. $V = \{a,\ b,\ c,\ e,\ f,\ g,\ h,\ k,\ m,\ n\}$, $E = \{(a,\ c),\ (a,\ h),\ (a,\ m),\ (a,\ k),\ (b,\ c),\ (b,\ k),\ (b,\ f),\ (b,\ m),\ (c,\ k),\ (c,\ m),\ (e,\ f),\ (e,\ g),\ (f,\ k),\ (f,\ n),\ (g,\ m),\ (g,\ h),\ (h,\ k),\ (h,\ m),\ (k,\ n)\}$. ⊛

333. Определить, является ли следующий граф $\mathfrak{G} = (V, E)$ двудольным. Если $\mathfrak{G}$ не двудольный, то удалить из $\mathfrak{G}$ наименьшее количество рёбер так, чтобы $\mathfrak{G}$ стал двудольным. $V = \{a,\ b,\ c,\ e,\ f,\ g,\ h,\ k,\ m,\ n\}$, $E = \{(a,\ h),\ (a,\ n),\ (a,\ k),\ (b,\ k),\ (b,\ f),\ (b,\ m),\ (c,\ k),\ (c,\ h),\ (e,\ f),\ (e,\ g),\ (f,\ a),\ (f,\ m),\ (g,\ m),\ (m,\ n)\}$. ⊛

334. Плоский граф $\mathfrak{G}$ с $n+1$ вершиной имеет вид правильного n-угольника, некоторые из вершин которого соединены с центром. Найти хроматическое число такого графа. ⊛

14. Деревья

Связный неориентированный граф $\mathfrak{T} = (V, E)$ без циклов называется (неориентированным) д е р е в о м. Эквивалентные определения неориентированного дерева таковы:

- для любых двух вершин в графе $\mathfrak{T}$ есть единственный соединяющий их путь;
- граф $\mathfrak{T}$ связен, но при удалении из E любого ребра граф перестаёт быть связным;
- граф $\mathfrak{T}$ связен и $|E| = |V| - 1$;
- граф $\mathfrak{T}$ ациклический и $|E| = |V| - 1$;
- граф $\mathfrak{T}$ ациклический, но добавление любого ребра к E порождает в графе цикл.

Ориентированный граф $\mathfrak{T} = (V, E)$ называется (ориентированным) д е р е в о м, если

1) в нём есть в точности один исток $r \in E$ — к о р е н ь дерева;
2) в каждую из остальных вершин входит ровно по одному ребру;
3) все вершины достижимы из корня.

Эквивалентное индуктивное определение ориентированного дерева:

- граф $\mathfrak{T} = (V, E)$, с единственной вершиной $V = \{v\}$ и пустым множеством рёбер $E = \varnothing$ является деревом. Вершина v называется корнем этого дерева;
- пусть графы $\mathfrak{T}_1 = (V_1, E_1), \ldots, \mathfrak{T}_k = (V_k, E_k)$ являются ориентированными деревьями с корнями $r_1, \ldots, r_k$ соответственно, множества вершин $V_1, \ldots, V_k$ попарно не пересекаются, а r_0 — новая вершина, не принадлежащая ни одному из $V_1, \ldots, V_k$. Тогда граф $\mathfrak{T} = (V, E)$

является ориентированным деревом с корнем r_0, где

$$V = \{r_0\} \cup \bigcup_{i=1}^{k} V_i, \qquad E = \{(r_0, r_i) : i = 1, \ldots, k\} \cup \bigcup_{i=1}^{k} E_i.$$

Стоки ориентированного дерева называют *листьями*. Путь из корня в лист называется *ветвью* дерева.

Глубина вершины v — это длина (единственного) пути из корня в v. *Высота вершины* v — это максимальная из длин путей из v в какой-либо лист. *Высота дерева* — это максимальная из длин его ветвей, эквивалентно: максимальная глубина вершины или максимальная высота вершины. Граф, являющийся объединением нескольких непересекающихся деревьев, называется *лесом*.

Для фиксированного множества из n вершин на них можно построить n^{n-1} ориентированных и n^{n-2} неориентированных деревьев (*теорема Кэли*).

В ориентированном дереве $\mathfrak{T} = (V, E)$

- вершина v называется *отцом* вершины u, а u — *сыном* вершины v, если $(v, u) \in E$;
- вершина v называется *предком* вершины u, а u — *потомком* вершины v, если есть путь из v в u;
- вершины с общим отцом называются *братьями* или *сёстрами*.

Ориентированное дерево называется *бинарным* (или *двоичным*), если каждая внутренняя вершина имеет не более двух сыновей, причём ведущие к ним рёбра помечены двумя разными метками. Обычно используются метки из пар: левый/правый, 0/1, +/−, да/нет и т. д. Бинарное дерево называется *полным*, если все его ветви имеют одинаковую длину.

Префиксный (или *прямой*) обход ориентированного дерева основан на принципе: «сначала родитель, затем дети». Определим индукцией по построению дерева $\mathfrak{T}$ его прямое представление $\mathrm{PRF}(\mathfrak{T})$:

- Если $\mathfrak{T} = (\{v\}, \varnothing)$, то $\mathrm{PRF}(\mathfrak{T}) = (v)$.
- Если $\mathfrak{T}$ получено из деревьев $\mathfrak{T}_1, \ldots, \mathfrak{T}_k$ и нового корня r_0, то

$$\mathrm{PRF}(\mathfrak{T}) = (r_0^{(k)}, \mathrm{PRF}(\mathfrak{T}_1), \ldots, \mathrm{PRF}(\mathfrak{T}_k)).$$

Индекс k у корня r_0 необходим для разнозначности префиксного представления.

С у ф ф и к с н ы й (или о б р а т н ы й) обход дерева основан на противоположном принципе: «сначала дети, затем родитель». Его индуктивное определение:

- Если $\mathfrak{T} = (\{v\}, \varnothing)$, то $\mathrm{SFF}(\mathfrak{T}) = (v)$.
- Если $\mathfrak{T}$ получено из деревьев $\mathfrak{T}_1, \ldots, \mathfrak{T}_k$ и нового корня r_0, то

$$\mathrm{SFF}(\mathfrak{T}) = (\mathrm{SFF}(\mathfrak{T}_1), \ldots, \mathrm{SFF}(\mathfrak{T}_k), r_0^{(k)}).$$

И н ф и к с н ы й (или в н у т р е н н и й) обход бинарного дерева основан на принципе: «сначала левый сын, затем родитель, а затем правый сын»:

- Если $\mathfrak{T} = (\{v\}, \varnothing)$, то $\mathrm{INF}(\mathfrak{T}) = (v)$.
- Если $\mathfrak{T}$ получено из деревьев $\mathfrak{T}_1$ и $\mathfrak{T}_2$ и нового корня r_0, то

$$\mathrm{INF}(\mathfrak{T}) = (\mathrm{INF}(\mathfrak{T}_1), r_0, \mathrm{INF}(\mathfrak{T}_2)).$$

  Здесь мы полагаем, что корень $\mathfrak{T}_1$ является левым сыном r_0, а корень $\mathfrak{T}_2$ — правым. Если $\mathfrak{T}_i$ отсутствует, то последовательность $\mathrm{INF}(\mathfrak{T}_i)$ пуста.

Задачи

335. Определить, является ли неориентированное дерево плоским, эйлеровым, полуэйлеровым, гамильтоновым графом? Найти его хроматическое число. ⊛

336. Найти формулу, связывающую количества вершин и рёбер для леса из k неориентированных деревьев. ⊛

337. Доказать, что в неориентированном дереве $\mathfrak{T} = (V, E)$, содержащем хотя бы одно ребро, количество висячих вершин равняется $2 + \sum\limits_{v \in V_2} (\deg v - 2)$, где V_2 — множество невисячих вершин. ⊛

338. Доказать, что если в неориентированном дереве имеется ровно две висячих вершины, то оно является «линией». ⊛

339. Доказать, что если в связном неориентированном графе количество вершин равно количеству рёбер, то можно выбросить одно из рёбер так, что после этого граф станет деревом. ⊛

340. Ц е н т р о м неориентированного графа $\mathfrak{G}$ называется вершина v, для которой длина максимального пути от неё до остальных вершин минимальна. Доказать, что в дереве (а) все центры смежные;

(б) более двух центров существовать не может. ⊛

341. Доказать, что в неориентированном дереве существует вершина, через которую проходят все максимальные простые пути. ⊛

342. Пусть $\mathfrak{T} = (V, E)$ — неориентированное дерево, $v \in V$ — произвольная его вершина. Для каждого ребра $(u, w) \in E$ выберем ориентацию от u к w, если расстояние от v до u меньше, чем от v до w. Доказать, что полученный ориентированный граф будет ориентированным деревом с корнем v. ⊛

343. Доказать следующее утверждение двумя способами: если в неориентированном дереве $\mathfrak{T} = (V, E)$ имеется вершина v степени $d > 1$, то в нём имеется по крайней мере d висячих вершин. ⊛

344. Для множества вершин $V = \{v_1, \ldots, v_n\}$ определить, сколько существует неориентированных деревьев, в которых

(а) вершина v_1 является висячей;

(б) обе вершины v_1 и v_2 являются висячими;

(в) все вершины $v_1, \ldots, v_k$ являются висячими;

(г) ровно две вершины являются висячими. ⊛

345. Пусть $\mathfrak{T} = (V, E)$ — это ориентированное дерево с корнем $v_0 \in V$. Определим для каждой вершины $v \in V$ подграф $\mathfrak{T}_v = (V_v, E_v)$ следующим образом: V_v — это множество вершин, достижимых из v в $\mathfrak{T}$, а E_v — это множество рёбер из E, оба конца которых входят в V_v. Доказать, что

(а) $\mathfrak{T}_v$ является деревом с корнем v;

(б) если две разные вершины v и u имеют одинаковую глубину, то деревья $\mathfrak{T}_v$ и $\mathfrak{T}_u$ не пересекаются. ⊛

346. Привести пример ориентированного графа, для которого выполнены первые два условия из определения дерева, но который не является деревом. ⊛

347. Некоторый слух распространялся следующим образом: его источник позвонил трём своим друзьям, каждый из них передал по телефону этот слух четырём своим друзьям, а каждый из них, в свою очередь, передал его пяти своим друзьям. Определить, сколько человек узнали слух, если никто из них не получил более одного звонка и никто не звонил источнику слуха? Найти, сколько всего звонков было произведено? ⊛

348. Пусть $\leqslant$ — отношение частичного порядка на конечном множестве V, которое обладает следующими двумя свойствами:

(а) существует наименьший элемент r;

(б) если элементы x и y множества V несравнимы, $a \geqslant x$, $b \geqslant y$, то a и b тоже несравнимы.

Бинарное отношение $E(x, y)$ на множестве V означает, что x — это максимальный из элементов, меньших y. Доказать, что граф (V, E) — это ориентированное дерево с корнем r. ⊛

349. Пусть $\mathfrak{T} = (V, E)$ — ориентированное дерево, а $x \leqslant y$ означает, что вершина y достижима из вершины x. Доказать, что $\leqslant$ — отношение нестрогого частичного порядка на V, удовлетворяющее свойствам (а) и (б) из предыдущей задачи. ⊛

350. Пусть $\mathfrak{G} = (V, E)$ — это ориентированный граф с не менее чем двумя вершинами. Доказать, что граф $\mathfrak{G}$ является (ориентированным) деревом тогда и только тогда, когда в $\mathfrak{G}$ нет циклов, имеется один исток r, а в каждую из остальных вершин $v \in V \setminus \{r\}$ входит ровно одно ребро. ⊛

351. Пусть $\mathfrak{G} = (V, E)$ — ориентированный граф. Доказать, что $\mathfrak{G}$ является (ориентированным) деревом тогда и только тогда, когда в $\mathfrak{G}$ есть вершина r (корень) такая, что в любую вершину v из r ведёт в точности один путь. ⊛

352. Пусть корень ориентированного дерева $\mathfrak{T}$ имеет пять сыновей, а каждая из остальных внутренних вершин имеет три или четыре сына, при этом количество вершин с тремя сыновьями вдвое больше количества вершин с четырьмя. Сколько всего вершин и рёбер в $\mathfrak{T}$, если известно, что количество его листьев равно 26? ⊛

353. Пусть корень ориентированного дерева $\mathfrak{T}$ имеет трёх сыновей, а каждая из остальных внутренних вершин имеет два или четыре сына, при этом количество вершин с двумя сыновьями вдвое меньше количества вершин с четырьмя. Сколько всего вершин в $\mathfrak{T}$, если известно, что количество его листьев равно 38? ⊛

354. Пусть $\mathfrak{F} = (\mathfrak{T}_1, \ldots, \mathfrak{T}_n)$ — лес деревьев. Доказать, что по последовательности $P = (\mathrm{PRF}(\mathfrak{T}_1), \ldots, \mathrm{PRF}(\mathfrak{T}_k))$ можно однозначно восстановить лес $\mathfrak{F}$. Доказать аналогичное утверждение для суффиксного обхода. ⊛

355. Доказать по индукции, что в каждом бинарном дереве количество n вершин с двумя сыновьями на единицу меньше количества ℓ листьев. ⊛

356. Найти количество листьев и вершин в полном бинарном дереве высоты h. ⊛

357. Построить дерево и ациклический ориентированный граф, представляющие следующую арифметическую формулу

$$\Phi = (a+b)/(c+a\times d) + ((c+a\times d) - (a+b)\times(c-d)).$$

Сколько вершин удалось сократить? ⊛

358. Построить дерево, представляющее следующую логическую формулу

$$\Psi = ((x \vee \neg y) \wedge \neg(z \to (x \wedge y))) \vee (\neg z \oplus y).$$

Для полученного дерева построить префиксный, суффиксный и инфиксный обходы (для вершин с отрицаниями единственного сына считать правым). ⊛

359. Определить префиксный, суффиксный и инфиксный обходы дерева $\mathfrak{T}_1$, изображённого на рис. 12 на следующей странице, считая, что рёбра, исходящие из одной вершины, пронумерованы слева направо. ⊛

360. Определить префиксный и суффиксный обходы дерева $\mathfrak{T}_2$, изображённого на рис. 13 на противоположной странице, считая, что рёбра, исходящие из одной вершины, пронумерованы слева направо. ⊛

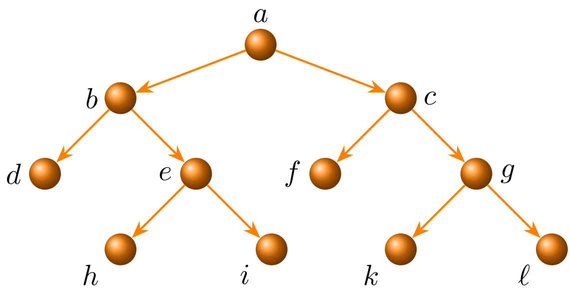


Рис. 12: Дерево $\mathfrak{T}_1$.

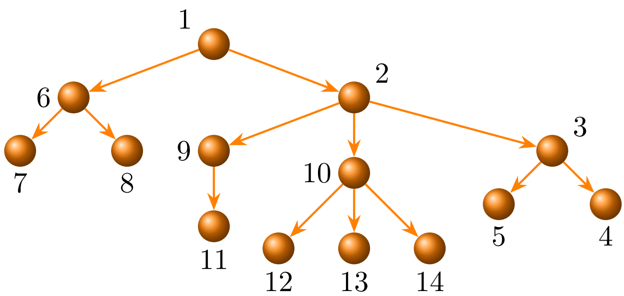


Рис. 13: Дерево $\mathfrak{T}_2$.

361. Пусть $(x, z, u, v, /, -, \times, y, z, -, x, y, +, \times, +)$ — это суффиксный обход дерева арифметической формулы, составленной из переменных x, y, z, u, v и знаков операций $+, -, \times, /$. Восстановить это дерево и формулу. ⊛

15. Алгоритмы на графах

Граф $\mathfrak{G}_1 = (V_1, E_1)$ называется *подграфом* графа $\mathfrak{G} = (V, E)$, если $V_1 \subseteq V$ и $E_1 \subseteq E$. *Остовным деревом* (неориентированного) связного графа $\mathfrak{G} = (V, E)$ называется его подграф $\mathfrak{S} = (V, T)$, являющийся деревом.

Пусть задана функция разметки рёбер $c : E \to \mathbb{R}$, приписывающая каждому ребру $e \in E$ некоторое число $c(e)$ — стоимость (вес, длину). Тогда *стоимость* $c(\mathfrak{S})$ дерева $\mathfrak{S}$ — это сумма стоимостей всех его рёбер, то есть $c(\mathfrak{S}) = \sum\limits_{e \in T} c(e)$. *Минимальным* остовным деревом называется остовное дерево минимальной стоимости. Минимальное остовное дерево может быть построено при помощи алгоритма Крускала (алгоритм МинОД, [3], § 12.1).

Другой способ построения остовного дерева — *поиск в глубину* (алгоритм ПОГ, [3], § 12.2). Дерево $\mathfrak{S} = (V, T)$, которое строится алгоритмом поиска в глубину, называется *глубинным* остовным деревом графа $\mathfrak{G} = (V, E)$. Рёбра графа $\mathfrak{G}$, попавшие в множество T, называются *прямыми*, а не попавшие — *обратными*.

Пусть $Num[w]$ — номер, который алгоритм поиска в глубину присваивает вершине w. Число $Up[w]$ — это минимум из $Num[w]$ и наименьшего из номеров вершин, к которым ведут обратные рёбра от w и её потомков. Ребро (v, w) глубинного остовного дерева $\mathfrak{S} = (V, T)$ является мостом графа $\mathfrak{G}$ тогда и только тогда, когда $Up[w] = Num[w]$.

Пусть $\mathfrak{G} = (V, E)$ — ориентированный граф, $c(e) \geqslant 0$ — длина (вес, стоимость) ребра $e \in E$. Тогда длина пути $p = (v_0, v_1, \ldots, v_k)$ — это сумма длин рёбер, входящих в этот путь: $c(p) = c(v_0, v_1) + \cdots + c(v_{k-1}, v_k)$. Если в $\mathfrak{G}$ имеется путь из вершины a в вершину b, то имеется и такой путь минимальной длины. Он называется *кратчайшим* путём из a в b. Для построения кратчайших путей из заданной вершины a во все остальные применяется алгоритм Дейкстры (алгоритм КрПути, [3], § 12.3).

Задачи

362. Построить минимальное остовное дерево для неориентированного графа $\mathfrak{G} = (V, E)$ с множествами вершин $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$ и рёбер $E = \{(v_1, v_2; 18), (v_1, v_3; 2), (v_3, v_2; 4), (v_3, v_4; 6), (v_3, v_5; 8), (v_4, v_6; 5), (v_5, v_4; 4), (v_6, v_1; 7), (v_6, v_8; 4), (v_6, v_7; 3), (v_7, v_5; 1), (v_7, v_8; 7), (v_8, v_1; 5), (v_8, v_9; 3), (v_9, v_1; 1)\}$. Третий параметр в скобках — вес ребра. ⊛

363. Найти минимальное остовное дерево для неориентированного графа $\mathfrak{G} = (V, E)$, в котором $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$ и $E = \{(v_1, v_2; 10), (v_1, v_4; 6), (v_2, v_3; 12), (v_2, v_4; 3), (v_3, v_5; 20), (v_3, v_6; 11), (v_4, v_5; 3), (v_4, v_7; 2), (v_5, v_6; 4), (v_5, v_7; 2), (v_5, v_8; 5), (v_5, v_9; 9), (v_6, v_8; 6), (v_6, v_9; 17), (v_8, v_9; 7)\}$. ⊛

364. Пусть e — ребро максимального веса в некотором простом цикле C графа $\mathfrak{G} = (V, E)$. Доказать, что

(а) существует минимальное остовное дерево графа $\mathfrak{G}' = (V, E \setminus \{e\})$, которое является и минимальным остовным деревом графа $\mathfrak{G}$;

(б) если в C нет других рёбер такого же веса, то никакое минимальное остовное дерево не содержит e. ⊛

365. Пусть $\mathfrak{T} = (V, T)$ — минимальное остовное дерево для нагруженного неориентированного графа $\mathfrak{G} = (V, E)$ с n вершинами, c — разметка рёбер. Пусть $(e_1, e_2, \ldots, e_{n-1})$ — это последовательность рёбер из T, упорядоченных по возрастанию $c(e_i)$. Пусть $\mathfrak{T}'$ — произвольное остовное дерево для $\mathfrak{G}$ с рёбрами $(d_1, d_2, \ldots, d_{n-1})$, упорядоченными по возрастанию $c(d_i)$. Показать, что $c(e_i) \leqslant c(d_i)$ для всех $i = 1, \ldots, n-1$.

Указание. Рассмотреть лес деревьев $\mathfrak{T}_{i,m}$ из рёбер e_j, $j < i$, и рёбра d_ℓ, $\ell \leqslant i$. ⊛

366. Пусть все рёбра дерева имеют попарно различные веса. При каком условии в алгоритме МинОД минимальное остовное дерево будет построено на самом последнем шаге? ⊛

367. Доказать корректность следующего алгоритма построения минимального остовного дерева. Пока в графе есть циклы, выбирать некоторый простой цикл и выкидывать одно из рёбер наибольшего веса, среди входящих в этот цикл. ⊛

368. Пусть задан неориентированный нагруженный граф $\mathfrak{G} = (V, E)$, в котором $V = \{a,\ b,\ c,\ d,\ e,\ f,\ g,\ h,\ k\}$ и $E = \{(a,\ b;\ 10),\ (a,\ c;\ 7),$ $(b,\ f;\ 21),\ (b,\ d;\ 9),\ (c,\ d;\ 8),\ (d,\ e;\ 21),\ (f,\ e;\ 7),\ (f,\ g;\ 8),\ (e,\ k;\ 12),$ $(e,\ h;\ 10),\ (g,\ h;\ 8)\}$. Какие из рёбер не могут попасть ни в какое минимальное остовное дерево? ⊛

369. Доказать корректность следующего алгоритма построения минимального остовного дерева, предложенного независимо В. Ярником, Р. Примом и Э. Дейкстрой.

(а) Выбрать произвольным образом вершину a, она сама по себе является деревом (без рёбер).

(б) На каждом шаге добавлять к имеющемуся дереву ребро наименьшего веса, одна из вершин которого принадлежит дереву, а вторая — нет. ⊛

370. Пусть $\mathfrak{T} = (V, T)$ — это глубинное остовное дерево, построенное алгоритмом обхода в глубину для графа $\mathfrak{G} = (V, E)$. Доказать, что для каждого обратного ребра $(u, v) \in E \setminus T$ или u является предком v в $\mathfrak{T}$, или v является предком u в $\mathfrak{T}$. ⊛

371. Модифицировать алгоритм поиска в глубину так, чтобы он вычислял $Up[v]$ и распечатывал список всех мостов графа. ⊛

372. Обойти (занумеровать) вершины заданного неориентированного графа $\mathfrak{G} = (V, E)$ с помощью алгоритма обхода в глубину, начиная с вершины v_1, и построить дерево этого обхода. $V = \{v_1,\ v_2,\ v_3,\ v_4,\ v_5,$ $v_6,\ v_7,\ v_8,\ v_9,\ v_{10}\}$, $E = \{(v_1,\ v_4),\ (v_1,\ v_5),\ (v_1,\ v_8),\ (v_1,\ v_{11}),\ (v_2,\ v_9),$ $(v_3,\ v_6),\ (v_3,\ v_7),\ (v_3,\ v_9),\ (v_3,\ v_{11}),\ (v_5,\ v_9),\ (v_5,\ v_{10}),\ (v_6,\ v_8),\ (v_7,\ v_8),$ $(v_9,\ v_{10})\}$. Какое обратное ребро $e \in E \setminus T$ и цикл в $\mathfrak{G}$ обнаружились в этом обходе первыми? Вычислить для каждой вершины v значение $Up[v]$ и определить все мосты графа $\mathfrak{G}$. Считать, что все списки смежности упорядочены по возрастанию номера вершины. ⊛

373. Пусть задан неориентированный граф $\mathfrak{G} = (V, E)$ с множеством вершин $V = \{a,\ b,\ c,\ d,\ e,\ f,\ g,\ h,\ i\}$ и множеством рёбер $E = \{(a,\ b),$ $(a,\ c),\ (b,\ d),\ (b,\ c),\ (d,\ e),\ (d,\ f),\ (f,\ g),\ (f, h), (f, i), (g, i)\}$. Используя вариант поиска в глубину, начиная с вершины a, с подсчётом функции Up, определить все мосты этого графа. Считать, что списки смежности упорядочены по алфавиту. ⊛

374. Начиная с вершины a, обойти (занумеровать) вершины неориентированного графа $\mathfrak{G} = (V, E)$ с помощью алгоритма обхода в глубину, начиная с вершины v_1, и построить дерево $\mathfrak{T} = (V, T)$ этого обхода. Граф $\mathfrak{G}$ имеет множество вершин $V = \{v_1,\ v_2,\ v_3,\ v_4,\ v_5,\ v_6,\ v_7,\ v_8,\ v_9,\ v_{10},\ v_{11}\}$ и множество рёбер $E = \{(v_1,\ v_2),\ (v_1,\ v_4),\ (v_1,\ v_8),\ (v_2,\ v_9),\ (v_2,\ v_4),\ (v_3,\ v_6),\ (v_3,\ v_7),\ (v_3,\ v_8),\ (v_5,\ v_6),\ (v_6,\ v_7),\ (v_7,\ v_8),\ (v_9,\ v_{10}),\ (v_9,\ v_{11}),\ (v_{10},\ v_{11})\}$. Какое обратное ребро $e \in E \setminus T$ и цикл в $\mathfrak{G}$ обнаружились в этом обходе первыми? Вычислить для каждой вершины v значение $Up(v)$ и определить все мосты графа $\mathfrak{G}$. Считать, что списки смежности упорядочены по возрастанию номера. ⊛

375. Обойти (занумеровать) вершины заданного неориентированного графа $\mathfrak{G} = (V, E)$ с помощью алгоритма обхода в глубину, начиная с вершины a, и построить дерево $\mathfrak{T} = (V, T)$ этого обхода. Граф задан списками смежности:

$$(a(e,f,h), b(e,h), c(d,f,g,i), d(c), e(a,b,h),$$
$$f(a,c,g,i), g(c,f), h(a,b,e), i(c,f)).$$

Какое обратное ребро $e \in E \setminus T$ и цикл в $\mathfrak{G}$ обнаружились в этом обходе первыми? Вычислить для каждой вершины v значение $Up[v]$ и определить все мосты графа $\mathfrak{G}$. ⊛

376. Изменить алгоритм обхода в глубину так, чтобы он позволил перечислить все связные компоненты неориентированного графа. ⊛

377. Если в ориентированном графе $\mathfrak{G} = (V, E)$ существует вершина r, из которой достижимы все остальные, то для него тоже можно определить понятие остовного (теперь ориентированного) дерева: дерево $\mathfrak{T} = (V, T)$ с корнем r. Определить, какие из алгоритмов построения остовного дерева можно приспособить для построения остовного дерева в ориентированном графе: алгоритм Крускала, алгоритм Ярника-Прима-Дейкстры, алгоритм поиска в глубину. ⊛

378. Определить для следующего нагруженного графа $\mathfrak{G} = (V, E)$ и вершины $a \in V$ длины кратчайших путей из a в остальные вершины $\mathfrak{G}$ и построить дерево этих путей. Здесь $V = \{a,\ b,\ c,\ d,\ e,\ f\}$, $E = \{(a,\ b;\ 154),\ (a,\ c;\ 17),\ (a,\ d;\ 214),\ (a,\ e;\ 63),\ (b,\ d;\ 25),\ (c,\ e;\ 33),\ (c,\ d;\ 192),\ (c,\ b;\ 123),\ (d,\ f;\ 5),\ (e,\ f;\ 140),\ (d,\ e;\ 10)\}$. ⊛

$$\left(\begin{array}{c|cccccc}
 & a & b & c & d & e & f \\
\hline
a & - & 25 & 5 & 26 & 53 & 75 \\
b & 12 & - & - & - & 120 & 40 \\
c & - & 15 & - & 20 & 47 & 60 \\
d & - & - & - & - & 20 & 45 \\
e & - & - & 75 & 20 & - & 20 \\
f & 40 & 15 & 15 & 26 & - & -
\end{array}\right)$$

Рис. 14: Матрица из задачи 379.

379. Задан размеченный граф $\mathfrak{G} = (V, E)$, в котором $V = \{a,\ b,\ c,\ d,\ e,\ f\}$, а длины рёбер заданы матрицей на рис. 14, «$-$» означает отсутствие ребра. Используя алгоритм Дейкстры, построить для $\mathfrak{G}$ дерево $\mathfrak{T}$ кратчайших путей из вершины a во все остальные. Определить сумму длин всех рёбер дерева $\mathfrak{T}$. $\circledast$

380. Дан ориентированный граф $\mathfrak{G} = (V, E)$, где $V = \{a,\ b,\ c,\ d,\ e,\ f,\ g,\ h\}$, $E = \{(a,\ b;\ 3),\ (a,\ g;\ 5),\ (b,\ a;\ 4),\ (b,\ d;\ 3),\ (b,\ h;\ 2),\ (c,\ g;\ 1),\ (c,\ h;\ 6),\ (d,\ c;\ 2),\ (d,\ e;\ 4),\ (d,\ h;\ 4),\ (e,\ c;\ 12),\ (e,\ d;\ 1),\ (f,\ c;\ 12),\ (f,\ d;\ 13),\ (f,\ g;\ 15),\ (f,\ h;\ 2),\ (h,\ a;\ 5),\ (h,\ e;\ 5)\}$ (третий элемент указывает вес соответствующего ребра). С помощью алгоритма Дейкстры по шагам найти кратчайшие пути из вершины f во все остальные. $\circledast$

381. Дан ориентированный граф $\mathfrak{G} = (V, E)$, где $V = \{a,\ b,\ c,\ d,\ e,\ f,\ g,\ h\}$, $E = \{(a,\ e;\ 11),\ (a,\ g;\ 100),\ (b,\ f;\ 46),\ (b,\ g;\ 16),\ (c,\ e;\ 71),\ (c,\ f;\ 16),\ (c,\ g;\ 111),\ (c,\ h;\ 21),\ (d,\ a;\ 8),\ (d,\ b;\ 429),\ (d,\ h;\ 3),\ (e,\ a;\ 1),\ (e,\ g;\ 25),\ (f,\ b;\ 8),\ (g,\ b;\ 2),\ (h,\ a;\ 28),\ (h,\ b;\ 154),\ (h,\ c;\ 7),\ (h,\ d;\ 9),\ (h,\ f;\ 126)\}$ (третий элемент указывает вес соответствующего ребра). С помощью алгоритма Дейкстры по шагам найти кратчайшие пути из вершины h во все остальные. $\circledast$

382. Доказать, что на каждом шаге алгоритма Дейкстры любой кратчайший путь из исходной вершины в любую вершину множества R проходит только через вершины множества R. $\circledast$

383. Доказать, что алгоритм Дейкстры не может использоваться в исходном виде, если веса рёбер графа могут быть отрицательными:

привести пример графа с отрицательными весами рёбер, для которого алгоритм Дейкстры даёт неверный ответ. ⊛

384. Сколько раз может меняться для одной вершины v значение $D[v]$ в ходе работы алгоритма Дейкстры для графа с шестью вершинами? Привести пример на каждый возможный случай. ⊛

385. Пусть в графе $\mathfrak{G}$ выбрана вершина a и для каждой вершины $v \in V$, достижимой из a, существует единственный кратчайший путь из a в v. Доказать, что рёбра всех этих путей образуют ориентированное дерево с корнем a. ⊛

16. Схемы из функциональных элементов

Зафиксируем некоторые множества переменных $\boldsymbol{V}$ и булевых функций $\mathcal{B}$, последнее далее будем называть базисом (это не имеет отношения к понятию базис, связанному с полнотой множества булевых функций). Схема из функциональных элементов в базисе $\mathcal{B}$ — это размеченный ориентированный мультиграф без циклов $\mathfrak{S}$, в котором

1) истоки $\mathfrak{S}$ называются входами, каждый из них помечен некоторой переменной из множества $\boldsymbol{V}$. При этом разные истоки должны иметь разные метки;
2) каждая из остальных вершин v помечена какой-то функцией $F_v \in \mathcal{B}$, количество входящих в неё рёбер равно n — местности F_v, а сами эти рёбра пронумерованы числами от 1 до n. Такие вершины называются функциональными элементами.

Далее мы будем использовать базис $\mathcal{B}_0 = \{\wedge, \vee, \neg\}$.

Глубина вершины v в схеме $\mathfrak{S}$ — это максимальная длина пути из входов $\mathfrak{S}$ в v. Глубиной $D(\mathfrak{S})$ схемы $\mathfrak{S}$ называется максимальная из глубин её вершин.

Пусть входы схемы $\mathfrak{S}$ помечены переменными $x_1, \ldots, x_n$. С каждой вершиной v схемы $\mathfrak{S}$ свяжем булеву функцию $f_v(x_1, \ldots, x_n)$, реализуемую в этой вершине. Определим f_v индукцией по глубине v:

- если вершина v имеет глубину 0, то она является входом, она помечена некоторой переменной x_i. Тогда $f_v(x_1, \ldots, x_n) = x_i$;
- если вершина v имеет глубину $k > 0$ и для всех вершин u глубины меньшей k функции f_u уже определены, то полагаем

$$f_v(x_1, \ldots, x_n) = F_v(f_{u_1}(x_1, \ldots, x_n), \ldots, f_{u_m}(x_1, \ldots, x_n)).$$

Здесь m — местность функции $F_v \in \mathcal{B}$, которой помечена вершина v, $u_1, \ldots, u_m$ — вершины, из которых рёбра с метками $1, \ldots, m$ соответственно ведут в v.

Сложность $L(\mathfrak{S})$ схемы $\mathfrak{S}$ — это количество функциональных элементов в $\mathfrak{S}$. Сложность $L(f)$ булевой функции f — это наименьшая сложность схемы, реализующей эту функцию в какой-либо вершине.

Линейная программа в базисе $\mathcal{B}$ — это последовательность слов вида $x \leftarrow F(x_1, \ldots, x_k)$, где $x, x_1, \ldots, x_k \in \boldsymbol{V}$ — это переменные, $F \in \mathcal{B}$ — k-местная базисная функция. Такие слова называют присваиваниями. В частности, для базиса $\mathcal{B}_0 = \{\wedge, \vee, \neg\}$ линейная программа состоит из присваиваний вида $z \leftarrow x \wedge y$, $z \leftarrow x \vee y$ и $z \leftarrow \neg x$, где $x, y, z \in \boldsymbol{V}$ — произвольные переменные. Переменная x линейной программы Π называется входной, если некоторое присваивание вида $y \leftarrow F(\ldots, x, \ldots)$ встречается в Π раньше, чем все присваивания вида $x \leftarrow F(\ldots)$.

При выполнении присваивания $x \leftarrow F(x_1, \ldots, x_k)$ значение переменной x меняется на $F(\sigma_1, \ldots, \sigma_k)$, где $\sigma_1, \ldots, \sigma_k$ — значения переменных $x_1, \ldots, x_k$ соответственно, а значения всех остальных переменных остаются неизменными. Обозначим с помощью $\Pi_z(\sigma_1, \ldots, \sigma_n)$ значение переменной z после выполнения линейной программы Π, если значения входных переменных $x_1, \ldots, x_n$ изначально были равны $\sigma_1, \ldots, \sigma_n$ соответственно. Линейная программа Π вычисляет в переменной z функцию $f(x_1, \ldots, x_n)$, если для любого набора значений входов $\sigma_1, \ldots, \sigma_n$ после завершения работы Π выполнено $\Pi_z(\sigma_1, \ldots, \sigma_n) = f(\sigma_1, \ldots, \sigma_n)$.

Задачи

386. Доказать, что минимальная схема для сложения по модулю два имеет сложность $L(\oplus) = 4$ в базисе $\mathcal{B}_0$.
Указание. Доказать это утверждение для функций $x_1 \oplus x_2$ и $x_1 \leftrightarrow x_2$ одновременно. ⊛

387. Определить, какую булеву функцию реализует схема $\mathfrak{S}$ на рис. 15 на следующей странице в вершине v? Можно ли для этой функции построить менее сложную схему? ⊛

388. Определить, какую булеву функцию реализует схема $\mathfrak{S}$ на рис. 16 на следующей странице в вершине v? Можно ли для этой функции построить менее сложную схему? Построить линейную программу, вычисляющую ту же функцию. ⊛

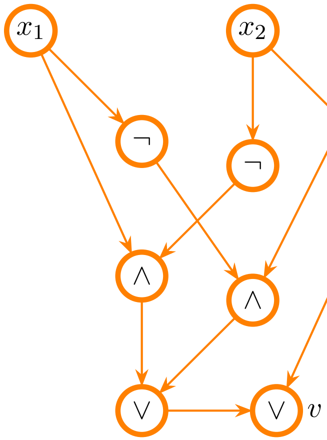


Рис. 15: Схема $\mathfrak{S}$ из задачи 387.

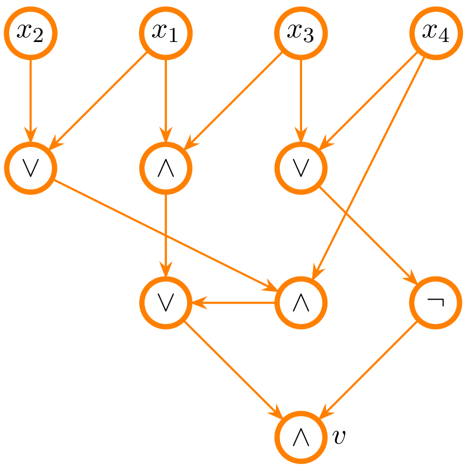


Рис. 16: Схема $\mathfrak{S}$ из задачи 388.

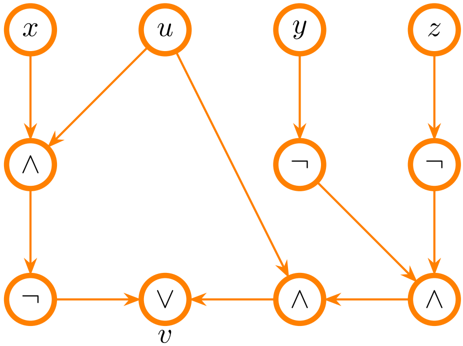


Рис. 17: Схема $\mathfrak{S}$ из задачи 391.

$u \leftarrow x \wedge u$	$f \leftarrow \neg y$	$y \leftarrow \neg y$
$u \leftarrow \neg u$	$g \leftarrow \neg z$	$z \leftarrow \neg z$
$y \leftarrow \neg y$	$e \leftarrow x \wedge u$	$z \leftarrow y \wedge z$
$z \leftarrow \neg z$	$k \leftarrow \neg e$	$z \leftarrow z \wedge u$
$y \leftarrow y \wedge z$	$h \leftarrow f \wedge g$	$u \leftarrow x \wedge u$
$z \leftarrow y \wedge u$	$i \leftarrow h \wedge u$	$u \leftarrow \neg u$
$z \leftarrow u \vee z$	$z \leftarrow h \vee k$	$z \leftarrow z \vee u$

Рис. 18: Программы из задачи 391.

389. Доказать, что в базисе $\mathcal{B}_0$ всякую булеву функцию f можно вычислить с помощью линейной программы, где присваивание выполняется не более чем трём переменным. Можно ли уменьшить это количество до двух? ⊛

390. Доказать, что в предыдущей задаче количество изменяемых переменных нельзя уменьшить до одной.
Указание. Рассмотреть функцию $x_1 \oplus x_2$. ⊛

391. Задана схема $\mathfrak{S}$ из функциональных элементов (рис. 17) и три линейных программы (рис. 18). Определить, какие из программ вычисляют в переменной z ту же функцию $f(x, y, z, u)$, что и схема $\mathfrak{S}$ в вершине v? ⊛

$$
\begin{aligned}
y &\leftarrow \neg x_1 \\
z &\leftarrow \neg x_2 \\
u &\leftarrow \neg x_3 \\
y &\leftarrow y \wedge x_2 \\
w &\leftarrow x_2 \wedge x_3 \\
y &\leftarrow y \wedge u \\
y &\leftarrow w \vee y \\
z &\leftarrow z \vee y
\end{aligned}
$$

Рис. 19: Линейная программа из задачи 392.

392. Пусть задана линейная программа Π с входными переменными x_1, x_2, x_3 (рис. 19). Построить схему $\mathfrak{S}_\Pi$ из функциональных элементов со входами x_1, x_2, x_3 и функциональными вершинами, соответствующими присваиваниям Π, вычисляющую ту же функцию, что и Π в выходной переменной z. Чему равна её глубина? Можно ли для вычисления той же функции построить более короткие схему и линейную программу? ⊛

393. Пусть базис $\mathcal{B}$ позволяет получить константу 0, а программа Π в базисе $\mathcal{B}$ кроме входных переменных $x_1, \ldots, x_n$ содержит $z_1, \ldots, z_m$, в каждой из которых вычисляется функция $f_1, \ldots, f_m$ соответственно. Доказать, что тогда можно эффективно построить схему $\mathfrak{S}_\Pi$ в базисе $\mathcal{B}$, содержащую в том числе вершины $v_1, \ldots, v_m$ такие, что $f_{v_i} = f_i$ для $i = 1, \ldots, m$. ⊛

394. Допустим, что функция f не является константой. Доказать, что сложность функции f в базисе $\mathcal{B}_0$ не превосходит сложности f в базисе $\mathcal{B}_1 = \mathcal{B}_0 \cup \{0, 1\}$. ⊛

395. Пусть SUM_n — это схема n-разрядного сумматора, выполняющего сложение двух n-разрядных двоичных чисел, результатом которого является $(n+1)$-разрядное число (см. [3], § 13.3). Используя схему SUM_n, построить схему, реализующую операцию вычитания двух n-разрядных двоичных чисел: $d = a - b$ (при условии, что $a \geqslant b$). Оценить сложность полученной схемы. ⊛

396. Два игрока независимо выбирают одно из четырёх чисел от 0 до 3. Первый игрок выигрывает, если их сумма является степенью двойки. Построить схему, определяющую выигрыш первого игрока. Её входы u_1, u_0 представляют в двоичной записи число, выбранное первым игроком, а v_1, v_0 — число, выбранное вторым игроком. ⊛

397. Построить схему $\mathfrak{C}_n$ для сравнения двух n-значных чисел, представленных в двоичном виде. Схема должна иметь входы $u_{n-1}, \ldots, u_0$ и $v_{n-1}, \ldots, v_0$ для исходных чисел и три выхода результата: больше, меньше или равно. ⊛

398. Показать, что схему для вычисления n-местной функции odd (задача 170 на стр. 54) можно реализовать в базисе $\mathcal{B}_0$, используя не более $\lfloor n/2 \rfloor$ отрицаний. ⊛

399. Пусть f — n-местная булева функция, которая реализуется схемой $\mathfrak{S}$ с k отрицаниями в базисе $\mathcal{B}_0$,

$$v_i = f(\underbrace{1, \ldots, 1}_{i \text{ единиц}}, 0, \ldots, 0).$$

Доказать, что для любых j и m в последовательности значений $s = (v_j, v_{j+1}, \ldots, v_{j+m})$ функции f существует не больше $2^k - 1$ позиций, где значение 1 меняется на 0.
Указание. Применить индукцию по k. ⊛

400. Показать, что оценка $2^k - 1$ из предыдущей задачи реально достижима.
Указание. Индукцией по k построить соответствующую функцию f_k с $2^{k+1} - 1$ аргументами. ⊛

401. Построить схему для умножения двух двухзначных двоичных чисел. Схема должна иметь входы a_1, a_0 и b_1, b_0 для исходных чисел и четыре выхода для разрядов результата. ⊛

402. Построить схему, определяющую результат голосования в комитете, состоящем из трёх членов и председателя. В случае равенства голосов голос председателя является решающим. ⊛

403. Построить логическую схему, определяющую результат голосования в комитете, состоящем из пяти членов, где у председателя и секретаря имеется по два голоса. То есть если x_1, x_2, x_3, y, z — это голоса «за» трёх обычных членов, председателя и секретаря соответственно, то

$$f(x_1, x_2, x_3, y, z) = 1 \iff x_1 + x_2 + x_3 + 2y + 2z \geqslant 4.$$

Определить сложность и глубину построенной схемы. ⊛

404. Пусть наборы аргументов булевой функции от трёх аргументов упорядочены лексикографически, а её значения задаются последовательностью из восьми нулей и единиц. Построить схемы, реализующие следующие функции: ⊛

(а) $f_1 = (1111\,1011)$; (б) $f_2 = (1001\,1001)$; (в) $f_3 = (0011\,1001)$.

405. Построить следующие логические схемы, определить их сложность и глубину:

(а) $\mathfrak{L}_n$, которая выполняет линейный сдвиг по команде: имеет входы $u_1, u_2, \ldots, u_n$ и v, выходы $w_1, w_2, \ldots, w_n, w_{n+1}$, значения $w_1, w_2, \ldots, w_n, w_{n+1}$ равны соответственно $u_1, u_2, \ldots, u_n, 0$ при $v = 0$ или $0, u_1, \ldots, u_{n-1}, u_n$ при $v = 1$;

(б) $\mathfrak{R}_n$, которая выполняет циклический сдвиг по команде: имеет входы $u_1, u_2, \ldots, u_n$ и v, выходы $w_1, w_2, \ldots, w_n$, значения $w_1, w_2, \ldots, w_n$ равны соответственно $u_1, u_2, \ldots, u_n$ при $v = 0$ или $u_n, u_1, \ldots, u_{n-1}$ при $v = 1$;

(в) $\mathfrak{C}_n$, которая выполняет подсчёт количества единиц: имеет входы $u_1, u_2, \ldots, u_n$, выходы $w_0, w_1, w_2, \ldots, w_n$, значение w_i равно единице тогда и только тогда, когда среди $u_1, u_2, \ldots, u_n$ имеется ровно i единиц. ⊛

406. Используя схемы из предыдущей задачи, построить логические схемы, реализующие следующие функции:

(а) $f_n(x_1, x_2, \ldots, x_n) = 1 \iff \sum\limits_{i=1}^{n} x_i \equiv 3 \pmod 4$;

(б) $g_n(x_1, x_2, \ldots, x_n) = 1 \iff \sum\limits_{i=1}^{n} x_i \not\equiv 0 \pmod 3$;

(в) п о р о г о в а я функция

$$T_k^n(x_1, x_2, \ldots, x_n) = 1 \iff \sum_{i=1}^{n} x_i \geqslant k.$$

Определить сложность и глубину построенных схем. ⊛

407. Пусть ориентированный граф $\mathfrak{G} = (\{1, 2, 3\}, E)$ без петель задан матрицей смежности A размера 3×3. Построить схему из функциональных элементов, которая по шести переменным $A_{i,j}$, $i \neq j$, определяет, является ли граф $\mathfrak{G}$ сильно связным. Определить сложность и глубину построенной схемы. ⊛

408. Пусть неориентированный граф $\mathfrak{G} = (\{1, 2, 3, 4\}, E)$ без петель задан матрицей смежности A размера 4×4. Построить схему из функциональных элементов, которая по шести переменным $A_{i,j}$, $i < j$, определяет, является ли граф $\mathfrak{G}$ двудольным. Определить сложность и глубину построенной схемы. ⊛

409. Пусть для слова w из четырёх букв переменная s_i означает, что i-я буква в w является гласной. Построить схему из функциональных элементов, которая по четырём переменным s_i, $i = 1, \ldots, 4$, определяет, что в слове w нет подряд идущих ни двух гласных, ни трёх согласных. Определить сложность и глубину построенной схемы. ⊛

410. С помощью переменных u_3, u_2, u_1, u_0 закодировано четырёхразрядное двоичное число. Построить схему из функциональных элементов, которая по переменным u_3, u_2, u_1, u_0, определяет, является ли это число простым. ⊛

17. Упорядоченные бинарные диаграммы решений

Бинарное дерево решений (сокращённо, БДР) — это бинарное дерево $\mathfrak{T} = (V, E)$, все внутренние вершины которого помечены переменными, а листья — константами 0 или 1. Из каждой внутренней вершины v выходят два ребра, одно имеет метку 0, другое — метку 1; вершина w_0, в которую ведёт первое ребро, помеченное 0, называется 0-сыном v, а вершина w_1, в которую ведёт второе ребро, помеченное 1, называется 1-сыном вершины v.

БДР $\mathfrak{T}$, вершины которого помечены переменными $x_1, \ldots, x_n$, реализует булеву функцию $f_{\mathfrak{T}}(x_1, \ldots, x_n)$, если для каждого набора значений переменных $\sigma_1, \sigma_2, \ldots, \sigma_n$ ветвь в дереве, соответствующая этому набору (из вершины x_i идём по ребру, помеченному σ_i), завершается листом с меткой $f(\sigma_1, \sigma_2, \ldots, \sigma_n)$.

Пусть зафиксирована некоторая перестановка π переменных $x_1, \ldots, x_n$: $x_{\pi(1)}, \ldots x_{\pi(n)}$. Упорядоченная бинарная диаграмма решений (сокращённо, УБДР) относительно перестановки переменных π — это ориентированный граф $\mathfrak{D}$ без циклов с одним истоком такой, что выполнены условия:

1) в $\mathfrak{D}$ существуют лишь два стока, которые помечены константами 0 и 1 соответственно;

2) остальные (внутренние) вершины $\mathfrak{D}$ помечены переменными и из каждой из них выходят в точности два ребра, одно имеет метку 0, 0-сын, другое — метку 1, 1-сын;

3) порядок, в котором переменные встречаются на любом пути в $\mathfrak{D}$ из истока в сток, совместим с π, то есть если из вершины, помеченной $x_{\pi(i)}$, есть путь в вершину, помеченную $x_{\pi(j)}$, то $i < j$.

УБДР $\mathfrak{D}$ реализует булеву функцию $f_{\mathfrak{D}}(x_1, \ldots, x_n)$, если для каждого набора значений переменных $\sigma_1, \sigma_2, \ldots, \sigma_n$ путь в $\mathfrak{D}$, начинающийся в истоке и соответствующий этому набору (из вершины x_i идём по ребру, помеченному σ_i), завершается стоком с меткой $f_{\mathfrak{D}}(\sigma_1, \sigma_2, \ldots, \sigma_n)$.

Сложностью УБДР $\mathfrak{D}$ называется число $L(\mathfrak{D})$ — количество внутренних вершин $\mathfrak{D}$.

УБДР $\mathfrak{D}$ называется сокращённой, если

1) у любой внутренней вершины v её 0-сын и 1-сын различны;
2) нет такой пары внутренних вершин u и v, для которых поддиаграммы с истоками u и v являются изоморфными (то есть взаимно однозначно отображаются друг на друга с сохранением всех меток).

Определим два типа эквивалентных преобразований УБДР:

Правило сокращения: если 0-сын и 1-сын вершины v совпадают и равны w, то удалить вершину v и перенаправить в вершину w все рёбра, которые ранее входили в v;

Правило слияния: если вершины v и w помечены одной переменной и имеют одинаковых 0-сыновей и 1-сыновей, то удалить вершину v, перенаправив все входящие в неё рёбра в вершину w.

УБДР $\mathfrak{D}$ является сокращённой в том и только том случае, когда к ней невозможно применить ни правило слияния, ни правило сокращения. Сокращённая УБДР одновременно является и минимальной (то есть имеет наименьшее возможное количество вершин) для заданного порядка переменных.

Задачи

411. Доказать, что всякую булеву функцию от n переменных можно реализовать в виде УБДР с не более чем $2^{n/2+2} - 1$ внутренней вершиной.
Указание. Показать, как преобразовать полное БДР. ⊛

412. Рассмотрим трёхместные булевы функции $f(x_1, x_2, x_3)$, зафиксируем порядок переменных $x_1 < x_2 < x_3$. Определить:

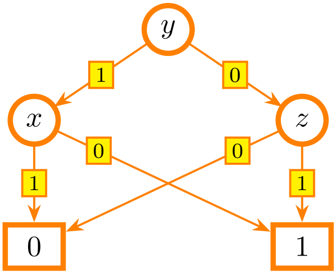


Рис. 20: УБДР $\mathfrak{D}_1$.

(а) наибольшее количество вершин в сокращённой УБДР для f;

(б) сколько существует функций с наибольшим количеством вершин в сокращённой УБДР. ⊛

413. Указать, как по УБДР для функции f построить УБДР для двойственной функции f^*. ⊛

414. Схемы из функциональных элементов естественным образом реализуются в виде линейных программ. Наоборот, для деревьев решений и УБДР естественным программным представлением являются ветвящиеся программы, включающие лишь условные операторы вида if v then Π_1 else Π_2 и присваивания $y \leftarrow 0$ и $y \leftarrow 1$ (см. раздел 22). Они соответствуют внутренним вершинам диаграмм и стокам соответственно. Здесь Π_1 и Π_2 — это снова ветвящиеся программы, а переменная y содержит результат.

Показать, как по УБДР построить ветвящуюся программу, вычисляющую ту же самую функцию.

Написать ветвящиеся программы, вычисляющие функции, представляемые УБДР $\mathfrak{D}_1$ на рис. 20 и $\mathfrak{D}_3$ на рис. 22 на стр. 125. ⊛

415. Доказать, что каждую УБДР $\mathfrak{D}$ можно представить в виде ациклической программы с метками Π (см. раздел 22) и при этом длина Π не превосходит количества вершин $\mathfrak{D}$. Программа Π ациклическая, если все метки программы Π пронумерованы и в каждом операторе метка оператора имеет номер меньший, чем метки перехода (это эквивалентно тому, что в блок-схеме нет циклов. Также можно сказать, что невозможны переходы «назад»). Присваивания имеют такой же вид, как в предыдущей задаче. ⊛

416. Доказать, что по любой УБДР $\mathfrak{D}$ с n вершинами, представляющей функцию f, можно построить линейную программу (и, следовательно, схему из функциональных элементов) размера $O(n)$ в базисе $\mathcal{B}_0 = \{\wedge, \vee, \neg\}$, вычисляющую ту же самую функцию f.
Указание. Использовать индукцию по высоте вершин. Высота вершины v в УБДР — это максимальная длина пути из v в какой-либо из стоков. ⊛

417. Построить минимальные УБДР для двухместных функций: $x \wedge y$, $x \vee y$, $x \oplus y$, $x \to y$, $x \uparrow y$. ⊛

418. Построить УБДР для функции $\mathrm{odd}(x_1, \ldots, x_n) = x_1 \oplus \cdots \oplus x_n$ и оценить её сложность. Сравнить со сложностью ДНФ этой же функции из задачи 170 на стр. 54. ⊛

419. Построить минимальные УБДР для функции

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = (x_1 \wedge x_2) \oplus (x_3 \wedge x_4) \oplus (x_5 \wedge x_6)$$

относительно двух упорядочений переменных:

(а) $x_1 < x_2 < x_3 < x_4 < x_5 < x_6$ и

(б) $x_1 < x_3 < x_5 < x_2 < x_4 < x_6$. ⊛

420. Используя алгоритм сокращения УБДР, построить сокращённую диаграмму, эквивалентную УБДР $\mathfrak{D}_2$ на рис. 21 на следующей странице. Определить, какую функцию реализует полученная схема. Построить её таблицу истинности. ⊛

421. Пусть Φ — это совершенная ДНФ для булевой функции f. Доказать, что существует УБДР для функции f, количество внутренних вершин которой не превосходит длины Φ независимо от порядка переменных. Под длиной Φ понимаем общее количество вхождений переменных в Φ.
Указание. Применить индукцию по количеству переменных n. ⊛

422. Доказать, что для функции

$$p_n(x_1, \ldots, x_n, y_n, \ldots, y_1) = (x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \ldots \vee (x_n \wedge y_n)$$

сложность сокращённой УБДР равна

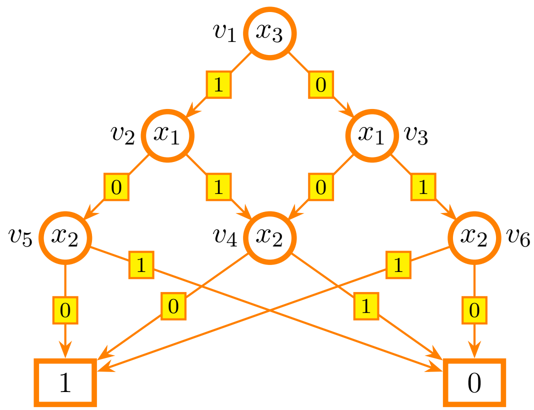


Рис. 21: УБДР $\mathfrak{D}_2$.

(а) $2n$ при упорядочении переменных $\pi_1 : x_1, y_1, \ldots, x_n, y_n$;

(б) $2^{n+1} - 2$ при упорядочении $\pi_2 : x_1, x_2, \ldots, x_n, y_n, \ldots, y_2, y_1$. ⊛

423. Пусть функция f' получена из функции f фиксированием значения переменной x_i:

$$f'(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = f(x_1, \ldots, x_{i-1}, \sigma, x_{i+1}, \ldots, x_n),$$

где $\sigma \in \{0, 1\}$.

(а) Предположим, что $\mathfrak{D}$ — это сокращённая УБДР для функции $f(x_1, \ldots, x_n)$ при указанном порядке переменных. Предложить процедуру, которая перестроит УБДР $\mathfrak{D}$ в УБДР $\mathfrak{D}'$ для функции $f'(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ при том же порядке. Всегда ли полученная УБДР $\mathfrak{D}'$ будет сокращённой?

(б) Применить процедуру из пункта (а) для получения из УБДР $\mathfrak{D}_3$ на рис. 22 на противоположной странице диаграмм для следующих функций: $f(0, x_2, x_3, x_4)$, $f(x_1, 1, x_3, x_4)$, $f(x_1, x_2, 0, x_4)$. ⊛

424. Пороговая функция T_k^n определена в задаче 406 на стр. 118.

(а) Построить УБДР для пороговой функции T_k^n в общем виде.

(б) Построить минимальную УБДР для пороговой функции T_4^6.

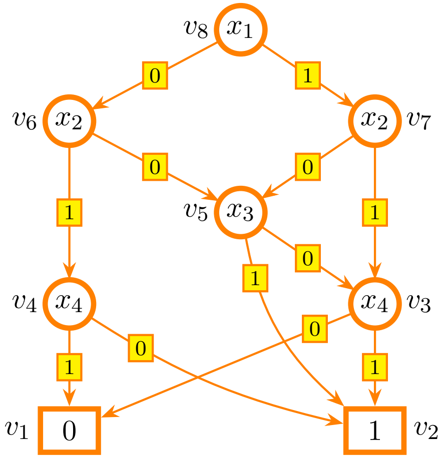


Рис. 22: УБДР $\mathfrak{D}_3$.

(в) Зависит ли сложность минимальной УБДР для пороговых функций от порядка переменных?

(г) Оценить сложность минимальной УБДР для пороговой функции T_k^n. ⊛

425. Построить минимальные УБДР, реализующие функции из задач 396, 402 и 403. ⊛

426. Построить УБДР для функций f_n и g_n из задачи 406 на стр. 118 и оценить их сложность. ⊛

427. Построить УБДР для решения задачи 407 на стр. 119. Определить сложность построенной диаграммы. ⊛

428. Построить УБДР для решения задачи 408 на стр. 119. Определить сложность построенной диаграммы. ⊛

429. Построить УБДР для решения задачи 409 на стр. 119. Определить сложность построенной диаграммы. ⊛

430. Построить УБДР для решения задачи 410 на стр. 119. Определить сложность построенной диаграммы. ⊛

18. Конечные автоматы: преобразователи и распознаватели

Конечный преобразователь $\mathfrak{M}$ — это пятёрка $(Q, \Sigma, \Omega, P, q_0)$, состоящая из следующих компонент:

1) $Q = \{q_0, \ldots, q_{n-1}\}$, $n \geqslant 1$, — конечное множество состояний;
2) $\Sigma = \{a_1, \ldots, a_m\}$, $m \geqslant 1$, — конечный входной алфавит;
3) $\Omega = \{b_1, \ldots, b_r\}$, $r \geqslant 1$, — конечный выходной алфавит;
4) P — программа преобразователя;
5) $q_0 \in Q$ — начальное состояние преобразователя.

Программа преобразователя — это конечное множество команд, то есть слов вида «$q, a \to p, y$», где $q, p \in Q$ — состояния, $a \in \Sigma$ — символ входного алфавита, $y \in \Omega^*$ — слово в выходном алфавите, может быть, пустое. При этом для каждых $q \in Q$ и $a \in \Sigma$ в P должна быть в точности одна команда описанного вида.

Диаграмма преобразователя $\mathfrak{M} = (Q, \Sigma, \Omega, P, q_0)$ — это размеченный ориентированный мультиграф $\mathfrak{D}_{\mathfrak{M}} = (Q, E)$ такой, что:

1) вершины $\mathfrak{D}_{\mathfrak{M}}$ являются состояниями $\mathfrak{M}$;
2) специальным образом отмечено начальное состояние q_0;
3) ребро (q, p) с меткой «$a : y$» есть тогда и только тогда, когда в P есть команда вида $q, a \to p, y$.

Для конечного преобразователя $\mathfrak{M} = (Q, \Sigma, \Omega, P, q_0)$ конфигурацией называется тройка вида (q, u, v), где $q \in Q$, $u \in \Sigma^*$, $v \in \Omega^*$. Конфигурация вида (q_0, u, ε) называется начальной, слово u в этом случае называется входным. Конфигурация вида (q, ε, v) для произвольного $q \in Q$ называется заключительной, а слово v в этом случае — выходным.

Произвольная конфигурация вида (q, au, v) переходит за один шаг в конфигурацию (p, u, vy) (записывается в виде $(q, au, v) \vdash_{\mathfrak{M}} (p, u, vy)$), если в программе P есть команда $q, a \to p, y$. Конфигурация σ переходит за k шагов в конфигурацию τ, если

1) $\sigma \vdash_{\mathfrak{M}}^{k} \tau$ при $\sigma = \tau$ и $k = 0$;

2) $\sigma \vdash_{\mathfrak{M}}^{k} \tau$ при $\sigma \vdash_{\mathfrak{M}}^{\ell} \rho$ и $\rho \vdash_{\mathfrak{M}} \tau$ для некоторый конфигурации ρ и $k = \ell + 1$.

Запись $\sigma \vdash_{\mathfrak{M}}^{*} \tau$ означает переход за произвольное количество шагов: $\sigma \vdash_{\mathfrak{M}}^{k} \tau$ для некоторого $k \in \omega$.

Автомат $\mathfrak{M}$ преобразует слово u в v, если $(q_0, u, \varepsilon) \vdash_{\mathfrak{M}}^{*} (q, \varepsilon, v)$ для некоторой заключительной конфигурации вида (q, ε, v). Слово v тогда называется результатом работы $\mathfrak{M}$ на u.

Детерминированный конечный автомат $\mathfrak{M}$ — это пятёрка вида (Q, Σ, P, q_0, F), в которой:

1) Q — конечное множество состояний;

2) Σ — конечный входной алфавит;

3) P — программа автомата;

4) $q_0 \in Q$ — начальное состояние;

5) $F \subseteq Q$ — множество принимающих состояний.

Программа детерминированного автомата — это конечное множество команд, то есть слов вида «$q, a \to p$», где $q, p \in Q$ — состояния, $a \in \Sigma$ — символ входного алфавита. При этом для каждых $q \in Q$ и $a \in \Sigma$ в P должна быть в точности одна команда описанного вида.

Диаграмма автомата $\mathfrak{M}$ — это ориентированный мультиграф $\mathfrak{D}_{\mathfrak{M}} = (Q, E)$ такой, что:

1) вершины $\mathfrak{D}_{\mathfrak{M}}$ являются состояниями $\mathfrak{M}$;

2) отмечено начальное состояние q_0;

3) отмечены все принимающие состояния из F;

4) если в P есть команда вида $q, a \to p$, то ребро (q, p) имеет метку a, других рёбер нет.

В настоящей книге мы отмечаем начальное состояние зелёным цветом, а принимающие — голубой окружностью.

Если $R = (q_1, q_2, \ldots, q_{k-1}, q_k)$ — путь в $\mathfrak{D}_{\mathfrak{M}}$, последовательно проходящий по рёбрам $e_1, \ldots, e_{k-1}$, то метки этих рёбер составляют некоторое слово w. В этом случае говорят, что путь R *несёт слово* w, а слово w *переводит* q_1 в q_k.

Для конечного автомата $\mathfrak{M} = (Q, \Sigma, P, q_0, F)$ *конфигурацией* называется пара вида (q, u), где $q \in Q$, $u \in \Sigma^*$. Конфигурация вида (q_0, u) называется *начальной*. Конфигурация вида (q, ε) называется *заключительной*, а если ещё при этом $q \in F$, то — *принимающей*,

Конфигурация вида (q, au) *переходит за один шаг* в конфигурацию (p, u) (записываем $(q, au) \vdash_{\mathfrak{M}} (p, u)$), если в программе P есть команда $q, a \to p$. Определение перехода за k шагов ($\sigma \vdash^k_{\mathfrak{M}} \tau$) и за произвольное количество шагов ($\sigma \vdash^*_{\mathfrak{M}} \tau$) не меняется.

Говорим, что $\mathfrak{M}$ *принимает* (или *распознаёт*) слово u, если $(q_0, u) \vdash^*_{\mathfrak{M}} (q, \varepsilon)$ для некоторого $q \in F$, то есть начальная конфигурация с входным словом u за какое-то количество шагов переходит в принимающую. Язык, *распознаваемый конечным автоматом* $\mathfrak{M}$ (обозначаем $L(\mathfrak{M})$), — это множество принимаемых $\mathfrak{M}$ слов:

$$L(\mathfrak{M}) = \{w \in \Sigma^* : \mathfrak{M} \text{ принимает } w\}.$$

Язык называется *автоматным*, если он распознаётся некоторым детерминированным конечным автоматом.

Пусть автоматы $\mathfrak{M}_1 = (Q_1, \Sigma, P_1, q_0^1, F_1)$ и $\mathfrak{M}_2 = (Q_2, \Sigma, P_2, q_0^2, F_2)$ распознают языки L_1 и L_2 соответственно. *Произведением* автоматов $\mathfrak{M}_1$ и $\mathfrak{M}_2$ называется всякий автомат $\mathfrak{N}$ следующего вида:

$$\mathfrak{N} = (Q_1 \times Q_2, \Sigma, P, (q_0^1, q_0^2), F),$$

где программа P содержит команду $(q_1, q_2), a \to (p_1, p_2)$, если $q_1, a \to p_1 \in P_1$ и $q_2, a \to p_2 \in P_2$. Множество принимающих состояний F определяется в зависимости от операции над языками L_1 и L_2, которую должен реализовать $\mathfrak{N}$. Например,

1) при $F = F_\cup = \{(q, p) : q \in F_1 \text{ или } p \in F_2\}$ будет $L(\mathfrak{N}) = L_1 \cup L_2$;

2) при $F = F_\cap = \{(q, p) : q \in F_1 \text{ и } p \in F_2\}$ будет $L(\mathfrak{N}) = L_1 \cap L_2$;

3) при $F = F_\setminus = \{(q, p) : q \in F_1 \text{ и } p \notin F_2\}$ будет $L(\mathfrak{N}) = L_1 \setminus L_2$.

Недетерминированный конечный автомат отличается от детерминированного тем, что кроме команд вида «$q, a \to p$» могут быть команды вида «$q, \varepsilon \to p$», где $q, p \in Q$ — состояния, а также нет ограничений на количество команд каждого вида. Команды вида «$q, \varepsilon \to p$» называют *пустыми* или *ε-переходами*.

$Q \backslash \Sigma$	P_1		P_2	
	0	1	0	1
q_0	$q_0 : T$	$q_1 : A$	$q_3 : 1$	$q_1 : 0$
q_1	$q_2 : P$	$q_3 : T$	$q_1 : 1$	$q_2 : 0$
q_2	$q_1 : A$	$q_2 : T$	$q_2 : 0$	$q_2 : 1$
q_3	$q_1 : A$	$q_3 : P$	$q_1 : 0$	$q_1 : 1$

Рис. 23: Программы преобразователей из задач 431 и 432.

Для недетерминированного автомата $\mathfrak{N} = (Q, \Sigma, P, q_0, F)$ конфигурация вида (q, au) переходит за один шаг в конфигурацию (p, u) (записываем $(q, au) \vdash_{\mathfrak{M}} (p, u)$), если в программе P есть команда $q, a \to p$. Также конфигурация вида (q, u) переходит за один шаг в конфигурацию (p, u), если в программе P есть команда $q, \varepsilon \to p$. Определения перехода за k шагов ($\sigma \vdash^k_{\mathfrak{M}} \tau$) и за произвольное количество шагов ($\sigma \vdash^*_{\mathfrak{M}} \tau$) не меняются. Также не меняются определения распознавания слов и языков. Отличие состоит в том, что у недетерминированного автомата может быть несколько различных путей вычисления на одном и том же входном слове w.

Задачи

431. Конечный преобразователь $\mathfrak{M} = (Q, \Sigma, \Omega, P_1, q_0)$ определён так: $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{0, 1\}$, $\Omega = \{A, P, T\}$, а программа P_1 задана таблицей на рис. 23.

(а) Найти результат работы преобразователя $\mathfrak{M}$ на слове 010100?

(б) На каком входе $\mathfrak{M}$ выдаст результат APAPAT? ⊛

432. Конечный преобразователь $\mathfrak{M}_c = (Q, \Sigma, \Omega, P_2, q_0)$ задан следующим образом: $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{0, 1\}$, $\Omega = \{0, 1\}$, а программа P_2 задана таблицей на рис. 23. Этот преобразователь вычитает из входного числа x в двоичной записи некоторую константу c, то есть выдаёт при $c \leqslant x$ выходное число $y = x - c$ в двоичной записи. Чему равна эта константа c? Разряды в двоичной записи чисел идут от младших к старшим, то есть число десять выглядит так: 0101. ⊛

433. Автомат по размену монет имеет щель для приёма монет и накопитель для их выдачи. Автомат принимает монеты достоинством в 1, 2 и 10 рублей и разменивает их монетами по 5 рублей. При этом на табло отображается уже внесённая сумма, а как только сумма оказывается достаточной, автомат сразу же выдаёт пятирублёвые монеты в нужном количестве.

Построить конечный преобразователь, моделирующий работу этого автомата. Определить входной и выходной алфавиты и построить его программу. ⊛

434. Торговый автомат по продаже кофе имеет щель для получения монет, кнопку, нажатие которой после уплаты достаточной суммы приводит к получению кофе, и накопитель, через который он выдаёт сдачу покупателю. Автомат принимает монеты достоинством в 1, 2 и 5 рублей. Чашка кофе стоит 8 рублей. Пока полученная сумма недостаточна, горит красная лампочка. Если сумма, полученная автоматом, становится больше или равна 8 рублям, то зажигается зелёная лампочка и после нажатия кнопки автомат наливает кофе и, если требуется, выдаёт сдачу наименьшим количеством монет. Если автомат получает монету, когда горит зелёная лампочка, то он немедленно её возвращает.

Построить конечный преобразователь, моделирующий работу этого автомата. Определить входной и выходной алфавиты и построить его программу. ⊛

435. Электронные часы имеют табло с указанием часов, минут и секунд, а также — две управляющие кнопки. Первая кнопка переводит часы из нормального режима в режим настройки времени — сначала в настройку часов, затем — минут, затем — секунд, а затем возвращает в нормальный режим. Вторая кнопка в нормальном режиме ничего не меняет, а в режиме настройки её нажатие увеличивает на единицу число настраиваемых часов, минут или секунд соответственно.

Построить конечный преобразователь, который моделирует работу часов. На вход он принимает сигналы нажатия от двух кнопок, а на выходе выдаёт сигналы изменения режима и увеличения соответствующего числа. Изобразить диаграмму преобразователя. ⊛

$Q \backslash \Sigma$	P_1		P_2		P_3		P_4		P_5	
	a	b	a	b	a	b	a	b	a	b
q_0	q_1	q_2	q_4	q_1	q_3	q_1	q_4	q_1	q_1	q_0
q_1	q_1	q_3	q_2	q_2	q_1	q_2	q_2	q_1	q_1	q_2
q_2	q_4	q_2	q_3	q_1	q_4	q_2	q_3	q_2	q_3	q_0
q_3	q_1	q_5	q_1	q_3	q_3	q_3	q_1	q_3	q_1	q_2
q_4	q_5	q_2	q_4	q_4	q_1	q_4	q_4	q_4		
q_5	q_5	q_5								

Рис. 24: Программы автоматов из задач 439, 440 и 441.

436. Построить конечный преобразователь, умножающий число в двоичной записи на 3. Цифры записаны в порядке возрастания разрядов (как в задаче 432 на стр. 129). ⊛

437. Построить конечный преобразователь, нацело делящий число в десятичной записи на 3. Цифры записаны в порядке убывания разрядов (в «обычном» порядке). ⊛

438. Определить, какие языки могут распознавать детерминированные конечные автоматы с алфавитом $\Sigma = \{0, 1\}$ и множеством состояний $Q = \{q_0, q_1\}$. ⊛

439. Пусть $\Sigma = \{a, b\}$, $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$, $F = \{q_3, q_4\}$ и функция переходов P_1 конечного автомата $\mathfrak{M} = (Q, \Sigma, P_1, q_0, F)$ задана таблицей, представленной на рис. 24. Определить, какие из следующих трёх слов распознаются автоматом $\mathfrak{M}$: $w = aaabbabab$, $v = babbbabba$, $u = ababaaab$? ⊛

440. Язык L состоит из всех слов, которые начинаются с буквы b и содержат количество букв a, кратное трём. Определить, какие из следующих трёх детерминированных конечных автоматов распознают язык L: $\mathfrak{M}_i = (\{q_0, q_1, q_2, q_3, q_4\}, \{a, b\}, P_i, q_0, \{q_1\})$, $i = 2, 3, 4$, программы P_i заданы таблицей, изображённой на рис. 24. ⊛

441. Составляющие конечного автомата $\mathfrak{M} = (Q, \Sigma, P_5, q_0, F)$ равны $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{a, b\}$, $F = \{q_3\}$, а программа P_5 задаётся таблицей на рис. 24. Доказать, что автомат $\mathfrak{M}$ распознаёт язык, состоя-

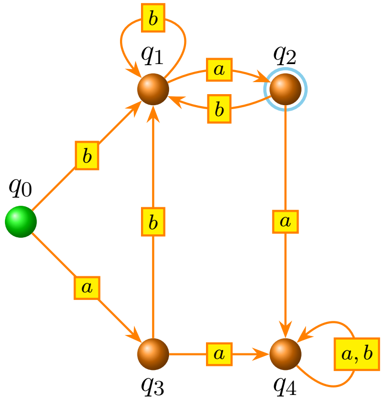


Рис. 25: Автомат из задачи 442.

щий в точности из тех слов в алфавите $\{a, b\}$, которые заканчиваются фрагментом aba. ⊛

442. Доказать, что приведённый на рис. 25 автомат, распознаёт язык, состоящий из всех слов, заканчивающихся на ba и не содержащих при этом aa. ⊛

443. Индукцией по длине слова w доказать следующее утверждение: $(q, wx) \vdash^*_{\mathfrak{M}} (p, x)$ тогда и только тогда, когда в $\mathfrak{D}_{\mathfrak{M}}$ есть путь из q в p, несущий w. ⊛

444. Построить детерминированные конечные автоматы, которые распознают следующие языки в алфавите $\Sigma = \{a, b\}$:

(а) $L = \{w :$ длина w делится на $5\}$;

(б) $L = \{w : w$ не содержит подслов «aab» и «bba»$\}$;

(в) $L = \{w : w$ содержит чётное количество букв a и нечётное количество букв $b\}$;

(г) $L = \{w :$ количество букв a в слове w делится на 3, а количество букв b — на $2\}$. ⊛

445. Построить детерминированный конечный автомат, который проверяет правильность сложения. На вход поступают «трёхэтажные» символы, в которых на верхнем этаже записан разряд первого слагаемого, на среднем — второго, на нижнем — предполагаемой суммы. Автомат должен принять слово, если записанный на нижнем этаже результат сложения верен. Цифры всех чисел записаны в порядке возрастания разрядов. ⊛

446. Построить детерминированные конечные автоматы, распознающие следующие языки в алфавите $\{0, 1\}$:

(а) L_1 — все слова, в которых на четвёртом с конца месте стоит 1;

(б) L_2 — все слова, содержащие подслова 111 и 010 и не содержащие подслово 001;

(в) L_3 — все слова, заканчивающиеся на 10, 11 или 01. ⊛

447. Построить детерминированный конечный автомат $\mathfrak{M}$ с алфавитом $\Sigma = \{\wedge, \vee, 0, 1\}$, который принимает только те слова, которые являются истинными ДНФ. ⊛

448. Даны два языка в алфавите $\Sigma = \{a, b\}$:

$$L_1 = \{w : \text{количество букв } b \text{ в } w \text{ кратно трём}\};$$
$$L_2 = \{w : w \text{ не содержит подслово } aaa\}.$$

(а) Построить детерминированные конечные автоматы $\mathfrak{A}$ и $\mathfrak{B}$, распознающие языки L_1 и L_2, соответственно.

(б) Построить произведение этих автоматов $\mathfrak{C} = \mathfrak{A} \times \mathfrak{B}$ и определить множества его допускающих состояний F_1, F_2 и F_3, для распознавания языков $R_1 = L_1 \cup L_2$, $R_2 = L_1 \cap L_2$, $R_3 = L_1 \setminus L_2$, соответственно. ⊛

449. Дан $\mathfrak{M} = (Q, \Sigma, P, q, \{p\})$ — недетерминированный конечный автомат, где $Q = \{q, p, s\}$, $\Sigma = \{0, 1\}$, $P = \{q,\ 0 \to p;\ q,\ 0 \to s;\ q,\ 1 \to q;\ p,\ 0 \to q;\ p,\ 0 \to p;\ s,\ 1 \to q;\ s,\ 1 \to p\}$.

Определить, какие из следующих трёх детерминированных автоматов $\mathfrak{M}_i = (Q_i, \Sigma, P_i, q, F_i)$ эквивалентны $\mathfrak{M}$:

(а) $Q_1 = \{q, ps, pq, pqs\}$, $F_1 = \{ps, pq, pqs\}$, $P_1 = \{q, 0 \to ps;\ q, 1 \to q;\ ps, 0 \to pq;\ ps, 1 \to pq;\ pq, 0 \to pqs;\ pq, 1 \to q;\ pqs, 0 \to pqs;\ pqs, 1 \to pq\}$.

(б) $Q_2 = \{q, p, s, ps, qs, pq, pqs, \varnothing\}$, $F_2 = \{ps, pq, pqs\}$,
$P_2 = \{q, 0 \to ps;\ q, 1 \to q;\ p, 0 \to pq;\ p, 1 \to q;\ ps, 0 \to qs;\ ps, 1 \to pq;\ pq, 0 \to pqs;\ pq, 1 \to q;\ qs, 0 \to ps;\ qs, 1 \to q;\ pqs, 0 \to pqs;\ pqs, 1 \to pq;\ \varnothing, 0 \to \varnothing;\ \varnothing, 1 \to \varnothing$.

(в) $Q_3 = \{q, p, s, ps, qs, pq, pqs, \varnothing\}$, $F_3 = \{p, ps, pq, pqs, s, qs\}$, $P_3 = \{q, 0 \to ps;\ q, 1 \to q;\ p, 0 \to pq;\ p, 1 \to q;\ s, 0 \to \varnothing;\ s, 1 \to pq;\ ps, 0 \to pq;\ ps, 1 \to pq;\ pq, 0 \to pqs;\ pq, 1 \to q;\ qs, 0 \to ps;\ qs, 1 \to q;\ pqs, 0 \to pqs;\ pqs, 1 \to pq;\ \varnothing, 0 \to \varnothing;\ \varnothing, 1 \to \varnothing$. ⊛

450. Используя процедуру детерминизации, построить детерминированные автоматы, эквивалентные следующим недетерминированным конечным автоматам $\mathfrak{N}$ с алфавитом $\Sigma = \{a, b\}$:

(а) $\mathfrak{N} = (\{q_0, q_1, q_2\}, \Sigma, P, q_0, \{q_2\})$ с программой $P = \{q_0,\ a \to q_1;\ q_0,\ \varepsilon \to q_1;\ q_1,\ b \to q_2;\ q_1,\ \varepsilon \to q_2;\ q_2,\ b \to q_2\}$;

(б) $\mathfrak{N} = (\{q_0, q_1, q_2\}, \Sigma, P, q_0, \{q_2\})$ с программой $P = \{q_0,\ a \to q_1;\ q_0,\ a \to q_2;\ q_1,\ b \to q_2;\ q_1,\ \varepsilon \to q_2;\ q_2,\ b \to q_0\}$. ⊛

451. Пусть $\mathfrak{M} = (Q, \Sigma, P, q_0, F)$ — это недетерминированный конечный автомат, в котором $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{a, b\}$, $F = \{q_1, q_3\}$, $P = \{q_0, a \to q_0;\ q_0, a \to q_1;\ q_0, b \to q_2;\ q_1, b \to q_0;\ q_2, a \to q_0;\ q_2, a \to q_3;\ q_2, b \to q_1;\ q_2, b \to q_2;\ q_2, b \to q_3;\ q_3, a \to q_0;\ q_3, a \to q_3\}$.

(а) Определить, какие из следующих слов распознаются автоматом $\mathfrak{M}$: $w_1 = aaabab$, $w_2 = ababbba$, $w_3 = bbaabba$, $w_4 = bbbaab$.

(б) Детерминизировать $\mathfrak{M}$.

(в) Определить, какой язык распознаёт $\mathfrak{M}$.

(г) Построить более простой вариант детерминированного автомата для распознавания того же языка. ⊛

452. Дан $\mathfrak{M} = (Q, \Sigma, P, q_0, F)$ — недетерминированный конечный автомат, где $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$, $\Sigma = \{0, 1\}$, $F = \{q_5\}$, а P определена в таблице на рис. 26 на противоположной странице (– означает отсутствие соответствующих переходов).

(а) Определить, какие из следующих слов распознаются автоматом $\mathfrak{M}$: $w = 101$, $v = 110000$, $u = 01$, $x = 10$, $y = 11$, $z = 1$.

$Q \backslash \Sigma$	0	1	ε
q_0	—	q_1, q_5	—
q_1	—	—	q_2, q_3
q_2	q_5	q_4	—
q_3	q_3	—	q_4
q_4	q_4, q_5	—	q_5
q_5	—	—	—

Рис. 26: Программа автомата из задачи 452.

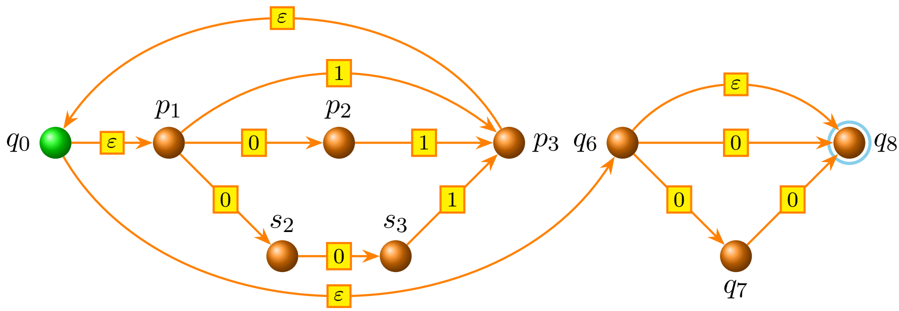


Рис. 27: Автомат из задачи 453.

(б) Доказать, что автомат $\mathfrak{M}$ распознаёт следующий язык

$$L = \{10^i : i \geqslant 0\} \cup \{110^i : i \geqslant 0\}.$$

(в) Построить детерминированный конечный автомат, эквивалентный $\mathfrak{M}$. ⊛

453. Применить процедуру детерминизации и построить детерминированный автомат эквивалентный недетерминированному конечному автомату, изображённому на рис. 27. ⊛

454. Язык L в алфавите $\Sigma = \{a_1, \ldots, a_k\}$ состоит из всех слов, в которых отсутствует хотя бы одна буква алфавита Σ.

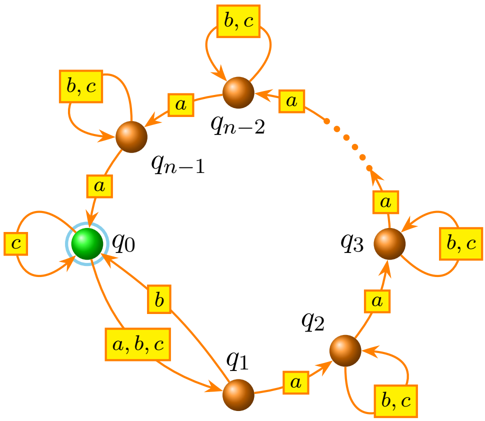


Рис. 28: Автомат из задачи 455.

(а) Построить недетерминированный автомат с $k+1$ состоянием, распознающий L.

(б) Доказать, что никакой детерминированный автомат с менее чем 2^k состояниями не может распознавать L. $\circledast$

455. Недетерминированный конечный автомат $\mathfrak{N} = (Q, \Sigma, P, q_0, F)$ для алфавита $\Sigma = \{a, b, c\}$ с n состояниями $Q = \{q_0, q_1, \ldots, q_{n-1}\}$ и множеством принимающих состояний $F = \{q_0\}$ изображён на рис. 28. Пусть $\mathfrak{M} = (\mathsf{P}(Q), \Sigma, P', \{q_0\}, F')$ — это автомат, полученный из $\mathfrak{N}$ при помощи процедуры детерминизации. Доказать, что

(а) для любого состояния $R \in \mathsf{P}(Q)$ автомата $\mathfrak{M}$ существует слово w_R, которое переводит начальное состояние $\{q_0\}$ в R;

(б) для любых состояний $R, S \in \mathsf{P}(Q)$, $R \neq S$, автомата $\mathfrak{M}$ существует слово $w_{R,S}$, которое переводит одно из состояний R или S в некоторое принимающее состояние, а другое — в непринимающее;

(в) не существует детерминированного конечного автомата, который был бы эквивалентен $\mathfrak{N}$ и имел бы меньше чем 2^n состояний. ⊛

19. Регулярные языки и конечные автоматы

Конкатенацией языков L_1 и L_2 в алфавите Σ называется язык

$$L = L_1 \mathbin{\&} L_2 = \{uv : u \in L_1, v \in L_2\}.$$

Если языки одинаковы, то используется степенная нотация: $L^0 = \{\varepsilon\}$, $L^{i+1} = L^i \mathbin{\&} L$. Таким образом, в L^i входят все слова, которые можно разбить на i подряд идущих слов из L.

Итерация языка L — это язык L^*, состоящий из всех слов, которые можно разбить на несколько подряд идущих слов из L:

$$L^* = \{w_1 \dots w_n : n \in \omega \text{ и } w_1, \dots, w_n \in L\}.$$

Её можно представить с помощью степеней:

$$L^* = \bigcup_{i=0}^{\infty} L^i.$$

Запись L^+ является сокращением для выражения $L \mathbin{\&} L^*$:

$$L^+ = L \mathbin{\&} L^* = \bigcup_{i=1}^{\infty} L^i.$$

Регулярное выражение над алфавитом Σ — слово одного из следующих видов. С каждым регулярным выражением r связан *представляемый* им язык $L(r)$:

1) $r = \varnothing$, $L(\varnothing) = \varnothing$;

2) $r = \varepsilon$, $L(\varepsilon) = \{\varepsilon\}$;

3) $r = a$ для $a \in \Sigma$, $L(a) = \{a\}$;
4) $r = (r_1 + r_2)$, $L(r) = L(r_1) \cup L(r_2)$;
5) $r = (r_1 r_2)$, $L(r) = L(r_1) \mathbin{\&} L(r_2)$;
6) $r = r_1^*$, $L(r) = (L(r_1))^*$.

Здесь r_1 и r_2 — произвольные уже построенные регулярные выражения.

Два регулярных выражения r и p называются *эквивалентными*, если совпадают представляемые ими языки, то есть $L(r) = L(p)$. В этом случае пишем $r \equiv p$.

Языки, которые представляются регулярными выражениями, называют *регулярными*. Классы автоматных и регулярных языков совпадают.

Задачи

456. Определить конкатенацию для следующих пар языков L_1 и L_2:

(а) $L_1 = \{a, ab, abb\}$ и $L_2 = \{\varepsilon, a, b, ab, ba\}$;

(б) $L_1 = \{\varepsilon, a, ab, abb\}$ и $L_2 = \{a, b, abb, ba\}$;

(в) $L_1 = \{\varepsilon, a, b, ab, aba\}$ и $L_2 = \{\varepsilon, a, b, ab, ba\}$. ⊛

457. Пусть $L = \{baa, bab, bba, bbb\}$. Какой из следующих языков является итерацией L^* этого языка?

(а) $\{w : w = bw'$ и $|w|$ делится на $3\} \cup \{\varepsilon\}$;

(б) $\{w : w = bw'$ и $|w| \geqslant 3\} \cup \{\varepsilon\}$;

(в) $\{w : w = x_1 x_2 x_3 \ldots x_{3n},\ x_i \in \{a, b\}$ и $x_{3i+1} = b$ для всех $i < n\} \cup \{\varepsilon\}$;

(г) $\{w : w = bw'$ и $|w| \geqslant 12\}$. ⊛

458. Какие из следующих регулярных выражений задают все слова из нулей и единиц, в которых нет двух подряд идущих 0?

(а) $(1 + 01)^*(\varepsilon + 0)$;

(б) $(1^*01^*)^*$;

(в) $(01)^*1^*01^*$;

(г) $1^*01(1 + 01)^*(\varepsilon + 0)$;

(д) $(1 + 01)^*(0 + 1)$;

(е) $1^*(011^*)^*(\varepsilon + 0)$. ⊛

459. Пусть регулярное выражение $b(ab)^*$ определяет некоторый язык над алфавитом $\Sigma = \{a, b\}$. Какие из следующих регулярных выражений задают тот же язык?

Q \ Σ	P_1			P_2		P_3	
	0	1	ε	0	1	0	1
q	p	r	s	p	r	p	s
p	r	—	—	r	—	r	t
r	—	—	q, s	p	r, s	p	s
s	—	t	—	—	—	p	s
t	—	—	—			t	t

Рис. 29: Программы автоматов из задачи 460.

Q \ Σ	P_4			P_5		P_6	
	0	1	ε	0	1	0	1
q	p	—	—	p	—	p	t
p	—	r, s	—	—	r, s	t	s
r	—	—	p, t	—	r, s	t	s
s	r	—	—	r	—	r	s
t	—	—	—			t	t

Рис. 30: Программы автоматов из задачи 461.

(а) $a(ba)^*$; (в) b^*ab^*; (д) $(ba)^*b(ab)^*$;

(б) $(ba)^*b$; (г) $b(ab + abab)^*$; (е) $(ab + ba)^*$. ⊛

460. Определить, какие из следующих трёх автоматов $\mathfrak{N}_1$, $\mathfrak{N}_2$, $\mathfrak{N}_3$ распознают язык, представляемый регулярным выражением $(00 + 1)^*1$:

(а) $\mathfrak{N}_1 = (\{q, p, r, s, t\}, \{0, 1\}, P_1, q, \{t\})$;

(б) $\mathfrak{N}_2 = (\{q, p, r, s\}, \{0, 1\}, P_2, q, \{s\})$;

(в) $\mathfrak{N}_3 = (\{q, p, r, s, t\}, \{0, 1\}, P_3, q, \{s\})$.

Программы автоматов заданы в таблице на рис. 29 (— означает отсутствие соответствующих переходов). ⊛

461. Определить, какие из следующих трёх автоматов $\mathfrak{N}_4$, $\mathfrak{N}_5$, $\mathfrak{N}_6$ распознают язык, представляемый регулярным выражением $0(10 + 1)^*$:

(а) $\mathfrak{N}_4 = (\{q, p, r, s, t\}, \{0, 1\}, P_4, q, \{t\})$;

(б) $\mathfrak{N}_5 = (\{q, p, r, s\}, \{0, 1\}, P_5, q, \{p, r\})$;

(в) $\mathfrak{N}_6 = (\{q, p, r, s, t\}, \{0, 1\}, P_6, q, \{p, r, s\})$.

Программы автоматов заданы в таблице на рис. 30 на противоположной странице (– означает отсутствие соответствующих переходов). ⊛

462. Доказать, что регулярное выражение $(1 + 01 + 001)^*(\varepsilon + 0 + 00)$ представляет язык, состоящий из всех слов в алфавите $\{0, 1\}$, которые не содержат подслово 000. ⊛

463. Определить, какой язык представляется следующими регулярными выражениями:

(а) 0^*1^*0;

(б) 01^*0;

(в) $(01^*)^*0$;

(г) $(00 + 11 + (01 + 10)(00 + 11)^*(01 + 10))^*$. ⊛

464. Доказать эквивалентности

(а) $r + p \equiv p + r$ (коммутативность объединения);

(б) $(r + p) + q \equiv r + (p + q)$ (ассоциативность объединения);

(в) $(rp)q \equiv r(pq)$ (ассоциативность конкатенации);

(г) $(r^*)^* \equiv r^*$ (идемпотентность итерации);

(д) $(r + p)q \equiv rq + pq$ (дистрибутивность). ⊛

465. Доказать следующие эквивалентности для регулярных выражений:

(а) $p^*(p + q)^* \equiv p^*(qp^*)^* \equiv (p + qp^*)^* \equiv (p + q)^*$;

(б) $p(qp)^* \equiv (pq)^*p$;

(в) $(p^*q^*)^* \equiv (q^*p^*)^*$;

(г) $(pq)^+(q^*p^* + q^*) \equiv (pq)^*pq^+p^*$. ⊛

466. Упростить следующие регулярные выражения:

(а) $00^*0 + (00)^*$; (в) $(0+\varepsilon)0^*1$;

(б) $(0+1)(\varepsilon+00)^+ + (0+1)$; (г) $(0^*+0)^*(01^*+0)$. ⊛

467. С помощью эквивалентных преобразований регулярных выражений упростить регулярное выражение ⊛

$$r = (aa)^* + a(aa)^*(ba(aa)^*)^*b(aa)^* + a(aa)^* + a(aa)^*(ba(aa)^*)^*ba(aa)^*.$$

468. Доказать, что каждое регулярное выражение r эквивалентно регулярному выражению s вида $s_1+\cdots+s_n$, где регулярные выражения $s_1,\ldots,s_n$ не содержат $+$. ⊛

469. Построить регулярное выражение, задающее язык L в алфавите $\Sigma = \{0,1\}$:

(а) $L = \{w : w$ содержит нечётное количество цифр 0 и чётное количество цифр $1\}$;

(б) $L = \{w : w$ содержит подслово 001 или подслово $110\}$;

(в) $L = \{w : w$ содержит по крайней мере два подряд идущих $0\}$;

(г) $L = \{w : w$ не содержит подслов 011 и $010\}$. ⊛

470. Пусть w — произвольное слово длины k, $m_1,\ldots,m_n$ — попарно различные натуральные числа, упорядоченные по возрастанию. Доказать, что язык $\{w^{m_1},\ldots,w^{m_n}\}$ можно описать регулярным выражением длины не большей $4km_n+4n-7$ (с учётом всех необходимых по определению символов). ⊛

471. Выше в задаче 445 на стр. 132 предлагалось построить автомат, который проверяет правильность сложения. Построить регулярное выражение, задающее распознаваемый этим автоматом язык S, то есть следующее множество слов в алфавите $\{0,1\}^3$: ⊛

$$S = \left\{ \begin{bmatrix} a_1 \\ b_1 \\ c_1 \end{bmatrix} \cdots \begin{bmatrix} a_n \\ b_n \\ c_n \end{bmatrix} : c_n\ldots c_1 \text{ — сумма двоичных чисел } a_n\ldots a_1 \text{ и } b_n\ldots b_1 \right\}.$$

472. Построить регулярное выражение для языка из задачи 447 на стр. 133. ⊛

473. Построить детерминированные конечные автоматы, распознающие языки, задаваемые следующими регулярными выражениями:

(а) $r_1 = aa(bb + aa)^*bb$;
(б) $r_2 = (a + bb)(a + b)^*(aa + bb)$;
(в) $r_3 = (a^+ + b^+)(ab + ba)^*$;
(г) $r_4 = (abb)^*(aab^* + bba^*)$;
(д) $r_5 = b^*(ab^*a)^*b^*$;
(е) $r_6 = (ab + ba)^*(aa + bb)^*$. ⊛

474. Для следующих пар регулярных выражений r_1 и r_2 построить регулярное выражение r, представляющее пересечение языков $L(r_1)$ и $L(r_2)$:

(а) $r_1 = aa(bab)^*bb$, $r_2 = a^*(bba)^*b^+$;
(б) $r_1 = (ba)^*(abb + ba)^*$, $r_2 = (abb)^*(a + bb)^*$;
(в) $r_1 = (b + ab)^*(aa + b)^*$, $r_2 = (ab + bb)^*(a^* + b^*)$;
(г) $r_1 = b^*(ab + abb)^*(a^* + bb^*)$, $r_2 = (b^* + (ab)^*)(b + aa)^*$. ⊛

475. Для следующих регулярных выражений r_1 построить регулярное выражение r, представляющее дополнение языка $L(r_1)$:

(а) $r_1 = (ba)^*(b + aa)^*$;
(б) $r_1 = (ab)^* + (ba + bba)^*$;
(в) $r_1 = (aa + ab)^*(b + a)^*$;
(г) $r_1 = (a + b)^*b(ba)^*b(bb)^*$. ⊛

476. Для следующих регулярных выражений r_1 построить регулярное выражение r с наименьшим количеством символов такое, чтобы выполнялось равенство $(L(r))^* = (L(r_1))^*$:

(а) $r_1 = (b + a)(a + bb)^*$;
(б) $r_1 = (a + ab + abb)^*b^*$;
(в) $r_1 = b^*(bba + ba)^*$;
(г) $r_1 = (baab + b(aa)^*)^*b^*$. ⊛

477. На рис. 31 на следующей странице представлены диаграммы четырёх недетерминированных конечных автоматов. Построить регулярные выражения, представляющие языки, которые распознаются этими автоматами. ⊛

478. Записать регулярное выражения для языка, который распознаётся автоматом на рис. 28 на стр. 136. ⊛

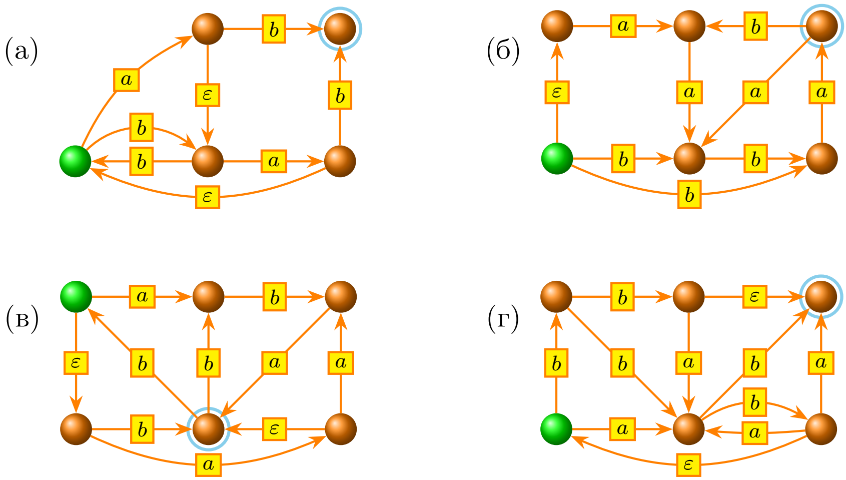


Рис. 31: Конечные автоматы из задачи 477.

20. Кодирования

Пусть Σ и Ω — два алфавита. Схемой кодирования из Σ в Ω называется конечное множество C пар вида (u, v), где $u \in \Sigma^*$, $v \in \Omega^*$. В частности, слова u, v могут быть пустыми.

Слово $y \in \Omega^*$ получено из $x \in \Sigma^*$ при помощи схемы кодирования C, если $x = u_1 \dots u_n$, $y = v_1 \dots v_n$ и $(u_1, v_1), \dots, (u_n, v_n) \in C$ для некоторых $u_1, \dots, u_n \in \Sigma^*$ и $v_1, \dots, v_n \in \Omega^*$. Ещё говорят, что y — это код слова x. В частности, при $n = 0$ из пустого слова всегда можно получить пустое.

С помощью $C[x]$ обозначаем множество кодов слова x, которые получены при помощи C.

Пусть C — схема кодирования из алфавита Σ в алфавит Ω, L — язык в алфавите Σ. Тогда с помощью $C(L)$ обозначаем код языка L, то есть множество всех слов, которые можно получить, кодируя слова из L при помощи C:

$$C(L) = \bigcup_{x \in L} C[x].$$

Построение кода языка или отдельного слова назовём кодированием. Класс автоматных языков замкнут относительно любых кодирований.

В частном случае схема кодирования C может быть функцией из алфавита Σ в множество слов Ω^*. Такие схемы называют гомоморфизмами. Таким образом, если $w = a_1 a_2 \dots a_n$, где $a_i \in \Sigma, i = 1, 2, \dots, n$, то множество кодов $C[w]$ содержит единственный элемент, который мы будем обозначать $C(w) = C(a_1)C(a_2)\dots C(a_n)$.

Образом $C(L)$ языка $L \subseteq \Sigma^*$ при гомоморфизме C называется язык $C(L) = \{C(w) : w \in L\}$, состоящий из образов всех слов языка L. Прообразом языка $K \subseteq \Omega^*$ при гомоморфизме C называется язык $C^{-1}(K) = \{w \in \Sigma^* : C(w) \in K\}$, состоящий из всех таких слов в алфавите Σ, чьи образы при гомоморфизме C попадают в K.

Ещё более частный случай, когда $\Omega \subseteq \Sigma$ и $C(a) = a$ при $a \in \Omega$, $C(a) = \varepsilon$ при $a \in \Sigma \setminus \Omega$. Такой гомоморфизм называется проекцией на алфавит

Ω, а образ слова получается из исходного вычёркиванием всех букв, не входящих в Ω.

Из замкнутости класса автоматных языков относительно кодирований следует его замкнутость относительно всех гомоморфизмов и их прообразов. Также класс автоматных языков замкнут относительно теоретико-множественных операций объединения, пересечения, дополнения, вычитания, а также относительно операций конкатенации, итерации.

Не для всякого гомоморфизма по образу можно однозначно восстановить исходное слово. В частности, это невозможно, если $C(a) = C(b)$ при $a \neq b$ или $C(a) = \varepsilon$.

В остальных случаях можно использовать критерий Маркова: гомоморфизм C разнозначен, если и только если в графе кодирования $\mathfrak{G}(C)$, кроме тривиальных петель, нет простых циклов, проходящих через вершину ε.

Графом кодирования для гомоморфизма C назовём размеченный ориентированный мультиграф $\mathfrak{G}(C) = (V, E)$, в котором ребро (x, y), $x, y \in V$, существует и помечено набором слов $(w_1, \ldots, w_k)$, если и только если $w_1, \ldots, w_k \in V_C$ и $xw_1 \ldots w_k y \in V_C$, где V_C — множество кодовых слов: $V_C = \{C(a) : a \in \Sigma\}$. Петлю $(\varepsilon, \varepsilon)$, помеченную одним кодовым словом $C(a)$, в графе $\mathfrak{G}(C)$ назовём тривиальной.

Разнозначными являются все беспрефиксные гомоморфизмы, то есть такие, что среди кодовых слов $C(a)$, $a \in \Sigma$, ни одно не является префиксом другого.

Разнозначный гомоморфизм из алфавита Σ в алфавит Ω с кодовыми словами длин $\ell_1, \ldots, \ell_n$, $n = |\Sigma|$, существует тогда и только тогда, когда выполнено неравенство Крафта-МакМиллана:

$$\sum_{i=1}^{n} |\Omega|^{-\ell_i} \leqslant 1.$$

Назовём беспрефиксный гомоморфизм $C : \Sigma^* \to \Omega^*$ оптимальным для слова $w \in \Sigma^*$, если $|C(w)| \leqslant |D(w)|$ для каждого беспрефиксного гомоморфизма $D : \Sigma^* \to \Omega^*$. Для построения оптимального гомоморфизма можно применять метод Хаффмана (алгоритм ОптКод, [3], § 17.3).

Задачи

479. Дана следующая схема кодирования из алфавита $\Sigma = \{a, b, c\}$ в алфавит $\Omega = \{0, 1, 2\}$: $C = \{(aa,\ 0),\ (abc,\ 11),\ (ba,\ 21),\ (c,\ \varepsilon),\ (bb,\ 2),\ (ab,\ 22),\ (b,\ 1)\}$. Найти множества кодов для следующих слов:

(а) $abaabac$;

(б) $bacbbaccab$;

(в) $aabcabccaab$;

(г) $abcbbaabc$. ⊛

480. Дана следующая схема кодирования из алфавита $\{0\}$ в алфавит $\{a, b\}$: $C = \{(0, a), (00, b)\}$. Доказать, что слово 0^n можно закодировать F_{n+1} способом, где F_i — соответствующее число Фибоначчи. ⊛

481. Пусть язык L в алфавите $\{a, b\}$, состоит из всех слов, которые начинаются на aa и содержат количество символов a кратное трём. Гомоморфизм $h : \{0, 1, 2\}^* \to \{a, b\}^*$ задан равенствами $h(0) = aaa$, $h(1) = ba$, $h(2) = \varepsilon$. Определить, какие из следующих трёх слов принадлежат прообразу $h^{-1}(L)$ языка L при гомоморфизме h: ⊛

(а) $w_1 = 21112$; (б) $w_2 = 20101012$; (в) $w_3 = 00211011$.

482. Пусть язык L в алфавите $\{a, b, c\}$ состоит из всех слов, которые начинаются на aa и содержат подслово bb. Какая из следующих фраз определяет язык $h(L)$, являющийся образом L при следующем гомоморфизме $h : \{a, b, c\}^* \to \{0, 1\}^*$: $h(a) = 01$, $h(b) = 11$, $h(c) = \varepsilon$?

(а) Все слова в алфавите $\{0, 1\}$, начинающиеся на 0101, с длиной, делящейся на 4.

(б) Все слова чётной длины в алфавите $\{0, 1\}$, начинающиеся на 0101.

(в) Все слова чётной длины в алфавите $\{0, 1\}$, начинающиеся на 0101, в которых каждый второй символ является единицей и которые содержат подслово 1111.

(г) Все слова в алфавите $\{0, 1\}$, начинающиеся на 0101, в которых на чётных местах стоят единицы и которые содержат подслово 1111. ⊛

483. Пусть язык L в алфавите $\{a, b, c\}$ состоит из всех слов, которые заканчиваются на bcc и содержат подслово aca. Какая из следующих фраз определяет язык $h(L)$, являющийся образом L при следующем гомоморфизме $h : \{a, b, c\}^* \to \{0, 1\}^*$: $h(a) = 00$, $h(b) = 10$, $h(c) = \varepsilon$?

(а) Все слова в алфавите $\{0, 1\}$, заканчивающиеся на 10, с длиной 12 или больше.

(б) Все слова чётной длины в алфавите $\{0, 1\}$, содержащие подслово 0000.

(в) Все слова чётной длины в алфавите $\{0, 1\}$, заканчивающиеся на 10, в которых каждый второй символ является нулём.

(г) Все слова в алфавите $\{0, 1\}$, заканчивающиеся на 10, в которых каждый второй символ является нулём и которые содержат подслово 0000.

(д) Все слова чётной длины в алфавите $\{0, 1\}$, заканчивающиеся на 10, в которых каждый шестой символ является нулём и которые содержат подслово 0000. $\circledast$

484. Дана следующая схема кодирования из алфавита $\Sigma = \{0, 1\}$ в алфавит $\Omega = \{a, b\}$: $C = \{(00, aa), (01, b), (1, a)\}$. Найти результаты кодирования следующих языков, представленных регулярными выражениями:

(а) $(0 + 1)^*$;

(б) $(01 + 10)^*$;

(в) $(000 + 11)^*$;

(г) $(101^* + 00)^*$. $\circledast$

485. Дана следующая схема кодирования из алфавита $\Sigma = \{0, 1\}$ в алфавит $\Omega = \{a, b\}$: $C = \{(000, ab), (10, a), (11, bb)\}$. Найти результаты кодирования следующих языков, представленных регулярными выражениями:

(а) $(0 + 1)^*$;

(б) $(10 + 0)^*(\varepsilon + 1)$;

(в) $(00 + 11)^*$;

(г) $(001 + 01 + 110)^*$. $\circledast$

486. Пусть даны детерминированный конечный автомат $\mathfrak{M} = (\{q_0, q_1, q_2, q_3\}, \{a, b, c\}, P, 0, \{2\})$ с программой $P = \{q_0, a \to q_1;\ q_0, b \to q_1;\ q_0, c \to q_0;\ q_1, a \to q_1;\ q_1, b \to q_2;\ q_1, c \to q_2;\ q_2, a \to q_3;\ q_2, b \to q_3;\ q_2, c \to q_2;\ q_3, a \to q_3;\ q_3, b \to q_3;\ q_3, c \to q_3\}$, а также — гомоморфизм $h : \{a, b, c\}^* \to \{0, 1\}^*$, для которого $h(a) = 01$, $h(b) = 11$, $h(c) = \varepsilon$. Какие из следующих трёх автоматов $\mathfrak{M}_1$, $\mathfrak{M}_2$, $\mathfrak{M}_3$ распознают гомоморфный образ $h(L_A)$?

(а) $\mathfrak{M}_1 = (\{q_i : i = 0, \ldots, 11\}, \{0, 1\}, P_1, q_0, F_1 = \{q_1, q_2\})$;

(б) $\mathfrak{M}_2 = (\{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}, \{0, 1\}, P_2, q_0, F_2 = \{q_1, q_2\})$;

Q / Σ	P_1											
	q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}
0	q_4	q_6	q_8	q_{10}	—	—	—	—	—	—	—	—
1	q_5	q_7	q_9	q_{11}	q_1	q_1	q_1	q_2	q_3	q_3	q_3	q_3

Q / Σ	P_2						
	q_0	q_1	q_2	q_3	q_4	q_5	q_6
0	q_4	q_5	q_3	q_3	q_3	q_3	q_3
1	q_4	q_6	q_3	q_3	q_1	q_1	q_2

Q / Σ	P_3						
	q_0	q_1	q_2	q_3	q_4	q_5	q_6
0	q_4	q_5	q_3	q_3	q_4	q_3	q_3
1	q_4	q_6	q_3	q_3	q_1	q_1	q_2

Рис. 32: Программы автоматов из задачи 486.

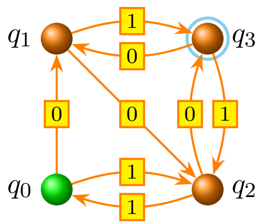


Рис. 33: Автомат из задачи 487.

(в) $\mathfrak{M}_3 = (\{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}, \{0, 1\}, P_3, q_0, F_3 = \{q_0, q_1, q_2\})$.

Программы автоматов заданы таблицами на рис. 32 (— означает отсутствие соответствующего перехода). ⊛

487. Дан детерминированный конечный автомат (рис. 33), распознающий некоторый язык L. Построить недетерминированный конечный автомат, распознающий язык $C(L)$ для схем кодирования C из задач (а) 484 и (б) 485. ⊛

488. Пусть гомоморфизм $C : \{0,1,2\}^* \to \{a,b\}^*$ определяется равенствами $C(0) = ab$, $C(1) = b$, $C(2) = aa$.

Построить детерминированный конечный автомат, который распознаёт образ $C(L)$ языка ⊛

$$L = \{w : w \text{ начинается не с } 1 \text{ и не содержит } 00\}.$$

489. Пусть гомоморфизм $C : \{0,1\}^* \to \{a,b,c\}^*$ определяется равенствами $C(0) = aa$, $C(1) = bc$.

Построить детерминированный конечный автомат, который распознаёт образ $C(L)$ языка ⊛

$$L = \{w : w \text{ начинается с } 0 \text{ и содержит } 11\}.$$

490. Пусть гомоморфизм $C : \{0,1,2\}^* \to \{a,b,c\}^*$ определяется равенствами $C(0) = ba$, $C(1) = \varepsilon$, $C(2) = bc$.

Построить детерминированный конечный автомат, который распознаёт язык $C(L)$ для языка ⊛

$$L = \{w : \text{цифры } 2 \text{ встречаются в } w \text{ блоками чётной длины}$$
$$\text{и хоть один такой блок имеется}\}$$

491. Пусть гомоморфизм $C : \{a,b,c\}^* \to \{0,1\}^*$ определяется равенствами $C(a) = 10$, $C(b) = 01$, $C(c) = \varepsilon$.

Построить детерминированный конечный автомат, который распознаёт язык $C^{-1}(L)$ для языка ⊛

$$L = \{w : w \text{ заканчивается на } 01$$
$$\text{и содержит чётное количество единиц}\}.$$

492. Пусть гомоморфизм $C : \{a,b,c\}^* \to \{0,1\}^*$ определяется равенствами $C(a) = 01$, $C(b) = 11$, $C(c) = \varepsilon$.

Построить детерминированный конечный автомат, который распознаёт прообраз $C^{-1}(L)$ языка ⊛

$$L = \{w : w \text{ начинается с } 11 \text{ и не содержит } 010\}.$$

493. Пусть гомоморфизм $C : \{a, b, c\}^* \to \{0, 1\}^*$ определяется равенствами $C(a) = 00$, $C(b) = 011$, $C(c) = 01$.

Построить детерминированный конечный автомат, который распознаёт язык $C^{-1}(L)$ для языка ⊛

$$L = \{w : w \text{ содержит подслово } 001 \text{ или подслово } 10\}.$$

494. Доказать, что любую операцию кодирования C для языка можно представить в виде суперпозиции прообраза гомоморфизма ψ и гомоморфизма φ: $C(L) = \varphi(\psi^{-1}(L))$ для некоторых ψ и φ. ⊛

495. Определим операцию ц и л и н д р и ф и к а ц и и, которая обратна операции проекции. Пусть C — проекция алфавита Σ на алфавит $\Omega \subseteq \Sigma$. Для любого языка $L \subseteq \Omega^*$ определим его цилиндрификацию до алфавита Σ как язык

$$Z_\Sigma(L) = \{w \in \Sigma^* : C(w) \in L\}.$$

Показать, что для автоматного языка L язык $Z_\Sigma(L)$ также является автоматным языком. Предложить процедуру перестройки автомата, распознающего L, в автомат, распознающий $Z_\Sigma(L)$. ⊛

496. Показать, что если гомоморфизм $C : \Sigma^* \to \Omega^*$ является беспрефиксным, то результат обратного кодирования C^{-1} может быть вычислен с помощью конечного преобразователя. Считаем, что $C^{-1}(w)$ может быть любым, если $w \notin \operatorname{rng} C$. ⊛

497. Используя критерий Маркова доказать, что с помощью схем кодирования из задач 484 и 485 каждое слово можно закодировать не более чем одним способом. ⊛

498. С помощью критерия Маркова определить, можно ли для схем кодирования из задач 484 и 485 по коду слова однозначно восстановить само слово. ⊛

499. Построить граф кодирования для первых шести кодовых слов азбуки Морзе: $C_m(a)$ = «·—»; $C_m(b)$ = «—···»; $C_m(c)$ = «—·—·»; $C_m(d)$ = «—··»; $C_m(e)$ = «·»; $C_m(f)$ = «··—·». ⊛

500. С помощью критерия Маркова проверить, будет ли следующий гомоморфизм разнозначным: $C(a) = 01$, $C(b) = 100$, $C(c) = 0110$, $C(d) = 11$, $C(e) = 0100$, $C(f) = 101$. Если нет, то найти слово, которое нельзя однозначно декодировать. ⊛

501. С помощью критерия Маркова проверить, будет ли следующий гомоморфизм разнозначным: $C(a) = 0101$, $C(b) = 100$, $C(c) = 1011$, $C(d) = 0010$, $C(e) = 110011$, $C(f) = 00$. Если нет, то найти слово, которое нельзя однозначно декодировать. ⊛

502. Доказать, что беспрефиксные гомоморфизмы разнозначны с помощью критерия Маркова. ⊛

503. Построить с помощью метода Хаффмана оптимальные двоичные кодирования для слов

(а) «каракатица»,
(б) «параллелепипед»,
(в) «телеаппаратура»,
(г) «индивидуальность»,
(д) «перераспределение»
(е) «обороноспособность»
(ж) «стронгилоцентротус»
(з) «тартароблатта».

Определить длину получившихся кодов слов. Вычислить, насколько оптимальное кодирование даёт результат короче, чем двоичное равномерное, то есть когда коды всех символов имеют одну и ту же длину. ⊛

504. Построить конечные преобразователи, выполняющие декодирование из задачи 503. ⊛

505. Индукцией по построению доказать, что для двоичного кода Хаффмана сумма в неравенстве Крафта-МакМиллана в точности равна единице. ⊛

506. Доказать, что для любого оптимального двоичного гомоморфизма сумма в неравенстве Крафта-МакМиллана в точности равна единице. Продемонстрировать, что для недвоичных гомоморфизмов это может быть неверно. ⊛

507. Пусть частоты символов a_i, $i \in I$, одинаковы. Доказать, что

(а) в любом оптимальном гомоморфизме длины кодовых слов $C(a_i)$ отличаются не более чем на 2;

(б) существует оптимальный гомоморфизм, в котором длины кодовых слов $C(a_i)$ отличаются не более чем на 1;

(в) в любом оптимальном двоичном гомоморфизме длины кодовых слов $C(a_i)$ отличаются не более чем на 1. ⊛

508. Требуется построить разнозначный двоичный гомоморфизм из алфавита $\{a, b, c, d, e, f, g, h\}$. По некоторым причинам для символов a, b, c, d, e, f решено использовать кодовые слова длин 5, 1, 3, 3, 4, 8 соответственно. Известно, что средние частоты символов g и h равны $1/8$ и $1/5$ соответственно. Найти оптимальные длины кодовых слов для g и h. ⊛

509. Требуется построить оптимальный двоичный гомоморфизм из алфавита $\{a, b, c, d, e, f, g, h\}$. Известно, что первые три символа встречаются одинаково часто, а остальные — намного реже, но тоже с одинаковой частотой. ⊛

510. Пусть C — схема кодирования из алфавита Σ в себя, L — автоматный язык алфавита Σ. Доказать, что язык $C^*(L)$ тоже является автоматным:

$$C^*(L) = \bigcup\{C[x] : x \in L, C[x] \neq \varnothing\} \cup \{x : x \in L, C[x] = \varnothing\}.$$

Таким образом, $C^*(L)$ содержит коды слов из L, которые можно закодировать, а также все слова из L, которые закодировать нельзя. ⊛

21. Замкнутость класса автоматных языков. Неавтоматные языки

Лемма о разрастании описывает необходимое условие для того, чтобы язык был автоматным. Пусть L — бесконечный автоматный язык. Тогда существует такая константа n, что в любом слове $\alpha u \beta \in L$, в котором $|u| \geqslant n$, фрагмент u можно разбить на три части x, y и z так, что

1) $u = xyz$;
2) $1 \leqslant |y| \leqslant n$;
3) слово $w_m = \alpha x y^m z \beta$ принадлежит L для любого натурального числа m.

Содержательно, лемма утверждает, что у всякого достаточно длинного фрагмента слова из автоматного языка имеется непустая часть, которую можно вырезать или повторить сколько угодно раз, оставаясь внутри языка.

Для доказательства неавтоматности L можно использовать схему доказательства от противного:

- предположим, что L автоматный язык. Тогда для него имеется константа n из утверждения леммы о разрастании;
- определим по n некоторое «специальное» слово w из L и его фрагмент u длины больше n: $w = \alpha u \beta$, $|u| \geqslant n$. Докажем, что для любого разбиения $u = xyz$, удовлетворяющего условиям 1) и 2) леммы, найдётся такое $m \geqslant 0$, что слово $w_m = \alpha x y^m z \beta$ не принадлежит L;
- на основании полученного противоречия делаем вывод, что L не автоматный язык.

Другой способ доказательства неавтоматности языка — использование свойств замкнутости и языков, про которые уже известно, что они неавтоматные. Пусть O — некоторая операция, относительно которой класс автоматных языков замкнут, а L' — неавтоматный язык. Тогда можно использовать такую схему рассуждений:

- допустим, что язык L — автоматный;
- подберём автоматные языки $L_1, \ldots, L_n$ таким образом, чтобы выполнялось $O(L, L_1, \ldots, L_n) = L'$;
- в силу замкнутости класса автоматных языков относительно O и автоматности языков $L, L_1, \ldots, L_n$ делаем вывод, что L' — автоматный;
- на основании полученного противоречия делаем вывод, что L — неавтоматный язык.

Примерами неавтоматных языков являются:

- $L_1 = \{w = 0^i 1^i : i \in \omega\}$;
- СКОБ — язык правильных скобочных последовательностей в алфавите $\{(,)\}$;
- $L_2 = \{w = 0^i 1^j : i \leqslant 2j + 1\}$;
- $L_3 = \{a^{i^2} : i \geqslant 0\}$ — язык «квадратов» в унарном алфавите $\{a\}$;
- $L_{pr} = \{a^p : p$ — простое число$\}$ — язык «простых чисел» в унарном алфавите $\{a\}$;
- $L_4 = \{0^i 1^j : i \neq j\}$.

Задачи

511. Обращением слова $w = a_1 a_2 \ldots a_k$, $a_i \in \Sigma$ при $i = 1, \ldots, k$, называется слово $w^{-1} = a_k \ldots a_2 a_1$. Показать, что для автоматного языка L его обращение — язык $L^{-1} = \{w^{-1} : w \in L\}$ — также является автоматным. ⊛

512. Пусть L — автоматный язык в алфавите Σ. Доказать, что автоматными являются и следующие языки:

(а) $\mathrm{PREF}(L) = \{w :$ есть такое слово $x \in \Sigma^*$, что $wx \in L\}$;

(б) $\mathrm{SUFF}(L) = \{w :$ есть такое слово $x \in \Sigma^*$, что $xw \in L\}$;

(в) $\mathrm{INF}(L) = \{w :$ есть такие слова $x, y \in \Sigma^*$, что $xwy \in L\}$;

(г) $\mathrm{MAX}(L) = \{w \in L : wx \notin L$ для всякого непустого слова $x\}$;

(д) $\mathrm{MIN}(L) = \{w \in L : x \notin L$ для всякого собственного префикса x слова $w\}$;

(е) $\mathrm{DEL}(L) = \{xz : xyz \in L$ для некоторого слова $y\}$;

(ж) $\mathrm{CYCLE}(L) = \{yx : xy \in L\}$. ⊛

513. Пусть L — автоматный язык в алфавите $\Sigma = \{a_1, \ldots, a_m\}$, а $L_1, \ldots, L_m$ — это автоматные языки в алфавите Δ. Доказать, что автоматным является и язык $\mathrm{SUBST}(L)$, полученный из слов L заменой каждой буквы a_i на некоторое слово из L_i. Таким образом, ⊛

$$\mathrm{SUBST}(L) = \{w_1 w_2 \ldots w_n : w_1 \in L_{i_1},\ \ldots,\ w_n \in L_{i_n} \text{ и существует } a_{i_1} a_{i_2} \ldots a_{i_n} \in L\}.$$

514. Пусть L_1 — автоматный язык в алфавите Σ, а L_2 — произвольный язык в том же алфавите. Доказать, что язык

$$L_1/L_2 = \{w \in \Sigma^* : \text{существует такое слово } x \in L_2, \text{ что } wx \in L_1\}$$

также является автоматным. ⊛

515. *Тасовкой* языков L_1 и L_2 называется язык

$$\mathrm{SHUFFLE}(L_1, L_2) = \{x_1 y_1 \ldots x_n y_n : x_1 \ldots x_n \in L_1, y_1 \ldots y_n \in L_2\}.$$

Здесь $x_1, \ldots, x_n, y_1, \ldots, y_n$ — произвольные слова. Доказать, что если языки L_1 и L_2 являются автоматными, то язык $\mathrm{SHUFFLE}(L_1, L_2)$ тоже будет автоматным. ⊛

516. Пусть язык L в алфавите $\{a, b, c\}$ состоит из всех слов, в которых количество букв b превосходит количество букв a не менее чем на 2. Предположим, что L — автоматный язык, а n — это константа, которая существует для него по утверждению леммы о разрастании. Какие из следующих «специальных» слов позволяют опровергнуть это предположение, то есть для какого из них не выполнено утверждение 3) леммы о разрастании?

(а) $c^n bbbaaabb$; (в) $cb^{n+2}a^2$; (д) $b^n ca^n bbb$;
(б) $ba^n b^{n+4} aaa$; (г) $b^{n+2} ca^n c$; (е) $(ab)^n ca^n bbb$. ⊛

517. Пусть язык L в алфавите $\{a, b\}$ состоит из всех слов нечётной длины, средней буквой в которых является a. Предположим, что L — автоматный язык, а n — это константа из леммы о разрастании. Какие из следующих «специальных» слов позволяют опровергнуть это предположение, то есть для какого из них не выполнено утверждение 3) леммы о разрастании?

(а) $bbbabbb$; (в) $a^n b^{n+1}$; (д) $b^n ab^n$; (ж) $(ab)^{2n} a$;
(б) $a^{n+1} b^n$; (г) $(ab)^n a$; (е) $a^n ba^n$; (з) a^{2n+1}. ⊛

518. Для каких из следующих языков L в алфавите $\Sigma = \{a, b, c\}$ слово $w = a^n (bc)^n a^n$ может быть использовано, чтобы опровергнуть автоматность L с помощью леммы о разрастании, если предположить, что n — это константа из леммы?

(а) $L = \{w :$ в слове w количества b и c равны$\}$;

(б) $L = \{w :$ в слове w количество a равно суммарному количеству остальных букв$\}$;

(в) $L = \{w :$ в слове w двумя средними буквами являются согласные$\}$;

(г) $L = \{w :$ в слове w количество a превосходит количество $b\}$;

(д) $L = \{w :$ слово w имеет чётную длину и его вторая половина содержит $a\}$;

(е) $L = \{w :$ в слове w все последовательности букв a имеют одинаковую длину$\}$;

(ж) $L = \{w :$ слово w имеет чётную длину и его первая половина не содержит $cc\}$;

(з) $L = \{w :$ в слове w количество a превосходит количество каждой из других букв$\}$. ⊛

519. Доказать, что следующие языки в алфавите $\Sigma = \{a, b, c\}$ не являются автоматными:

(а) множество всех слов, в которых букв a на 3 больше, чем букв b;

(б) $L = \{a^n c b^m : m > 3n\}$;

(в) $L = \{wcw^{-1} : w = a^2 b^n a$ для некоторого $n > 0\}$;

(г) $L = \{w : |w| = 2^n$ для некоторого натурального $n\}$;

(д) $L = \{wc^{|w|} : w \in \{a, b\}^*$, $|w|$ — длина слова $w\}$. ⊛

520. ДНФ записываются с использованием алфавита $\Sigma = \{\wedge, \vee, \neg, a\}$, переменная x_i обозначается повторением i раз буквы a, например, $x_1 \wedge \neg x_2 \vee x_3 \wedge x_2$ выглядит так: $a \wedge \neg aa \vee aaa \wedge aa$. Определить, будет ли автоматным язык:

(а) L_0, состоящий из всех ДНФ;

(б) L_1, состоящий из тождественно истинных ДНФ;

(в) L_2, состоящий из тождественно ложных ДНФ;

(г) L_3, состоящий из выполнимых ДНФ. ⊛

521. Пусть $\boldsymbol{V}$ — это конечное множество переменных, $\boldsymbol{L} = \{$«(», «)», «λ»$\}$. Тогда *λ-выражение* — это слово в алфавите $\boldsymbol{V} \cup \boldsymbol{L}$, определяемое индуктивно: либо переменная $x \in \boldsymbol{V}$, либо «$\lambda x\, e_1$», либо «$(e_1\, e_2)$», где $x \in \boldsymbol{V}$, e_1, e_2 — λ-выражения. Например, слова «$\lambda x\, x$», «$\lambda x\, (x\, x)$», «$\lambda x\, \lambda x\, (\lambda x\, (x\, x)\, \lambda x\, (x\, x))$» — это λ-выражения, а слова «$(x\, \lambda x)$», «$\lambda x\, (\lambda x)$» и «$\lambda x\, ((x\, x)$» λ-выражениями не являются. Доказать, что множество λ-выражений в алфавите $\boldsymbol{V} \cup \boldsymbol{L}$ не является автоматным. ⊛

522. Выше в задаче 445 на стр. 132 строился автомат, который проверял правильность сложения двоичных чисел. Доказать, что для операции умножения двоичных чисел такого автомата не существует. Точнее, следующий язык в алфавите трёхэтажных символов не является автоматным: ⊛

$$\left\{ \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \ldots \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} : x_i, y_i, z_i \in \{0, 1\} \text{ для } i = 1, \ldots, n \text{ и } z_n \ldots z_1 \text{ — это произведение двоичных чисел } x_n \ldots x_1 \text{ и } y_n \ldots y_1 \right\}.$$

523. Пусть f — это некоторая m-местная функция на множестве натуральных чисел. Предположим, что существует конечный автомат, который проверяет корректность f по аналогии с задачами 445 на стр. 132 и 522 на предыдущей странице. Это означает, что на вход этому автомату подаются $(m+1)$-этажные символы, на верхних этажах записаны в двоичном виде аргументы, на нижнем — предполагаемый результат. Доказать, что тогда существует константа k, для которой имеет место оценка $f(x_1,\ldots,x_m) \leqslant k\max\{x_1,\ldots,x_m\}$ для произвольных натуральных чисел $x_1,\ldots,x_m$. ⊛

524. Пусть f — некоторая одноместная функция на натуральных числах, которая монотонно не убывает. Предположим, что существует конечный автомат, который проверяет корректность f по аналогии с задачей 523. Доказать что либо функция f ограничена, либо существует константа $k>0$, для которой выполнена оценка $f(x) \geqslant kx$ для любого x. ⊛

525. Доказать, что условие монотонного неубывания функции f в задаче 524 является существенным. Если его исключить, то утверждение может быть неверным. ⊛

526. Доказать, что следующий язык в алфавите $\Sigma = \{a,b,c\}$ не является автоматным: ⊛

$$L = \{w \in \Sigma^* : \text{количества букв } a \text{ и } b \text{ в слове } w \text{ различны}\}.$$

527. Используя лемму о разрастании, установить, какие из следующих языков в алфавите $\{a,b\}$ не являются автоматными.

(а) $L_1 = \{a^2b^na^2 : n>0\}$;

(б) $L_2 = \{vv : v = a^2b^na^2, n>0\}$;

(в) $L_3 = \{v'v'' : v' = a^2b^na^2,\ v'' = b^2a^mb^2$ для некоторых $n,m>0\}$;

(г) $L_4 = \{(ab)^i(ab)^i : i>0\}$;

(д) $L_5 = \{(ab)^iba(ab)^i : i>0\}$;

(е) $L_6 = \{a^{\lfloor\sqrt{n}\rfloor} : n \in \omega\}$;

(ж) $L_7 = \{a^{\lfloor n|\sin(n^2+n)|\rfloor} : n \in \mathbb{R}, n \geqslant 0\}$;

(з) $L_8 = \{a^{n^2+n} : n \in \omega\}$. ⊛

528. Пусть C — схема кодирования из алфавита Σ в себя. Привести пример, показывающий, что следующий язык может не быть автоматным:

$$L = \{x \in \Sigma^* : x \in C[x]\},$$

то есть L — множество слов, которые при кодировании могут переходить в себя же. ⊛

529. Операция *коммутативного замыкания* COMM заключается в произвольной перестановке букв слов языка:

$$\mathrm{COMM}(L) = \{a_{f(1)}a_{f(2)}\dots a_{f(n)} : f \text{ — перестановка множества } \{1, 2, \dots, n\} \text{ и существует } a_1a_2\dots a_n \in L\}.$$

Верно ли такое утверждение: если язык L автоматный, то и язык COMM(L) тоже автоматный? ⊛

530. Доказать, что для односимвольного алфавита лемма о разрастании является не только необходимым, но и достаточным признаком автоматного языка: если указанная в лемме константа n для языка L существует, то язык L автоматный.

Указание. Рассмотреть слова короче n и все остальные, последние разбить на классы в соответствием с остатком от деления длины слова на $n!$. ⊛

531. Пусть $C : \Sigma^* \to \Omega^*$ — гомоморфизм языков. Доказать, что проверка корректности φ при помощи конечного автомата (по аналогии с задачами 523 и 524 на предшествующей странице) возможна тогда и только тогда, когда имеет место один из двух следующих случаев:

(а) $|C(a)| = 1$ для всех $a \in \Sigma$;

(б) $|C(a)| = 0$ для всех $a \in \Sigma$.

При проверке условия $C(u) = v$ мы считаем, что более короткое слово дополняется справа специальным символом Λ. Например, для проверки $C(ab) = aacb$ на вход автомату подаётся слово ⊛

$$\begin{bmatrix}\!\begin{bmatrix} a \\ a \end{bmatrix}\!\end{bmatrix} \begin{bmatrix}\!\begin{bmatrix} b \\ a \end{bmatrix}\!\end{bmatrix} \begin{bmatrix}\!\begin{bmatrix} \Lambda \\ c \end{bmatrix}\!\end{bmatrix} \begin{bmatrix}\!\begin{bmatrix} \Lambda \\ b \end{bmatrix}\!\end{bmatrix}.$$

532. Доказать, что для любого натурального числа n существует язык L_n, который может быть распознан (не)детерминированным конечным автоматом с $n+1$ состоянием, но не может быть распознан никаким автоматом с n состояниями. ⊛

533. Пусть k — мощность алфавита. Доказать, что для любого натурального числа n недетерминированный конечный автомат с n состояниями не может распознавать никакой конечный язык, содержащий больше чем

(а) n слов при $k = 1$;

(б) $\dfrac{k^n - 1}{k - 1}$ слов при $k > 1$. ⊛

534. Доказать аналог леммы о разрастании для конечных преобразователей. Пусть $f : \Sigma^* \to \Omega^*$ — функция, вычисляемая некоторым конечным преобразователем $\mathfrak{M}$. Тогда существуют константы n и m такие, что для любого слова $w \in \Sigma^*$ и любого его фрагмента u длины n или более: $w = \alpha u \beta$, $|u| \geqslant n$, выполнено следующее. Существует разбиение $u = xyz$, $1 \leqslant |y| \leqslant n$, и разбиение $f(w) = XYZ \in \Omega^*$ такие, что $f(\alpha x y^i z \beta) = XY^iZ$ для всех натуральных i, $|X| \leqslant m|\alpha x|$, $|Y| \leqslant m|y|$ и $|Z| \leqslant m|z\beta|$. ⊛

535. Пользуясь задачей 534, показать, что не существует конечных преобразователей, которые выполняли бы перевод числа из унарной записи в двоичную и наоборот (см. раздел 24). ⊛

536. Пользуясь задачей 534, показать, что не существует конечных преобразователей, которые выполняли бы «переворачивание» слова в алфавите $\{a, b\}$. ⊛

537. Найти разнозначный гомоморфизм $C : \Sigma^* \to \Omega^*$, который вычисляется некоторым конечным преобразователем, но обратное к C кодирование C^{-1} никаким конечным преобразователем выполнено быть не может. Считаем, что $C^{-1}(w)$ может быть любым, если $w \notin \operatorname{rng} C$. ⊛

22. Программная вычислимость

Структурированные программы. Пусть зафиксировано некоторое счётно бесконечное множество переменных $\boldsymbol{V}$. *Оператором присваивания* называется строка одного из следующих видов, где $x, y, z \in \boldsymbol{V}$ — переменные:

1) $x \leftarrow y;$
2) $x \leftarrow 0;$
3) $x \leftarrow s(y);$
4) $x \leftarrow (y < z);$

Точки с запятой являются частями соответствующих строк. Каждый оператор присваивания является структурированной программой. Если Π_1 и Π_2 — структурированные программы, $x \in \boldsymbol{V}$ — переменная, то следующие строки тоже являются структурированными программами:

1) $\Pi_1\,\Pi_2$
2) $\mathsf{if}\ x\ \mathsf{then}\ \Pi_1\ \mathsf{end};$
3) $\mathsf{if}\ x\ \mathsf{then}\ \Pi_1\ \mathsf{else}\ \Pi_2\ \mathsf{end};$
4) $\mathsf{while}\ x\ \mathsf{do}\ \Pi_1\ \mathsf{end};$

Точки с запятой снова являются частями соответствующих строк. Программы первого вида называют *следованиями*, второго — *неполными ветвлениями*, третьего — *полными ветвлениями*, четвёртого — *циклами*.

С помощью $\mathrm{Var}(\Pi)$ обозначается множество переменных, входящих в программу Π.

Конфигурация σ структурированной программы Π — это отображение множества $\mathrm{Var}(\Pi)$ в множество натуральных чисел ω.

Семантикой структурированной программы Π называется (частичное) отображение множества конфигураций Π в себя, значение которого для конфигурации σ обозначается $\Pi(\sigma)$ и определяется следующим образом:

1) если Π — это $x \leftarrow y;$, то $\tau = \Pi(\sigma)$ совпадает с σ, кроме того, что $\tau(x) = \sigma(y)$;
2) если Π — это $x \leftarrow 0;$, то $\tau = \Pi(\sigma)$ совпадает с σ, кроме того, что $\tau(x) = 0$;
3) если Π — это $x \leftarrow s(y);$, то $\tau = \Pi(\sigma)$ совпадает с σ, кроме того, что $\tau(x) = \sigma(y) + 1$;
4) если Π — это $x \leftarrow (y < z);$, то $\tau = \Pi(\sigma)$ совпадает с σ, кроме того, что $\tau(x) = 1$ при $\sigma(y) < \sigma(z)$ и $\tau(x) = 0$ в противном случае;
5) если Π — это $\Pi_1\,\Pi_2$, то $\Pi(\sigma) = \Pi_2(\Pi_1(\sigma))$;
6) если Π — это $\mathsf{if}\ x\ \mathsf{then}\ \Pi_1\ \mathsf{end};$, то $\Pi(\sigma) = \Pi_1(\sigma)$ при $\sigma(x) \neq 0$ и $\Pi(\sigma) = \sigma$ в противном случае;
7) если Π — это $\mathsf{if}\ x\ \mathsf{then}\ \Pi_1\ \mathsf{else}\ \Pi_2\ \mathsf{end};$, то $\Pi(\sigma) = \Pi_1(\sigma)$ при $\sigma(x) \neq 0$ и $\Pi(\sigma) = \Pi_2(\sigma)$ в противном случае;
8) если Π — это $\mathsf{while}\ x\ \mathsf{do}\ \Pi_1\ \mathsf{end};$, то $\Pi(\sigma) = \sigma_n$, если n — это наименьшее натуральное число такое, что $\sigma_n(x) = 0$, где $\sigma_0 = \sigma$, $\sigma_{i+1} = \Pi_1(\sigma_i)$. Если такого n не существует, то $\Pi(\sigma)$ неопределено, говорят также, что программа зацикливается.

Программы с метками. Пусть зафиксированы некоторые счётно бесконечные множества переменных $\boldsymbol{V}$ и меток $\boldsymbol{L}$, которые не пересекаются. Программой с метками называется конечная последовательность строк одного из двух видов:

1) $\alpha\ A\ \beta$
2) $\alpha\ \mathsf{if}\ x\ \mathsf{then}\ \beta\ \mathsf{else}\ \gamma$

Здесь A — оператор присваивания, $x \in \boldsymbol{V}$ — переменная, $\alpha, \beta, \gamma \in \boldsymbol{L}$ — метки. При этом α называется меткой оператора, β и γ — метками перехода. В программе не должно быть двух одинаковых меток оператора. Строки первого вида называют присваиваниями, второго — ветвлениями. Как и ранее, $\mathrm{Var}(\Pi)$ — множество переменных, входящих в программу Π.

Конфигурацией программы с метками Π называется пара (σ, α), где σ — отображение $\mathrm{Var}(\Pi)$ в ω, $\alpha \in \boldsymbol{L}$ — метка.

Семантикой программы с метками П называется (частичное) отображение множества конфигураций П в себя, значение которого для конфигурации (σ, α) обозначается $\Pi(\sigma, \alpha)$ и равняется (σ_n, α_n), если n — наименьшее натуральное число, для которого α_n не является меткой оператора программы П. Если такого n не существует, то $\Pi(\sigma, \alpha)$ неопределено. Здесь $(\sigma_0, \alpha_0) = (\sigma, \alpha)$, а $(\sigma_{i+1}, \alpha_{i+1})$ определяется так:

1) если в П есть присваивание $\alpha_i\ A\ \beta_i$, то $\sigma_{i+1} = A(\sigma_i)$, $\alpha_{i+1} = \beta_i$;
2) если в П есть ветвление α_i if x then β_i else γ_i, то $\sigma_{i+1} = \sigma_i$, $\alpha_{i+1} = \beta_i$ при $\sigma_i(x) \neq 0$ и $\alpha_{i+1} = \gamma_i$ при $\sigma_i(x) = 0$.

Если α — метка первого оператора программы П, $\Pi(\sigma, \alpha) = (\tau, \beta)$ для некоторой метки β, то полагаем $\Pi(\sigma) = \tau$.

Вычислимые функции. Пусть П — программа (структурированная или с метками). Выделим каким-либо образом среди переменных $\mathrm{Var}(\Pi)$ программы П входные переменные $x_1, \ldots, x_n$ и выходную переменную y. Говорим, что программа П вычисляет (частичную) функцию f, если для любых натуральных $a_1, \ldots, a_n$ выполнено одно из двух:

1) $f(a_1, \ldots, a_n) = b$ и $\Pi(\sigma) = \tau$ для некоторого τ такого, что $\tau(y) = b$;
2) $f(a_1, \ldots, a_n)$ неопределено и $\Pi(\sigma)$ неопределено.

Здесь $\sigma(x_i) = a_i$ при $i = 1, \ldots, n$ и $\sigma(z) = 0$ для остальных переменных z.

С помощью структурированных программ и программ с метками можно вычислять одни и те же (частичные) функции. (Частичные) функции, для которых существуют вычисляющие их программы, называются (алгоритмически, программно) вычислимыми.

Языки программирования (структурированные или с метками) можно расширять, допуская присваивания вида $y \leftarrow f(x_1, \ldots, x_n);$, где f — n-местная вычислимая функция. Выразительные возможности языков при этом не изменяются.

Задачи

538. Определить, какие из следующих строк являются синтаксически правильными структурированными программами:

(a) $x \leftarrow s(x);$ $z \leftarrow y;$ $u \leftarrow (s(x) < z);$ if u then $y \leftarrow z;$ else $y \leftarrow x;$ end;

(б) $x \leftarrow s(x)$; $z \leftarrow y$; $z \leftarrow s(z)$; if $x \leftarrow z$ then $y \leftarrow z$; else $y \leftarrow x$; end;

(в) $x \leftarrow y$; $u \leftarrow z$; $u \leftarrow s(u)$; $x \leftarrow (u > z)$; while x do $y \leftarrow z$; $u \leftarrow s(u)$; end;

(г) $x \leftarrow s(z)$; $z \leftarrow y$; $x \leftarrow (x < x)$; if x then $y \leftarrow z$; else $x \leftarrow y$; end;

(д) $x \leftarrow x$; $x \leftarrow x$; $y \leftarrow y$; $y \leftarrow y$;

(е) $x \leftarrow y$; $u \leftarrow z$; $u \leftarrow s(u)$; $x \leftarrow (z < u)$; while x do $y \leftarrow z$; $u \leftarrow s(u)$; end; $x \leftarrow y$;

(ж) $x \leftarrow s(x)$; $z \leftarrow y$; $z \leftarrow s(z)$; $x \leftarrow (x \leftarrow z)$; if x then $y \leftarrow z$; $z \leftarrow s(z)$; else $y \leftarrow x$; end; ⊛

539. Для следующей структурированной программы Π найти $\Pi(\sigma)$, где $\sigma = \{(x, 3); (y, 4); (z, 2)\}$: ⊛

$$x \leftarrow y;\ x \leftarrow s(x);\ y \leftarrow z;\ y \leftarrow s(y);\ z \leftarrow s(z);$$
$$y \leftarrow s(x);\ z \leftarrow y;\ z \leftarrow s(z);\ x \leftarrow s(x);$$

540. Для структурированной программы Π на рис. 34 на следующей странице (слева) найти $\Pi(\sigma)$, где $\sigma = \{(x,\ 0);\ (y,\ 3);\ (z,\ 5);\ (u,\ 4);\ (v,\ 2)\}$. ⊛

541. Для структурированной программы Π на рис. 34 на следующей странице (справа) найти $\Pi(\sigma)$, где $\sigma = \{(x,\ 2);\ (y,\ 3);\ (z,\ 7);\ (u,\ 5);\ (v,\ 0)\}$. ⊛

542. Написать структурированную программу Π_+, которая имеет входные переменные x и y и вычисляет функцию $x + y$ в переменной x, не изменяя y. ⊛

543. Пусть Π_+ — это программа, которая вычисляет функцию $x + y$ в переменной x, не изменяя y и используя дополнительные переменные z_0 и z_1. Какие из структурированных программ на рис. 35 на следующей странице вычисляют в переменной x произведение xy? ⊛

544. Пусть $\Pi_\times$ — это программа, которая вычисляет функцию xy в переменной x, используя вспомогательные переменные z_i, $i = 0, 1, \ldots, m$,

```
x ← y; x ← s(x); y ← u;
y ← s(y); v ← z; v ← s(v);
u ← (x < v); v ← (x < y);
if u then
    if v then
        z ← y; z ← s(z);
    else
        z ← x;
    end;
else
    z ← s(x);
end;
```

```
x ← y; x ← s(x); v ← u;
v ← s(v);
z ← (x < v);
while z do
    z ← (y < x);
    if z then
        y ← s(y);
    else
        x ← s(x); u ← s(u);
    end;
    z ← (x < v);
end;
```

Рис. 34: Программы из задач 540 и 541.

```
z3 ← x;
z2 ← 0; x ← 0;
z1 ← (z2 < z3);
while z1 do
    z1 ← 0; Π+
    z0 ← 0;
    z2 ← s(z2);
    z1 ← (z2 < z3);
end;
```

```
z3 ← y; y ← x;
x ← 0;
z0 ← (z2 < z3);
while z0 do
    z0 ← 0; z1 ← 0;
    Π+ z2 ← s(z2);
    z0 ← (z2 < z3);
end;
```

```
z1 ← y;
if y then
    while z1 do
        z0 ← 0;
        z1 ← 0; Π+
        z2 ← s(z2);
        z1 ← (z2 < y);
    end;
else
    x ← 0;
end;
```

Рис. 35: Программы из задачи 543.

```
u ← x;  x ← 0;
while x do
    Π_z Π_×
    y1 ← y;
    y ← s(y);
    x ← (x < u);
end;
u ← (u < x);
if u then
    x ← y1;
else
    x ← y;
end;
```

```
u ← x;
while x do
    y1 ← y;
    y ← s(y);
    x ← y;
    Π_z Π_×
    x ← (x < u);
end;
u ← (u < x);
x ← y;
if u then
    x ← y1;
end;
```

```
u ← x;  u ← s(u);
while x do
    y1 ← y;
    y ← s(y);
    x ← y;
    Π_z Π_×
    x ← (x < u);
end;
x ← y1;
```

Рис. 36: Программы из задачи 544.

Π_z — программа, обнуляющая переменные z_i: $z_0 \leftarrow 0; \ldots z_m \leftarrow 0;$. Какие из структурированных программ на рис. 36 вычисляют в переменной x квадратный корень из x, то есть функцию $\mathrm{sqrt}(x) = \lfloor\sqrt{x}\rfloor$? ⊛

545. Пусть $\Pi_\times$ и Π_z — программы из задачи 544 на стр. 165. Какие из структурированных программ на рис. 37 на следующей странице вычисляют в переменной x целую часть частного: $\lfloor {}^x/_y \rfloor$? При $y = 0$ результат тоже должен быть равен нулю. ⊛

546. Построить программы, вычисляющие в переменной x следующие функции, и доказать их корректность. Для двухместных функций значение переменной y должно остаться неизменным.

(а) $\mathrm{mult}(x, y) = xy$;

(б) $\mathrm{dec}(x) = x \dot{-} 1$, где $0 \dot{-} 1 = 0$ и $(x+1) \dot{-} 1 = x$;

(в) $\mathrm{sub}(x, y) = x \dot{-} y$, где $x \dot{-} y = x - y$, если $x \geqslant y$, и $x \dot{-} y = 0$ иначе;

(г) $\mathrm{fact}(x) = x!$ — факториал;

(д) $\mathrm{div}(x, y) = \lfloor {}^x/_y \rfloor$ — целочисленное частное;

```
u ← s(x); x ← 0;
i ← (x < u);
while i do
    j1 ← j;
    j ← s(j);
    Π_z Π_×
    i ← (x < u);
end;
x ← j1;
```

```
u ← s(x); x ← 0;
i ← y;
while i do
    x ← j;
    Π_z Π_×
    i ← (x < u);
    if i then
        j1 ← j;
    end;
    j ← s(j);
    i ← (j < u);
end;
x ← j1;
```

```
u ← x; i ← y;
while i do
    j1 ← j;
    j ← s(j);
    x ← j;
    Π_z Π_×
    i = (x < u);
end;
i = (u < x);
x ← j;
if i then
    x ← j1;
end;
```

Рис. 37: Программы из задачи 545.

(е) $\operatorname{sqrt}(x) = \lfloor\sqrt{x}\rfloor$;

(ж) $\operatorname{pow}(x, y) = x^y$ — возведение в степень;

(з) $\log(y, x) = \lfloor\log_y x\rfloor$;

(и) $\operatorname{mod}(x, y)$ — остаток от деления x на y;

(к) $\tau(x)$ — количество различных делителей числа x, $\tau(0) = 0$. ⊛

547. Построить программу с метками, которая по натуральному числу i вычисляет число Фибоначчи F_i. ⊛

548. Построить программы, вычисляющие в переменной z следующие функции: ⊛

(а) $f_1(x, y) = \begin{cases} \lfloor (x+y)/2 \rfloor & \text{при } x < y, \\ x^2 y^3 & \text{в противном случае;} \end{cases}$

(б) $f_2(x, y) = \begin{cases} 3^y & \text{при } \log_2(x+1) \geqslant y, \\ |x - y| & \text{в противном случае;} \end{cases}$

(в) $f_3(x,y) = \begin{cases} \lfloor xy/2 \rfloor & \text{при } \log_2 x \leqslant y+2, \\ x^{\lfloor y/2 \rfloor} & \text{в противном случае;} \end{cases}$

(г) $f_4(x,y) = \begin{cases} \lfloor \log_3(x+y+1) \rfloor & \text{при } 2x \leqslant y^3+y, \\ \lfloor \sqrt{x+y} \rfloor & \text{в противном случае;} \end{cases}$

(д) $f_5(x,y) = \begin{cases} 1, & \text{если } x \text{ — простое число,} \\ 0 & \text{в противном случае.} \end{cases}$

549. Для каждой из заданных ниже рекурсивными соотношениями функций $A(x,y)$, $B(x,y)$, $C(x,y)$, $D(x,y), E(x,y), G(x,y)$ и $H(x,y)$ построить вычисляющую её структурированную программу:

(а) $A(0,y) = y+1,$
$A(x+1,0) = 2A(x,1),$
$A(x+1,y+1) = A(x,y) + xy;$

(б) $B(0,y) = 2y,$
$B(x+1,0) = B(x,1)+1,$
$B(x+1,y+1) = x + B(x+1,y);$

(в) $C(0,y) = y+2,$
$C(x+1,0) = C(x,1)+1,$
$C(x+1,y+1) = C(x+1,y) \dot{-} x;$

(г) $D(x,0) = x,\ D(0,y) = y,$
$D(x+1,y+1) = D(x,y+1) + x^y;$

(д) $E(0,y) = y+1,$
$E(x+1,0) = E(x,1),$
$E(x+1,y+1) = xE(x+1,y);$

(е) $G(0,y) = 3,$
$G(x+1,0) = G(x,0) + \lfloor \log_2(x+1) \rfloor,$
$G(x+1,y+1) = G(x+1,y) + x^2 y;$

(ж) $H(0,y) = y,$
$H(x+1,0) = H(x,1)+2,$
$H(x+1,1) = H(x,0)+1,$
$H(x+1,y+2) = y + H(x+1,y).$ ⊛

550. Написать программу, которая будет реализовывать присваивание $a \leftarrow (b < c);$, используя присваивания видов $x \leftarrow 0;$ $x \leftarrow y;$ $x \leftarrow s(y);$ и $x \leftarrow \mathrm{eq}(y, z);$. Последнее изменяет значение x на 1, если значения y и z равны, или на 0 в противном случае. ⊛

551. Доказать, что для вычисления любой вычислимой функции можно написать структурированную программу без ветвлений.
Указание. Сначала показать, что можно написать программу без полных ветвлений. ⊛

552. Пусть программа П с одной входной переменной x вычисляет в переменной y некоторую всюду определённую взаимно однозначную функцию f, область значений которой совпадает с множеством всех натуральных чисел ω. Пусть $\mathrm{Var}(\Pi) = \{x, y, z_1, \ldots, z_m\}$. Построить программу, которая вычисляет обратную к f функцию f^{-1}: $f^{-1}(y) = x$, если $f(x) = y$. ⊛

553. Пусть П — программа и $|\,\mathrm{Var}(\Pi)| = m$. Из определений следует, что при различном выборе входных переменных и выходных переменных программа может вычислять различные функции.

(а) Каково максимальное количество функций от $n \leqslant m$ переменных, которое может вычислять П? Сколько всего разных функций может вычислить П? Функции, имеющие различное количество аргументов, считаем разными a priori.

(б) Построить программу $\Pi_{m,n}$, которая вычисляет максимальное количество различных функций от $n \leqslant m$ переменных.

(в) Построить программу Π_m, $|\,\mathrm{Var}(\Pi_m)| = m$, которая для каждого $n \leqslant m$ вычисляет максимальное количество различных функций от n переменных. ⊛

554. Определить, сколько всего существует попарно неэквивалентных программ с метками, имеющих не более n операторов и только одну переменную, которая является одновременно входной и выходной. ⊛

555. Определить, какие всюду определённые функции могут вычисляться программами с метками, которые имеют только одну переменную, являющуюся одновременно входной и выходной, и используют только ветвления и присваивания видов $x \leftarrow s(x);$ и $x \leftarrow \mathrm{dec}(x);$. ⊛

556. Пусть программа с метками содержит только пропозициональные переменные $x_1, \ldots, x_n$, ветвления и присваивания с булевыми связками: $x_i \leftarrow x_j \wedge x_k$; $x_i \leftarrow x_j \vee x_k$; $x_i \leftarrow x_j \rightarrow x_k$; $x_i \leftarrow \neg x_j$;. Предложить способ, который позволяет определить, какую функцию вычисляет такая программа. ⊛

557. Доказать, что никакая из функций $x + y$, $x \dot{-} y$, xy и $\lfloor x/y \rfloor$ не вычисляется никакой структурированной программой без циклов. ⊛

23. Частично рекурсивные функции

Базисными рекурсивными функциями называются:

1) функция 0, которая не имеет аргументов. Значение этой функции равняется нулю;
2) одноместная функция s. Значение $s(x)$ равняется $x+1$ для всех x;
3) всевозможные проектирующие n-местные функции id_m^n, где $n, m \in \omega$, $0 < m \leqslant n$. Значение $\mathrm{id}_m^n(x_1, \ldots, x_n)$ равняется x_m.

Пусть даны m-местная функция g и n-местные функции $f_1, \ldots, f_m$. Функция h получена из $g, f_1, \ldots, f_m$ суперпозицией, если

$$h(x_1, \ldots, x_n) = g(f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)).$$

Суперпозиция обозначается так: $h = g(f_1, \ldots, f_m)$.

Допустим, что имеются $(n-1)$-местная функция f и $(n+1)$-местная функция g. Функция h получена из f и g примитивной рекурсией, если

1) $h(x_1, \ldots, x_{n-1}, 0) = f(x_1, \ldots, x_{n-1})$;
2) $h(x_1, \ldots, x_{n-1}, i{+}1) = g(x_1, \ldots, x_{n-1}, i, h(x_1, \ldots, x_{n-1}, i))$ для каждого $i \in \omega$.

Функция h обозначается в этом случае с помощью $\mathrm{Pr}[f, g]$.

Пусть $f(x_1, \ldots, x_{n-1}, x_n)$ — n-местная функция. Частичная $(n-1)$-местная функция h получена из f с помощью минимизации: $h = \mu f$, если $h(x_1, \ldots, x_{n-1}) = y$ в том и только том случае, когда

1) $f(x_1, \ldots, x_{n-1}, z)$ определено и не равно нулю при всех $z < y$;
2) $f(x_1, \ldots, x_{n-1}, y) = 0$.

Следовательно, если $f(x_1, \ldots, x_{n-1}, y) \neq 0$ для всех натуральных y, то $h(x_1, \ldots, x_{n-1})$ неопределено.

Если при определении значения какой-либо функции f требуется определить значение функции g и оно не определено, то значение функции f также неопределено.

Частичная функция, которая может быть получена из базисных рекурсивных функций при помощи операторов суперпозиции, примитивной рекурсии и минимизации, называется частично рекурсивной функцией (сокращённо: ч. р. ф.). Если частично рекурсивная функция является всюду определённой, то она называется общерекурсивной (сокращённо: о. р. ф.).

Например, нульместные константы $1^{(0)}, 2^{(0)}, \ldots$ легко определяются по индукции с помощью суперпозиции функции s и предыдущей константы:

$$1 = s(0); \quad 2 = s(1); \quad \ldots$$

Индукцией по местности m легко построить функции-константы произвольной местности:

$$n^{(m+1)} = \mathrm{Pr}[n^{(m)}, \mathrm{id}^{m+2}_{m+2}].$$

Характеристическая функция нуля $\mathrm{not}(x)$, превращающая нуль в единицу, а остальные числа — в нуль, тоже строится при помощи примитивной рекурсии:

$$\mathrm{not}^{(1)} = \mathrm{Pr}[1^{(0)}, 0^{(2)}].$$

Характеристическая функция множества положительных натуральных чисел $\mathrm{test}(x)$ легко получается суперпозицией not с собой же:

$$\mathrm{test} = \mathrm{not}(\mathrm{not}).$$

Двухместная функция сложения определяется как

$$+ = \mathrm{Pr}[\mathrm{id}^1_1, s(\mathrm{id}^3_3)],$$

а умножение — аналогичным образом через сложение

$$\times = \mathrm{Pr}[0^{(1)}, \mathrm{id}^3_3 + \mathrm{id}^1_3].$$

Функция $\mathrm{dec}(x)$, ограниченного декремента: $\mathrm{dec}(x+1) = x$, $\mathrm{dec}(0) = 0$, строится так:

$$\mathrm{dec} = \mathrm{Pr}\left[0^{(0)}, \mathrm{id}^2_1\right],$$

а функция ограниченного вычитания $x \dot{-} y$ — с помощью dec:

$$\dot{-} = \mathrm{Pr}\left[\,\mathrm{id}_1^1, \mathrm{dec}\left(\,\mathrm{id}_3^3\,\right)\right].$$

Чтобы построить характеристическую функцию для отношения «меньше» $\mathrm{less}(x, y)$, можно прибегнуть к помощи ограниченного вычитания:

$$\mathrm{less} = \mathrm{test}\left(\mathrm{id}_2^2 \dot{-} \mathrm{id}_1^2\right).$$

Далее легко получить другие сравнения: «меньше или равно» $\mathrm{leq}(x, y)$

$$\mathrm{leq} = \mathrm{not}\left(\,\mathrm{less}\left(\,\mathrm{id}_2^2, \mathrm{id}_1^2\,\right)\right)$$

и равенство $\mathrm{eq}(x, y)$

$$\mathrm{eq} = \mathrm{leq}\left(\,\mathrm{id}_1^2, \mathrm{id}_2^2\,\right) \cdot \mathrm{leq}\left(\,\mathrm{id}_2^2, \mathrm{id}_1^2\,\right).$$

Нигде не определённая функция $\delta^{(1)}$ может быть задана при помощи минимизации ненулевой двухместной функции-константы: $\delta = \mu 1^{(2)}$.

Если $(m+1)$-местная функция $f(\bar{x}, y)$ является ч. р. ф. (соответственно о. р. ф.), то и следующие функции являются ч. р. ф. (соответственно о. р. ф.):

$$g(\bar{x}, z) = \sum_{y<z} f(\bar{x}, y); \quad h(\bar{x}, z) = \prod_{y<z} f(\bar{x}, y).$$

Их определения выглядят так:

$$g = \mathrm{Pr}\left[0^{(m)}, \mathrm{id}_{m+2}^{m+2} + f\left(\,\mathrm{id}_1^{m+2}, \ldots, \mathrm{id}_m^{m+2}, \mathrm{id}_{m+1}^{m+2}\,\right)\right];$$
$$h = \mathrm{Pr}\left[1^{(m)}, \mathrm{id}_{m+2}^{m+2} \cdot f\left(\,\mathrm{id}_1^{m+2}, \ldots, \mathrm{id}_m^{m+2}, \mathrm{id}_{m+1}^{m+2}\,\right)\right].$$

Задачи

558. Пусть заданы три функции: $f(x, y, z) = xy + z$, $g(x, y) = 2x + y$, $h(x) = 2x^2$. Определить, какой будет функция $F^{(2)}$, задаваемая так: $f(h(\mathrm{id}_1^2), g(h(\mathrm{id}_2^2), \mathrm{id}_2^2), \mathrm{id}_2^2)$? ⊛

559. Определить, чему равно значение следующих о. р. ф.:

(а) $f = \mathrm{Pr}[0, s(\mathrm{id}_1^2 + \mathrm{id}_1^2) + \mathrm{id}_2^2]$;

(б) $f = \mathrm{Pr}[0, s(\mathrm{id}_1^2) + s(\mathrm{id}_1^2) + \mathrm{id}_2^2]$;

(в) $f = \mathrm{Pr}[0, s(3 \cdot \mathrm{id}_1^2 \cdot s(\mathrm{id}_1^2)) + \mathrm{id}_2^2]$. ⊛

560. О. р. ф. $f(x)$ задана следующим образом:

$$\mathrm{Pr}[0, s(\mathrm{id}_1^2) + s(\mathrm{id}_1^2) + \mathrm{id}_1^2 + \mathrm{id}_1^2 + \mathrm{id}_2^2 + \mathrm{id}_2^2].$$

Доказать, что $f(x) = 3 \cdot 2^{x+1} - 4x - 6$. ⊛

561. Найти значение $F(5)$ для следующих функций:

(а) $F = \mathrm{Pr}[1, h]$, где $h(y, z) = \left\lfloor 2^z / z^2 \right\rfloor$;

(б) $F = \mathrm{Pr}[1, h]$, где $h(y, z) = \left\lfloor 2^z / z \right\rfloor$;

(в) $F = \mathrm{Pr}[2, h]$, где $h(y, z) = 3z \dot{-} (2y + 1)$. ⊛

562. Показать, что следующие функции являются о. р. ф.:

(а) $\mathrm{pow}(x, y) = x^y$ — возведение в степень;

(б) $\mathrm{fact}(x) = x!$ — факториал;

(в) $\mathrm{if}(x, y, z)$ — условная операция,

$$\mathrm{if}(x, y, z) = \begin{cases} y, & \text{если } x \neq 0, \\ z, & \text{иначе;} \end{cases}$$

(г) $\min^{(n)}(x_1, \ldots, x_n)$ — наименьший из аргументов, $n > 0$;

(д) $\max^{(n)}(x_1, \ldots, x_n)$ — наибольший из аргументов, $n > 0$;

(е) $\mathrm{diff} = |x - y|$ — модуль разности;

(ж) $\mathrm{mod}(x, y)$ — остаток от деления x на y;

(з) $\mathrm{div}(x, y) = \left\lfloor x/y \right\rfloor$ — целочисленное частное. ⊛

563. Доказать, что если функция $f(x_1, \ldots, x_n)$ является ч. р. ф., то и функция $g(x_1, \ldots, x_n) = f(x_{i_1}, \ldots, x_{i_n})$ является ч. р. ф. для каждой перестановки $(i_1, \ldots, i_n)$ чисел $1, 2, \ldots, n$. ⊛

564. Говорят, что функция $h^{(n+1)}$ получена *ограниченной минимизацией* функции $f^{(n+1)}$ (это обозначается $h = \mu_{\lim} f$), если $h(\bar{x}, y) = y$, когда $f(\bar{x}, y) \neq 0$ для всех $z < y$, иначе $h(\bar{x}, y)$ равняется наименьшему $z < y$, для которого $f(\bar{x}, z) = 0$. Доказать, что функцию $\mu_{\lim} f$ можно построить из f и базисных функций при помощи суперпозиции и примитивной рекурсии. ⊛

565. Пусть $g(x_1, \ldots, x_{n-1}, x_n)$ — ч. р. ф., a и $b > 0$ — натуральные числа. Доказать, что функция

$$f(x_1, \ldots, x_{n-1}, x_n) = \begin{cases} a, & \text{если } x_n < b \\ g(x_1, \ldots, x_{n-1}, x_n - b), & \text{если } x_n \geqslant b \end{cases}$$

тоже является ч. р. ф. ⊛

566. Найти значение $F(9)$ и $F(17)$ для $F = \mu f$ и следующих функций $f^{(2)}$. Определить, для какой из них будет выполнено $F(x) = \lfloor \sqrt{x} \rfloor$:

(а) $f(x,y) = y^2 \dot{-} x$;

(б) $f(x,y) = x \dot{-} y^2$;

(в) $f(x,y) = (x+1) \dot{-} y^2$;

(г) $f(x,y) = (x+1) \dot{-} (y+1)^2$;

(д) $f(x,y) = x \dot{-} (y+1)^2$. ⊛

567. Определить, для какой из следующих функций $f^{(3)}$ будет выполнено $F(x) = \lfloor y/x \rfloor$, где $F = \mu f$ и $x \neq 0$:

(а) $f(x,y,i) = y \dot{-} ix$;

(б) $f(x,y,i) = y \dot{-} (i+1)x$;

(в) $f(x,y,i) = (y+1) \dot{-} ix$;

(г) $f(x,y,i) = (y+1) \dot{-} i(x+1)$;

(д) $f(x,y,i) = (y+1) \dot{-} (i+1)x$. ⊛

568. Какие из следующих выражений определяют количество $\tau(x)$ различных делителей числа x?

(а) $x - \sum\limits_{i<x} \mathrm{test}(\mathrm{mod}(x, s(i)))$;

(б) $x - \sum\limits_{i<x} \mathrm{not}(\mathrm{mod}(x, s(i)))$;

(в) $\sum\limits_{i<x} \mathrm{test}(\mathrm{mod}(x, s(i)))$;

(г) $\sum\limits_{i<x} \mathrm{not}(\mathrm{mod}(x, s(i)))$. ⊛

569. Показать, что следующие функции являются ч. р. ф.:

(а) $\mathrm{rt}(n,x) = \lfloor \sqrt[n]{x} \rfloor$ — целая часть корня n-й степени из x;

(б) $\log(i,x) = \lfloor \log_i x \rfloor$;

(в) $p(x) = 1$, если x — простое число, и $p(x) = 0$ в противном случае;

(г) $\mathrm{pn}(k)$ — k-е простое число в порядке возрастания, $\mathrm{pn}(0) = 2$;

(д) $\sigma(x)$ — сумма делителей числа x, считать, что $\sigma(0) = 0$;

(е) $\mathrm{digit}(n,m,i)$ — i-я цифра в m-ичном представлении числа n: то есть если $n = \sum\limits_{i=0}^{\infty} a_i m^i$, где $0 \leqslant a_i < m$, то $\mathrm{digit}(n,m,i) = a_i$;

(ж) НОД(x, y) — наибольший общий делитель чисел x и y. ⊛

570. Определить ч. р. ф. $l^{(1)}$, значение которой равно $\log_2 x$, когда x является степенью двойки, и неопределено в противном случае. ⊛

571. На евклидовой координатной плоскости дан круг C с центром в точке $(100, 100)$ радиусом 50. Доказать, что следующая функция $r^{(2)}$ является частично рекурсивной: $r(x, y)$ определено тогда и только тогда, когда точка (x, y) находится вне круга C, при этом её значение равно целой части расстояния от (x, y) до круга (то есть до ближайшей к (x, y) точки круга C). ⊛

572. О. р. ф., которая может быть построена без использования минимизации, называется п р и м и т и в н о р е к у р с и в н о й. Пусть функции g и f примитивно рекурсивны, $h = \mu g$, h — о. р. ф., и h мажорируется функцией f: $h(\bar{x}) < f(\bar{x})$ для всех $\bar{x}$. Доказать, что h тоже является примитивно рекурсивной. ⊛

573. Доказать, что если значения о. р. ф. $f(x)$ изменить на конечном множестве, то получившаяся функция $f'(x)$ также будет о. р. ф. ⊛

574. Доказать, что из константы 0 и функций id^n_m с помощью суперпозиции и примитивной рекурсии нельзя получить функцию $s(x) = x+1$ и функцию $d(x) = 2x$.

Указание. Индукцией по построению функции доказать, что для всех таким образом построенных функций выполнено неравенство $f(x_1, \ldots, x_n) < 2$ для произвольных $x_1, \ldots, x_n < 2$. ⊛

575. Пусть f — взаимно однозначная на ω о. р. ф. Доказать, что обратная функция f^{-1} тоже является о. р. ф., явно её построив. ⊛

576. Пусть $g(x_1, \ldots, x_n, y)$ — о. р. ф. Доказать, что функция

$$f(x_1, \ldots, x_n, y, z) = \begin{cases} \sum\limits_{i=0}^{z} g(x_1, \ldots, x_n, y+i), & \text{при } y \leqslant z, \\ 0, & \text{при } y > z \end{cases}$$

общерекурсивна. ⊛

577. Доказать, что если функции $f(x_1, \ldots, x_n, y), g(x_1, \ldots, x_n, y)$ и $h(x_1, \ldots, x_n, y)$ общерекурсивны, то функция

$$F(x_1,\dots,x_n)=\min\{y : f(x_1,\dots,x_n,y)=0 \text{ или } \\ g(x_1,\dots,x_n,y)>h(x_1,\dots,x_n,y)\}$$

является ч. р. ф. ⊛

578. Допустим, что все пары (x, y) натуральных чисел упорядочены по возрастанию суммы $x+y$, а пары с одинаковой суммой — по возрастанию координаты x. Этот порядок выглядит так:

$$(0,0),(0,1),(1,0),(0,2),(1,1),(2,0),\dots \\ \dots,(0,x+y),(1,x+y-1),\dots,(x,y),\dots,(x+y,0),\dots$$

Пусть $\langle x,y\rangle$ — это номер пары (x, y) в этом порядке (будем считать, что пара $(0,0)$ имеет номер 0). Тогда функция $\langle,\rangle$ взаимно однозначно нумерует все пары натуральных чисел.

(а) Доказать, что $\langle x,y\rangle=\dfrac{(x+y)(x+y+1)}{2}+x$.

(б) Найти обратные функции $\langle\rangle_1$ и $\langle\rangle_2$ такие, что будут выполнены равенства $\langle\langle x,y\rangle\rangle_1=x$, $\langle\langle x,y\rangle\rangle_2=y$ и, следовательно, $\langle\langle z\rangle_1,\langle z\rangle_2\rangle=z$.

(в) Показать, что все эти функции общерекурсивны. ⊛

579. Показать, что функция $F(x)$ из задачи 547 на стр. 168 является о. р. ф.
Указание. Показать сначала, что функция $g(x)=\langle F(x),F(x+1)\rangle$ является о. р. ф. ⊛

580. Другая функция, позволяющая нумеровать пары натуральных чисел, выглядит так: $c(x,y)=2^x(2y+1)-1$. Доказать, что сама c и обратные к ней функции c_1 и c_2 (то есть $c_1(c(x,y))=x$ и $c_2(c(x,y))=y$) являются общерекурсивными. ⊛

581. Если задана взаимно однозначная нумерация пар k (например, из задач 578 или 580), то можно индуктивно пронумеровать упорядоченные n-ки произвольной длины:

$$K()=0,$$
$$K(x_1,\dots,x_n,x_{n+1})=k(k(k_1(K(x_1,\dots,x_n)),x_{n+1}),n+1).$$

(а) Доказать, что функция K взаимно однозначна.

(б) Доказать, что функции $K^{(n)}(x_1, \ldots, x_n)$ общерекурсивны.

(в) Доказать, что обратная к K функция K' тоже является общерекурсивной: ⊛

$$K'(K(x_0, \ldots, x_n), m) = \begin{cases} x_m, & \text{при } m \leqslant n, \\ 0, & \text{иначе.} \end{cases}$$

582. Ещё один способ кодирования конечных последовательностей натуральных чисел использует двоичную запись: последовательность $(n_0, n_1, \ldots, n_k)$ кодируется двоичным числом $1^{n_k}01^{n_{k-1}}0\ldots01^{n_1}01^{n_0}$ (последовательности, которые получаются добавлением или удалением конечных нулей, не различаются).

(а) Найти коды последовательностей $(2, 3, 0, 1)$ и $(0, 0, 2, 1, 2)$.

(б) Определить, кодами каких последовательностей являются числа 169 и 19783.

(в) Доказать, что функция $k(x, y)$ является общерекурсивной. Здесь x — код последовательности, y — номер элемента, значение функции равно этому элементу. ⊛

583. Функция Аккермана задана следующим образом:

$$A(0, y) = y + 1,$$
$$A(x + 1, 0) = A(x, 1),$$
$$A(x + 1, y + 1) = A(x, A(x + 1, y)).$$

(а) Найти $A(1, y)$ и $A(2, y)$ для всех натуральных y.

(б) Доказать, что $A(3, y) = 2^{y+3} - 3$.

(в) Доказать, что функция Аккермана является общерекурсивной.

Указание. С помощью функций K и K' (задача 581 на противоположной странице) строить последовательности $(x_1, x_2, \ldots, x_n, z)$ такие, что ⊛

$$A(x, y) = A(x_1, A(x_2, \ldots, A(x_n, z) \ldots)).$$

24. Машины Тьюринга

Машина Тьюринга $\mathfrak{M}$ — это четвёрка вида (Q, Σ, P, q_0), в которой:

1) $Q = \{q_0, \ldots, q_{n-1}\}$, $n \geqslant 1$, — конечное множество состояний;
2) $\Sigma = \{a_1, \ldots, a_m\}$, $m \geqslant 1$, — конечный алфавит, при этом одним из символов Σ обязательно является Λ;
3) P — программа машины Тьюринга;
4) $q_0 \in Q$ — начальное состояние.

Символ Λ называется *пустым* (или *пробельным*).

Программа машины Тьюринга — это множество *команд*, то есть строк вида «$q, a \to p, b, s$», где $q, p \in Q$ — состояния, $a, b \in \Sigma$ — символы алфавита, $s \in \{+1, -1, 0\}$ — направление сдвига. При этом для всех $q \in Q$ и $a \in \Sigma$ в P должно быть не более одной команды описанного вида.

Как и для конечных автоматов, программу P можно задавать с помощью таблицы размера $n \times m$, строки которой соответствуют состояниям из Q, а столбцы — символам из входного алфавита Σ, в которой на пересечении строки q и столбца a стоит тройка p, b, s — правая часть команды «$q, a \to p, b, s$».

Конфигурацией машины Тьюринга $\mathfrak{M}$ называется тройка (q, i, α), где $q \in Q$ — *текущее состояние*, $i \in \mathbb{Z}$ — целое число, *положение головки*, $\alpha : \mathbb{Z} \to \Sigma$ — функция, называемая *лентой*.

В дальнейшем мы всегда считаем, что «слово» — это конечная последовательность символов, *не содержащая пустого*. Пустой символ мы используем как *разделитель* между словами. Поэтому, например, в последовательности символов $\Lambda aa\Lambda ba\Lambda\Lambda abba$ записаны три слова: aa, ba и $abba$, из которых первые два разделены одним пустым символом, а следующие — двумя.

Если лента α содержит только пустые символы: $\alpha(i) = \Lambda$ для всех $i \in \mathbb{Z}$, то говорим, что *на ленте записано* пустое слово. Иначе *записанное на ленте* слово — это максимальная последовательность непустых символов вида $\alpha(k)\alpha(k+1)\ldots\alpha(\ell-1)\alpha(\ell)$. Максимальность означает, что эту последовательность нельзя продолжить ни в одну сторону, то есть она окружена пустыми символами. Из определения непосредственно следует, что на ленте может быть записано несколько слов.

Начальная конфигурация машины $\mathfrak{M}$ с входными словами $w_1,\ldots,w_n$ — это конфигурация вида $(q_0,0,\alpha)$, где на ленте α, начиная с нулевой ячейки, записаны слова $w_1,\ldots,w_n$, разделённые одним пустым символом. Следовательно, в начальной конфигурации конец входа легко опознать по двум подряд идущим пустым символам.

Конфигурация вида (q,i,α) называется заключительной для $\mathfrak{M}$, если в программе P нет команды вида $q,\alpha(i)\to\ldots$

Конфигурация $\sigma=(q,i,\alpha)$ переходит за один шаг в конфигурацию $\tau=(p,j,\beta)$, если

1) в программе P есть команда $q,a\to p,b,s$, где $a=\alpha(i)$;
2) $j=i+s$;
3) $\beta(i)=b$, $\beta(k)=\alpha(k)$ при $k\neq i$.

Записываем это как обычно: $\sigma\vdash_{\mathfrak{M}}\tau$.

Если при выполнении команды некоторый непустой символ заменяется на пустой, то есть команда имеет вид $q,a\to p,\Lambda,s$, то будем говорить, что символ a стирается.

Как для конечных автоматов определяется переход за k шагов $\vdash^k_{\mathfrak{M}}$, за произвольное количество шагов $\vdash^*_{\mathfrak{M}}$ и понятие вычисления.

Если вычисление $\mathfrak{M}$ на входных словах $w_1,w_2,\ldots,w_n$ завершается заключительной конфигурацией τ, причём в конфигурации τ на ленте написано слово v, то говорим, что v является результатом вычисления $\mathfrak{M}$ на входе $w_1,\ldots,w_n$.

В том случае, когда в заключительной конфигурации на ленте записано несколько слов $v_1,\ldots,v_k$, разделённых одним или несколькими пустыми символами, считаем, что результатом является набор $(v_1,\ldots,v_k)$.

В силу детерминированности у машины Тьюринга может быть не более одного результата вычисления. Результат может быть неопределён, если в вычислении заключительная конфигурация никогда не образуется. Тогда говорят, что вычисление зацикливается.

Скажем, что машина Тьюринга $\mathfrak{M}$ вычисляет частичную словарную функцию $f:\Omega_1^*\to\Omega_2^*$ (с неопределённым количеством аргументов), если $\Omega_1,\Omega_2\subseteq\Sigma$ и для всех слов $w_1,\ldots,w_n\in\Omega_1^*$ выполнено одно из двух:

1) $f(w_1,\ldots,w_n)$ определено и равно v, результатом вычисления $\mathfrak{M}$ на $w_1,\ldots,w_n$ тоже является v (v может быть одним словом или их набором);

2) $f(w_1, \ldots, w_n)$ неопределено и результат вычисления $\mathfrak{M}$ на $w_1, \ldots, w_n$ неопределён.

Если словарная функция f вычисляется некоторой машиной Тьюринга, то она называется *вычислимой по Тьюрингу*.

Если машины Тьюринга $\mathfrak{M}$ и $\mathfrak{N}$ вычисляют одну и ту же функцию $f : \Omega_1^* \to \Omega_2^*$, то они *$\Omega_1$-эквивалентны*.

Назовём *стандартной* заключительную конфигурацию вида $(q, 0, \alpha)$, если слова на ленте α записаны, начиная с нулевой ячейки, и разделены в точности одним пустым символом. Говорим, что машина Тьюринга $\mathfrak{M}$ имеет *стандартную заключительную конфигурацию*, если для всех начальных конфигураций σ, на которых машина $\mathfrak{M}$ останавливается, заключительная конфигурация τ, в которую σ перейдёт, будет стандартной. Таким образом, если машина $\mathfrak{M}$ остановится, то обязательно в стандартной заключительной конфигурации. Любая машина Тьюринга эквивалентна машине со стандартной заключительной конфигурацией.

Машина Тьюринга $\mathfrak{M} = (Q, \Sigma, P, q_0)$ называется *односторонней* в случае, если

1) алфавит ленты Σ содержит символ #;

2) в программе P нет команд ни одного из трёх следующих видов: «$q, \# \to p, \#, -1$»; «$q, \# \to p, a, s$»; «$q, a \to p, \#, s$» ни для каких $q, p \in Q$, $a \in \Sigma \setminus \{\#\}$, $s \in \{+1, -1, 0\}$.

Таким образом, односторонняя машина не может сдвинуть головку левее #, не может стереть # и не может записать его в ячейку, где его не было.

При рассмотрении односторонних машин Тьюринга считаем, что слова на ленте не могут содержать символ #. Таким образом, они должны быть ограничены с обеих сторон символами Λ или #. В частности, в начальной конфигурации и в стандартной заключительной конфигурации слова должны быть записаны сразу после #.

Любая машина Тьюринга эквивалентна односторонней.

Унарная система счисления — это способ записи числа n соответствующим количеством палочек или других одинаковых символов. Считаем, что в унарной системе натуральное число n представляется в виде слова из $n+1$ палочки, например, число 5 кодируется словом «||||||». Таким образом, унарная запись любого натурального числа является непустым словом.

Пусть зафиксирована некоторая система счисления π, унарная или k-ичная, где $k \geqslant 2$ (двоичная, десятичная и т. д.). Машина Тьюринга $\mathfrak{M}$ *вычисляет* частичную арифметическую функцию $f : \omega^n \to \omega$, если для всех наборов значений аргументов $a_1, \ldots, a_n \in \omega$ выполнено одно из двух:

1) $f(a_1, \ldots, a_n)$ определено и равно b, и $\mathfrak{M}$ останавливается на входе $(\pi(a_1), \ldots, \pi(a_n))$, а выходным словом при этом является $\pi(b)$;

2) $f(a_1, \ldots, a_n)$ неопределено и $\mathfrak{M}$ на входе $(\pi(a_1), \ldots, \pi(a_n))$ не останавливается.

Арифметическая (частичная) функция $f(x_1, \ldots, x_n)$ *вычислима по Тьюрингу*, если существует машина Тьюринга, которая эту функцию вычисляет.

Пусть Σ — произвольный алфавит машин Тьюринга. Тогда существует Σ-*универсальная* машина Тьюринга $\mathfrak{U} = (Q^{\mathfrak{U}}, \Sigma, P^{\mathfrak{U}}, q_0^{\mathfrak{U}})$, то есть такая, что для каждой машины Тьюринга $\mathfrak{M} = (Q, \Sigma, P, q_0)$ можно найти слово $c_{\mathfrak{M}} \in \Sigma^*$, и при этом для всех слов $w_1, \ldots, w_n \in \Sigma^*$ выполнено $\mathfrak{U}(c_{\mathfrak{M}}, w_1, \ldots, w_n) = \mathfrak{M}(w_1, \ldots, w_n)$.

Задачи

584. Дана машина Тьюринга $\mathfrak{M} = (\{q, p, r\}, \{\Lambda, 0, 1\}, P, q)$ со следующей программой $P = \{q, 0 \to q, 0, +1;\ q, 1 \to q, 1, +1;\ q, \Lambda \to p, \Lambda, -1;\ p, 0 \to p, 1, -1;\ p, 1 \to r, 0, -1;\ r, 0 \to r, 0, -1;\ r, 1 \to r, 1, -1\}$. Какой будет заключительная конфигурация этой машины при работе на входе 1100? Сколько шагов при этом будет сделано? ⊛

585. Имеются три машины Тьюринга $\mathfrak{M}_i = (Q, \Sigma, P_i, q)$, $i = 1, 2, 3$, которые имеют общие алфавит ленты $\Sigma = \{\Lambda, a, b\}$, множество состояний $Q = \{q, p, r, s, t\}$ и начальное состояние q. Их программы P_i представлены в таблицах на рис. 38 на следующей странице. Определить, какие из этих машин переводят любое входное слово вида $a^{2n}b$, $n > 0$, в выходное ba^n. ⊛

586. Пусть Σ — произвольный алфавит, не содержащий Λ. Требуется построить машину Тьюринга $\mathfrak{C}$, которая меняет местами два аргумента, точнее переводит любой вход вида $x\Lambda y$ (x и y — слова в алфавите Σ) в выход $y\Lambda x$ со стандартной заключительной конфигурацией.

Определить, какие из следующих программ можно использовать для машины $\mathfrak{C}$. В текстах программ q_0 — начальное состояние, α, β — это произвольные символы из Σ, γ, δ — это произвольные символы из $\Sigma \cup \{u, v\}$, где u, v — новые символы.

$Q \backslash \Sigma$	P_1			P_2		
	a	b	Λ	a	b	Λ
q	$p, \Lambda, +1$			$p, \Lambda, +1$		
p	$p, a, +1$	$r, b, +1$		$p, a, +1$	$r, b, +1$	
r	$r, a, +1$	$t, \Lambda, 0$	$s, a, -1$	$r, a, +1$	$q, \Lambda, 0$	$s, a, -1$
s	$s, a, -1$	$s, b, -1$	$t, \Lambda, +1$	$s, a, -1$	$t, b, -1$	$q, \Lambda, +1$
t	$q, \Lambda, +1$	$t, b, -1$		$t, a, +1$	$q, \Lambda, 0$	$q, \Lambda, +1$

$Q \backslash \Sigma$	P_3		
	a	b	Λ
q	$p, \Lambda, +1$		
p	$r, \Lambda, +1$	$r, b, +1$	
r	$r, a, +1$	$t, b, +1$	
s	$s, a, -1$	$s, b, -1$	$q, \Lambda, +1$
t	$t, a, +1$	$q, \Lambda, 0$	$s, a, -1$

Рис. 38: Программы машин из задачи 585.

(а) $P_1 = \{q_0,\ \alpha \to q_1^\alpha,\ \Lambda,\ +1;\ q_1^\alpha,\ \beta \to q_1^\alpha,\ \beta,\ +1;\ q_1^\alpha,\ \Lambda \to q_2,\ \alpha,\ +1;$ $q_2,\ \alpha \to q_3^\alpha,\ \Lambda,\ -1;\ q_2,\ \Lambda \to q_4,\ \Lambda,\ -1;\ q_3^\alpha,\ \beta \to q_3^\alpha,\ \beta,\ -1;$ $q_3^\alpha,\ \Lambda \to q_0,\ \alpha,\ +1;\ q_4,\ \alpha \to q_4,\ \alpha,\ -1;\ q_4,\ \Lambda \to q_5,\ \Lambda,\ -1;$ $q_5,\ \alpha \to q_5,\ \alpha,\ -1;\ q_5,\ \Lambda \to q_6,\ \Lambda,\ +1\};$

(б) $P_2 = \{q_0,\ \alpha \to q_0,\ \alpha,\ +1;\ q_0,\ \Lambda \to q_1,\ u,\ +1;\ q_1,\ \alpha \to q_1,\ \alpha,\ +1;$ $q_1,\ \Lambda \to q_2,\ v,\ 0;\ q_2,\ \gamma \to q_3^\gamma,\ \Lambda,\ -1;\ q_3^\gamma,\ \delta \to q_3^\delta,\ \gamma,\ -1;$ $q_3^\alpha,\ \Lambda \to q_9^\alpha,\ \Lambda,\ +1;\ q_3^u,\ \Lambda \to q_6,\ \Lambda,\ +1;\ q_6,\ \alpha \to q_6,\ \alpha,\ +1;$ $q_6,\ v \to q_7,\ \Lambda,\ -1;\ q_7,\ \alpha \to q_7,\ \alpha,\ -1;\ q_7,\ \Lambda \to q_8,\ \Lambda,\ +1;$ $q_9^\alpha,\ \delta \to q_9^\alpha,\ \delta,\ +1;\ q_9^\alpha,\ \Lambda \to q_2,\ \alpha,\ 0\};$

(в) $P_3 = \{q_0,\ \alpha \to q_0,\ \alpha,\ +1;\ q_0,\ \Lambda \to q_1,\ u,\ -1;\ q_1,\ \alpha \to q_1,\ \alpha,\ -1;$ $q_1,\ \Lambda \to q_2,\ \Lambda,\ +1;\ q_2,\ \alpha \to q_3^\alpha,\ v,\ +1;\ q_3^\gamma,\ \delta \to q_3^\delta,\ \gamma,\ +1;$ $q_3^\gamma,\ \Lambda \to q_6,\ \gamma,\ 0;\ q_6,\ \alpha \to q_7^\alpha,\ \Lambda,\ -1;\ q_6,\ u \to q_8,\ \Lambda,\ -1;$ $q_7^\alpha,\ \delta \to q_7^\alpha,\ \delta,\ -1;\ q_7^\alpha,\ \Lambda \to q_9^\alpha,\ \Lambda,\ +1;\ q_8,\ \alpha \to q_8,\ \alpha,\ -1;$ $q_8,\ u \to q_8,\ \Lambda,\ -1;\ q_8,\ v \to q_8,\ \Lambda,\ -1;\ q_8,\ \Lambda \to q_4,\ \Lambda,\ +1;$ $q_9^\alpha,\ \delta \to q_3^\delta,\ \alpha,\ +1\}.$ ⊛

587. Построить машину Тьюринга, которая по каждому входному слову вида $a^n b^k$ выдаёт результат $a^{\lfloor (n+k)/2 \rfloor} b^{\lfloor (n+k)/2 \rfloor}$. Как работает машина на других входах — несущественно. Оценить время работы построенной машины. ⊛

588. Построить машину Тьюринга, которая по каждому входному слову вида $0^n 1^k$, где $n > k > 0$, выдаёт результат 0, а для всех остальных слов — результат 1. Оценить время работы построенной машины. ⊛

589. Построить машину Тьюринга, которая переводит любое входное слово $w \in \{a,b\}^*$ в выходное $a^n \Lambda b^m$, где n — количество букв a, а m — количество букв b в слове w. Другими словами, выполняется сортировка букв слова с последующим их разделением. Оценить время работы построенной машины. ⊛

590. Пусть алфавит Σ не содержит символов Λ и $*$. Построить программу машины Тьюринга, которая выполняла бы перенос последнего слова на ленте в алфавите Σ слева направо в место ленты, отмеченное $*$. Точнее, из ленты $\alpha \Lambda w \Lambda^n *$ должно быть получено $\alpha \Lambda \Lambda^n w$, где $w \in \Sigma^*$. ⊛

591. Построить машину Тьюринга, выполняющую следующую задачу: по входу, состоящему из одного или нескольких слов $w_1 \Lambda \ldots \Lambda w_n$ в алфавите Σ (возможно, что $n = 1$), построить выход, удвоив последнее из слов: $w_1 \Lambda \ldots \Lambda w_n \Lambda w_n$. ⊛

592. Доказать, что для любой константы $k > 1$ можно построить машину Тьюринга, которая решает задачу 591 за время не большее $n^2/k + O(n)$, где n — это суммарная длина входных слов. ⊛

593. Построить машину Тьюринга, сравнивающую два входных слова в алфавите $\{a,b,c\}$ лексикографически. Эта машина Тьюринга должна вычислять словарную функцию: ⊛

$$f(x,y) = \begin{cases} a, & \text{если } x < y, \\ b, & \text{если } x = y, \\ c, & \text{если } y < x. \end{cases}$$

594. Построить программы машин Тьюринга, вычисляющих следующие словарные функции в алфавите $\Sigma = \{a,b,c\}$:

(а) циклическая перестановка букв: $f(xw) = wx$, $x \in \Sigma$;

(б) циклическая перестановка слов:

$$f(w_1, \ldots, w_n) = (w_2, \ldots, w_n, w_1);$$

(в) «переворачивание» последовательности слов:

$$f(w_1, \ldots, w_n) = (w_n, \ldots, w_1);$$

(г) удвоение каждой буквы: $f(x_1 \ldots x_n) = x_1 x_1 \ldots x_n x_n$;

(д) нахождение образа при гомоморфизме φ: $\varphi(a) = a$, $\varphi(b) = cb$, $\varphi(c) = \varepsilon$;

(е) удаление всех одиночных букв a, стоящих на чётных позициях;

(ж) проверка, содержат ли два слова одно и то же количество букв a (результат равен 1, если ответ «да», 0, если «нет»);

(з) проверка, является ли слово палиндромом (то есть симметричным);

(и) проверка, есть ли в слове две последовательности букв a одной и той же длины;

(к) проверка, образуют ли в слове количества букв a, b и c возрастающую арифметическую прогрессию. ⊛

595. Построить машину Тьюринга для перевода записи числа из двоичной системы в унарную, входной алфавит $\{0, 1\}$. ⊛

596. Построить машину Тьюринга, которая по унарной записи трёх чисел, то есть входу вида $|^{n+1}\Lambda|^{k+1}\Lambda|^{m+1}$, выдаёт результат 0, если $0 < n < k + m$, и 1 в противном случае. Оценить время работы построенной машины. ⊛

597. Построить программы односторонних машин Тьюринга, вычисляющих следующие арифметические функции в унарной системе:

(а) $x + y$;

(б) $x \dot{-} y$;

(в) xy;

(г) сравнение $x < y$;

(д) возведение в степень: x^y;

(е) квадратный корень: $\lfloor\sqrt{x}\rfloor$;

(ж) логарифм: $\lfloor\log_2 x\rfloor$;

(з) деление нацело: $\lfloor {}^{x}/_{y} \rfloor$;

(и) остаток: $x \bmod y$;

(к) функция выбора m-го аргумента: id_m, m — заранее заданная константа. ⊛

598. Даны машины Тьюринга со стандартной заключительной конфигурацией $\mathfrak{M}_i$, каждая из которых вычисляет словарную функцию $f_i(x)$ за время $t_i(x)$, $i = 1, \ldots, n$. Указать, как построить, и оценить время работы

(а) машины $\mathfrak{M}_1; \ldots; \mathfrak{M}_n$, вычисляющей функцию

$$g_1(x) = f_n(\ldots f_1(x) \ldots);$$

(б) машины $\mathrm{par}(a, \mathfrak{M}_1, \ldots, \mathfrak{M}_n)$, вычисляющей функцию

$$g_2(x_1 a x_2 a \ldots a x_n) = f_1(x_1) a f_2(x_2) a \ldots a f_n(x_n);$$

(в) машины $\mathrm{if}(\mathfrak{M}_1, \mathfrak{M}_2, \mathfrak{M}_3)$, вычисляющей функцию

$$g_3(x) = \begin{cases} \mathfrak{M}_2(x), & \text{если } \mathfrak{M}_1(x) \text{ непусто,} \\ \mathfrak{M}_3(x), & \text{если } \mathfrak{M}_1(x) \text{ пусто;} \end{cases}$$

(г) машины $\mathrm{while}(\mathfrak{M}_1, \mathfrak{M}_2)$, вычисляющей функцию $g_4(x) = x_q$, где q — наименьшее натуральное число, для которого $\mathfrak{M}_1(x_q)$ пусто, при этом $x_0 = x$ и $x_{j+1} = \mathfrak{M}_2(x_j)$. Считать, что $|x_j| \leqslant L(x)$ и $t_i(x_j) \leqslant T_i(x)$. ⊛

599. Даны следующие машины Тьюринга со стандартной заключительной конфигурацией:

- $\mathrm{copy}(a)$ — копирует вход, используя символ a как разделитель между оригиналом и копией: $w \mapsto waw$;
- $\mathrm{replace}(a, b)$ — заменяет самое левое вхождение символа a на b, если символа a нет, то вход не меняется;

- add — складывает два числа, записанных в унарной системе счисления;
- mult — умножает два числа, записанных в унарной системе счисления;
- nop — не изменяет вход.

Определить, какие арифметические функции вычисляются (в унарной системе счисления) следующими машинами Тьюринга, построенными методами из задачи 598 на предыдущей странице:

(а) copy($*$); par($*$, copy(Λ), copy(Λ)); par($*$, mult, add); replace($*$, Λ); add;

(б) copy($*$); par($*$, copy(Λ), copy(Λ)); par($*$, mult, add); par($*$, (copy(Λ); add), nop); replace($*$, Λ); add;

(в) copy($*$); par($*$, (copy(Λ); mult), nop); par($*$, (copy(Λ); add), nop); replace($*$, Λ); add. ⊛

600. Даны машины Тьюринга со стандартной заключительной конфигурацией из задачи 599 на предшествующей странице, а также следующие:

- id_i — возвращает i-й аргумент или пустое слово, если количество аргументов меньше i;
- pos_i — проверяет, является ли i-й аргумент положительным (числа записаны в унарной системе), возвращает пустое слово, если он равен нулю, или $|$, если он положителен;
- dec — уменьшает аргумент на единицу, если он положителен, в противном случае вход не изменяется (числа записаны в унарной системе).

Определить, какие арифметические функции (в унарной системе) вычисляются каждой из машин Тьюринга, программы которых схематично изображены на рис. 39 на следующей странице (см. задачу 598 на предшествующей странице). ⊛

601. Используя машины Тьюринга из предыдущих задач, построить программы машин Тьюринга, вычисляющих следующие функции:

```
replace(Λ, ∗);
replace(Λ, |);
replace(Λ, |);
while pos₁ do
    par(∗, dec,
       (copy(#);
          replace(#, |);
          dec; dec))
end;
id₂
```

```
replace(Λ, ∗);
replace(Λ, |);
replace(Λ, |);
while pos₁ do
    par(∗, dec,
       (copy(#);
          par(#, nop,
             copy(&));
          replace(&, |);
          replace(#, |);
          dec; dec;
          dec; dec))
end;
id₂
```

```
if pos₁ then
    copy(∗);
    replace(Λ, ∗);
    replace(Λ, |);
    replace(Λ, |);
    while pos₁ do
        par(∗, dec,
           copy(#), nop);
        par(#, nop, replace(∗, Λ));
        par(#, nop, mult);
        replace(#, ∗)
    end;
    id₃
else
    replace(Λ, |)
end;
```

```
if pos₁ then
    copy(∗);
    replace(Λ, ∗);
    replace(Λ, |);
    while pos₁ do
        par(∗, dec,
           copy(#), nop);
        par(#, nop, replace(∗, Λ));
        par(#, nop, add);
        replace(#, ∗)
    end;
    id₃
else
    nop
end;
```

Рис. 39: Программы машин из задачи 600.

(а) $f_1(x,y) = \begin{cases} x^2y^3, & \text{если } x < y, \\ \left\lfloor (x+y)/2 \right\rfloor, & \text{в противном случае;} \end{cases}$

(б) $f_2(x,y) = \begin{cases} \left\lfloor \sqrt{x} \right\rfloor, & \text{если } x+1 \geqslant y, \\ 2(x+y), & \text{в противном случае;} \end{cases}$

(в) $f_3(x,y) = \begin{cases} \left\lfloor \log_2 \left(1 + \left\lfloor x/y \right\rfloor\right) \right\rfloor, & \text{если } x > 2y, \\ xy, & \text{в противном случае;} \end{cases}$

(г) $f_4(x,y) = \begin{cases} \left\lfloor \sqrt{x} \right\rfloor, & \text{если } 2x \geqslant y, \\ x \bmod y, & \text{в противном случае.} \end{cases}$ ⊛

602. Показать, как на обычной машине Тьюринга можно организовать выполнение следующих команд:

(а) $q, a \to p, b, \mathrm{return}$ — возврат головки в ту из соседних ячеек, из которой она пришла в текущую;

(б) $q, a \to p, b, \mathrm{zero}$ — возврат головки в нулевую ячейку;

(в) $q, a \to p, b, \mathrm{mirror}$ — сдвиг головку в ячейку $-i$, если она была в ячейке с номером i;

(г) $q, a \to p, b, \mathrm{double}$ — сдвиг головку в ячейку $2i$, если она была в ячейке с номером i;

(д) $q, a \to p, b, \mathrm{next}$ — сдвиг головки в ближайшую справа к текущей ячейку, содержащую a (если она есть, иначе головка остаётся на месте);

(е) $q, a \to p, \mathrm{insert}\, b$ — сдвиг текущей и всех ячеек справа от неё на одну позицию вправо и запись символа b в освободившуюся ячейку, головка оказывается в новой ячейке;

(ж) $q, a \to p, \mathrm{restore}, s$ — замена символа a на тот, который находился в ячейке в начальной конфигурации. ⊛

603. Построить машину Тьюринга для задачи 591 на стр. 185, не увеличивая исходный алфавит Σ. ⊛

604. Реализовать следующие машины Тьюринга без расширения алфавита:

(а) стереть вторую половину слова, оставив центральный символ, если длина была нечётной;

(б) сложить два числа, записанных в двоичной системе. ⊛

605. Один из подходов к моделированию двухсторонней ленты на односторонней заключается в том, чтобы содержимое неотрицательной половины ленты $\mathfrak{M}$ хранить в ячейках с нечётными номерами, а содержимое левой половины — с чётными. То есть новая лента будет иметь вид $\beta(0) = \#$, $\beta(2x+1) = \alpha(x)$ и $\beta(2x) = \alpha(-x)$ при $x > 0$. Построить программу односторонней машины $\mathfrak{N}$, реализующую этот подход. ⊛

606. Доказать, что всякую арифметическую функцию $f(x)$, вычислимую на некоторой машине Тьюринга $\mathfrak{M} = (Q, \Sigma, P, q_0)$ в унарной системе счисления, можно также вычислить на машине Тьюринга $\mathfrak{N}$, алфавит ленты которой содержит лишь два символа: Λ и $|$.

Указание. Использовать для моделирования одного символа a_i алфавита Σ блок из нескольких подряд идущих ячеек, содержащих слово $|^i\Lambda^{k-i}$, где $\Sigma = \{a_0, \ldots, a_k\}$. Заменить каждую команду $\mathfrak{M}$ группой команд, обрабатывающих соответствующий блок ячеек. ⊛

607. k-ленточная машина Тьюринга имеет k лент, по каждой из которых перемещается своя головка. Команда k-ленточной машины Тьюринга имеет вид

$$q, a_1, a_2, \ldots, a_k \to p, b_1, b_2, \ldots, b_k, s_1, s_2, \ldots, s_k,$$

где $q, p \in Q$ — состояния, $a_1, \ldots, a_k$ — символы, наблюдаемые головками на соответствующих лентах, $b_1, \ldots, b_k$ — символы, записываемые головками на соответствующих лентах, $s_1, \ldots, s_k$ — направления сдвигов головок на соответствующих лентах. Входные данные записываются на первой ленте, а остальные в начальной конфигурации пусты. Результат также располагается на первой ленте.

Доказать, что любую функцию, вычислимую на k-ленточной машине Тьюринга, можно также вычислить и на обычной машине Тьюринга. На сколько при этом увеличится время вычисления? ⊛

608. Показать, что палиндромы (симметричные слова) можно распознать на двухленточной машине Тьюринга за время, пропорциональное их длине. ⊛

25. Тезис Тьюринга-Чёрча и неразрешимые проблемы

Поскольку классы функций программно вычислимых, частично рекурсивных и вычислимых по Тьюрингу совпадают, то возникает вопрос о том, насколько общим является этот результат? На эти вопросы теория алгоритмов отвечает следующей эвристической гипотезой, которую называют тезисом Тьюринга-Чёрча:

> всякий алгоритм (в интуитивном смысле) может быть формализован в виде соответствующей машины Тьюринга (частично рекурсивного определения), а класс вычислимых (в интуитивном смысле) функций совпадает с классом функций, вычислимых на машинах Тьюринга (с классом частично рекурсивных функций).

Алгоритмической проблемой называется произвольное множество A (натуральных чисел, слов и т. д.) или, что эквивалентно, характеристическая функция этого множества.

Решением алгоритмической проблемы A называется алгоритм $\mathfrak{M}$ (машина Тьюринга, программа с метками, ч. р. ф. и т. д.), вычисляющий характеристическую функцию I_A этого множества:

$$\mathfrak{M}(x) = I_A(x) = \begin{cases} 1, \text{если } x \in A, \\ 0, \text{если } x \notin A. \end{cases}$$

Проблема, для которой существует решение, называется (алгоритмически) разрешимой, а в противном случае — (алгоритмически) неразрешимой.

Если проблема A алгоритмически неразрешима, то её дополнение $\bar{A}$ тоже неразрешимо.

Множество (или проблема) A (алгоритмически) сводится к множеству (проблеме) B, если существует о. р. ф. f такая, что для всех x

$$x \in A \iff f(x) \in B.$$

В этом случае записывают $A \leqslant_m B$ посредством f.

Если A сводится к B и проблема A неразрешима, то и проблема B неразрешима.

Далее перечисляются некоторые примеры неразрешимых проблем и невычислимых функций. Предположим, зафиксирован некоторый «разумный» способ π представления машин Тьюринга с алфавитом ленты Σ в виде слов того же алфавита.

Машина Тьюринга $\mathfrak{M} = (Q, \Sigma, P, q_0)$ называется самоприменимой, если она останавливается на входном слове $\pi(\mathfrak{M})$. Проблемой самоприменимости называется следующая словарная функция SELF:

$$\mathrm{SELF}(\pi(\mathfrak{M})) = \begin{cases} 1, & \text{если } \mathfrak{M} \text{ самоприменима,} \\ 0, & \text{если } \mathfrak{M} \text{ несамоприменима.} \end{cases}$$

Проблемой остановки называется двухместная функция HALT:

$$\mathrm{HALT}(\pi(\mathfrak{M}), x) = \begin{cases} 1, & \text{если } \mathfrak{M} \text{ останавливается на входе } x, \\ 0, & \text{если } \mathfrak{M} \text{ не останавливается на входе } x. \end{cases}$$

Существуют конкретные машины Тьюринга $\mathfrak{N}$, для которых соответствующие проблемы остановки $\mathrm{HALT}_{\mathfrak{N}}(x) = \mathrm{HALT}(\pi(\mathfrak{N}), x)$ является алгоритмически неразрешимыми. Такой, например, является универсальная машина Тьюринга $\mathfrak{U}$.

Проблемой тотальности называется следующая одноместная функция TOTAL:

$$\mathrm{TOTAL}(\pi(\mathfrak{M})) = \begin{cases} 1, & \text{если } \mathfrak{M} \text{ останавливается на всех входах,} \\ 0, & \text{иначе.} \end{cases}$$

Пусть далее алфавит ленты всех машин Тьюринга зафиксирован.

Если запустить произвольную машину Тьюринга $\mathfrak{M}$ на пустой ленте, то она либо остановится, либо зациклится. В первом случае потенциалом машины $\mathfrak{M}$ назовём количество непустых символов на ленте в момент остановки, во втором — считаем, что потенциал равен нулю. Функция «усердного бобра» bb определяется так: bb(n) — это самый высокий потенциал среди машин, у которых не больше n состояний.

Функция оптимального сжатия arc(w) равна натуральному числу n, если n — это наименьшее количество состояний машины Тьюринга, порождающей слово (или набор слов) w, начав работать на пустой ленте.

Задачи

609. Найти точное количество машин Тьюринга, которые имеют вид $\mathfrak{M} = (Q, \Sigma, P, q_0)$ с заранее зафиксированными множеством состояний Q и алфавитом ленты Σ. ⊛

610. Пусть зафиксирован такой алфавит машин Тьюринга: $\{\Lambda, |\}$. Для функции «усердного бобра» bb

(а) найти bb(1);

(б) доказать, что bb$(2) \geqslant 4$. ⊛

611. Доказать, что отношение алгоритмической сводимости $\leqslant_m$ является рефлексивным и транзитивным. ⊛

612. Доказать алгоритмическую неразрешимость проблемы полноты тестовых данных TEST. Проблема состоит из троек $(\pi(\mathfrak{M}), x, q)$ таких, что в вычислении машины Тьюринга $\mathfrak{M}$ на входе x встречается состояние q. ⊛

613. Доказать алгоритмическую неразрешимость проблемы нуля ZERO. Проблема ZERO состоит из кодов $\pi(\mathfrak{M})$ машин Тьюринга $\mathfrak{M}$ таких, что $\mathfrak{M}(x) = \varepsilon$ для всех x.
Указание. Свести TOTAL к ZERO. ⊛

614. Доказать алгоритмическую неразрешимость следующей проблемы эквивалентности EQU. Проблема EQU состоит из пар $(\pi(\mathfrak{M}), \pi(\mathfrak{N}))$ таких, что машины $\mathfrak{M}$ и $\mathfrak{N}$ эквивалентны.
Указание. Свести ZERO к EQU. ⊛

615. Доказать алгоритмическую неразрешимость проблемы константы CONST. Проблема CONST состоит из таких $\pi(\mathfrak{M})$, что машина Тьюринга $\mathfrak{M}$ на любом входе возвращает один и тот же результат, если вообще останавливается.
Указание. Свести дополнение SELF к CONST. ⊛

616. Доказать алгоритмическую неразрешимость проблемы возрастания GROW. Проблема GROW состоит из таких $\pi(\mathfrak{M})$, что машина Тьюринга $\mathfrak{M}$ вычисляет арифметическую функцию и при этом выполнено $\mathfrak{M}(x) < \mathfrak{M}(y)$ для всех $x < y$, когда оба значения определены.
Указание. Свести дополнение SELF к GROW. ⊛

617. Доказать, что

(а) пересечение двух разрешимых множеств является разрешимым множеством;

(б) объединение двух разрешимых множеств является разрешимым множеством;

(в) декартово произведение двух разрешимых множеств является разрешимым множеством. ⊛

618. Доказать, что для двух разрешимых множеств A и B натуральных чисел их «сумма» $A+B = \{x+y : x \in A, y \in B\}$ и «произведение» (не декартово!) $A \cdot B = \{x \cdot y : x \in A, y \in B\}$ также являются разрешимыми множествами. ⊛

619. Доказать, что для двух разрешимых языков L и K в алфавите Σ их конкатенация LK и итерация L^* тоже будут разрешимыми языками. ⊛

620. Пусть A — разрешимое множество, а $g(x)$ и $h(x)$ являются о. р. ф. Доказать, что функция

$$F(x) = \begin{cases} g(x), & \text{если } x \in A, \\ h(x), & \text{в противном случае} \end{cases}$$

также является общерекурсивной. ⊛

621. Доказать, что проблема ограниченной остановки разрешима. Проблема состоит из троек вида $(\pi(\mathfrak{M}), x, t)$ таких, что вычисление машины Тьюринга $\mathfrak{M}$ на входе x останавливается не более чем за t шагов. ⊛

622. Показать, что при построении проекции язык из разрешимого может стать неразрешимым. ⊛

623. Показать, что функция arc неограниченно возрастает, но не монотонна: может быть $\mathrm{arc}(u) < \mathrm{arc}(v)$, даже если слово v короче слова u. ⊛

624. Показать, что функция arc растёт медленнее каждой вычислимой функции: если для о. р. ф. f выполнено $f(x) \leqslant \mathrm{arc}(x)$ для всех x, то f ограничена. ⊛

625. Пусть $T(n)$ — наибольшее время работы машины Тьюринга с n состояниями на пустой ленте. Доказать, что функция $T(n)$ невычислима. ⊛

626. Рабочей областью машины Тьюринга $\mathfrak{M}$ на входе x назовём множество ячеек, в которых побывала головка машины до её остановки. Обозначим с помощью $S(\mathfrak{M}, x)$ размер рабочей области машины $\mathfrak{M}$ на входе x. Если машина $\mathfrak{M}$ на x не останавливается, то значение $S(\mathfrak{M}, x)$ может быть произвольным. Допустим, что для машины $\mathfrak{M}$ функция $S(\mathfrak{M}, x)$ мажорируется общерекурсивной функцией $S^*(x)$: $S(\mathfrak{M}, x) \leqslant S^*(x)$ для всех x. Доказать, что проблема $\mathrm{HALT}_{\mathfrak{M}}$ остановки для машины $\mathfrak{M}$ алгоритмически разрешима. ⊛

627. Доказать, что функция $S(\pi(\mathfrak{M}), x)$ из предыдущей задачи невычислима. ⊛

628. Для любой пары множеств A и B определим

$$A \vee B = \{2x : x \in A\} \cup \{2x + 1 : x \in B\}.$$

Доказать, что для любого множества C условие $A \vee B \leqslant_m C$ выполнено тогда и только тогда, когда $A \leqslant_m C$ и $B \leqslant_m C$. ⊛

629. Непустое множество A называется *рекурсивно перечислимым*, если оно является областью значений некоторой о. р. ф. f: $A = \{f(0), f(1), f(2), \ldots\}$. Доказать, что

(а) любое непустое разрешимое множество является рекурсивно перечислимым;

(б) непустое пересечение рекурсивно перечислимого множества с разрешимым множеством также является рекурсивно перечислимым множеством;

(в) объединение двух рекурсивно перечислимых множеств также является рекурсивно перечислимым множеством;

(г) для любого рекурсивно перечислимого множества A существует ч. р. ф. g такая, что $\operatorname{dom} g = A$. ⊛

630. Доказать, что множество TOTAL не является рекурсивно перечислимым.

Указание. Для универсальной машины Тьюринга $\mathfrak{U}$ рассмотреть функцию $\mathfrak{U}(f(x), x) \mathbin{\&} a$ (конкатенация с непустым символом). ⊛

631. Доказать, что существует ч. р. ф. f, которую нельзя дополнить ни до какой о. р. ф. g, то есть такой, что $f(x) = g(x)$ для всех $x \in \operatorname{dom} f$.

Указание. Для универсальной машины Тьюринга $\mathfrak{U}$ рассмотреть арифметическую функцию $f(x) = \mathfrak{U}(x, x) + 1$. ⊛

Ответы и решения

Множества, отношения и функции

1. $\mathsf{P}(\varnothing) = \{\varnothing\}$, $\mathsf{P}(\{\varnothing\}) = \{\varnothing, \{\varnothing\}\}$, $\mathsf{P}(\{1, 2, 3\}) = \{\varnothing, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$, $\mathsf{P}(\{a, \{1, 2\}, \varnothing\}) = \{\varnothing, \{a\}, \{\{1, 2\}\}, \{\varnothing\}, \{a, \{1, 2\}\}, \{a, \varnothing\}, \{\{1, 2\}, \varnothing\}, \{a, \{1, 2\}, \varnothing\}\}$.

2. D, F и G.

3. $(A \cup B) \setminus C = \{\{\varnothing\}, \{a, c, d\}, e, \{a\}, \{b\}\}$, $|(A \cup B) \setminus C| = 5$.

4. $(A \setminus B) \cap C = \{b, \{\varnothing\}\}$, $|(A \setminus B) \cap C| = 2$.

5. $A \times B = \{(0, a), (0, b), (0, c), (1, a), (1, b), (1, c)\}$, $B \times A = \{(a, 0), (a, 1), (b, 0), (b, 1), (c, 0), (c, 1)\}$.

6. $A \times B = B \times A$ тогда и только тогда, когда $A = B$ или одно из множеств A, B пусто. В обратную сторону очевидно. В прямую сторону: предположим, что $A \neq B$, $A \neq \varnothing$, $B \neq \varnothing$. Тогда существует такой элемент x, что либо $x \in A$, $x \notin B$, либо $x \notin A$, $x \in B$. Предположим, что имеет место первый случай. Пусть $y \in B$. Получаем $(x, y) \in A \times B$, $(x, y) \notin B \times A$. Второй случай рассматривается аналогично.

7. **(а)** $A \setminus B = \{0, 1\}$, $C \cap D = \{a, c\}$, $F_1 = \{(0, a), (0, c), (1, a), (1, c)\}$,
(б) $A \cap B = \{2\}$, $C \cap D = \{a, c\}$, $F_2 = \{(2, a), (2, c)\}$,
(в) $B \setminus A = \{3\}$, $C \setminus D = \{b\}$, $F_3 = \{(3, b)\}$.

8. **(а)** Если $x \in A \cap B$, то $x \in A$ и $x \in B$, то есть $x \in A$. Если $x \in A$, то $x \in A$ или $x \in B$, то есть $x \in A \cup B$.
(б) Если $x \in A \setminus B$, то $x \in A$ и $x \notin B$, то есть $x \in A$.

9. **(б)** Докажем, что $A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C)$. Пусть $x \in A \cap (B \cup C)$. По определению пересечения множеств это значит, что $x \in A$ и $x \in B \cup C$. По определению объединения $x \in B$ или $x \in C$. Если $x \in B$, то $x \in A \cap B$, а значит, $x \in (A \cap B) \cup (A \cap C)$. Если же $x \in C$, то $x \in A \cap C$ и снова $x \in (A \cap B) \cup (A \cap C)$. Докажем, что $(A \cap B) \cup (A \cap C) \subseteq A \cap (B \cup C)$. Пусть $x \in (A \cap B) \cup (A \cap C)$. По определению объединения множеств это значит, что $x \in A \cap B$ или $x \in A \cap C$. Если $x \in A \cap B$, то $x \in A$ и $x \in B$. Тогда $x \in B \cup C$, и следовательно, $x \in A \cap (B \cup C)$. Если же $x \in A \cap C$, то $x \in A$ и $x \in C$. Значит, $x \in B \cup C$, и следовательно, $x \in A \cap (B \cup C)$.

(г) Докажем $A \setminus (B \cup C) \subseteq (A \setminus B) \cap (A \setminus C)$. Если $x \in A \setminus (B \cup C)$, то $x \in A$ и $x \notin B \cup C$, то есть $x \notin B$ и $x \notin C$. Следовательно, $x \in A \setminus B$ и $x \in A \setminus C$, что означает $x \in (A \setminus B) \cap (A \setminus C)$. Докажем $A \setminus (B \cup C) \supseteq (A \setminus B) \cap (A \setminus C)$. Если $x \in (A \setminus B) \cap (A \setminus C)$, то $x \in A \setminus B$ и $x \in A \setminus C$. Следовательно, $x \in A$, но $x \notin B$ и $x \notin C$. Получаем $x \notin B \cup C$ и $x \in A \setminus (B \cup C)$.

(е) Докажем $(A \setminus B) \cap C \subseteq (A \cap C) \setminus B$. Если $x \in (A \setminus B) \cap C$, то $x \in A \setminus B$ и $x \in C$. Следовательно, $x \in A$ и $x \notin B$. Таким образом, $x \in A \cap C$ и $x \in (A \cap C) \setminus B$. Докажем $(A \setminus B) \cap C \supseteq (A \cap C) \setminus B$. Если $x \in (A \cap C) \setminus B$, то $x \in A \cap C$ и $x \notin B$. Получаем, что $x \in A$ и $x \in C$. Следовательно, $x \in A \setminus B$ и $x \in (A \setminus B) \cap C$.

(ж) Докажем $A \setminus (B \setminus C) \subseteq (A \setminus B) \cup (A \cap C)$. Если $x \in A \setminus (B \setminus C)$, то $x \in A$ и $x \notin B \setminus C$. Последнее означает, что $x \notin B$ или $x \in C$. В случае $x \notin B$ получаем $x \in A \setminus B$ и, следовательно, $x \in (A \setminus B) \cup (A \cap C)$. В случае $x \in B$ получаем $x \in A \cap C$ и, следовательно, $x \in (A \setminus B) \cup (A \cap C)$. Докажем $A \setminus (B \setminus C) \supseteq (A \setminus B) \cup (A \cap C)$. Если $x \in (A \setminus B) \cup (A \cap C)$, то $x \in A \setminus B$ или $x \in A \cap C$. В случае $x \in A \setminus B$ получаем $x \in A$, но $x \notin B$. Следовательно, $x \notin B \setminus C$ и $x \in A \setminus (B \setminus C)$. В случае $x \in A \cap C$ получаем $x \in A$ и $x \in C$. Следовательно, $x \notin B \setminus C$ и $x \in A \setminus (B \setminus C)$.

10. Используем тождества из задачи 9.

(а) $\overline{A \cap B} = U \setminus (A \cap B) = (U \setminus A) \cup (U \setminus B) = \bar{A} \cup \bar{B}$;

(б) $\overline{A \cup B} = U \setminus (A \cup B) = (U \setminus A) \cap (U \setminus B) = \bar{A} \cap \bar{B}$;

(в) $A \setminus \bar{B} = A \setminus (U \setminus B) = (A \setminus U) \cup (A \cap B) = \varnothing \cup (A \cap B) = A \cap B$;

(г) $\bar{A} \setminus B = (U \setminus A) \setminus B = U \setminus (A \cup B) = \overline{A \cup B} = \bar{A} \cap \bar{B}$;

(д) $\bar{A} \setminus \bar{B} = (U \setminus A) \setminus (U \setminus B) = ((U \setminus A) \setminus U) \cup ((U \setminus A) \cap B) = ((U \setminus (A \cup U)) \cup ((U \cap B) \setminus A)) = ((U \setminus U) \cup (B \setminus A)) = \varnothing \cup (B \setminus A) = B \setminus A$.

11. (а) Докажем включение $A \times (B \cup C) \subseteq (A \times B) \cup (A \times C)$. Пусть $x \in A \times (B \cup C)$. По определению декартова произведения это значит, что $x = (y, z)$, $y \in A$, $z \in B \cup C$. Тогда $z \in B$ или $z \in C$. Если $z \in B$, то $(y, z) \in A \times B$, поэтому $x \in (A \times B) \cup (A \times C)$. Если $z \in C$, то $(y, z) \in A \times C$, значит, и в этом случае $x \in (A \times B) \cup (A \times C)$. Пусть теперь $x \in (A \times B) \cup (A \times C)$, то есть $x \in A \times B$ или $x \in A \times C$. В первом случае по определению декартова произведения получаем $x = (y, z)$ для $y \in A$, $z \in B$. Тогда $z \in B \cup C$ и $x = (y, z) \in A \times (B \cup C)$. Аналогично для случая $z \in C$.

Остальные включения доказываются аналогично.

12. Пусть $A \subseteq B$ и $x \in \mathsf{P}(A)$, тогда $x \subseteq A$, следовательно, $x \subseteq B$ и $x \in \mathsf{P}(B)$. Если $\mathsf{P}(A) \subseteq \mathsf{P}(B)$ и $x \in A$, то $\{x\} \subseteq A$, поэтому $\{x\} \subseteq B$ и $x \in B$.

13. (а) Пусть $A \subseteq B \cap C$. Тогда для любого $x \in A$ получаем $x \in B \cap C$, то есть $x \in B$ и $x \in C$. Это означает, что $A \subseteq B$ и $A \subseteq C$. Пусть теперь $A \subseteq B$ и $A \subseteq C$. Тогда для любого $x \in A$ получаем $x \in B$ и $x \in C$, то есть $x \in B \cap C$. Следовательно,, что $A \subseteq B \cap C$.
(б) Пусть $A \subseteq B \setminus C$. Тогда для любого $x \in A$ получаем $x \in B \setminus C$, то есть $x \in B$ и $x \notin C$. Это означает, что $A \subseteq B$, а A и C не имеют общих элементов, $A \cap C = \varnothing$. Пусть $A \subseteq B$ и $A \cap C = \varnothing$. Тогда для любого $x \in A$ получаем $x \in B$ и $x \notin C$, то есть $x \in B \setminus C$. Поэтому $x \in B \setminus C$.

14. (а) Пусть $x \in \mathsf{P}(A \cap B)$, значит, $x \subseteq A \cap B$, следовательно, $x \subseteq A$ и $x \subseteq B$, поэтому $x \in \mathsf{P}(A)$ и $x \in \mathsf{P}(B)$, то есть $x \in \mathsf{P}(A) \cap \mathsf{P}(B)$. Пусть $x \in \mathsf{P}(A) \cap \mathsf{P}(B)$, тогда $x \in \mathsf{P}(A)$ и $x \in \mathsf{P}(B)$, значит, $x \subseteq A$ и $x \subseteq B$, поэтому $x \subseteq A \cap B$ и $x \in \mathsf{P}(A \cap B)$.
(б) Если $x \in \mathsf{P}(A) \cup \mathsf{P}(B)$, то $x \in \mathsf{P}(A)$ или $x \in \mathsf{P}(B)$, значит, $x \subseteq A$ или $x \subseteq B$, поэтому $x \subseteq A \cup B$ и $x \in \mathsf{P}(A \cup B)$. Пусть $A = \{a\}$, $B = \{b\}$, где $a \neq b$. Тогда $\mathsf{P}(A \cup B) = \{\varnothing, \{a\}, \{b\}, \{a, b\}\}$, но $\mathsf{P}(A) \cup \mathsf{P}(B) = \{\varnothing, \{a\}, \{b\}\}$.
(в) Пусть $x \in \mathsf{P}(A \setminus B)$, значит, $x \subseteq A \setminus B$, следовательно, $x \subseteq A$ и $x \cap B = \varnothing$. Получаем $x \in \mathsf{P}(A)$, а если $x \in \mathsf{P}(B)$, то $x = \varnothing$. Тогда $x \in (\mathsf{P}(A) \setminus \mathsf{P}(B)) \cup \{\varnothing\}$. Пусть $A = \{a, b\}$, $B = \{b\}$, где $a \neq b$. Тогда $\mathsf{P}(A \setminus B) = \{\varnothing, \{a\}\}$, но $(\mathsf{P}(A) \setminus \mathsf{P}(B)) \cup \{\varnothing\} = \{\varnothing, \{a\}, \{a, b\}\}$.

15. (а) $\operatorname{dom} R = \omega \setminus \{0\}$, $\operatorname{rng} R = \omega$, $R^{-1} = \{(x, y) : x, y \in \omega$ и x делится на $y\}$, $R \circ R = R$, $R \circ R^{-1} = \{(x, y) : x, y \in \omega \setminus \{0\}\}$;
(б) $\operatorname{dom} R = \{0, 1, \ldots, 10\}$, $\operatorname{rng} R = \{0, 1, \ldots, 10\}$, $R^{-1} = R$, $R \circ$ $\circ\, R = R \circ R^{-1} = \{(x, y) : 0 \leqslant x, y \leqslant 10\}$;
(в) $\operatorname{dom} R = \omega$, $\operatorname{rng} R = \{3x + 1 : x \in \omega\}$, $R^{-1} = \{(x, y) : x = 3y + 1\}$, $R \circ R = \{(x, y) : x, y \in \omega$ и $y = 9x + 4\}$, $R \circ R^{-1} = \{(x, x) : x \in \omega\}$;
(г) $\operatorname{dom} R = \{x \in \omega : x \leqslant 10\}$, $\operatorname{rng} R = \{x^2 : x \in \omega$ и $x \leqslant 10\}$, $R^{-1} = \{(x^2, x) : x \in \omega$ и $x \leqslant 10\}$, $R \circ R = \{(x, x^4) : x \in \omega$ и $x \leqslant 10\}$, $R \circ R^{-1} = \{(x, x) : x \in \omega$ и $x \leqslant 10\}$;
(д) $\operatorname{dom} R = \{a, b, c, d\}$, $\operatorname{rng} R = \{b, c, d\}$, $R^{-1} = \{(b, a), (c, b), (d, b), (d, c), (b, d)\}$, $R \circ R = \{(a, c), (a, d), (b, b), (b, d), (c, b), (d, c), (d, d)\}$, $R \circ R^{-1} = \{(a, a), (a, d), (b, b), (b, c), (c, b), (c, c), (d, a), (d, d)\}$.

16. (а) R_1 рефлексивно, симметрично, транзитивно;
(б) R_2 рефлексивно, антисимметрично, транзитивно;
(в) R_3 антисимметрично, транзитивно.

17. $A = \{a, b\}$, $R = \{(a, b), (b, a)\}$. На множествах $\varnothing$ и $\{a\}$ существуют только транзитивные отношения $\varnothing$ и $\{(a, a)\}$.

18. P и R.

19. Все отношения рефлексивны.

20. Все отношения симметричны.
21. (а) и (г).
22. (а) и (г).
23. **(а)** $L = \{((x, y), (u, v)) : 1 \leqslant x, y, u, v \leqslant 8,\ (x = u, y \neq v)$ или $(x \neq u, y = v)\}$;
(б) $K = \{((x, y), (u, v)) : 1 \leqslant x, y, u, v \leqslant 8,\ |x - u| \cdot |y - v| = 2\}$.
L и K не являются отношениями эквивалентности, так как они не рефлексивны. $L \circ L = \{((x, y), (u, v)) : 1 \leqslant x, y, u, v \leqslant 8\} = S \times S$.
24. **(а)** Эквивалентность. Классы: $\{1, 4\}$, $\{2\}$, $\{3\}$, $\{5\}$.
(б) Отношение не рефлексивно.
(в) Отношение не транзитивно.
(г) Эквивалентность. Классы: $\{1, 2, 4\}$, $\{3, 5\}$.
(д) Отношение не рефлексивно.
(е) Эквивалентность. Классы: $\{1, 3, 5\}$, $\{2, 4\}$.
(ж) Отношение не рефлексивно.
25. **(а)** Да. **(б)** Нет.
26. **(а)** Рефлексивно, антисимметрично, транзитивно.
(б) Рефлексивно, симметрично.
(в) Антирефлексивно.
27. **(а)** R_1 рефлексивно, симметрично.
(б) R_2 рефлексивно, транзитивно.
(в) R_3 рефлексивно, симметрично, транзитивно.
28. Будут (а), (б), (д), (ж).
29. Будут все кроме (д) и (е).
30. **(а)** R_1 является отношением частичного порядка, но не является отношением линейного порядка.
(б) R_2 является отношением линейного порядка.
31. (б), (д), (ж), (и).
32. **(а)** $\operatorname{rng} f = \mathbb{R}$, взаимно однозначная;
(б) $\operatorname{rng} f = [1, \infty)$;
(в) $\operatorname{rng} f = \mathbb{R}$, взаимно однозначная;
(г) $\operatorname{rng} f = (0, \infty)$, разнозначная;
(д) $\operatorname{rng} f = [0, \infty)$;
(е) $\operatorname{rng} f = [-1, 1]$, разнозначная;
(ж) $\operatorname{rng} f = [0, 1]$;
(з) $\operatorname{rng} f = [-1, 1]$, сюръективная;
(и) $\operatorname{rng} f = \mathbb{R}$, сюръективная.

33. (а) Неверно.
(б) Верно. Нужно проверить, что для любого y существует x такой, что $(f \circ g)(x) = y$, то есть $f(g(x)) = y$. Так как f — сюръективная функция, то существует z такой, что $f(z) = y$. Так как g — сюръективная функция, то существует x такой, что $g(x) = z$. Тогда $f(g(x)) = y$.
(в) Верно.
(г) Неверно.
(д) Верно.
(е) Верно.

34. (а) Если $y \in f[A \cap B]$, то $y = f(x)$ для некоторого $x \in A \cap B$, следовательно, $x \in A$ и $x \in B$, поэтому $y = f(x) \in f[A]$ и $y = f(x) \in f[B]$, то есть $y \in f[A] \cap f[B]$. Пусть $f(x) = x^2$, $A = \{1\}$, $B = \{-1\}$, тогда $f[A \cap B] = f[\varnothing] = \varnothing$, но $f[A] \cap f[B] = \{1\}$.
(б) Если $y \in f[A \cup B]$, то $y = f(x)$ для некоторого $x \in A \cup B$, следовательно, $x \in A$ или $x \in B$. В первом случае получаем $y = f(x) \in f[A]$ и $y \in f[A] \cup f[B]$, во втором — $y = f(x) \in f[B]$ и $y \in f[A] \cup f[B]$. Если $y \in f[A] \cup f[B]$, то $y \in f[A]$ или $y \in f[B]$. В первом случае получаем $y = f(x)$ для некоторого $x \in A$, следовательно, $x \in A \cup B$, поэтому $y = f(x) \in f[A \cup B]$. Аналогично во втором случае.
(в) Если $y \in f[A] \setminus f[B]$, то $y \in f[A]$, но $y \notin f[B]$. Получаем $y = f(x)$ для некоторого $x \in A$. При этом не может быть $x \in B$, иначе было бы $x \in f[B]$. Следовательно, $x \in A \setminus B$ и $y = f(x) \in f[A \setminus B]$. Пусть $f(x) = x^2$, $A = \{-1, 1\}$, $B = \{-1\}$, тогда $f[A \setminus B] = f[\{1\}] = \{1\}$, но $f[A] \setminus f[B] = \{1\} \setminus \{1\} = \varnothing$.
(г) Если $x \in f^{-1}[A \cap B]$, то $f(x) \in A \cap B$, следовательно, $f(x) \in A$ и $f(x) \in B$. Получаем $x \in f^{-1}[A]$ и $x \in f^{-1}[B]$, то есть $x \in f^{-1}[A] \cap f^{-1}[B]$. Если $x \in f^{-1}[A] \cap f^{-1}[B]$, то $x \in f^{-1}[A]$ и $x \in f^{-1}[B]$. Следовательно, $f(x) \in A$ и $f(x) \in B$, поэтому $f(x) \in A \cap B$ и $x \in f^{-1}[A \cap B]$.
(д) Если $x \in f^{-1}[A \cup B]$, то $f(x) \in A \cup B$, следовательно, $f(x) \in A$ или $f(x) \in B$. В первом случае получаем $x \in f^{-1}[A]$ и $x \in f^{-1}[A] \cup f^{-1}[B]$, аналогично во втором. Если $x \in f^{-1}[A] \cup f^{-1}[B]$, то $x \in f^{-1}[A]$ или $x \in f^{-1}[B]$. Следовательно, $f(x) \in A$ или $f(x) \in B$. В первом случае получаем $f(x) \in A \cup B$, следовательно, $x \in f^{-1}[A \cup B]$. Аналогично во втором случае.
(е) Если $x \in f^{-1}[A \setminus B]$, то $f(x) \in A \setminus B$, следовательно, $f(x) \in A$, но $f(x) \notin B$. Получаем $x \in f^{-1}[A]$ и $x \notin f^{-1}[B]$, то есть $x \in f^{-1}[A] \setminus f^{-1}[B]$. Если $x \in f^{-1}[A] \setminus f^{-1}[B]$, то $x \in f^{-1}[A]$ и $x \notin f^{-1}[B]$. Следовательно, $f(x) \in A$ и $f(x) \notin B$, поэтому $f(x) \in A \setminus B$ и $x \in f^{-1}[A \setminus B]$.

35. Если f — частичный порядок, то он является транзитивным: из $(x, y), (y, z) \in f$ следует $(x, z) \in f$ для любых $x, y, z \in A$. Но из определения функции получаем, что $(x, y), (x, z) \in f$ влечёт $y = z$. Следовательно, $f(f(x)) = f(y) = z = y = f(x)$. Наоборот, пусть $f(f(x)) = f(x)$ для всех $x \in A$. Если $(x, y), (y, x) \in f$, то $f(x) = y$ и $f(y) = x$, что означает $x = f(y) = f(f(x)) = f(x) = y$, это доказывает антисимметричность. Если $(x, y), (y, z) \in f$, то $f(x) = y$ и $f(y) = z$, далее получаем $z = f(y) = f(f(x)) = f(x) = y$, то есть выполнено $(x, z) \in f$, что доказывает транзитивность.

36. Из разнозначности f сразу следует разнозначность g. Пусть $b \in B$, то есть $b \in A_i$ для всех $i \in \omega$. Так как $b \in A_1$, то $b = f(a)$ для некоторого $a \in A$. Из разнозначности f получаем, что ни для какого x отличного от a равенство $f(x) = b$ не выполняется. Следовательно, если бы было $a \notin A_i$ для какого-то i, то мы бы получили $b \notin A_{i+1}$ и $b \notin B$, противоречие. Значит, $a \in A_i$ для всех $i \in \omega$ и $a \in B$. Поэтому для каждого $b \in B$ существует прообраз из B и функция g сюръективна.

37. **(а)** Расположить элементы $A \times B$ в прямоугольной таблице размером $|A| \times |B|$.

(б) $A \cup B = (A \setminus B) \cup (A \cap B) \cup (B \setminus A)$, $A = (A \setminus B) \cup (A \cap B)$ и $B = (B \setminus A) \cup (A \cap B)$, где объединяемые множества не пересекаются. Следовательно, $|A \cup B| = |A \setminus B| + |A \cap B| + |B \setminus A|$, $|A| = |A \setminus B| + |A \cap B|$ и $|B| = |B \setminus A| + |A \cap B|$. Получаем, $|A \cup B| = |A \setminus B| + |A \cap B| + |B \setminus A| = |A| + |B \setminus A| = |A| + |B| - |A \cap B|$.

(в) Каждый набор из n нулей и единиц является двоичной записью в точности одного числа из диапазона $[0, 2^n - 1]$, который содержит 2^n чисел.

(г) Пусть $A = \{a_1, \ldots, a_n\}$, тогда для каждого $x \in \mathsf{P}(A)$ можно взаимно однозначно построить двоичное число $b_1 \ldots b_n$, где $b_i = 1$ при $a_i \in x$ и $b_i = 0$ при $a_i \notin x$. Таких чисел $2^n = 2^{|A|}$.

38. 2^{n^k}, количество подмножеств множества A^k.

39. **(а)** 2^{n^2}, количество подмножеств множества $A \times A$.

(б) 2^{n^2-n}. Все пары вида (a, a) принадлежат отношению, поэтому остаётся $n^2 - n$ пар, которые можно выбирать произвольно.

(в) 2^{n^2-n}. Аналогично (б).

(г) $2^{(n^2+n)/2}$. Пусть $A = \{a_1, a_2, \ldots, a_n\}$. При $i \neq j$ либо обе пары (a_i, a_j), (a_j, a_i) принадлежат отношению, либо обе не принадлежат, поэтому нужно выбрать несколько пар вида (a_i, a_j) для $i < j$. Существует $(n^2 - n)/2$ таких пар. Пары вида (a_i, a_i) можно выбирать произвольно, их количество равно n. Поэтому отношение определяется выбором нескольких элементов из множества мощности $(n^2 - n)/2 + n = (n^2 + n)/2$.

(д) $2^n \cdot 3^{(n^2-n)/2}$. Как и в пункте (г), пары вида (a_i, a_i) выбираются произвольно, для этого есть 2^n вариантов. Для любых двух пар (a_i, a_j), (a_j, a_i) при $i \neq j$ существуют три варианта: отношение содержит либо только первую пару, либо только вторую, либо не содержит ни одной из них, поэтому существует $3^{(n^2-n)/2}$ вариантов. Общее количество отношений составляет $2^n \cdot 3^{(n^2-n)/2}$.

(е) $n!$, количество перестановок.

40. (а) Определим вес точки (x, y) как $w(x, y) = |x| + |y|$. Для любого натурального числа k существует лишь конечное количество точек веса k. Поэтому все точки с целочисленными координатами можно упорядочить по весу, а точки одного веса — произвольным образом (например, по первой координате, а точки с равной первой координатой — по второй координате). Тогда взаимно однозначное соответствие множества ω и множества точек задаётся функцией f такой, что $f(i)$ — это $(i+1)$-й элемент в определённом выше упорядочении.

(б)–(в) Аналогично (а).

(г) Упорядочим слова по возрастанию длины, а слова одной длины, которых конечно много, — произвольным образом. Далее как в пункте (а).

(д) Любое рациональное число (кроме 0) единственным образом представляется в виде несократимой дроби $^p/_q$, где p — целое число, q — натуральное число. Ноль будем записывать в виде $^0/_1$. Определим вес рационального числа как $w(^p/_q) = |p| + q$. Далее как в пункте (а).

(е) Следует из (в), так как каждый многочлен задаётся набором $(a_0, a_1, \ldots, a_n)$ своих коэффициентов.

(ж) Пронумеруем все многочлены натуральными числами. Для каждого u зафиксируем некоторый многочлен $p_u(x)$, корнем которого u является. Расположим все u по возрастанию номера многочлена $p_u(x)$, а корни одного многочлена, которых конечно много, — в произвольном порядке. Далее как в пункте (а).

Метод математической индукции

41. (а) Базис: $2 \cdot 3 + 1 = 7 \leqslant 2^3$. Шаг: пусть $2n + 1 \leqslant 2^n$. Очевидно, что $2 \leqslant 2^n$ при $n \geqslant 3$. Тогда $2(n+1)+1 = 2n+1+2 \leqslant 2^n+2 \leqslant 2^n+2^n = 2^{n+1}$.

(б) Базис: $4^2 = 16 \leqslant 16 = 2^4$. Шаг: пусть $n^2 \leqslant 2^n$. Тогда $(n+1)^2 = n^2 + 2n + 1 \leqslant 2^n + 2^n \leqslant 2^{n+1}$.

42. (а) Базис: $7^0 - 1 = 0$ делится на 6. Шаг: пусть $7^n - 1$ делится на 6. Тогда $7^{n+1} - 1 = 7(7^n - 1) + 6$ делится на 6.

(б) Шаг: $11^{n+1} - 6 = 11(11^n - 6) + 60$ делится на 5, если $11^n - 6$ делится.

(в) Шаг: $67^{n+1} - 23^{n+1} = 67(67^n - 23^n) + 44 \cdot 23^n$ делится на 4, если $67^n - 23^n$ делится.

(г) Шаг: $3^{n+1} + 7^{n+1} - 2 = 3(3^n + 7^n - 2) + 4(7^n + 1)$. Первое слагаемое делится на 8 по индукционному предположению, второе — так как является произведением 4 на чётное число.

(д) Базис: при $n = 1$ получаем $3^1 + 1^2 = 4$, делится на 4. Шаг: пусть для нечётного n утверждение верно. Тогда для $n+2$ (следующее нечётное число) получаем $3^{n+2} + (n+2)^2 = 9 \cdot 3^n + n^2 + 4n + 4 = 8 \cdot 3^n + (3^n + n^2) + 4n + 4$ делится на 4, так как все слагаемые делятся.

43. Базис: $n = 0$, $0 = 0$. Если для n доказано, то $1^4 + 2^4 + \ldots + n^4 + (n+1)^4 = n(n+1)(2n+1)(3n^2+3n-1)/30 + (n+1)^4 = (n+1)(n(2n+1)(3n^2+3n-1)+30(n+1)^3)/30 = (n+1)(6n^4+6n^3-2n^2+3n^3+3n^2-n+30n^3+90n^2+90n+30)/30 = (n+1)(6n^4+39n^3+91n^2+89n+30)/30$ и $(n+1)((n+1)+1)(2(n+1)+1)(3(n+1)^2+3(n+1)-1)/30 = (n+1)(n+2)(2n+3)(3n^2+9n+5)/30 = (n+1)(2n^2+7n+6)(3n^2+9n+5)/30 = (n+1)(6n^4+18n^3+10n^2+21n^3+63n^2+35n+18n^2+54n+30)/30 = (n+1)(6n^4+39n^3+91n^2+89n+30)/30$.

44. Базис: $n = 0$, $0 = 0$. Если для n доказано, то $1 \cdot 2 + 2 \cdot 3 + \ldots + n(n+1) + (n+1)(n+2) = n(n+1)(n+2)/3 + (n+1)(n+2) = (n(n+1)(n+2) + 3(n+1)(n+2))/3 = (n+1)(n+2)(n+3)/3$.

45. Базис: $n = 0$, $0^2 = 0$. Если для n доказано, то $(1 + \cdots + n + (n+1))^2 = (1 + \cdots + n)^2 + 2(1 + \cdots + n)(n+1) + (n+1)^2 = (1^3 + \cdots + n^3) + 2n(n+1)(n+1)/2 + (n+1)^2 = (1^3 + \cdots + n^3) + n(n+1)^2 + (n+1)^2 = (1^3 + \cdots + n^3) + (n+1)^3$.

46. Базис: $n = 0$, $0 = 0$. Если для n доказано, то $1 \cdot (n+1) + 2 \cdot n + \cdots + n \cdot 2 + (n+1) \cdot 1 = 1 \cdot (n+1) + 2 \cdot (n-1+1) + \cdots + n \cdot (2-1+1) + (n+1) \cdot (1-1+1) = (1 \cdot n + 2 \cdot (n-1) + \cdots + n \cdot 1 + (n+1) \cdot 0) + (1 + 2 + \cdots + n + (n+1)) = n(n+1)(n+2)/6 + (n+1)(n+2)/2 = (n+1)(n+2)(n+3)/6$.

47. Базис: $n = 0$, $1 = (x-1)/(x-1)$. Если для n доказано, то $1 + x + x^2 + x^3 + \ldots + x^n + x^{n+1} = (x^{n+1}-1)/(x-1) + x^{n+1} = ((x^{n+1}-1) + x^{n+1}(x-1))/(x-1) = (x^{n+1} - 1 + x^{n+2} - x^{n-1})/(x-1) = (x^{n+2}-1)/(x-1)$.

48. Базис: $n = 0$, $z^0 = 1 = r^0(\cos 0\varphi + i \sin 0\varphi)$. Шаг: пусть для n утверждение верно, тогда $z^{n+1} = z^n \cdot z = r^n(\cos n\varphi + i \sin n\varphi) \cdot r(\cos\varphi + i \sin\varphi) = r^{n+1}(\cos n\varphi \cos\varphi + i \cos n\varphi \sin\varphi + i \sin n\varphi \cos\varphi + i^2 \sin n\varphi \sin\varphi) = r^{n+1}((\cos n\varphi \cos\varphi - \sin n\varphi \sin\varphi) + i(\cos n\varphi \sin\varphi + \sin n\varphi \cos\varphi)) = r^{n+1}(\cos(n+1)\varphi + i \sin(n+1)\varphi)$.

49. Доказательство индукцией по n. Базис: если $n = 0$, то имеется всего одна область, которую можно покрасить в любой из цветов. Шаг: предположим, что для n прямых требуемая раскраска найдена. Пусть проведена ещё одна прямая. Если изменить цвета всех областей, находящихся по одну сторону от новой прямой, на противоположные, то получится правильная раскраска. Действительно, если граница двух областей не лежит на новой прямой, то они либо обе не меняли цвет, либо обе поменяли его. В обоих случаях цвета областей останутся разными. Если же граница двух областей лежит на новой прямой, то раньше они образовывали одну область одного цвета. После перекрашивания цвет одной из новых областей изменится, а цвет другой останется прежним, поэтому они будут покрашены в разные цвета.

50. Базис: при $n = 0$ есть в точности одна область. Шаг: $(n+1)$-я прямая имеет n точек пересечения с предыдущими n прямыми. Значит, она проходит через $n+1$ область и разбивает каждую из них на две части. Поэтому к имеющимся добавится $n+1$ область. Общее количество областей станет равно $(n^2+n+2)/2+n+1 = ((n+1)^2+(n+1)+2)/2$.

51. Базис: при $n = 4$ есть в точности одна точка пересечения, $\frac{1\cdot2\cdot3\cdot4}{24} = 1$. Шаг: если из $(n+1)$-й точки провести диагональ d в любую другую, то она разобьёт все остальные вершины многоугольника на две части, лежащие по разные стороны от d. Если одна часть содержит k вершин, а другая ℓ, то их сумма равна $n+1-2 = n-1$, а d пересечёт $k\ell$ диагоналей. Таким образом, общее количество точек пересечения возрастёт на $1\cdot(n-2)+2\cdot(n-3)+\dots+(n-2)\cdot1 = \frac{(n-2)(n-1)n}{6}$ (задача 46). Получаем $D(n+1) = D(n)+\frac{(n-2)(n-1)n}{6} = \frac{(n-3)(n-2)(n-1)n}{24}+\frac{4(n-2)(n-1)n}{24} = \frac{(n-2)(n-1)n(n+1)}{24}$.

52. Индукция по n — стоимости письма. Базис: при $n = 6$ получаем $n = 2+2+2$; при $n = 7$ требуется одна семирублёвая марка. Шаг: пусть при всех $k < n$ утверждение доказано и $n > 7$. В частности, $6 \leqslant n-2 < n$, поэтому сумму в $n-2$ рубля можно составить из двух- и семирублёвых марок. Тогда сумма в n рублей составляется добавлением одной двухрублёвой марки.

53. В доказательстве пропущен базис индукции. Неравенство справедливо при всех $n \geqslant 3$.

54. Индукционный шаг неверен при переходе от $n = 0$ к $n = 1$, так как количество областей возрастает на 1, а не на 0. $N(n) = 2+n(n-1)$ при $n \geqslant 1$.

55. Базис: если $n = 1$, то имеются только два куска. Шаг: пусть для n сфер утверждение верно, $N(n) = 2n+\frac{n(n-1)(n-2)}{3}$. Тогда при пересечении $(n+1)$-й сферы с предыдущими на ней образуется n окружностей, удовлетворяющих условию задачи 54. Следовательно, эти окружности разделят $(n+1)$-ю сферу на $2+n(n-1)$ частей. Каждая из этих частей будет делить кусок пространства, в котором располагается, на два, то есть количество кусков возрастёт на $2+n(n-1)$. Поэтому $N(n+1) = N(n)+2+n(n-1) = 2n+\frac{n(n-1)(n-2)}{3}+2+n(n-1) = 2(n+1)+\frac{n(n-1)(n+1)}{3}$.

56. Базис: если $T = 0$, то все бактерии имеют $t = 0$, следовательно, они не могут делиться, на каждом шаге какая-то из них будет погибать, их количество будет уменьшаться на 1, и рано или поздно все они погибнут. Шаг $(*)$: пусть для всех $T' < T$ утверждение доказано, докажем его для

T. Применим теперь индукцию по количеству n бактерий, для которых $t = T$. Базис: $n = 1$, то есть имеется только одна бактерия b, способная делиться T раз. Так как остальные бактерии могут делиться меньше чем T раз, то они не могут существовать бесконечно долго по индукционному предположению $(*)$. Следовательно, рано или поздно бактерия b должна будет делиться. Но после этого для всех бактерий максимальное количество делений будет меньше T, поэтому по индукционному предположению $(*)$ колония вымрет. Шаг $(**)$: пусть для n бактерий с количеством делений T утверждение доказано, докажем для $n+1$. Как и в базисе остальные бактерии, способные к делению менее T раз, не могут существовать бесконечно долго по предположению $(*)$. Следовательно, когда-то одна из бактерий с $t = T$ должна будет делиться. После этого количество таких бактерий станет равно n, поэтому колония вымрет по предположению $(**)$.

57. (a) Базис: при $n = 0$ получаем

$$F_0 = 0 = \frac{1}{\sqrt{5}}[1-1] = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^0 - \left(\frac{1-\sqrt{5}}{2}\right)^0\right];$$

при $n = 1$ получаем

$$F_1 = 1 = \frac{1}{\sqrt{5}}\left[\frac{1+\sqrt{5}}{2} - \frac{1-\sqrt{5}}{2}\right].$$

Шаг: пусть для всех $j < n$ утверждение верно. Тогда

$$F_n = F_{n-1} + F_{n-2} = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n-1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n-1}\right] +$$

$$\frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n-2} - \left(\frac{1-\sqrt{5}}{2}\right)^{n-2}\right] =$$

$$\frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n-1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n-1} + \left(\frac{1+\sqrt{5}}{2}\right)^{n-2} - \left(\frac{1-\sqrt{5}}{2}\right)^{n-2}\right] =$$

$$\frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n-2}\left(\frac{1+\sqrt{5}}{2}+1\right) - \left(\frac{1-\sqrt{5}}{2}\right)^{n-2}\left(\frac{1-\sqrt{5}}{2}+1\right)\right] =$$

$$\frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n-2}\left(\frac{1+2\sqrt{5}+5}{4}\right) - \left(\frac{1-\sqrt{5}}{2}\right)^{n-2}\left(\frac{1-2\sqrt{5}+5}{4}\right)\right] =$$

$$\frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n-2}\left(\frac{1+\sqrt{5}}{2}\right)^{2}-\left(\frac{1-\sqrt{5}}{2}\right)^{n-2}\left(\frac{1-\sqrt{5}}{2}\right)^{2}\right]=$$
$$\frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n}-\left(\frac{1-\sqrt{5}}{2}\right)^{n}\right].$$

(б) Базис: при $n=0$ получаем равенство $0=0=1-1=F_2-1$. Шаг: пусть для n утверждение верно, тогда $F_1+F_2+F_3+\cdots+F_n+F_{n+1}=(F_{n+2}-1)+F_{n+1}=F_{n+3}-1$.

(в) Базис: при $n=1$ получаем равенство $1\cdot F_1=1=5-1-3=F_5-1-3$. Шаг: пусть для n утверждение верно, тогда $(n+1)\cdot F_1+\cdots+2\cdot F_n+1\cdot F_{n+1}=(n\cdot F_1+\ldots+1\cdot F_n)+(F_1+\cdots+F_{n+1})=(F_{n+4}-n-3)+(F_{n+3}-1)=F_{n+5}-(n+1)-3$.

(г) Базис: при $n=1$: $F_{1-1}F_{1+1}=0\cdot1=0=1-1=F_1^2+(-1)^1$. Шаг: если для n доказано, то $F_{n+1-1}F_{n+1+1}=F_nF_{n+2}=F_n(F_{n+1}+F_n)=F_nF_{n+1}+F_n^2=F_nF_{n+1}+F_{n-1}F_{n+1}-(-1)^n=(F_n+F_{n-1})F_{n+1}-(-1)^n=F_{n+1}^2+(-1)^{n+1}$.

(д) Базис: числа $F_1=1$ и $F_2=1$ взаимно просты. Шаг: пусть для всех $j<n$ утверждение доказано и $n\geqslant 2$. Если бы числа F_n и F_{n+1} имели общий множитель $d>1$: $F_n=dF'_n$ и $F_{n+1}=dF'_{n+1}$, то получили бы $F_{n-1}=F_{n+1}-F_n=d(F'_{n+1}-F'_n)$, то есть F_{n-1} и F_n тоже имели бы общий множитель $d>1$, что невозможно по индукционному предположению.

(е) Базис: $F_0=0$ — чётное, $F_1=F_2=1$ — нечётные. Если для всех $j<n$ доказано и n делится на 3, то $n-1$ и $n-2$ не делятся на 3, F_{n-1} и F_{n-2} — нечётные, F_n — чётное. Если n не делится на 3, то одно и только одно из чисел $n-1$ или $n-2$ делится на 3, поэтому одно из чисел F_{n-1}, F_{n-2} чётное, другое — нечётное, следовательно, F_n — нечётное.

(ж) Индукция по m. Базис: $m=0$, $F_{n+0}=F_n=F_{n-1}\cdot0+F_n\cdot1=F_{n-1}\cdot F_0+F_n\cdot F_1$; $m=1$, $F_{n+1}=F_{n-1}+F_n=F_{n-1}\cdot1+F_n\cdot1=F_{n-1}\cdot F_1+F_n\cdot F_2$. Шаг: пусть для меньших m чисел утверждение верно. $F_{n+m}=F_{n+m-2}+F_{n+m-1}=F_{n-1}F_{m-2}+F_nF_{m-1}+F_{n-1}F_{m-1}+F_nF_m=F_{n-1}(F_{m-2}+F_{m-1})+F_n(F_{m-1}+F_m)=F_{n-1}F_m+F_nF_{m+1}$.

58. Базис: $n=1$, $F_2=1$, $F_1=1$, $F_0=0$. Шаг: пусть для n утверждение верно, тогда

$$\begin{pmatrix}1&1\\1&0\end{pmatrix}^{n+1}=\begin{pmatrix}1&1\\1&0\end{pmatrix}^{n}\cdot\begin{pmatrix}1&1\\1&0\end{pmatrix}=\begin{pmatrix}F_{n+1}&F_n\\F_n&F_{n-1}\end{pmatrix}\cdot\begin{pmatrix}1&1\\1&0\end{pmatrix}=$$
$$\begin{pmatrix}F_{n+1}\cdot1+F_n\cdot1&F_{n+1}\cdot1+F_n\cdot0\\F_n\cdot1+F_{n-1}\cdot1&F_n\cdot1+F_{n-1}\cdot0\end{pmatrix}=\begin{pmatrix}F_{n+2}&F_{n+1}\\F_{n+1}&F_n\end{pmatrix}.$$

Отсюда следует другое решение предыдущей задачи, пункт (г). Из равенства матриц получается равенство определителей, то есть $F_{n+1}F_{n-1} - F_n^2 = (-1)^n$.

59. Обозначим через A_n количество таких последовательностей, которые заканчиваются на единицу, а через B_n — количество таких последовательностей, которые не заканчиваются на единицу. Тогда $A_0 = 0$, $B_0 = 1$ (существует пустая последовательность), $A_1 = 1$, $B_1 = 1$. Чтобы получить последовательность длины $n + 1$, нужно приписать один символ в конец последовательности длины n, при этом если на конце стоит единица, то нельзя добавлять единицу, в противном случае можно добавить любой символ. Поэтому $A_{n+1} = B_n$, $B_{n+1} = A_n + B_n$. Отсюда $B_0 = 1$, $B_1 = 1$, $B_{n+1} = B_{n-1} + B_n$, поэтому $B_n = F_{n+1}$, $A_n = B_{n-1} = F_n$, а общее количество последовательностей равно $A_n + B_n = F_{n+1} + F_n = F_{n+2}$.

60. **(а)** Доказательство индукцией по n. Базис: один диск можно переложить за одну операцию. Шаг: пусть для n дисков найден способ переноса с одного стержня на другой не более чем за $2^n - 1$ операций. Тогда $n+1$ диск можно переместить следующим образом: переносим n верхних дисков на третий стержень ($2^n - 1$ операций), переносим самый большой диск на второй (одна операция), переносим n дисков с третьего стержня на второй ($2^n - 1$ операций). Общее количество операций равно $(2^n - 1) + 1 + (2^n - 1) = 2^{n+1} - 1$.

(б) Доказательство индукцией по n. Базис: переместить один диск меньше чем за одну операцию невозможно. Шаг: пусть доказано, что n дисков нельзя переместить с одного стержня на другой менее чем за $2^n - 1$ операций. Рассмотрим $n + 1$ диск. Перенести самый большой диск с одного стержня на другой возможно, только если эти два стержня ничего другого не содержат, то есть все остальные диски находятся на третьем стержне. По индукционному предположению, чтобы перенести все остальные диски на третий стержень, нужно не менее $2^n - 1$ операций. Далее требуется переложить самый большой диск (одна операция) и переложить оставшиеся n дисков на него (не менее $2^n - 1$ операций). Поэтому общее количество операций оказывается не меньше чем $2^{n+1} - 1$.

(в) Согласно пункту (б) нужно не менее $2^{64} - 1$ операций, чтобы перенести 64 диска. Предположим, что на перекладывание одного диска тратится одна секунда. Оценим 2^{64} секунд снизу. Один год — это не более $366 \cdot 24 \cdot 60 \cdot 60 = 31622400$ секунд, то есть менее $2^{25} = 33554432$ секунд. Следовательно, 2^{64} секунд — это более $2^{64}/2^{25} = 2^{39}$ лет. Получаем $2^{39} = 2^{40}/2 \geqslant 10^{12}/2$, мы воспользовались неравенством $2^{10} = 1024 > 1000 = 10^3$. Итак, мир просуществует не менее 500 миллиардов лет. А согласно современным данным возраст Вселенной составляет около 14 млрд. лет. Поэтому ожидать конца света в ближайшем будущем не стоит.

61. Базис. Наименьшее значение $x + y$ равно d при $x = d, y = 0$ или $x = 0, y = d$. В этом случае $d = 1x + 0y$ или $d = 0x + 1y$ соответственно. Индукционный шаг. Допустим, что для всех x', y', когда $x' + y' < x + y$, утверждение доказано и $y \leqslant x$. Если $x = y$, то НОД$\{x, y\} = x = d$, откуда $x = y = d$ и $d = 1x + 0y$. Иначе пусть $y < x$. Если бы $y = 0$, то НОД$\{x, y\} = x = d$ и мы бы получили базисный случай. Пусть теперь $y > 0$, тогда $(x - y) + y = x < x + y$, следовательно, по индукционному предположению НОД$\{x - y, y\} = a'(x - y) + b'y$ для каких-то целых a' и b'. Получаем $d = $ НОД$\{x, y\} = $ НОД$\{x - y, y\} = a'(x - y) + b'y = a'x + (b' - a')y$.

Элементы комбинаторики

62. **(а)** 3^{3^2}; **(б)** 3^{3^3}; **(в)** 2^{3^3}.

63. **(а)** $C_8^3 = 56$;

(б) $C_7^4 = 35$, к всевозможным четырёхэлементным подмножествам из букв $A \setminus \{b\}$ добавляем b;

(в) $C_7^5 = 21$, всевозможные пятиэлементные подмножества из букв $A \setminus \{b\}$;

(г) $C_5^4 = 5$, к всевозможным четырёхэлементным подмножествам из букв $A \setminus \{c, f, h\}$ добавляем c.

(д) Всего существует $2^8 = 256$ подмножеств множества A. Меньше трёх элементов содержат $C_8^0 + C_8^1 + C_8^2 = 37$ подмножеств. Поэтому хотя бы три элемента содержат $256 - 37 = 219$ подмножеств.

(е) Более шести элементов содержат $C_8^7 + C_8^8 = 9$ подмножеств. Поэтому не более шести элементов содержат $256 - 9 = 247$ подмножеств.

64. Последовательность однозначно определяется множеством чисел, которые входят в последовательность между первым и последним элементами. Поэтому (а) 2^5; (б) 2^6; (в) 2^{m-1}.

65. Выбрать элемент из множества A_i можно $k_i + 1$ способами (имеется k_i элементов, но можно не брать ни один из них). Всего существует $(k_1 + 1)(k_2 + 1) \ldots (k_n + 1)$ способов.

66. Если q — делитель p, то $q = p_1^{i_1} p_2^{i_2} \ldots p_r^{i_r}$, где $i_j \leqslant k_j$. Согласно предыдущей задаче существует $\prod\limits_{i=1}^{r} (k_i + 1)$ делителей. Число n является квадратом тогда и только тогда, когда все показатели степеней k_i чётны. А это имеет место тогда и только тогда, когда $t(n)$ нечётно.

67. Суммируя по всем m и подставляя значение A_n^m, получаем $A_n = \sum\limits_{m=0}^{n} A_n^m = \sum\limits_{m=0}^{n} n!/(n-m)! = n! \sum\limits_{m=0}^{n} 1/(n-m)! = n! \sum\limits_{m=0}^{n} 1/m! = e_n n!$. Так как e_n стремится к e при возрастании n, то при больши́х n получаем $A_n \approx en!$.

68. (а) $C_{n-1}^k+C_{n-1}^{k-1}=(n-1)!/k!(n-1-k)!+(n-1)!/(k-1)!(n-1-(k-1))!=(n-1)!(n-k)/k!(n-k)!+(n-1)!k/k!(n-k)!=(n-1)!(n-k+k)/k!(n-k)!=n!/k!(n-k)!=C_n^k$;
(б) $nC_{n-1}^{k-1}=n\cdot(n-1)!/(k-1)!(n-1-(k-1))!=n!/(k-1)!(n-k)!=n!k/k!(n-k)!=kC_n^k$;
(в) $C_n^kC_{n-k}^{m-k}=n!/k!(n-k)!\cdot(n-k)!/(m-k)!(n-k-(m-k))!=n!/k!(m-k)!(n-m)!=n!m!/k!m!(m-k)!(n-m)!=m!/k!(m-k)!\cdot n!/m!(n-m)!=C_m^kC_n^m$.

69. Базис: для $n=0$: $(x+y)^0=1=C_0^0x^0y^0$. Шаг: если для n утверждение уже доказано, то для $n+1$ получаем $(x+y)^{n+1}=(x+y)^n(x+y)=(\sum\limits_{i=0}^{n}C_n^ix^iy^{n-i})(x+y)=(\sum\limits_{i=0}^{n}C_n^ix^{i+1}y^{n-i})+(\sum\limits_{i=0}^{n}C_n^ix^iy^{n-i+1})=(\sum\limits_{i=0}^{n+1}C_n^{i-1}x^{i-1+1}y^{n-(i-1)})+(\sum\limits_{i=0}^{n+1}C_n^ix^iy^{n+1-i})=(\sum\limits_{i=0}^{n+1}C_n^{i-1}x^iy^{n+1-i})+(\sum\limits_{i=0}^{n+1}C_n^ix^iy^{n+1-i})=\sum\limits_{i=0}^{n+1}(C_n^{i-1}+C_n^i)x^iy^{n+1-i}=\sum\limits_{i=0}^{n+1}C_{n+1}^ix^iy^{n+1-i}$.

70. $C_n^{i+1}/C_n^i={}^{n!i!(n-i)!}/{}_{n!(i+1)!(n-i-1)!}={}^{(n-i)}/{}_{(i+1)}$; при $i<\lfloor{}^n/{}_2\rfloor$ получаем ${}^{(n-i)}/{}_{(i+1)}>1$, поэтому $C_n^{i+1}/C_n^i>1$ и $C_n^i<C_n^{i+1}$; при $i\geqslant\lceil{}^n/{}_2\rceil$ получаем ${}^{(n-i)}/{}_{(i+1)}<1$, поэтому $C_n^{i+1}/C_n^i<1$ и $C_n^i>C_n^{i+1}$.

71. Всего должно быть сделано последовательно $x+y$ шагов, из них x — горизонтальных, остальные — вертикальные. Общее количество вариантов составляет C_{x+y}^x.

72. (а) $C_7^3\cdot 2^4=560$; **(б)** $C_6^3\cdot 5^3\cdot(-1)^3=-2500$.

73. Произведение содержит 15 множителей $1+x^2+x^5$. Получить x^{20} можно тремя способами. Первый способ: x^{20} равно произведению четырёх множителей x^5. Нужно взять четыре скобки, из которых выбирается x^5, из оставшихся выбирается 1, есть C_{15}^4 вариантов. Второй способ: x^{20} равно произведению двух множителей x^5 и пяти множителей x^2. Нужно взять две скобки, из которых выбирается x^5, а из оставшихся — пять скобок, из которых выбирается x^2, есть $C_{15}^2\cdot C_{13}^5$ вариантов. Третий способ: x^{20} равно произведению десяти множителей x^2. Нужно взять десять скобок, из которых выбирается x^2, есть C_{15}^{10} вариантов. Так как все коэффициенты многочлена $1+x^2+x^5$ равны 1, то коэффициент при x^{20} равен $C_{15}^4+C_{15}^2\cdot C_{13}^5+C_{15}^{10}=139503$.

74. Базис: $k=1$, тогда существует одно разбиение, $n_1=n$ и ${}^{n!}/{}_{n_1!}=1$. Если для k доказано, то любое разбиение на $k+1$ множество можно получить так: сначала отделить $(k+1)$-е множество от остального, а затем разбить остаток на k множеств. Количество вариантов для $(k+1)$-го:

$C_n^{n_{k+1}} = {}^{n!}/_{n_{k+1}!(n-n_{k+1})!}$, количество вариантов для оставшегося разбиения по индукционному предположению равно ${}^{(n-n_{k+1})!}/_{n_1!\cdots n_k!}$. Всего вариантов: ${}^{n!}/_{n_{k+1}!(n-n_{k+1})!} \cdot {}^{(n-n_{k+1})!}/_{n_1!\cdots n_k!} = {}^{n!}/_{n_1!\cdots n_k! n_{k+1}!}$.

75. Нужно выбрать в слове позиции для каждой из букв, применяем задачу 74: ${}^{11!}/_{5!2!2!} = 83160$.

76. Произведение содержит n множителей $x_1 + x_2 + \cdots + x_k$. Чтобы получить $x_1^{n_1}x_2^{n_2}\ldots x_k^{n_k}$, нужно взять n_i скобок, из которых выбирается x_i, для $i = 1, 2, \ldots, k$. Коэффициент при $x_1^{n_1}x_2^{n_2}\ldots x_k^{n_k}$ равен количеству способов такого выбора, согласно задаче 74 это количество равно $\frac{n!}{n_1!n_2!\ldots n_k!}$.

77. **(а)** $C_{52}^{13} \cdot C_{39}^{13} \cdot C_{26}^{13} \cdot C_{13}^{13} = {}^{52!}/_{(13!)^4}$;
(б) ${}^{32!}/_{(10!)^3}$; **(в)** ${}^{36!}/_{(6!)^4}$;
(г) (i) ${}^{N!}/_{(r!)^k(N-kr)!}$; (ii) предыдущее умножаем на количество способов упорядочить прикуп: ${}^{N!}/_{(r!)^k(N-kr)!} \cdot (N-kr)! = {}^{N!}/_{(r!)^k}$.

78. Нужно найти наименьшее n такое, что $C_n^3 \geqslant 20$. Следовательно, $n = 6$: $C_6^3 = 20$.

79. Существует C_n^5 пятиэлементных подмножеств и C_{n-1}^4 пятиэлементных подмножеств, содержащих 7. По условию имеем $C_n^5 = 4C_{n-1}^4$. Решая уравнение, находим $n = 20$.

80. 2^{30} вариантов: (а)–(г) да, (д) нет.

81. Индукция по m. Базис, при $m = 0$: $C_{n+0}^k = C_n^k C_0^0 = \sum\limits_{i=0}^{k} C_n^i C_0^{k-i}$, так как при $k \neq i$ будет $C_0^{k-i} = 0$. Шаг: если для m доказано, то $C_{n+(m+1)}^k = C_{n+m}^k + C_{n+m}^{k-1} = \sum\limits_{i=0}^{k} C_n^i C_m^{k-i} + \sum\limits_{i=0}^{k-1} C_n^i C_m^{k-1-i} = C_n^k C_m^0 + \sum\limits_{i=0}^{k-1} C_n^i C_m^{k-i} + \sum\limits_{i=0}^{k-1} C_n^i C_m^{k-1-i} = C_n^k C_{m+1}^0 + \sum\limits_{i=0}^{k-1} C_n^i (C_m^{k-i} + C_m^{k-1-i}) = C_n^k C_{m+1}^{k-k} + \sum\limits_{i=0}^{k-1} C_n^i C_{m+1}^{k-i} = \sum\limits_{i=0}^{k} C_n^i C_{m+1}^{k-i}$.
Другой способ доказательства: обе части этого равенства задают количество вариантов выбора k человек из группы, состоящей из n женщин и m мужчин (правая часть равенства — всевозможные варианты распределения количеств мужчин и женщин в этой группе).

82. Базис: $n = 0$, $\rho(k) \subseteq \rho(0)$ выполнено только при $k = 0$, при этом $C_0^0 = 1$ и $C_0^k = 0$ при других k. Аналогично для $n = 1$, $\rho(k) \subseteq \rho(1)$ выполнено при $k \in \{0, 1\}$, при этом $C_1^0 = C_1^1 = 1$ и $C_1^k = 0$ при других k. Шаг: пусть для всех чисел меньших n доказано и $n = 2^u + m$, где $1 \leqslant m \leqslant 2^u$. Используем тождество Коши: $C_n^k = C_{2^u+m}^k = \sum\limits_{i=0}^{k} C_{2^u}^i C_m^{k-i}$. Так как 2^u и m меньше n, то для них можно пользоваться индукционным предположением. Если $k < 2^u$, то условие $\rho(i) \subseteq \rho(2^u)$ выполнено только при $i = 0$, поэтому единственное слагаемое, которое может быть нечётным, — это $C^0_{2^u}C^k_m = C^k_m$. Оно нечётно тогда и только тогда, когда $\rho(k) \subseteq \rho(m)$, то есть $\rho(k) \subseteq \rho(n)$, поскольку единственный элемент $\rho(n) \setminus \rho(m)$ равен u и в $\rho(k)$ не содержится. Если $k = 2^u$, то условие $\rho(i) \subseteq \rho(2^u)$ выполнено при $i = 0$ и при $i = 2^u$, поэтому есть только два слагаемых, которые могут быть нечётными, — это $C^0_{2^u}C^k_m = C^k_m$ и $C^{2^u}_{2^u}C^0_m = 1$. Второе нечётно всегда, а первое — при $m = 2^u$. Таким образом, для $k = 2^u$ число C^k_n чётно при $n = 2^{u+1}$ и нечётно при $n < 2^{u+1}$. Последнее условие эквивалентно $\rho(k) \subseteq \rho(n)$. Если $k > 2^u$, то условие $\rho(i) \subseteq \rho(2^u)$ также выполнено при $i = 0$ и при $i = 2^u$, поэтому есть только два слагаемых, которые могут быть нечётными, — это $C^0_{2^u}C^k_m = C^k_m$ и $C^{2^u}_{2^u}C^\ell_m = C^\ell_m$, где $\ell = k - 2^u$. Но $m \leqslant 2^u < k$, поэтому первое из них равно нулю, следовательно, C^k_n нечётно тогда и только тогда, когда нечётно C^ℓ_m, что по индукционному предположению эквивалентно $\rho(\ell) \subseteq \rho(m)$, то есть $\rho(k) = \rho(\ell) \cup \{u\} \subseteq \rho(m) \cup \{u\} = \rho(n)$.

83. Если мультимножество содержит k_i элементов a_i для $i = 1, \ldots, n$, то его можно однозначно представить в виде слова в алфавите $\{0, 1\}$ следующим образом: $0^{k_1}10^{k_2}1 \ldots 0^{k_{n-1}}10^{k_n}$. Оно имеет длину $n + k - 1$ и содержит $n - 1$ единицу. Таких слов существует $C^{n-1}_{n+k-1} = C^k_{n+k-1}$.

84. Каждое решение можно представить в виде 10-элементного мультимножества из букв x, y, z, u, количество которых является значением соответствующей переменной. С помощью задачи 83 получаем $C^{10}_{4+10-1} = C^3_{13} = 286$.

85. Если $x_i = m + y_i$, то $y_1 + \cdots + y_n = k - nm$, где y_i — любые натуральные числа. Аналогично предыдущей задаче получаем $C^{k-nm}_{n+k-nm-1}$.

86. Если $x_i = m - y_i$, то $y_1 + \cdots + y_n = nm - k$, где y_i — любые натуральные числа. Так как $nm - k \leqslant m$, то автоматически получаем $x_i = m - y_i \geqslant 0$ для любого решения. Аналогично предыдущей задаче количество решений равняется $C^{nm-k}_{n+nm-k-1}$.

87. Каждая партия должна иметь хотя бы одно место, иначе одна из двух других получит больше половины. Пусть i-я партия получает $200 - x_i$ мест, тогда $x_i \geqslant 0$ и $x_i \leqslant 199$. Значит, всего они получают $(200 - x_1) + (200 - x_2) + (200 - x_3) = 401$ место, иначе говоря, $x_1 + x_2 + x_3 = 199$. Аналогично задаче 84 находим количество решений такого уравнения: $C^{199}_{3+199-1} = C^2_{201}$.

88. Покупка — мультимножество, составленное из элементов четырёх видов, используем задачу 83:

(а) C^6_{4+6-1};

(б) C^7_{4+7-1};

(в) C^3_{4+3-1}, покупаем по одному пирожному каждого вида, а потом — три любых.

89. $A_{16}^3 \cdot C_{13}^2$.

90. Поставить белые шары, а затем чёрные на $n-m+1$ место между ними и по краям: C_{n-m+1}^m.

91. Левая часть равенства — это общее количество способов расставить чёрные шары между n белыми так, чтобы никаких два чёрных шара не оказались рядом. Согласно задаче 59 это количество равно F_{n+2}.

92. Аналогично предыдущей задаче: C_{n-m-1}^m, только крайние элементы выбирать не допускается.

93. Если пронумеровать места за столом натуральными числами $1, \ldots, n$, то требуется из n выбрать k мест так, чтобы не были выбраны одновременно первое и последнее, а также — любые два числа подряд. Используем задачу 90: если выбрать 1, то останется выбрать $k-1$ число из $3, \ldots, n-1$, то есть C_{n-k-1}^{k-1}. Если 1 не выбирать, то останется выбрать k чисел из $2, \ldots, n$, то есть C_{n-k}^k. Итого: $C_{n-k-1}^{k-1} + C_{n-k}^k$.

94. Взять в качестве множества объектов $\bigcup\limits_{i=1}^{n} A_i$, а в качестве p_i — свойство «не принадлежать A_i». Тогда $\left|\bigcap\limits_{i\in I} A_i\right| = N\{p_i : i \in I\}$ и $m' = 0$. Применяя принцип включения и исключения, получаем $0 = \sum\limits_{I\subseteq\{1,\ldots,n\}} (-1)^{|I|} \left|\bigcap\limits_{i\in I} A_i\right|$. Выделяя из суммы слагаемое для $I = \varnothing$, получаем требуемое.

95. $100 - \lfloor {}^{100}/_{2} \rfloor - \lfloor {}^{100}/_{3} \rfloor - \lfloor {}^{100}/_{5} \rfloor + \lfloor {}^{100}/_{6} \rfloor + \lfloor {}^{100}/_{10} \rfloor + \lfloor {}^{100}/_{15} \rfloor - \lfloor {}^{100}/_{30} \rfloor = 26$. В первой тысяче их 266.

96. $x + y = 11 - z$, $x + y \geqslant 4$, $4 - x \leqslant y \leqslant 3$. При $x = 1$ — одно решение, при $x = 2$ — два, при $x = 3$ — три, при $x = 4$ — четыре. Всего — десять.

97. Общее количество перестановок $n!$. Количество перестановок, при котором выбранные k элементов остаются неподвижными: $(n-k)!$. Получаем: $n! - C_n^1(n-1)! + C_n^2(n-2)! - C_n^3(n-3)! + \ldots = n! - {}^{n!}/_{1!(n-1)!} \cdot (n-1)! + {}^{n!}/_{2!(n-2)!} \cdot (n-2)! - {}^{n!}/_{3!(n-3)!} \cdot (n-3)! + \ldots = n! - {}^{n!}/_{1!} + {}^{n!}/_{2!} - {}^{n!}/_{3!} + \ldots = n!(1 - {}^{1}/_{1!} + {}^{1}/_{2!} - {}^{1}/_{3!} + \ldots) \approx {}^{n!}/_{e}$.

98. **(а)** $8! = 40320$.

(б) Существует $6! = 720$ способов переставить буквы $\{f, m, p, r, t, x\}$. Подсчитаем количество способов вставить буквы a и c так, чтобы между ними были две буквы. Имеется семь позиций для вставки. Первая вставляемая буква может занимать позиции с первой по пятую. С учётом того, что буквы a и c можно вставить в разном порядке, получаем 10 способов вставки. Поэтому количество перестановок равно 7200. Аналогично получаем, что количество перестановок, в которых между a и c стоят три буквы равно

5760. Значит, общее количество перестановок равно 12960.
(в) $8! - 12960 = 27360$.
(г) $4! \cdot 4! = 576$.
(д) Существует $4! = 24$ перестановки букв $\{a, c, f, r\}$. Будем рассматривать блок из этих символов как целую букву. Тогда существует 5! перестановок, если не учитывать порядок букв $\{a, c, f, r\}$. С учётом порядка получается $5! \cdot 4! = 2880$ перестановок.

99. Любой вариант можно однозначно закодировать словом в алфавите $\{1, 2\}$, каждый символ в котором означает победу соответствующей команды в очередной партии. Найдём количество слов, соответствующих победе первой команды; одно слово длины 3: 111, $C_3^1 = 3$ слова длины 4 (размещаем одну двойку в слове 111 не на последнем месте), $C_4^2 = 6$ слов длины 5 (размещаем две двойки в слове длины 5 не на последнем месте). Итого 10 вариантов при выигрыше первой команды. Столько же при выигрыше второй, всего 20.

100. $A_n^3 \cdot C_{n-3}^5$, выбрать трёх призёров и ещё пять лауреатов.

101. Существует $n!$ способов поставить n девушек в ряд. Из любого хоровода можно получить n рядов, разбивая хоровод в n местах. Поэтому существует $n!/n = (n-1)!$ хороводов.

102. **(а)** Аналогично задаче 98 (д) получаем $6! \cdot 5! = 86400$.
(б) Имеется $A_5^2 = 20$ способов выбрать крайних юношей. Остальных людей можно пересадить $8! = 40320$ способами. Всего получается $A_5^2 \cdot 8! = 806400$.
(в) Девушек можно посадить относительно друг друга $5! = 120$ способами. При фиксированном положении девушек есть шесть мест, куда можно посадить юношей, причём на каждое место можно посадить не более одного юноши. Поэтому посадить юношей можно A_6^5 способами. Всего существует $5! \cdot A_6^5 = 86400$ способов.
(г) Существует $6! \cdot 5!$ перестановок, когда все юноши сидят рядом, и столько же перестановок, когда все девушки сидят рядом. В $2 \cdot 5! \cdot 5!$ перестановках и юноши, и девушки сидят рядом. Значит, имеется $2 \cdot 6! \cdot 5! - 2 \cdot 5! \cdot 5! = 144000$ перестановок, когда либо юноши, либо девушки сидят рядом. Поэтому количество перестановок, в которых ни юноши, ни девушки не сидят все вместе равно $10! - 144000 = 3484800$.
(д) Аналогично пункту (а) получаем $9! \cdot 2 = 725760$.
(е) Сначала посадим 8 человек, не участвующих в ограничении. Это можно сделать $8! = 40320$ способами. Оставшихся юношу и девушку можно посадить на 9 мест (между двумя людьми или по краям). При этом их нельзя сажать в одну позицию, чтобы они не оказались рядом. Для этого есть $A_9^2 = 72$ варианта. Всего получается $8! \cdot A_9^2 = 2903040$ способов.

103. Пусть свойство p_a слова w означает, что в w отсутствует буква a. Тогда количество слов длины n, обладающих i различными свойствами p_a, равно $(k-i)^n$. Используем принцип включения и исключения: искомое количество равно $\sum\limits_P (-1)^{|P|}(k-|P|)^n$. Но i свойств можно выбрать C_k^i способами, поэтому, сгруппировав слагаемые с одинаковым $|P|=i$, получаем $\sum\limits_{i=0}^{k-1}(-1)^i C_k^i (k-i)^n$.

104. Есть три способа представить число 8 в виде суммы пяти положительных целых слагаемых. (а) $8=4+1+1+1+1$. Количество вариантов: $(5\cdot C_8^4)\cdot 4!$, выбираем сотрудника, который получит четыре задания, выбираем эти задания, распределяем оставшиеся. (б) $8=3+2+1+1+1$. Количество вариантов: $(5\cdot C_8^3)\cdot(4\cdot C_5^2)\cdot 3!$, выбираем сотрудника, который получит три задания, выбираем эти задания, выбираем сотрудника, который получит два задания, выбираем эти задания, распределяем оставшиеся. (в) $8=2+2+2+1+1$. Количество вариантов: $C_5^3\cdot(C_8^2\cdot C_6^2\cdot C_4^2)\cdot 2!$, выбираем сотрудников, которые получат по два задания, выбираем эти задания для каждого из них, распределяем оставшиеся. Для получения ответа найденные количества нужно просуммировать. Другой способ решения. Закодируем выдачу заданий последовательностью чисел от 1 до 5 длины 8. На i-м месте стоит число j, если i-е задание досталось сотруднику j. Нужно найти количество слов длины 8 в алфавите $\{1,2,3,4,5\}$, в которых каждая буква встречается хотя бы один раз. Согласно задаче 103 существует $U^*(8,5)$ таких слов.

Логика высказываний и булевы функции

105. Индукция по n для всех k сразу. Базис: $n=0$. Есть одна нульмерная грань (единственная точка), нет никаких других. Шаг: пусть $F(n,k)=C_n^k\cdot 2^{n-k}$. При построении $(n+1)$-мерного куба количество k-мерных граней удваивается, а также каждая $(k-1)$-мерная грань даёт одну k-мерную. Следовательно, $F(n+1,k)=2F(n,k)+F(n,k-1)=2C_n^k\cdot 2^{n-k}+C_n^{k-1}\cdot 2^{n-(k-1)}=C_n^k\cdot 2^{n-k+1}+C_n^{k-1}\cdot 2^{n-k+1}=(C_n^k+C_n^{k-1})\cdot 2^{n-k+1}=C_{n+1}^k\cdot 2^{n+1-k}$. Общее количество граней $F(n)=\sum\limits_{k=0}^{n}F(n,k)=\sum\limits_{k=0}^{n}C_n^k 2^{n-k}=(1+2)^n=3^n$. Другой способ. В координаты вершин k-мерной грани могут принимать такие значения: k координат изменяются от 0 до 1, а остальные $n-k$ фиксированы. Существует $C_n^{n-k}=C_n^k$ способов выбрать фиксированные координаты. Каждая такая координата равна либо 0, либо 1, поэтому получается 2^{n-k} вариантов для одного набора. Следовательно, общее количество k-мерных граней равно $C_n^k \cdot 2^{n-k}$. Общее количество всех граней равно 3^n, так как для каждой координаты в грани есть три варианта: равна 0, равна 1, меняется от 0 до 1.

106. **(а)** Неравенство $H(\bar{a}, \bar{b}) \geqslant 0$ очевидно. Если $\bar{a} = \bar{b}$, то $|a_i - b_i| = 0$ для всех i и $H(\bar{a}, \bar{b}) = 0$. Наоборот, сумма неотрицательных чисел равна нулю, только если все слагаемые равны нулю, то есть $|a_i - b_i| = 0$ для всех i и $\bar{a} = \bar{b}$.

(б) Следует из того, что $|a_i - b_i| = |b_i - a_i|$.

(в) Из неравенства $|x + y| \leqslant |x| + |y|$ получаем $|a_i - b_i| = |a_i - c_i + c_i - b_i| \leqslant |a_i - c_i| + |c_i - b_i|$, поэтому $H(\bar{a}, \bar{b}) = \sum\limits_{i=1}^{n} |a_i - b_i| \leqslant \sum\limits_{i=1}^{n} |a_i - c_i| + \sum\limits_{i=1}^{n} |c_i - b_i| = H(\bar{a}, \bar{c}) + H(\bar{c}, \bar{b})$.

107. **(а)** $H(\bar{a}, \bar{b})$ — это количество несовпадающих координат точек $\bar{a}$ и $\bar{b}$. Поскольку при перемещении по одному ребру изменяется ровно одна координата, то количество шагов не может быть меньше k. Существование пути длины k доказывается индукцией по k. При $k = 0$ вершины совпадают и никуда идти не нужно. Если $H(\bar{a}, \bar{b}) = k + 1$, то выберем одну из несовпадающих координат i и построим набор $\bar{c}$, заменив в $\bar{a}$ элемент a_i на b_i. Теперь сделаем шаг из точки $\bar{a}$ в соседнюю точку $\bar{c}$. Тогда $H(\bar{c}, \bar{b}) = k$, и по индукционному предположению существует путь из $\bar{c}$ в $\bar{b}$ длины k. Поэтому путь из $\bar{a}$ в $\bar{b}$ имеет длину $k + 1$.

(б) Доказательство индукцией по k. Базис: $k = 0$, точки совпадают, существует единственный пустой путь, $k! = 1$. Шаг: пусть для k доказано и $H(\bar{a}, \bar{b}) = k+1$. Тогда набор $\bar{c}$ можно построить $k+1$ способом, а из каждого такого $\bar{c}$ есть $k!$ путей по индукционному предположению. Следовательно, общее их количество равно $(k + 1) \cdot k! = (k + 1)!$.

108. Допустим, что система имеет два решения $\bar{x}$ и $\bar{y}$, которые отличаются в j-й позиции: $x_j = 0 \neq 1 = y_j$. Пусть в последовательности при переходе от $\bar{a}^{(i)}$ к $\bar{a}^{(i+1)}$ в j-й позиции ноль меняется на единицу. Тогда $H(\bar{x}, \bar{a}^{(i+1)}) = H(\bar{x}, \bar{a}^{(i)}) + 1$, $H(\bar{y}, \bar{a}^{(i+1)}) = H(\bar{y}, \bar{a}^{(i)}) - 1$. Если $H(\bar{x}, \bar{a}^{(i)}) = k_i = H(\bar{y}, \bar{a}^{(i)})$, то $H(\bar{x}, \bar{a}^{(i+1)}) \neq H(\bar{y}, \bar{a}^{(i+1)})$, следовательно, наборы $\bar{x}$ и $\bar{y}$ не могут быть решением системы одновременно.

109. **(а)** $(1, 0, 0, 1, 0)$; **(б)** $(0, 0, 1, 1, 1)$;

(в) такой точки нет, сумма первого и последнего расстояния должна быть равна 5;

(г) $(1, 1, 0, 0, 1)$; **(д)** $(0, 1, 0, 1, 0)$;

(е) такой точки нет, как показано в предыдущей задаче, на каждом шаге расстояние должно меняться на 1 в ту или другую сторону.

110. **(а)** {(0, 0, 0, 0), (0, 0, 0, 1), (0, 0, 1, 0), (0, 0, 1, 1), (0, 1, 0, 0), (0, 1, 0, 1), (0, 1, 1, 0), (0, 1, 1, 1)};

x_1	x_2	x_3	RS-триггеры				D-триггер
			T_1	T_2	T_3	T_4	
0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0
0	1	0	0	0	0	0	0
0	1	1	0	0	1	1	1
1	0	0	1	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	0	0	0	1
1	1	1	0	1	0	1	1

Рис. 40: Триггеры.

(б) $\{(0,\ 0,\ 0,\ 1),\ (0,\ 0,\ 1,\ 1),\ (0,\ 1,\ 0,\ 1),\ (0,\ 1,\ 1,\ 1),\ (1,\ 0,\ 0,\ 1),\ (1,\ 0,\ 1,\ 1),\ (1,\ 1,\ 0,\ 1),\ (1,\ 1,\ 1,\ 1)\}$;
(в) $\{(0,\ 0,\ 0,\ 0),\ (0,\ 1,\ 0,\ 0),\ (1,\ 0,\ 0,\ 0),\ (1,\ 1,\ 0,\ 0),\ (0,\ 1,\ 0,\ 1),\ (1,\ 0,\ 0,\ 1),\ (1,\ 1,\ 0,\ 1),\ (0,\ 1,\ 1,\ 0),\ (1,\ 0,\ 1,\ 0),\ (1,\ 1,\ 1,\ 0),\ (1,\ 1,\ 1,\ 1)\}$;
(г) $\mathbb{B}^4 \setminus \{(1,\ 1,\ 0,\ 0),\ (1,\ 1,\ 0,\ 1),\ (1,\ 1,\ 1,\ 0)\}$.

111. **(а)** 0 только в строке $(0,1,0)$;
(б) 1 в строках $(0,0,0)$, $(0,1,1)$, $(1,0,1)$, $(1,1,0)$;
(в) 0 в строках $(0,1,1)$, $(1,0,0)$;
(г) 1 в строках $(0,0,1)$, $(0,1,1)$, $(1,1,0)$, $(1,1,1)$.

112. Рис. 40.

113. **(а)** Глубина 3, рис. 41.
(б) Глубина 3, рис. 42.
(в) Глубина 4, рис. 43.
(г) Глубина 3, рис. 44.
(д) Глубина 3, рис. 45.

114. Рис. 46 и 47. Формула Ψ_2 ложна в интерпретации $I(x) = I(y) = I(z) = 0$. Формула Ψ_4 ложна в интерпретации $I(x) = I(y) = 1$, $I(z) = 0$.

115. Только (д). (а) истинна для цифры 8, (б) и (г) истинны для цифры 3, (в) истинна для цифры 2, (е) ложно для ошибочного кода $(1,1,1,1)$.

116. **(а)** $\neg A \wedge B \wedge C \wedge \neg D$;
(б) $\neg C \wedge \neg D$;
(в) $((A \leftrightarrow D) \wedge (B \leftrightarrow C)) \vee ((A \leftrightarrow B) \wedge (C \leftrightarrow D))$;
(г) $\neg A \wedge ((\neg B \wedge C) \vee (B \wedge D))$;

x_1	x_2	x_3	$(($ x_1	$\to$	$\neg$	$x_3)$	$\vee$	$(x_2$	$\oplus$	$x_3))$
0	0	0	0	1	1	0	1	0	0	0
0	0	1	0	1	0	1	1	0	1	1
0	1	0	0	1	1	0	1	1	1	0
0	1	1	0	1	0	1	1	1	0	1
1	0	0	1	1	1	0	1	0	0	0
1	0	1	1	0	0	1	1	0	1	1
1	1	0	1	1	1	0	1	1	1	0
1	1	1	1	0	0	1	0	1	0	1

Рис. 41: Таблица истинности Ψ_1.

x_1	x_2	$(\neg$	$(x_1$	$\uparrow$	$x_2)$	$\leftrightarrow$	$(\neg$	x_1	$\wedge$	$x_2))$
0	0	0	0	1	0	1	1	0	0	0
0	1	0	0	1	1	0	1	0	1	1
1	0	0	1	1	0	1	0	1	0	0
1	1	1	1	0	1	0	0	1	0	1

Рис. 42: Таблица истинности Ψ_2.

x_1	x_2	x_3	$((x_2$	$\oplus$	$\neg$	$x_3)$	$\wedge$	$((x_1$	$\vee$	$x_2)$	$\to$	$(x_1$	$\leftrightarrow$	$\neg$	$x_3)))$
0	0	0	0	1	1	0	1	0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	0	0	0	0	1	0	1	0	1
0	1	0	1	0	1	0	0	0	1	1	0	0	0	1	0
0	1	1	1	1	0	1	1	0	1	1	1	0	1	0	1
1	0	0	0	1	1	0	1	1	1	0	1	1	1	1	0
1	0	1	0	0	0	1	0	1	1	0	0	1	0	0	1
1	1	0	1	0	1	0	0	1	1	1	1	1	1	1	0
1	1	1	1	1	0	1	0	1	1	1	0	1	0	0	1

Рис. 43: Таблица истинности Ψ_3.

x_1	x_2	x_3	$((\neg$	x_1	$\wedge$	$x_2)$	$\downarrow$	$(x_1$	$\to$	$\neg$	$x_3))$
0	0	0	1	0	0	0	0	0	1	1	0
0	0	1	1	0	0	0	0	0	1	0	1
0	1	0	1	0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	0	0	1	0	1
1	0	0	0	1	0	0	0	1	1	1	0
1	0	1	0	1	0	0	1	1	0	0	1
1	1	0	0	1	0	1	0	1	1	1	0
1	1	1	0	1	0	1	1	1	0	0	1

Рис. 44: Таблица истинности Ψ_4.

x_1	x_2	x_3	$(((x_1$	$\oplus$	$x_2)$	$\oplus$	$x_3)$	$\uparrow$	$(\neg$	x_2	$\downarrow$	$x_3))$
0	0	0	0	0	0	0	0	1	1	0	0	0
0	0	1	0	0	0	1	1	1	1	0	0	1
0	1	0	0	1	1	1	0	0	0	1	1	0
0	1	1	0	1	1	0	1	1	0	1	0	1
1	0	0	1	1	0	1	0	1	1	0	0	0
1	0	1	1	1	0	0	1	1	1	0	0	1
1	1	0	1	0	1	0	0	1	0	1	1	0
1	1	1	1	0	1	1	1	1	0	1	0	1

Рис. 45: Таблица истинности Ψ_5.

(д) $A \vee (B \wedge \neg D)$;
(е) $\neg(A \vee B) \vee (\neg C \wedge ((A \oplus B) \wedge (A \oplus D)))$.
117. **(а)** $(A_1 \leftrightarrow A_2) \wedge (B_1 \leftrightarrow B_2) \wedge (C_1 \leftrightarrow C_2) \wedge (D_1 \leftrightarrow D_2)$;
(б) $(\neg A_1 \wedge A_2) \vee ((A_1 \leftrightarrow A_2) \wedge ((\neg B_1 \wedge B_2) \vee ((B_1 \leftrightarrow B_2) \wedge ((\neg C_1 \wedge C_2) \vee ((C_1 \leftrightarrow C_2) \wedge (\neg D_1 \wedge D_2))))))$;
(в) $(\neg A_1 \wedge A_2) \vee ((A_1 \leftrightarrow A_2) \wedge ((\neg B_1 \wedge B_2) \vee ((B_1 \leftrightarrow B_2) \wedge ((\neg C_1 \wedge C_2) \vee ((C_1 \leftrightarrow C_2) \wedge (D_1 \to D_2))))))$.
118. $p_D = (D_1 \wedge D_2)$, $p_C = (C_1 \wedge D_2) \oplus (C_2 \wedge D_1)$, $p_B = (((C_1 \wedge \neg D_1) \wedge C_2) \vee ((C_2 \wedge \neg D_2) \wedge C_1))$, $p_A = (((C_1 \wedge D_1) \wedge C_2) \wedge D_2)$.

$x\ y\ z$	$((x \to \neg\, y) \to ((\neg\, y \to z) \to (x \to z)))$
0 0 0	1 1 1 1 0 1 1
0 0 1	1 1 1 1 1 1 1
0 1 0	1 0 1 0 1 1 1
0 1 1	1 0 1 0 1 1 1
1 0 0	1 1 1 1 0 1 0
1 0 1	1 1 1 1 1 1 1
1 1 0	0 0 1 0 1 0 0
1 1 1	0 0 1 0 1 1 1

Рис. 46: Таблица истинности Ψ_1.

$x\ y\ z$	$((x \to \neg\, y) \to ((x \to (\neg\, y \to z)) \to (x \to z)))$
0 0 0	1 1 1 1 1 0 1 1
0 0 1	1 1 1 1 1 1 1 1
0 1 0	1 0 1 1 0 1 1 1
0 1 1	1 0 1 1 0 1 1 1
1 0 0	1 1 1 0 1 0 1 0
1 0 1	1 1 1 1 1 1 1 1
1 1 0	0 0 1 1 0 1 0 0
1 1 1	0 0 1 1 0 1 1 1

Рис. 47: Таблица истинности Ψ_3.

119. **(а)** $(((x_1 \wedge x_2) \vee (x_1 \wedge x_3)) \vee ((x_1 \wedge x_4) \vee (x_2 \wedge x_3))) \vee ((x_2 \wedge x_4) \vee (x_3 \wedge x_4))$.
(б) $(x_1 \vee x_2) \vee (x_3 \vee x_4)$.
(в) Нельзя, так как если все переменные имеют значение 1, то конъюнкция и дизъюнкция тоже дадут значение 1, в то время как функция должна принимать значение 0.
120. **(а)** Две константы; **(б)** две: x_n и $\neg x_n$.
121. $C^k_{2^n}$.
122. Две: значение такой функции на любом наборе $\bar{x}$ однозначно определяется значением функции на нулевом наборе и количеством единиц в $\bar{x}$. Последнее от функции не зависит.

123. Значение функции на произвольном наборе $\bar{x}$ зависит только от количества единиц в $\bar{x}$ и не зависит от их расположения, это количество может быть равно $0, 1, \ldots, n$, поэтому количество функций будет равно 2^{n+1}.
124. $2^{2^{n-1}}$.
125. Существует $2^{n/2}$ парных наборов. Всего $2^{2^n - 2^{n/2}} \cdot 2$.
126. Без связок вообще — три, с помощью 0 и 1 — по одной, с помощью $\neg$ — три, с помощью $\wedge$, $\vee$, $\oplus$, $\leftrightarrow$, $\uparrow$, $\downarrow$ — по $C_3^2 = 3$ (в силу коммутативности), с помощью $\to$ — $A_3^2 = 6$. Всего 32.
127. При записи формул используются знаки: имена функций, $|F|$ штук, имена переменных, n штук, две скобки и запятая, всего $|F| + n + 3$ знаков. Из них можно составить $(|F| + n + 3)^k$ слов длины k и $\frac{(|F|+n+3)^k - 1}{(|F|+n+3)-1}$ слов длины меньшей k (сумма геометрической прогрессии). Таким образом, количество формул, в которых меньше $2^n/n = 2^{n - \log_2 n}$ символов, не превосходит $\frac{(|F|+n+3)^{2^{n-\log_2 n}} - 1}{(|F|+n+3)-1}$. Учитывая равенство $|F| + n + 3 = 2^{\log_2(n+c)}$ для $c = |F| + 3$, получаем, что количество таких формул не превосходит $(2^{\log_2(n+c)})^{2^{n-\log_2 n}} = 2^{2^{n-\log_2 n + \log_2 \log_2(n+c)}}$. Функция $\log_2 n$ растёт намного быстрее $\log_2 \log_2(n + c)$, поэтому для достаточно бóльших n будет выполнено $\varepsilon \log_2 n \geqslant \log_2 \log_2(n+c)$ и количество «коротких» формул не превосходит $2^{2^{n-(1-\varepsilon)\log_2 n}} = 2^{2^n/n^{1-\varepsilon}}$.
128. Количество булевых функций, у которых аргументы $x_{i_1}, \ldots, x_{i_k}$ являются фиктивными, равняется $2^{2^{n-k}}$, так как для каждой группы из 2^k строк с одинаковыми значениями $x_{i_1}, \ldots, x_{i_k}$ будет одно и то же значение f. Количество способов выбора аргументов $x_{i_1}, \ldots, x_{i_k}$ равняется C_n^k. По формуле включений и исключений получаем $\sum\limits_{k=0}^{n} (-1)^k C_n^k 2^{2^{n-k}}$. В частности, при $n = 2$: $2^{2^2} - 2 \cdot 2^{2^1} + 2^{2^0} = 10$; при $n = 3$: $2^{2^3} - 3 \cdot 2^{2^2} + 3 \cdot 2^{2^1} - 2^{2^0} = 218$; при $n = 4$: $2^{2^4} - 4 \cdot 2^{2^3} + 6 \cdot 2^{2^2} - 4 \cdot 2^{2^1} + 2^{2^0} = 64594$.
129. Базис: $\operatorname{depth}(x)^x_\Psi = \operatorname{depth}(\Psi) = 0 + \operatorname{depth}(\Psi) = \operatorname{depth}\Phi + \operatorname{depth}\Psi$; $\operatorname{depth}(y)^x_\Psi = \operatorname{depth}(y) = 0 \leqslant \operatorname{depth}\Phi + \operatorname{depth}\Psi$. Шаг для конъюнкции: $\operatorname{depth}(\Phi_1 \wedge \Phi_2)^x_\Psi = \operatorname{depth}((\Phi_1)^x_\Psi \wedge (\Phi_2)^x_\Psi) = \max\{\operatorname{depth}(\Phi_1)^x_\Psi, \operatorname{depth}(\Phi_2)^x_\Psi\} + 1 \leqslant \max\{\operatorname{depth}(\Phi_1) + \operatorname{depth}(\Psi), \operatorname{depth}(\Phi_2) + \operatorname{depth}(\Psi)\} + 1 = \max\{\operatorname{depth}(\Phi_1), \operatorname{depth}(\Phi_2)\} + 1 + \operatorname{depth}(\Psi) = \operatorname{depth}(\Phi_1 \wedge \Phi_2) + \operatorname{depth}(\Psi)$. Аналогично для остальных булевых связок. $\operatorname{depth}(y)^x_{x\wedge y} = \operatorname{depth} y = 0 < 1 = 0 + 1 = \operatorname{depth}(y) + \operatorname{depth}(x \wedge y)$.
130. a — A корректна, b — B корректна, c — C корректна. $(b \to \neg a) \wedge (b \vee c) \wedge (a \vee c) \wedge (\neg a \to (b \oplus c))$. B ошибочна, единственность установить невозможно.

a b c	$(b \to c) \wedge ((a \vee c) \wedge \neg(a \wedge c)) \wedge (\neg c \to \neg a)$			Итог
0 0 0	1	0	1	0
0 0 1	1	1	1	1
0 1 0	0	0	1	0
0 1 1	1	1	1	1
1 0 0	1	1	0	0
1 0 1	1	0	1	0
1 1 0	0	1	0	0
1 1 1	1	0	1	0

Рис. 48: Таблица истинности из задачи 134.

131. a — «$x < 0$», b — «$y < 0$», c — «$z < 0$». $(c \to (a \vee \neg b)) \wedge (\neg a \vee b) \wedge (b \to (a \oplus c))$. Нельзя.

132. $y \wedge ((x_1 \wedge (x_2 \vee x_3 \vee x_4)) \vee x_2 \wedge (x_3 \vee x_4) \vee x_3 \wedge x_4)$.

133. **(а)** a — «Миша сделает ремонт», b — «Миша сменит обои», c — «Миша покрасит пол», d — «Миша сменит ламинат», e — «Миша сменит дверь», f — «Миша сменит стеклопакет», g — «Миша установит натяжной потолок», h — «Миша сменит люстру», i — «Миша сменит мебель», $a \wedge (b \vee c) \wedge (b \to d \wedge e \wedge f) \wedge (c \to g \wedge h \wedge i) \wedge (d \to i) \wedge \neg(\neg b \wedge \neg i) \wedge \neg(\neg i \wedge \neg h)$.

(б) a — «будет мороз», b — «будет солнечно», c — «пойдёт снег», d — «пойдёт дождь», e — «будет сыро», f — «будет грязно», $(a \to (b \vee (\neg b \wedge c))) \wedge (\neg a \to (b \vee (\neg b \wedge d))) \wedge ((c \vee d) \to e) \wedge (d \to f) \wedge (b \to (\neg e \wedge \neg f))$.

(в) a — «Вова дома», b — «кошка ест», c — «кошка спит», d — «кошка играет», e — «Вова купит корм», f — «Вова купит мебель», g — «Вова купит обои», h — «Вова посетит в зоомагазин», i — «Вова посетит в хозяйственный магазин», j — «Вова поедет на такси», $(a \to (b \vee c)) \wedge (\neg a \to ((b \vee c) \wedge d)) \wedge (b \to e) \wedge (d \to (f \vee g)) \wedge (e \to h) \wedge ((f \vee g) \to i) \wedge ((h \vee i) \to j) \wedge \neg(j \wedge i)$.

134. Пусть переменные a, b, c означают, что A, B, C соответственно является преступником. (а) $(b \to c)$; (б) $((a \vee c) \wedge \neg(a \wedge c))$; (в) $(\neg c \to \neg a)$. Рис. 48. C является преступником в любом случае. Утверждать, что он действовал в одиночку, нельзя, возможно, B был сообщником.

135. Пусть переменные a, b, c означают, что A, B, C соответственно является преступником. Показания A: $\Phi_A = (b \to \neg c)$. Показания B: $\Phi_B = (a \to c)$. Показания C: $\Phi_C = a$. Всё вместе: $(\Phi_A \to \neg\Phi_B) \wedge \neg\Phi_C = ((b \to \neg c) \to \neg(a \to c)) \wedge \neg a$. Рис. 49. Преступники B и C.

a b c	$((b \to \neg c) \to \neg (a \to c)) \wedge \neg a$						Итог
0 0 0	1	1	0	0	1	1	0
0 0 1	1	0	0	0	1	1	0
0 1 0	1	1	0	0	1	1	0
0 1 1	0	0	1	0	1	1	1
1 0 0	1	1	1	1	0	0	0
1 0 1	1	0	0	0	1	0	0
1 1 0	1	1	1	1	0	0	0
1 1 1	0	0	1	0	1	0	0

Рис. 49: Таблица истинности из задачи 135.

ℓ n g	$((\ell \wedge n \to \neg g) \wedge (n \wedge \neg g \to \ell) \wedge (\ell \wedge g \to n))$			Итог
0 0 0	1	1	1	1
0 0 1	1	1	1	1
0 1 0	1	0	1	0
0 1 1	1	1	1	1
1 0 0	1	1	1	1
1 0 1	1	1	0	0
1 1 0	1	1	1	1
1 1 1	0	1	1	0

Рис. 50: Таблица истинности из задачи 136.

136. Пусть переменные ℓ, n, g означают наличие львов, носорогов и жирафов соответственно. Тогда утверждение записывается формулой $((\ell \wedge n \to \neg g) \wedge (n \wedge \neg g \to \ell) \wedge (\ell \wedge g \to n))$. Рис. 50. Да, может (пятая строка).

137. Пусть переменные k, u, v означают наличие карманов, узкие джинсы, вылинявшие соответственно. Тогда формула, описывающая одинаковые джинсы: $(k \vee u \vee (v \wedge \neg k)) \wedge (\neg k \vee (v \wedge u \wedge k)) \wedge (\neg u \vee (\neg v \wedge u \wedge k))$. Рис. 51. На девушках вылинявшие джинсы-клёши без карманов.

138. Пусть переменные m, n, ℓ означают «мы едем в Манчестер», «мы едем из Ньюкасла», «мы остановились в Ливерпуле» соответственно. Тогда фразы отца: $\Phi_O = m$ и $\Psi_O = n$, фразы матери: $\Phi_M = \neg m \wedge n$ и $\Psi_M = \neg \ell \wedge n$, фразы

k u v	$(k \vee u \vee (v \wedge \neg k))$	$\wedge (\neg k \vee (v \wedge u \wedge k))$	$\wedge (\neg u \vee (\neg v \wedge u \wedge k))$	Итог
0 0 0	0	1	1	0
0 0 1	1	1	1	1
0 1 0	1	1	0	0
0 1 1	1	1	0	0
1 0 0	1	0	1	0
1 0 1	1	0	1	0
1 1 0	1	0	1	0
1 1 1	1	1	0	0

Рис. 51: Таблица истинности из задачи 137.

m n ℓ	$(m \vee n)$	$\wedge((\neg m \wedge n)$	$\vee$	$(\neg \ell \wedge n))$	$\wedge (\neg n \vee \ell)$	Итог
0 0 0	0	0	0	0	1	0
0 0 1	0	0	0	0	1	0
0 1 0	1	1	1	1	0	0
0 1 1	1	1	1	0	1	1
1 0 0	1	0	0	0	1	0
1 0 1	1	0	0	0	1	0
1 1 0	1	0	1	1	0	0
1 1 1	1	0	0	0	1	0

Рис. 52: Таблица истинности из задачи 138.

дочери: $\Phi_D = \neg n$ и $\Psi_D = \ell$. Всё вместе: $(\Phi_O \vee \Psi_O) \wedge (\Phi_M \vee \Psi_M) \wedge (\Phi_D \vee \Psi_D) = (m \vee n) \wedge ((\neg m \wedge n) \vee (\neg \ell \wedge n)) \wedge (\neg n \vee \ell)$. Рис. 52. Следовательно, семья едет из Ньюкасла, останавливалась в Ливерпуле, но не едет в Манчестер. Садиться в машину не нужно.

139. Пусть переменные e, o, s, k означают «злоумышленник уехал в экипаже», «свидетель ошибся», «злоумышленник имел сообщника», «у злоумышленника был ключ» соответственно. Тогда вся фраза описывается формулой $((e \wedge \neg o) \vee (\neg e \wedge o)) \wedge (s \to e) \wedge ((\neg s \wedge \neg k) \vee (s \wedge k)) \wedge k$. Из истинности k и второй дизъюнкции получаем, что истинно s. Из истинности s и импликации получаем истинность e. Из истинности e и первой дизъюнкции получаем истинность $\neg o$, то есть ложность o. Следовательно, злоумышленник уехал в экипаже, свидетель не ошибся, злоумышленник имел сообщника, у злоумышленника был ключ.

140. Пусть переменные i, p, b, n означают «инвестиции увеличатся», «возрастут правительственные расходы», «возникнет безработица», «налоги снизятся» соответственно. Тогда (а) $(\neg i \to p \vee b)$; (б) $(\neg p \to n)$; (в) $(n \wedge i \to \neg b)$. Всё утверждение записывается формулой $(\neg i \to p \vee b) \wedge (\neg p \to n) \wedge (n \wedge i \to \neg b) \wedge \neg i$. Вывод сделать нельзя: формула будет истинна в такой интерпретации: $I(n) = I(b) = 1$, $I(i) = I(p) = 0$.

Эквивалентность формул

141. Всегда выполнено $\Phi \wedge \Psi \Rightarrow \Phi$. Если $\Phi \Rightarrow \Psi$, то $\Phi \Rightarrow \Phi \wedge \Psi$. Следовательно, $\Phi \wedge \Psi \equiv \Phi$. Если $\Phi \wedge \Psi \equiv \Phi$, то $\Phi \Rightarrow \Phi \wedge \Psi$. Это означает, что $\Phi \Rightarrow \Psi$. Всегда выполнено $\Psi \Rightarrow \Phi \vee \Psi$. Если $\Phi \Rightarrow \Psi$, то $\Phi \vee \Psi \Rightarrow \Psi$. Следовательно, $\Phi \vee \Psi \equiv \Psi$. Если $\Phi \vee \Psi \equiv \Psi$, то $\Phi \vee \Psi \Rightarrow \Psi$. Последнее означает $\Phi \Rightarrow \Psi$.

143. **(а)** Использовать эквивалентность $\Phi \wedge 0 \equiv 0$.
(б) Использовать эквивалентность $\Phi \vee 1 \equiv 1$.
(в) Использовать эквивалентность $\Phi \wedge 1 \equiv \Phi$.
(г) Использовать эквивалентность $\Phi \vee 0 \equiv \Phi$.

144. **(а)** $\neg(x \vee \neg y) \wedge (x \to \neg y) \equiv (\neg x \wedge \neg\neg y) \wedge (\neg x \vee \neg y) \equiv (\neg x \wedge y) \wedge (\neg x \vee \neg y) \equiv \neg x \wedge y \wedge \neg x \vee \neg x \wedge y \wedge \neg y \equiv \neg x \wedge y \vee \neg x \wedge 0 \equiv \neg x \wedge y \vee 0 \equiv \neg x \wedge y$;
(б) $\neg[(x \wedge \neg y) \to (\neg x \vee z)] \equiv \neg[\neg(x \wedge \neg y) \vee (\neg x \vee z)] \equiv \neg\neg(x \wedge \neg y) \wedge \neg(\neg x \vee z) \equiv (x \wedge \neg y) \wedge (\neg\neg x \wedge \neg z) \equiv x \wedge \neg y \wedge x \wedge \neg z \equiv x \wedge \neg y \wedge \neg z$;
(в) $(x \oplus y) \to (x \wedge \neg y) \equiv ((x \vee y) \wedge (\neg x \vee \neg y)) \to (x \wedge \neg y) \equiv \neg((x \vee y) \wedge (\neg x \vee \neg y)) \vee (x \wedge \neg y) \equiv (\neg(x \vee y) \vee \neg(\neg x \wedge \neg y)) \vee (x \wedge \neg y) \equiv (\neg x \wedge \neg y) \vee (\neg\neg x \wedge \neg\neg y) \vee (x \wedge \neg y) \equiv (\neg x \wedge \neg y) \vee (x \wedge y) \vee (x \wedge \neg y) \equiv (\neg x \wedge \neg y) \vee x \wedge (y \vee \neg y) \equiv (\neg x \wedge \neg y) \vee x \wedge 1 \equiv (\neg x \wedge \neg y) \vee x$.

145. $\Phi \vee (\Phi \wedge \Psi) \equiv (\Phi \wedge 1) \vee (\Phi \wedge \Psi) \equiv (\Phi \wedge (\neg\Psi \vee \Psi)) \vee (\Phi \wedge \Psi) \equiv (\Phi \wedge \neg\Psi) \vee (\Phi \wedge \Psi) \vee (\Phi \wedge \Psi) \equiv (\Phi \wedge \neg\Psi) \vee (\Phi \wedge \Psi) \equiv \Phi \wedge (\neg\Psi \vee \Psi) \equiv \Phi \wedge 1 \equiv \Phi$.
$(\Phi \wedge \Psi) \vee (\neg\Phi \wedge \Theta) \vee (\Psi \wedge \Theta) \equiv (\Phi \wedge \Psi) \vee (\neg\Phi \wedge \Theta) \vee (1 \wedge \Psi \wedge \Theta) \equiv (\Phi \wedge \Psi) \vee (\neg\Phi \wedge \Theta) \vee ((\Phi \vee \neg\Phi) \wedge \Psi \wedge \Theta) \equiv (\Phi \wedge \Psi) \vee (\neg\Phi \wedge \Theta) \vee (\Phi \wedge \Psi \wedge \Theta) \vee (\neg\Phi \wedge \Psi \wedge \Theta) \equiv ((\Phi \wedge \Psi) \vee (\Phi \wedge \Psi \wedge \Theta)) \vee ((\neg\Phi \wedge \Theta) \vee (\neg\Phi \wedge \Psi \wedge \Theta)) \equiv (\Phi \wedge \Psi) \vee (\neg\Phi \wedge \Theta)$.

146. **(а)** $\Phi \to 0 \equiv \neg\Phi \vee 0 \equiv \neg\Phi$;
(б) $1 \to \Phi \equiv \neg 1 \vee \Phi \equiv 0 \vee \Phi \equiv \Phi$;
(в) $(\Phi \to \Psi) \to \Psi \equiv \neg(\Phi \to \Psi) \vee \Psi \equiv \neg(\neg\Phi \vee \Psi) \vee \Psi \equiv (\neg\neg\Phi \wedge \neg\Psi) \vee \Psi \equiv (\Phi \wedge \neg\Psi) \vee \Psi \equiv (\Phi \vee \Psi) \wedge (\neg\Psi \vee \Psi) \equiv (\Phi \vee \Psi) \wedge 1 \equiv \Phi \vee \Psi$;
(г) $\neg\Phi \to \Phi \equiv \neg\neg\Phi \vee \Phi \equiv \Phi \vee \Phi \equiv \Phi$;
(д) $(\Phi \to \Phi) \to \Phi \equiv \Phi \vee \Phi \equiv \Phi$, использована эквивалентность (в);
(е) построить таблицы истинности;

(ж) $\Phi \vee (\neg\Phi \wedge \Psi) \equiv (\Phi \vee \neg\Phi) \wedge (\Phi \vee \Psi) \equiv 1 \wedge (\Phi \vee \Psi) \equiv \Phi \vee \Psi$;

(з) $(\Phi \to \Psi) \oplus (\Psi \to \Phi) \equiv ((\Phi \to \Psi) \vee (\Psi \to \Phi)) \wedge (\neg(\Phi \to \Psi) \vee \neg(\Psi \to \Phi))(\neg\Phi \vee \Psi \vee \neg\Psi \vee \Phi) \wedge ((\Phi \wedge \neg\Psi) \vee (\Psi \wedge \neg\Phi)) \equiv 1 \wedge ((\Phi \wedge \neg\Psi) \vee (\Psi \wedge \neg\Phi)) \equiv (\Phi \wedge \neg\Psi) \vee (\Psi \wedge \neg\Phi) \equiv \Phi \oplus \Psi$;

(и) $\Phi \to (\Psi \vee \Theta) \equiv \neg\Phi \vee (\Psi \vee \Theta) \equiv (\neg\Phi \vee \Theta) \vee \Psi \equiv \neg(\Phi \wedge \neg\Theta) \vee \Psi \equiv (\Phi \wedge \neg\Theta) \to \Psi$;

(к) $(\Phi \wedge \Theta) \to \Psi \equiv \neg(\Phi \wedge \Theta) \vee \Psi \equiv (\neg\Phi \vee \neg\Theta) \vee \Psi \equiv \neg\Phi \vee (\Psi \vee \neg\Theta) \equiv \Phi \to (\Psi \vee \neg\Theta)$.

147. (а) $(x \to (y \to x)) \equiv (\neg x \vee (\neg y \vee x)) \equiv (\neg x \vee x) \vee \neg y \equiv 1 \vee \neg y \equiv 1$;

(б) $((x \wedge y) \to x) \equiv (\neg(x \wedge y) \vee x) \equiv ((\neg x \vee \neg y) \vee x) \equiv (\neg x \vee x) \vee \neg y \equiv 1 \vee \neg y \equiv 1$;

(в) $(x \to (x \vee y)) \equiv (\neg x \vee (x \vee y)) \equiv (\neg x \vee x) \vee y \equiv 1 \vee y \equiv 1$;

(г) $((x \to \neg y) \to \neg(x \wedge y)) \equiv ((\neg x \vee \neg y) \to (\neg x \vee \neg y)) \equiv (\neg(\neg x \vee \neg y) \vee (\neg x \vee \neg y)) \equiv 1$;

(д) $(((x \to y) \wedge (y \to z)) \to (x \to z)) \equiv (((\neg x \vee y) \wedge (\neg y \vee z)) \to (\neg x \vee z)) \equiv (\neg((\neg x \vee y) \wedge (\neg y \vee z)) \vee (\neg x \vee z)) \equiv ((\neg(\neg x \vee y) \vee \neg(\neg y \vee z)) \vee (\neg x \vee z)) \equiv (((\neg\neg x \wedge \neg y) \vee (\neg\neg y \wedge \neg z)) \vee (\neg x \vee z)) \equiv (((x \wedge \neg y) \vee (y \wedge \neg z)) \vee (\neg x \vee z)) \equiv (((x \wedge \neg y) \vee \neg x) \vee ((y \wedge \neg z) \vee z)) \equiv (((x \vee \neg x) \wedge (\neg y \vee \neg x)) \vee ((y \vee z) \wedge (\neg z \vee z))) \equiv ((1 \wedge (\neg y \vee \neg x)) \vee ((y \vee z) \wedge 1)) \equiv ((\neg y \vee \neg x) \vee (y \vee z)) \equiv ((\neg y \vee y) \vee (\neg x \vee z)) \equiv 1$;

(е) $(((x \to y) \vee (x \to z)) \to (x \to (y \vee z))) \equiv (((\neg x \vee y) \vee (\neg x \vee z)) \to (\neg x \vee (y \vee z))) \equiv (\neg(\neg x \vee y \vee z) \vee (\neg x \vee y \vee z)) \equiv 1$;

(ж) $((x \to y) \to ((x \to \neg y) \to \neg x)) \equiv ((\neg x \vee y) \to ((\neg x \vee \neg y) \to \neg x)) \equiv ((\neg x \vee y) \to (\neg(\neg x \vee \neg y) \vee \neg x)) \equiv (\neg(\neg x \vee y) \vee (\neg(\neg x \vee \neg y) \vee \neg x)) \equiv ((\neg\neg x \wedge \neg y) \vee ((\neg\neg x \wedge \neg\neg y) \vee \neg x)) \equiv ((x \wedge \neg y) \vee ((x \wedge y) \vee \neg x)) \equiv ((x \wedge (\neg y \vee y)) \vee \neg x) \equiv ((x \wedge 1) \vee \neg x) \equiv (x \vee \neg x) \equiv 1$;

(з) $((x \to (y \to z)) \to ((x \to y) \to (x \to z))) \equiv \neg(x \to (y \to z)) \vee (\neg(x \to y) \vee (x \to z)) \equiv (x \wedge \neg(y \to z)) \vee (x \wedge \neg y) \vee (\neg x \vee z) \equiv ((x \wedge y \wedge \neg z) \vee \neg x) \vee ((x \wedge \neg y) \vee \neg x) \vee z \equiv ((y \wedge \neg z) \vee \neg x) \vee (\neg y \vee \neg x) \vee z \equiv ((y \wedge \neg z) \vee z) \vee \neg y \vee \neg x \equiv (y \vee z) \vee \neg y \vee \neg x \equiv (y \vee \neg y) \vee z \vee \neg x \equiv 1 \vee z \vee \neg x \equiv 1$.

148. Индукция по построению Φ. Базис: если $\Phi = x$, то $(\Phi)^x_\Theta = \Theta$ и $I((\Phi)^x_\Theta) = I(\Theta) = J(x) = J(\Phi)$; если $\Phi = y$, то $(\Phi)^x_\Theta = y$ и $I((\Phi)^x_\Theta) = I(y) = J(y) = J(\Phi)$. Шаг для конъюнкции: $I((\Phi_1 \wedge \Phi_2)^x_\Theta) = I((\Phi_1)^x_\Theta \wedge (\Phi_2)^x_\Theta) = I((\Phi_1)^x_\Theta) \wedge I((\Phi_2)^x_\Theta) = J(\Phi_1) \wedge J(\Phi_2) = J(\Phi_1 \wedge \Phi_2) = J(\Phi)$. Аналогично для других связок.

149. В условиях предыдущей задачи $I(\Phi)^x_\Theta = J(\Phi)$, $I(\Psi)^x_\Theta = J(\Psi)$. Тогда $I(\Phi)^x_\Theta = J(\Phi) = J(\Psi) = I(\Psi)^x_\Theta$.

150. Пусть $x_i = 0$. Тогда Φ эквивалентна Φ_0. Получаем $(\neg x_i \wedge \Phi_0) \vee (x_i \wedge \Phi_1) \equiv (\neg 0 \wedge \Phi_0) \vee (0 \wedge \Phi_1) \equiv (1 \wedge \Phi_0) \vee 0 \equiv \Phi_0 \equiv \Phi$; $(\neg x_i \to \Phi_0) \wedge (x_i \to \Phi_1) \equiv$

$(\neg 0 \to \Phi_0) \wedge (0 \to \Phi_1) \equiv (1 \to \Phi_0) \wedge 1 \equiv \Phi_0 \equiv \Phi$. Пусть $x_i = 1$. Тогда Φ эквивалентна Φ_1. Получаем $(\neg x_i \wedge \Phi_0) \vee (x_i \wedge \Phi_1) \equiv (\neg 1 \wedge \Phi_0) \vee (1 \wedge \Phi_1) \equiv (0 \wedge \Phi_0) \vee \Phi_1 \equiv 0 \vee \Phi_1 \equiv \Phi_1 \equiv \Phi$; $(\neg x_i \to \Phi_0) \wedge (x_i \to \Phi_1) \equiv (\neg 1 \to \Phi_0) \wedge (1 \to \Phi_1) \equiv (0 \to \Phi_0) \wedge \Phi_1 \equiv 1 \wedge \Phi_1 \equiv \Phi_1 \equiv \Phi$.

151. **(а)** $f^{**}(x_1,\ldots,x_n) = \neg f^*(\neg x_1,\ldots,\neg x_n) = \neg\neg f(\neg\neg x_1,\ldots,\neg\neg x_n) = f(x_1,\ldots,x_n)$.

(б) $h^*(x_1,\ldots,x_n) = \neg h(\neg x_1,\ldots,\neg x_n) = \neg f(g_1(\neg x_1,\ldots,\neg x_n),\ldots,g_m(\neg x_1,\ldots,\neg x_n)) = \neg f(\neg\neg g_1(\neg x_1,\ldots,\neg x_n),\ldots,\neg\neg g_m(\neg x_1,\ldots,\neg x_n)) = \neg f(\neg g_1^*(x_1,\ldots,x_n),\ldots,\neg g_m^*(x_1,\ldots,x_n)) = f^*(g_1^*(x_1,\ldots,x_n),\ldots,g_m^*(x_1,\ldots,x_n))$.

152. **(а)** $\neg^* x = \neg(\neg(\neg x)) = \neg x$;

(б) $x \to^* y = \neg(\neg x \to \neg y) = \neg(\neg\neg x \vee \neg y) = \neg\neg\neg x \wedge \neg\neg y = \neg x \wedge y$;

(в) $x \oplus^* y = \neg(\neg x \oplus \neg y) = ((x\oplus 1)\oplus(y\oplus 1))\oplus 1 = (x\oplus y)\oplus 1 = \neg(x\oplus y) = x \leftrightarrow y$;

(г) $x \uparrow^* y = \neg(\neg x \uparrow \neg y) = \neg\neg(\neg x \wedge \neg y) = \neg x \wedge \neg y = \neg(x \vee y) = x \downarrow y$.

153. **(а)** $(x \to \neg(\neg y \oplus x \oplus z))^* = (x \to^* \neg^*(\neg^* y \oplus^* x \oplus^* z)) = (\neg x \wedge \neg(\neg y \leftrightarrow x \leftrightarrow z)) = (\neg x \wedge \neg\neg(\neg y \oplus (x \leftrightarrow z))) = (\neg x \wedge (\neg y \oplus \neg(x \oplus z))) = (\neg x \wedge ((y \oplus 1) \oplus (x \oplus z \oplus 1))) = (\neg x \wedge (y \oplus x \oplus z)) = (\neg x \wedge y) \oplus (\neg x \wedge x) \oplus (\neg x \wedge z) = (\neg x \wedge y) \oplus (\neg x \wedge z) = \neg x \wedge (y \oplus z)$;

(б) $((x \uparrow y) \oplus (\neg x \to z))^* = ((x \uparrow^* y) \oplus^* (\neg^* x \to^* z)) = ((x \downarrow y) \leftrightarrow (\neg x \to^* z)) = ((x \downarrow y) \leftrightarrow (\neg\neg x \wedge z)) = ((x \downarrow y) \leftrightarrow (x \wedge z))$;

(в) $((x \vee \neg y) \to (\neg x \downarrow y))^* = ((x \vee^* \neg^* y) \to^* (\neg^* x \downarrow^* y)) = (\neg(x \wedge \neg y) \wedge (\neg x \uparrow y)) = ((\neg x \vee y) \wedge (\neg\neg x \vee \neg y)) = ((x \to y) \wedge (x \vee \neg y)) = ((x \to y) \wedge (y \to x)) = x \leftrightarrow y$.

154. **(а)** Существует только $2^{2^2} = 16 < 20$ булевых функций и, соответственно, попарно неэквивалентных формул с двумя переменными.

(б) Неверно, может быть 4 пары эквивалентных и ещё 12 неэквивалентных.

(в) $k/2^{2^n}$ с округлением вверх.

155. **(а)** Взять значение всех переменных равным 1.

(б) Индукция по количеству переменных. Базис: для формулы $\Phi = x_1$ — очевидно. Индукционный шаг: пусть $\Phi = \Psi \to \Theta$, причём Ψ и Θ не имеют общих переменных. По индукционному предположению Θ принимает нулевое значение хотя бы на одном наборе $\bar\tau$, Ψ принимает значение 1 хотя бы на одном наборе $\bar\sigma$. Тогда значение Φ будет равно нулю на наборе $(\bar\sigma, \bar\tau)$.

(в) Пусть $\Phi = \Psi \to \Theta$. Если переменная x_k находится в Ψ, то сделать так, чтобы значение Θ было равно 1. Если переменная x_k находится в Θ, то сделать так, чтобы значение Ψ было равно 0.

(г) Индукция по количеству импликаций в формуле Φ. Базис: в функции x_1 единственный аргумент фиктивным не является. Индукционный шаг. Пусть $\Phi = \Psi \to \Theta$. Формула Ψ принимает значение 1 хотя бы на одном наборе, а формула Θ не имеет фиктивных аргументов по индукционному предположению. Следовательно, среди переменных формулы Θ фиктивных аргументов для Φ нет. Формула Θ принимает значение 0 хотя бы на одном наборе, а формула Ψ не имеет фиктивных аргументов по индукционному предположению. Следовательно, среди переменных формулы Ψ фиктивных аргументов для Φ тоже нет.

(д) Для формул разной длины утверждение следует из предыдущего пункта. Для формул одинаковой длины применим индукцию по n. Базис. Для $n < 3$ доказывать нечего, так как различная расстановка скобок невозможна. Формулы $(x_1 \to x_2) \to x_3$ и $x_1 \to (x_2 \to x_3)$ принимают разные значения на наборе $(0,1,0)$. Индукционный шаг. Пусть $\Phi_1 = \Psi_1 \to \Theta_1$ и $\Phi_2 = \Psi_2 \to \Theta_2$ — две разные импликативные формулы. Пусть длины Θ_1 и Θ_2 равны. Тогда равны длины Ψ_1 и Ψ_2. Если Θ_1 и Θ_2 отличаются, то по индукционному предположению они не эквивалентны, то есть принимают разные значения в некоторой интерпретации. Если теперь придать всем переменным формул Ψ_1 и Ψ_2 значение 1, то значения формул Φ_1 и Φ_2 тоже будут разными. Если Θ_1 и Θ_2 одинаковы, то так как Φ_1 и Φ_2 отличаются, Ψ_1 и Ψ_2 тоже должны отличаться. По индукционному предположению Ψ_1 и Ψ_2 не эквивалентны, то есть принимают разные значения в некоторой интерпретации. Если теперь придать всем переменным формул $\Theta_1 = \Theta_2$ значения так, чтобы $\Theta_1 = \Theta_2$ были ложными, то значения формул Φ_1 и Φ_2 тоже будут разными. Пусть длина Ψ_1 меньше длины Ψ_2. Если Ψ_1 содержит две или более переменных, то положим значение x_1 равным 1. Тогда импликация вида $x_1 \to \Xi$ будет эквивалентна Ξ и мы получим две импликативные формулы $\Phi_1' = \Psi_1' \to \Theta_1$ и $\Phi_2' = \Psi_2' \to \Theta_2$ с $n-1$ переменной. Так как длины Θ_1 и Θ_2 отличаются, то по индукционному предположению Φ_1' и Φ_2' не эквивалентны. Следовательно, Φ_1 и Φ_2 тоже не эквивалентны. Пусть теперь Ψ_1 содержит одну переменную: $\Phi_1 = x_1 \to (\Psi_1' \to \Theta_1')$. Если Θ_1' отличается от Θ_2, то для $x_1 = 1$ снова применяем индукционное предположение, получаем, что Φ_1' и Φ_2' не эквивалентны, как и Φ_1 и Φ_2. Если же $\Theta_1' = \Theta_2$, то возьмём значение $x_1 = 0$ расширим его до набора $\bar{\sigma}$, на котором Ψ_2 истинно, далее расширим его до набора $(\bar{\sigma}, \bar{\tau})$, на котором $\Theta_1' = \Theta_2$ ложно. Теперь формула Φ_1 истинна на $(\bar{\sigma}, \bar{\tau})$, а формула Φ_2 — ложна.

156. В прямую сторону: индукция по построению формулы Φ, задающей функцию g. Базис: $\Phi = x$, $\Phi \equiv x \vee x$. Индукционный шаг. Если $\Phi = \Theta \to \Xi$, то по индукционному предположению $\Xi \equiv x \vee \Upsilon$, следовательно, $\Phi \equiv \neg\Theta \vee \Xi \equiv x \vee (\Upsilon \vee \neg\Theta)$. Обратно: индукция по постоению Ψ. Базис: если $\Psi = y$, то $x \vee y \equiv x \vee (\neg x \wedge y) \equiv x \vee \neg(y \to x) \equiv (y \to x) \to x$. Индукционный шаг. Для отрицания: $x \vee \neg\Psi \equiv x \vee (\neg x \wedge \neg\Psi) \equiv x \vee \neg(x \vee \Psi)$. По индукционному предположению $x \vee \Psi$ эквивалентно некоторой формуле Φ', которая получена из импликативной подстановокой переменных. Следовательно, $x \vee \neg(x \vee \Psi) \equiv x \vee \neg\Phi' \equiv \Phi' \to x$. Для конъюнкции: $x \vee (\Psi \wedge \Theta) \equiv x \vee (x \vee (\Psi \wedge \Theta)) \equiv x \vee ((x \vee \Psi) \wedge (x \vee \Theta))$. По индукционному предположению $x \vee \Psi$ и $x \vee \Theta$ эквивалентны некоторым формулам Φ' и Φ'' соответственно, которые получены из импликативных подстановками переменных. Следовательно, $x \vee ((x \vee \Psi) \wedge (x \vee \Theta)) \equiv x \vee (\Phi' \wedge \Phi'') \equiv x \vee (\neg x \wedge \Phi' \wedge \Phi'') \equiv x \vee \neg(x \vee \neg\Phi' \vee \neg\Phi'') \equiv x \vee \neg(\Phi' \to (\Phi'' \to x)) \equiv (\Phi' \to (\Phi'' \to x)) \to x$. Остальные случаи сводятся к предыдущим: $\Psi \vee \Theta \equiv \neg(\neg\Psi \wedge \neg\Theta)$, $\Psi \to \Theta \equiv \neg(\Psi \wedge \neg\Theta)$.

Дизъюнктивные и конъюнктивные нормальные формы

157. $1^1 = 1$, $1^0 = \neg 1 = 0$; $0^1 = 0 = \neg 1$, $0^0 = \neg 0 = 1$.

158. **(а)** $C_n^k \cdot 2^k$: выбрать k переменных из n, определить, какие из них с отрицаниями.

(б) $3^n - 1$: каждая переменная может присутствовать без отрицания, присутствовать с отрицанием или отсутствовать. Отсутствие всех переменных разом недопустимо.

159. Для ДНФ. Если $f(\sigma_1, \ldots, \sigma_n) = 1$, то $\mathcal{D}_f$ содержит конъюнкцию $\bigwedge x_i^{\sigma_i}$. Согласно (5) при $x_i = \sigma_i$ будет $x_i^{\sigma_i} = 1$, следовательно, $\bigwedge x_i^{\neg\sigma_i} = 1$, поэтому $\mathcal{D}_f(\sigma_1, \ldots, \sigma_n) = 1$. Если $\mathcal{D}_f(\sigma_1, \ldots, \sigma_n) = 1$, то существует конъюнкция $\bigwedge x_i^{\tau_i}$ равная единице. Следовательно, $x_i^{\tau_i} = 1$ для всех x_i. Согласно (5) это означает, что $x_i \neq \neg\tau_i$, то есть $\sigma_i = x_i = \tau_i$. Итак, $\mathcal{D}_f$ содержит конъюнкцию $\bigwedge x_i^{\sigma_i}$, по определению $\mathcal{D}_f$ это означает $f(\sigma_1, \ldots, \sigma_n) = 1$. Для КНФ. Если $f(\sigma_1, \ldots, \sigma_n) = 0$, то $\mathcal{C}_f$ содержит дизъюнкцию $\bigvee x_i^{\neg\sigma_i}$. Согласно (5) при $x_i = \sigma_i$ будет $x_i^{\neg\sigma_i} = 0$, следовательно, $\bigvee x_i^{\neg\sigma_i} = 0$, поэтому $\mathcal{C}_f(\sigma_1, \ldots, \sigma_n) = 0$. Если $\mathcal{C}_f(\sigma_1, \ldots, \sigma_n) = 0$, то существует дизъюнкция $\bigvee x_i^{\neg\tau_i}$ равная нулю. Следовательно, $x_i^{\neg\tau_i} = 0$ для всех x_i. Согласно (5) это означает, что $x_i \neq \neg\tau_i$, то есть $\sigma_i = x_i = \tau_i$. Итак, $\mathcal{C}_f$ содержит дизъюнкцию $\bigvee x_i^{\neg\sigma_i}$, по определению $\mathcal{C}_f$ это означает $f(\sigma_1, \ldots, \sigma_n) = 0$.

160. **(а)** Если K — конъюнкция, а $x_1, \ldots, x_n$ — новые переменные, то $\mathcal{D} = \bigvee\limits_{(\sigma_1, \ldots, \sigma_n) \in \mathbb{B}^n} K \wedge \bigwedge\limits_i x_i^{\sigma_i}$.

(б) Аналогично.

161. Если $f(\sigma_1, \ldots, \sigma_k, \sigma_{k+1}, \ldots, \sigma_n) = 1$, то при $x_i = \sigma_i$ получаем $x_i^{\sigma_i} = 1$ и $x_1^{\sigma_1} \wedge \ldots \wedge x_k^{\sigma_k} \wedge f(\sigma_1, \ldots, \sigma_k, x_{k+1}, \ldots, x_n) = 1$, поэтому вся правая дизъюнкция истинна. Если правая дизъюнкция истинна при $x_i = \sigma_i$, то $x_1^{\tau_1} \wedge \ldots \wedge x_k^{\tau_k} \wedge f(\tau_1, \ldots, \tau_k, x_{k+1}, \ldots, x_n) = 1$ для каких-то $\tau_1, \ldots, \tau_k$. Из истинности $x_i^{\tau_i} = 1$ и (5) получаем $\tau_i = \sigma_i$. Следовательно, $f(\sigma_1, \ldots, \sigma_k, \sigma_{k+1}, \ldots, \sigma_n) = 1$.

162. $\mathcal{C}(x) = \mathcal{D}(x) = x$, $\mathcal{C}(\neg\Phi) \equiv \neg\mathcal{D}(\Phi)$, $\mathcal{D}(\neg\Phi) \equiv \neg\mathcal{C}(\Phi)$, $\mathcal{C}(\Phi \wedge \Psi) = \mathcal{C}(\Phi) \wedge \mathcal{C}(\Psi)$, $\mathcal{D}(\Phi \vee \Psi) = \mathcal{D}(\Phi) \vee \mathcal{D}(\Psi)$, $\mathcal{C}(\Phi \vee \Psi) = \bigwedge\limits_{D_1 \in \mathcal{C}(\Phi)} \bigwedge\limits_{D_2 \in \mathcal{C}(\Psi)} (D_1 \vee D_2)$, $\mathcal{D}(\Phi \wedge \Psi) = \bigvee\limits_{K_1 \in \mathcal{D}(\Phi)} \bigvee\limits_{K_2 \in \mathcal{D}(\Psi)} (K_1 \wedge K_2)$, $\mathcal{C}(\Phi \to \Psi) = \bigwedge\limits_{K_1 \in \mathcal{D}(\Phi)} \bigwedge\limits_{D_2 \in \mathcal{C}(\Psi)} (\neg K_1 \vee D_2)$, $\mathcal{D}(\Phi \to \Psi) = \neg\mathcal{C}(\Phi) \vee \mathcal{D}(\Psi)$.

163. **(а)** $(\neg x \wedge \neg y \wedge \neg z) \vee (\neg x \wedge \neg y \wedge z) \vee (x \wedge \neg y \wedge \neg z) \vee (x \wedge \neg y \wedge z) \vee (x \wedge y \wedge z)$, $(x \vee \neg y \vee z) \wedge (x \vee \neg y \vee \neg z) \wedge (\neg x \vee \neg y \vee z)$;

(б) $(\neg x \wedge \neg y \wedge z) \vee (x \wedge \neg y \wedge \neg z) \vee (x \wedge \neg y \wedge z) \vee (x \wedge y \wedge \neg z) \vee (x \wedge y \wedge z)$, $(x \vee y \vee z) \wedge (x \vee \neg y \vee z) \wedge (x \vee \neg y \vee \neg z)$;

(в) $(\neg x \wedge \neg y \wedge z) \vee (\neg x \wedge y \wedge z) \vee (x \wedge \neg y \wedge z)$, $(x \vee y \vee z) \wedge (x \vee \neg y \vee z) \wedge (\neg x \vee y \vee z) \wedge (\neg x \vee \neg y \vee z) \wedge (\neg x \vee \neg y \vee \neg z)$.

164. **(а)** $(\neg x \wedge \neg y \wedge z) \vee (\neg x \wedge y \wedge z) \vee (x \wedge \neg y \wedge z) \vee (x \wedge y \wedge \neg z) \vee (x \wedge y \wedge z)$;

(б) $(\neg x \wedge \neg y \wedge z) \vee (\neg x \wedge y \wedge \neg z) \vee (\neg x \wedge y \wedge z) \vee (x \wedge \neg y \wedge \neg z) \vee (x \wedge \neg y \wedge z) \vee (x \wedge y \wedge \neg z) \vee (x \wedge y \wedge z)$;

(в) $(\neg x \wedge \neg y \wedge \neg z) \vee (\neg x \wedge \neg y \wedge z) \vee (\neg x \wedge y \wedge z) \vee (x \wedge \neg y \wedge \neg z) \vee (x \wedge \neg y \wedge z) \vee (x \wedge y \wedge \neg z) \vee (x \wedge y \wedge z)$.

165. **(а)** $(x \vee y \vee z) \wedge (x \vee y \vee \neg z) \wedge (x \vee \neg y \vee \neg z) \wedge (\neg x \vee y \vee z) \wedge (\neg x \vee \neg y \vee z)$;

(б) $(x \vee y \vee z) \wedge (x \vee y \vee \neg z) \wedge (x \vee \neg y \vee z) \wedge (\neg x \vee y \vee z) \wedge (\neg x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee z) \wedge (\neg x \vee \neg y \vee \neg z)$;

(в) $(x \vee y \vee z) \wedge (\neg x \vee y \vee z) \wedge (\neg x \vee \neg y \vee z) \wedge (\neg x \vee \neg y \vee \neg z)$.

166. $(0000\,1111)$, $\mathcal{D}_f = (x \wedge \neg y \wedge \neg z) \vee (x \wedge \neg y \wedge z) \vee (x \wedge y \wedge \neg z) \vee (x \wedge y \wedge z)$, $\mathcal{C}_f = (x \vee y \vee z) \wedge (x \vee y \vee \neg z) \wedge (x \vee \neg y \vee z) \wedge (x \vee \neg y \vee \neg z)$.

167. **(а)** $(x \wedge y) \vee (\neg x \wedge y \wedge z) \vee (x \wedge z) \equiv (x \wedge y) \vee (\neg x \wedge y \wedge z) \vee (x \wedge z) \vee (y \wedge z) \equiv (x \wedge y) \vee (x \wedge z) \vee (y \wedge z) \equiv (x \vee x \vee y) \wedge (x \vee x \vee z) \wedge (x \vee z \vee y) \wedge (x \vee z \vee z) \wedge (y \vee x \vee y) \wedge (y \vee x \vee z) \wedge (y \vee z \vee y) \wedge (y \vee z \vee z) \equiv (x \vee y) \wedge (x \vee z) \wedge (y \vee z)$;

(б) $(\neg x \wedge \neg y \wedge z) \vee (\neg x \wedge y \wedge z) \vee (x \wedge y \wedge \neg z) \equiv ((\neg x \wedge z) \wedge (\neg y \vee y)) \vee (x \wedge y \wedge \neg z) \equiv (\neg x \wedge z) \vee (x \wedge y \wedge \neg z) \equiv (\neg x \vee x) \wedge (\neg x \vee y) \wedge (\neg x \vee \neg z) \wedge (z \vee x) \wedge (z \vee y) \wedge (z \vee \neg z) \equiv (\neg x \vee y) \wedge (\neg x \vee \neg z) \wedge (z \vee x) \wedge (z \vee y)$;

(в) $x \vee (\neg x \wedge z) \vee (\neg x \wedge y \wedge z) \vee (\neg x \wedge \neg y \wedge \neg z) \equiv x \vee z \vee (y \wedge z) \vee (\neg y \wedge \neg z) \equiv x \vee z \vee \neg y$.

168. Являются импликантами (а), (б), (г), (е). (в) и (д) истины при $x = z = 1$ и $y = 0$, когда f имеет значение 0. Минимальными являются (б) и (г).

169. Все элементарные конъюнкции в ДНФ являются импликантами. Если бы среди них был неминимальный, то его можно было бы заменить более коротким и получить более короткую ДНФ.

170. При изменении значения любой переменной значение всей функции всегда меняется. Если бы какая-то элементарная конъюнкция не содержала ни переменную x_i, ни её отрицание, то её значение не изменялось бы истинной при изменении значения этой переменной. Поэтому когда все элементы этой конъюнкции истинны, значение функции равно 1 и не зависит от x_i, что невозможно. Следовательно, все импликанты содержат все переменные и должны иметь длину n. Поскольку функция принимает значение 1 в точности на 2^{n-1} наборах, то в ДНФ этих импликантов именно столько и должно быть.

171. Если количество единиц среди значений булевой функции f не превосходит 2^{n-1}, то совершенная ДНФ состоит из не более чем 2^{n-1} конъюнкций, и сокращённая ДНФ имеет не больше 2^{n-1} конъюнкций. Если количество единиц среди значений булевой функции f больше 2^{n-1}, то среди наборов, на которых f принимает значение 1, будут как минимум два, отличающихся значением только одной переменной. Тогда при построении сокращённой ДНФ соответствующие две элементарные конъюнкции будут заменены одной конъюнкцией меньшей длины.

172. **(а)** $(\neg x \wedge \neg y \wedge \neg z) \vee (\neg x \wedge y \wedge \neg z) \vee (\neg x \wedge y \wedge z) \vee (x \wedge y \wedge \neg z) \vee (x \wedge y \wedge z)$; $(x \vee y \vee \neg z) \wedge (\neg x \vee y \vee z) \wedge (\neg x \vee y \vee \neg z)$; $y \vee (\neg x \wedge \neg z)$;

(б) $(\neg x \wedge y \wedge \neg z) \vee (\neg x \vee y \vee z) \wedge (x \wedge \neg y \wedge \neg z) \wedge (x \wedge y \wedge z)$; $(x \vee y \vee z) \wedge (x \vee y \vee \neg z) \wedge (\neg x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee z)$; $(\neg x \wedge y) \vee (y \wedge z) \vee (x \wedge \neg y \wedge \neg z)$;

(в) $(\neg x \wedge \neg y \wedge \neg z) \vee (\neg x \wedge \neg y \wedge z) \vee (\neg x \wedge y \wedge \neg z) \vee (x \wedge \neg y \wedge \neg z) \vee (x \wedge y \wedge \neg z) \vee (x \wedge y \wedge z)$; $(x \vee \neg y \vee \neg z) \wedge (\neg x \vee y \vee \neg z)$; $\neg z \vee (\neg x \wedge \neg y) \vee (x \wedge y)$;

(г) $(\neg x \wedge y \wedge \neg z) \vee (\neg x \wedge y \wedge z) \vee (x \wedge \neg y \wedge \neg z) \vee (x \wedge y \wedge \neg z) \vee (x \wedge y \wedge z)$; $(x \vee y \vee z) \wedge (x \vee y \vee \neg z) \wedge (\neg x \vee y \vee \neg z)$; $y \vee (x \wedge \neg z)$;

(д) $(\neg x \wedge y \wedge z) \vee (x \wedge \neg y \wedge z) \vee (x \wedge y \wedge \neg z) \vee (x \wedge y \wedge z)$; $(x \vee y \vee z) \wedge (x \vee y \vee \neg z) \wedge (x \vee \neg y \vee z) \wedge (\neg x \vee y \vee z)$; $(x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$;

(е) $(\neg x \wedge \neg y \wedge z) \vee (\neg x \wedge y \wedge \neg z) \vee (\neg x \wedge y \wedge z) \vee (x \wedge \neg y \wedge z) \vee (x \wedge y \wedge z)$; $(x \vee y \vee z) \wedge (\neg x \vee y \vee z) \wedge (\neg x \vee \neg y \vee z)$; $z \vee (\neg x \wedge y)$.

173. **(а)** $y \vee (\neg x \wedge \neg z)$;

(б) $x_1 \wedge \neg x_2 \wedge x_3$;

(в) $(x \wedge \neg y) \vee (\neg x \wedge y \wedge \neg z) \vee (\neg x \wedge \neg z \wedge u) \vee (\neg y \wedge \neg z \wedge u)$;

(г) $x \wedge \neg y \wedge z$;

(д) $\neg y \vee (x \wedge \neg z)$;

(е) $\neg z \vee (\neg x \wedge \neg y)$.

174. Пусть C — минимальный импликант Φ. Придадим всем переменным, входящим в C без отрицаний, значение 1, а остальным переменным — 0. Тогда K истинна, следовательно, $\mathcal{D}$ тоже истинна, значит, в $\mathcal{D}$ есть конъюнкция C', которая содержит только истинные переменные и не содержит ничего другого. Но все эти переменные входят и в C, причём без отрицаний. Следовательно, C' является частью K. Но так как C — минимальный импликант, то $C' = C$. Следовательно, $\mathcal{D}$ содержит все минимальные импликанты. Других элементарных конъюнкций $\mathcal{D}$ не содержжит по условию, то есть является сокращённой ДНФ.

$$
\begin{array}{llllllllll}
\alpha_0 & & & & & & & & = 1\\
\alpha_0 \oplus \alpha_1 & & & & & & & & = 0\\
\alpha_0 & \oplus \alpha_2 & & & & & & & = 0\\
\alpha_0 \oplus \alpha_1 & \oplus \alpha_2 & \oplus \alpha_3 & & & & & & = 1\\
\alpha_0 & & & \oplus \alpha_4 & & & & & = 0\\
\alpha_0 \oplus \alpha_1 & & & \oplus \alpha_4 & \oplus \alpha_5 & & & & = 0\\
\alpha_0 & \oplus \alpha_2 & & \oplus \alpha_4 & & \oplus \alpha_6 & & & = 0\\
\alpha_0 \oplus \alpha_1 & \oplus \alpha_2 & \oplus \alpha_3 & \oplus \alpha_4 & \oplus \alpha_5 & \oplus \alpha_6 & \oplus \alpha_7 & & = 1
\end{array}
$$

Рис. 53: Система уравнений из задачи 178.

175. $\mathcal{D} = (x \wedge z) \vee (y \wedge \neg z)$, минимальным импликантом будет $x \wedge y$.

Многочлены Жегалкина

176. Построить истинностные таблицы.
177. **(а)** $1 \oplus x \oplus xyz$;
(б) $x \oplus xy$;
(в) $1 \oplus yz \oplus xz \oplus xyz$;
(г) $1 \oplus x \oplus xz \oplus xyz$;
(д) $xz \oplus xyz$.
178. Рис. 53; значения $\alpha_0, \ldots, \alpha_7$ равны $1, 1, 1, 0, 1, 1, 1, 1$ соответственно; $f = 1 \oplus x_3 \oplus x_2 \oplus x_1 \oplus x_1x_3 \oplus x_1x_2 \oplus x_1x_2x_3$.
179. **(а)** $y \oplus yz \oplus x \oplus xyz$;
(б) $1 \oplus yz \oplus xy \oplus xyz$;
(в) $1 \oplus y \oplus x$;
(г) $z \oplus y \oplus x \oplus xy \oplus xyz$.
180. **(а)** $1 \oplus y \oplus xy \oplus yz \oplus xyz$;
(б) $1 \oplus u \oplus yu \oplus yz \oplus x \oplus xz \oplus xy \oplus xyz \oplus xyzu$;
(в) $x \oplus yz \oplus yzu \oplus xyz \oplus xyzu$.
181. Формула под знаком суммирования эквивалентна элементарной конъюнкции, соответствующей строке функции f при построении совершенной ДНФ: $x^\sigma \equiv x \oplus \sigma \oplus 1$. Поскольку все эти конъюнкции попарно несовместны, то их сумма по модулю два совпадает с их же дизъюнкцией, то есть эквивалентной f совершенной ДНФ.
182. **(а)** $(x \oplus 1)(y \oplus 1)z \oplus (x \oplus 1)y(z \oplus 1) \oplus x(y \oplus 1)(z \oplus 1) = xyz \oplus xz \oplus yz \oplus z \oplus xyz \oplus yz \oplus xy \oplus y \oplus xyz \oplus xy \oplus xz \oplus x = z \oplus xyz \oplus y \oplus x$.

(б) $(x \oplus 1)(y \oplus 1)(z \oplus 1) \oplus (x \oplus 1)y(z \oplus 1) \oplus x(y \oplus 1)(z \oplus 1) \oplus x(y \oplus 1)z = xyz \oplus xy \oplus xz \oplus yz \oplus x \oplus y \oplus z \oplus 1 \oplus xyz \oplus xy \oplus yz \oplus y \oplus xyz \oplus xy \oplus xz \oplus x \oplus xyz \oplus xz = xy \oplus z \oplus 1 \oplus xz$.

(в) Значения функции: (1111 1011). Построим значения противоположной функции (0000 0100): $x(y \oplus 1)z = xyz \oplus xz$; добавим единицу: $xyz \oplus xz \oplus 1$.

183. Если все α_i равны нулю, то и сумма всех слагаемых будет равна нулю. Если какой-то из α_i не равен нулю, то выбрав α_i, для которого слагаемое содержит меньше всего переменных, и придав значение 1 только этим переменных, мы получим значение многочлена равное 1, так как все остальные слагаемые будут равны нулю.

184. Общее количество всевозможных слагаемых в многочлене Жегалкина равняется 2^n. Чтобы многочлен был равен нулю на нулевом наборе, нужно, чтобы среди слагаемых не было единицы. Следовательно, k штук из них можно выбрать $C^k_{2^n-1}$ способами. На единичном наборе многочлен будет равен нулю тогда и только тогда, когда количество слагаемых чётно. Следовательно, требуемых многочленов существует $C^k_{2^n-1}$ для чётных k и ни одного для нечётных.

185. Согласно задаче 181 этот многочлен равен $(x_1 \oplus 1 \oplus \sigma_1) \ldots (x_n \oplus 1 \oplus \sigma_n)$. Его длина равна 2^m, где m — это количество нулей среди $\sigma_1, \ldots, \sigma_n$, так как только они дают двучлены в произведении.

186. $\neg x_1 \wedge \cdots \wedge \neg x_n$, см. предыдущую задачу.

187. Сумма нового и старого многочленов даст многочлен из задачи 186. Следовательно, сумма новой и старой функций будет равна единице только на нулевых аргументах. Значит, новая функция получается из старой изменением значения на противоположное на нулевых аргументах.

188. Согласно задаче 181 этот многочлен равен $(x_1 \oplus 1 \oplus \sigma_1) \ldots (x_n \oplus 1 \oplus \sigma_n) \oplus (x_1 \oplus \sigma_1) \ldots (x_n \oplus \sigma_n)$. Если после раскрытия скобок слагаемое, не содержащее x_i, присутствует в первой части, то в ней есть множитель $(x_i \oplus 1)$, значит, во второй части есть множитель x_i и она такое слагаемое не содержит. Следовательно, будет в точности одна пара одинаковых слагаемых: $x_1 \ldots x_n$. Таким образом, после сокращений длина многочлена будет равна $2^{m_1} + 2^{m_0} - 2$, где m_1 и m_0 — это количества единиц и нулей соответственно среди $\sigma_1, \ldots, \sigma_n$.

189. В обратную сторону: многочлен не меняет вида при произвольной перестановке переменных, следовательно, функция коммутативна. В прямую сторону: количество таких функций равно 2^{n+1}, задача 123. Количество таких многочленов также равно 2^{n+1}: для каждого $k = 0, \ldots, n$ есть два варианта. Если бы для какой-то функции такого многочлена не нашлось, то согласно первому пункту какая-то функция задавалась бы двумя разными приведёнными многочленами, что невозможно.

190. Базис: при $n = 1$ соответствующими $k = 0, 1, 2$ многочленами являются $0, x, 1$. Шаг: пусть для n утверждение доказано и многочлены p_n^k найдены. Тогда для $n+1$ и $k \leqslant 2^n$ можно взять те же многочлены $p_{n+1}^k = p_n^k$. При $k > 2^n$ полагаем $p_{n+1}^k = x_{n+1} p_n^{2^{n+1}-k} \oplus 1$. В самом деле, при $x_{n+1} = 0$ многочлен p_{n+1}^k примет 2^n единичных значений. При $x_{n+1} = 1$ многочлен $p_n^{2^{n+1}-k}$ примет $2^{n+1} - k$ единичных значений, следовательно, $2^n - (2^{n+1} - k) = k - 2^n$ нулевых. Именно на них p_{n+1}^k будет равен единице. Следовательно, всего единичных значений для p_{n+1}^k будет $2^n + k - 2^n = k$.

191. Базис: при $n = 1$ многочлен $1 \oplus x_1$ принимает одно единичное значение и $\frac{1}{3}(2^{1+1} + (-1)^1) = \frac{1}{3}(4 - 1) = 1$. Шаг: пусть для n утверждение доказано. Заметим, что $p_{n+1}(x_1, \bar{x}) = 1 \oplus x_1 p_n(\bar{x})$. Таким образом, количество единиц среди значений многочлена $p_{n+1}(x_1, \bar{x})$ равно 2^n при $x_1 = 0$ и совпадает с количеством нулей $p_n(\bar{x})$ при $x_1 = 1$. Последнее по индукционному предположению равно $2^n - \frac{1}{3}(2^{n+1} + (-1)^n)$. Всего получаем $2^n + 2^n - \frac{1}{3}(2^{n+1} + (-1)^n) = 2^{n+1} - \frac{1}{3}2^{n+1} - \frac{1}{3}(-1)^n = \frac{2}{3}2^{n+1} + \frac{1}{3}(-1)^{n+1} = \frac{1}{3}2^{n+2} + \frac{1}{3}(-1)^{n+1} = \frac{1}{3}(2^{n+2} + (-1)^{n+1})$.

192. Базис: при $n = k$ получаем $2^{n-k} = 1$. Ненулевой многочлен принимает хотя бы одно единичное значение (задача 183). Шаг: пусть для n утверждение доказано. Тогда $p(\bar{x}, y) = p_0(\bar{x}) \oplus y p_1(\bar{x})$ для некоторых многочленов $p_0(\bar{x})$ и $p_1(\bar{x})$ с n переменными. При $y = 0$ количество единиц совпадает с количеством единиц для $p_0(\bar{x})$ и по индукционному предположению не меньше 2^{n-k}. При $y = 1$ количество единиц совпадает с количеством единиц для $p_0(\bar{x}) \oplus p_1(\bar{x})$ и по индукционному предположению не меньше 2^{n-k}. Следовательно, общее количество единиц многочлена $p(\bar{x}, y)$ не меньше $2^{n-k} \cdot 2 = 2^{(n+1)-k}$.

Полные множества функций и теорема Поста

193. Самодвойственные (б) и (в).

194. Монотонные (а) и (б).

195. $\neg x = x \downarrow x$, $x \wedge y = (x \downarrow x) \downarrow (y \downarrow y)$, $x \vee y = (x \downarrow y) \downarrow (x \downarrow y)$.

196. $\{\vee, \wedge, \to\} \subseteq \mathcal{S}_1$, можно добавить 0.

197. $f_3, f_4, f_5, f_6 \in \mathcal{S}_0$, $f_2, f_3, f_4, f_5 \in \mathcal{S}_1$, $f_3, f_4, f_5, f_7 \in \mathcal{S}$, $f_2, f_4, f_5, f_7 \in \mathcal{L}$, $f_5 \in \mathcal{M}$.

198. $\{f_1, f_i, f_j\}$, $1 < i < j$; $\{f_2, f_i, f_6\}$, $i = 3, 4, 5, 7$; $\{f_i, f_6, f_7\}$, $i < 6$; $\{f_1, f_i\}$, $i > 1$; $\{f_2, f_6\}$; $\{f_6, f_7\}$; $\{f_1\}$. Базисы — три последних множества.

199. Удалить f и h, $g = \uparrow$. Функцию g удалить невозможно, так как это единственная функция, не сохраняющая ноль.

	$\mathcal{S}_0$	$\mathcal{S}_1$	$\mathcal{L}$	$\mathcal{S}$	$\mathcal{M}$
f_1	$+$	$-$	$+$	$-$	$-$
f_2	$-$	$+$	$+$	$-$	$-$
f_3	$+$	$+$	$+$	$+$	$-$
f_4	$-$	$-$	$+$	$+$	$-$
f_5	$-$	$-$	$-$	$+$	$-$
f_6	$+$	$+$	$-$	$+$	$-$
f_7	$-$	$+$	$-$	$-$	$-$
f_8	$-$	$-$	$-$	$-$	$-$
f_9	$+$	$+$	$-$	$-$	$+$
f_{10}	$+$	$+$	$-$	$-$	$+$

Рис. 54: Таблица из задачи 203.

200. Только $\{0, \to\}$. 0 — единственная функция, не сохраняющая единицу, $\to$ — единственная немонотонная.

201. $h \in \mathcal{S}_0$, $g \in \mathcal{S}_1$, $f \in \mathcal{S}$, $f \in \mathcal{L}$, все функции немонотонны. Базисы: $\{f, g\}$, $\{f, h\}$, $\{g, h\}$. В базисе $\{f, g\}$: $\neg x = f(x, x, x)$, $1 = g(x, x, x)$, $0 = f(g(x, x, x), g(x, x, x), g(x, x, x))$, $x \to y = g(g(x, x, x), f(x, x, x), y)$. В базисе $\{f, h\}$: $0 = h(x, x, x)$, $\neg x = f(x, x, x)$, $1 = f(h(x, x, x), h(x, x, x), h(x, x, x))$, $x \to y = f(h(h(x, x, x), x, y), h(h(x, x, x), x, y), h(h(x, x, x), x, y))$. В базисе $\{g, h\}$: $0 = h(x, x, x)$, $1 = g(x, x, x)$, $\neg x = g(0, x, 0) = g(h(x, x, x), x, h(x, x, x))$, $x \to y = \neg h(0, x, y) = \neg h(h(x, x, x), x, y)$.

202. $1 = x \to x$, $0 = \neg 1$, $x \vee y = \neg x \to y$, $x \wedge y = \neg(\neg x \vee \neg y)$, $x \leftrightarrow y = (x \wedge y) \vee (\neg x \wedge \neg y)$.

203. Принадлежность функций классам показана на рис.54. Общие действия: $0 = f_1(x, x, x)$; $1 = f_2(x, x, x) = f_7(x, x, x)$; $\neg x = f_4(x, x, x) = f_5(x, x, x) = f_8(x, x, x)$. Базисы. $\{f_8\}$: $0 = f_8(x, \neg x, x)$, $1 = \neg 0$, $x \vee y = \neg f_8(1, x, y)$. $\{f_1, f_5\}$: $1 = \neg 0$, $x \vee y = \neg f_5(1, x, y)$. $\{f_1, f_2, f_6\}$: $\neg x = f_2(0, 0, x)$, $x \wedge y = f_6(1, x, y)$. $\{f_1, f_2, f_9\}$: $\neg x = f_2(0, 0, x)$, $x \vee y = f_9(1, x, y)$. $\{f_1, f_2, f_{10}\}$: $\neg x = f_2(0, 0, x)$, $x \wedge y = f_{10}(0, x, y)$. $\{f_1, f_7\}$: $\neg x = f_7(0, 0, x)$, $x \vee y = f_7(1, x, \neg y)$. $\{f_1, f_4, f_6\}$: $1 = \neg 0$, $x \wedge y = f_6(1, x, y)$. $\{f_2, f_5\}$: $0 = \neg 1$, $x \vee y = \neg f_5(1, x, y)$. $\{f_2, f_4, f_6\}$: $0 = \neg 1$, $x \wedge y = f_6(1, x, y)$. $\{f_4, f_7\}$: $0 = \neg 1$, $x \vee y = f_7(1, x, \neg y)$. $\{f_4, f_9\}$: $1 = f_9(x, x, \neg x)$, $0 = \neg 1$, $x \vee y = f_9(1, x, y)$. $\{f_4, f_{10}\}$: $0 = f_{10}(x, x, \neg x)$, $1 = \neg 0$, $x \wedge y = f_{10}(0, x, y)$. $\{f_5, f_7\}$: $0 = \neg 1$, $x \vee y = f_7(1, x, \neg y)$. $\{f_5, f_9\}$: $1 = f_9(x, x, \neg x)$, $0 = \neg 1$, $x \vee y = f_9(1, x, y)$. $\{f_5, f_{10}\}$: $0 = f_{10}(x, x, \neg x)$, $1 = \neg 0$, $x \wedge y = f_{10}(0, x, y)$.

204. ДНФ сохраняет ноль, если нет элементарных конъюнкций, содержащих только переменные с отрицаниями, ДНФ сохраняет единицу, если есть элементарная конъюнкция, содержащая только переменные без отрицаний. КНФ сохраняет ноль, если есть элементарная дизъюнкция, содержащая только переменные без отрицаний, КНФ сохраняет единицу, если нет элементарных дизъюнкций, содержащих только переменные с отрицаниями.
205. Все функции сохраняют единицу, следовательно, замыкание этого множества включено в $\mathcal{S}_1$. Если f сохраняет единицу, то по предыдущей задаче она представляется в виде КНФ, где каждая элементарная дизъюнкция содержит хотя бы одну переменную без отрицания. Если переменные с отрицаниями есть, то от них можно избавиться с помощью эквивалентности $\neg x_1 \vee \dots \vee \neg x_n \vee D \equiv \neg(x_1 \wedge \dots \wedge x_n) \vee D \equiv (x_1 \wedge \dots \wedge x_n) \to D$, где D — дизъюнкция переменных без отрицаний. Можно, $x \vee y \equiv (x \to y) \to y$.
206. Аналогично предыдущей. Воспользоваться эквивалентностями $(x \leftrightarrow y) \leftrightarrow (x \vee y) \equiv x \wedge y$, $x \leftrightarrow (x \wedge y) \equiv x \to y$, $x \leftrightarrow (y \to x) \equiv x \vee y$.
207. Множество $\mathcal{S}_0$: многочлены Жегалкина без свободного члена.
208. Если $f = g_1 \oplus \alpha_1 x_1 = g_2 \oplus \alpha_2 x_2$, то $g_1 \oplus \alpha_2 x_2 = g_2 \oplus \alpha_1 x_1 = g_{12}$. Первая часть не зависит от x_1, вторая — от x_2, поэтому обе они не зависят от x_1 и x_2 и мы получаем $f = g_{12} \oplus \alpha_1 x_1 \oplus \alpha_2 x_2$. Далее по индукции. Обратно: в силу коммутативности и ассоциативности $\oplus$.
209. В прямую сторону — по определению операции $\oplus$. Обратно: от противного, допустим, f нелинейна по x_i. Найдём в приведённом многочлене Жегалкина самое короткое произведение p, содержащее x_i и ещё хотя бы один множитель x_j. Придадим значение $\sigma_k = 1$ всем переменным x_k из p, кроме x_i и x_j, и значение $\sigma_k = 0$ всем переменным x_k, не входящим в p. Останется многочлен вида $f'(x_i, x_j) = x_i x_j \oplus \sigma_j x_i \oplus \alpha x_j \oplus \beta$. При $x_j = \sigma_j$ значение f' (и, следовательно, f) от x_i не зависит, при $x_j = \neg\sigma_j$ — меняется при изменении x_i.
210. Очевидно, что $\leftrightarrow \in \mathcal{S}_1 \cap \mathcal{L}$, поэтому, в силу замкнутости $\mathcal{S}_1$ и $\mathcal{L}$, все составленные с помощью $\leftrightarrow$ функции тоже принадлежат этим классам. Если $f \in \mathcal{S}_1 \cap \mathcal{L}$, то f имеет один из двух видов $f^{(2n)} = 1 \oplus x_1 \oplus \dots \oplus x_{2n}$ или $f^{(2n+1)} = x_1 \oplus \dots \oplus x_{2n+1}$ (фиктивные аргументы пропущены). Используем индукцию по количеству аргументов: $f^{(0)} = 1 = (x \leftrightarrow x)$, $f^{(n+1)} = f^{(n)} \oplus 1 \oplus x_{n+1} = (f^{(n)} \leftrightarrow x_{n+1})$.
211. Если $x_i = 0$, то первая конъюнкция равна нулю, результат равен второй. Если $x_i = 1$, то первая в силу монотонности имеет значение не меньшее, чем вторая, поэтому результат равен первой. Для ДНФ: по индукции $f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \equiv \bigvee\limits_j C_j^0$, $f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n) \equiv \bigvee\limits_j C_j^1$,

где C_i^0 и C_i^1 — элементарные конъюнкции без отрицаний. Тогда f задаётся как $\left(\bigvee\limits_j (x_i \wedge C_j^1)\right) \vee \bigvee\limits_j C_j^0$ — ДНФ без отрицаний. Для КНФ: по индукции $f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) \equiv \bigwedge\limits_j D_j^0$, $f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n) \equiv \bigwedge\limits_j D_j^1$, где D_i^0 и D_i^1 — элементарные дизъюнкции без отрицаний. Тогда f задаётся как $(x_i \wedge \bigwedge\limits_j D_j^1) \vee \bigwedge\limits_j D_j^0 \equiv \left(\bigwedge\limits_j (x_i \vee D_j^0)\right) \wedge \bigwedge\limits_{j_0, j_1} (D_{j_1}^1 \vee D_{j_0}^0)$ — КНФ без отрицаний.

212. В прямую сторону — по определению, так как наборы удовлетворяют неравенствам для всех аргументов. В обратную: взять наборы $\sigma_i \leqslant \rho_i$, $i = 1, \ldots, n$, и использовать индукцию по количеству i, для которых $\sigma_i < \rho_i$, то есть $\sigma_i = 0$, $\rho_i = 1$. На каждом шаге изменяется только один аргумент, поэтому значение функции не уменьшается.

213. Если минимальный импликант для f содержит отрицание $C \wedge \neg a$, то C импликантом для f не является. Значит, существует интерпретация I такая, что C равна единице, а f — нулю. Тогда $I(\neg a) = 0$ и $I(a) = 1$. Если теперь изменить значение a на ноль, то получим интерпретацию J, в которой $J(C \wedge \neg a) = 1$ и f равно единице. Но тогда f немонотонна. В обратную сторону утверждение следует из монотонности конъюнкции и дизъюнкции, а также замкнутости множества монотонных функций.

214. 0; 1; x; $x \vee y$; $x \vee z$; $x \vee y \vee z$; $x \vee (y \wedge z)$; y; $y \vee z$; $y \vee (x \wedge z)$; z; $z \vee (x \wedge y)$; $x \wedge y$; $x \wedge z$; $y \wedge z$; $(x \wedge y) \vee (x \wedge z)$; $(x \wedge y) \vee (y \wedge z)$; $(y \wedge z) \vee (x \wedge z)$; $(x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$; $x \wedge y \wedge z$.

215. Множество монотонных функций (задача 213).

216. Если исключить две функции-константы $f = 0$ и $f = 1$, а также пустое множество $V = \varnothing$, то существует биекция: $V \leftrightarrow \bigvee\limits_{U \in V} \bigwedge\limits_{x \in U} x$. В правой части стоят всевозможные сокращённые ДНФ без отрицаний (задача 174).

217. Если f самодвойственна, то $f^*(x_1, \ldots, x_n) = \neg f(\neg x_1, \ldots, \neg x_n) = \neg\neg f(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$. Обратно, если $f^* = f$, то $f(\neg x_1, \ldots, \neg x_n) = f^*(\neg x_1, \ldots, \neg x_n) = \neg f(\neg\neg x_1, \ldots, \neg\neg x_n) = \neg f(x_1, \ldots, x_n)$.

218. Пусть f самодвойственна. ДНФ содержит конъюнкцию $(x_1^{\sigma_1} \wedge \cdots \wedge x_n^{\sigma_n})$ тогда и только тогда, когда $f(\sigma_1, \ldots, \sigma_n) = 1$, тогда и только тогда, когда $f(\neg\sigma_1, \ldots, \neg\sigma_n) = 0$, тогда и только тогда, когда КНФ содержит дизъюнкцию $(x_1^{\sigma_1} \vee \cdots \vee x_n^{\sigma_n})$. Пусть f не самодвойственна, тогда существуют $\sigma_1, \ldots, \sigma_n$ такие, что $f(\sigma_1, \ldots, \sigma_n) = f(\neg\sigma_1, \ldots, \neg\sigma_n)$. ДНФ содержит конъюнкцию $(x_1^{\sigma_1} \wedge \cdots \wedge x_n^{\sigma_n})$ тогда и только тогда, когда это значение равно 1, а КНФ содержит дизъюнкцию $(x_1^{\sigma_1} \vee \cdots \vee x_n^{\sigma_n})$ тогда и только тогда, когда значение равно 0. Поэтому КНФ не получается из ДНФ заменой.

219. Если $f = \alpha \oplus \alpha_1 x_1 \oplus \dots \oplus \alpha_n x_n$, то $f^* = \alpha \oplus \alpha_1 x_1 \oplus \alpha_1 \oplus \dots \oplus \alpha_n x_n \oplus \alpha_n \oplus 1 = \beta \oplus \alpha_1 x_1 \oplus \dots \oplus \alpha_n x_n$. Если $(\sigma_1, \dots, \sigma_n) \leqslant (\rho_1, \dots, \rho_n)$, то $(\neg\sigma_1, \dots, \neg\sigma_n) \geqslant (\neg\rho_1, \dots, \neg\rho_n)$, поэтому $f(\neg\sigma_1, \dots, \neg\sigma_n) \geqslant f(\neg\rho_1, \dots, \neg\rho_n)$ и $\neg f(\neg\sigma_1, \dots, \neg\sigma_n) \leqslant \neg f(\neg\rho_1, \dots, \neg\rho_n)$. Свойства сохранения нуля и единицы меняются местами.

220. Пусть f самодвойственна. При $x_1 = 0$ в силу равенства $y \oplus x_1 = y$ получаем $f(x_1, x_2, \dots, x_n) = f(x_1 \oplus x_1, x_2 \oplus x_1, \dots, x_n \oplus x_1) \oplus x_1 = f(0, x_2 \oplus x_1, \dots, x_n \oplus x_1) \oplus x_1$. При $x_1 = 1$ в силу равенства $y \oplus x_1 = \neg y$ получаем $f(x_1, x_2, \dots, x_n) = \neg f(\neg x_1, \neg x_2, \dots, \neg x_n) = f(x_1 \oplus x_1, x_2 \oplus x_1, \dots, x_n \oplus x_1) \oplus x_1 = f(0, x_2 \oplus x_1, \dots, x_n \oplus x_1) \oplus x_1$. Следовательно, $g(u_1, \dots, u_{n-1}) = f(0, x_2 \oplus x_1, \dots, x_n \oplus x_1)$. В обратную сторону используем равенство $\neg y \oplus \neg z = y \oplus z$: $\neg(g(\neg x_1 \oplus \neg x_2, \dots, \neg x_1 \oplus \neg x_n) \oplus \neg x_1) = \neg(g(x_1 \oplus x_2, \dots, x_1 \oplus x_n) \oplus \neg x_1) = \neg(g(x_1 \oplus x_2, \dots, x_1 \oplus x_n) \oplus (x_1 \oplus 1)) = \neg((g(x_1 \oplus x_2, \dots, x_1 \oplus x_n) \oplus x_1) \oplus 1) = g(x_1 \oplus x_2, \dots, x_1 \oplus x_n) \oplus x_1$. Следовательно, функция $g(x_1 \oplus x_2, \dots, x_1 \oplus x_n) \oplus x_1$ самодвойственна.

221. $|\mathcal{S}_0| = |\mathcal{S}_1| = 2^{2^n - 1}$; $|\mathcal{S}| = 2^{2^{n-1}}$; $|\mathcal{L}| = 2^{n+1}$.

222. Всего 2^n. Самодвойственными будут линейные многочлены Жегалкина, в которых количество слагаемых-переменных нечётно.

223. Всего $n+2$: x_i и константы. Иначе f имеет вид $x_i \oplus x_j \oplus \dots$ или $1 \oplus x_i \oplus \dots$ Пусть для удобства $i = 1$, $j = 2$. Тогда в первом случае $f(1, 0, 0, \dots, 0) = 1 > 0 = f(1, 1, 0, \dots, 0)$, а во втором $f(0, 0, \dots, 0) = 1 > 0 = f(1, 0, \dots, 0)$. То есть в обоих случаях функция не монотонна.

224. **(а)** $2^{2^n - 2}$; **(б)** $2^{2^n} - 2^{2^n - 2} = 3 \cdot 2^{2^n - 2}$; **(в)** $2^{2^{n-1} - 1}$; **(г)** 2^n; **(д)** 2^n; **(е)** $2^{2^{n-1} - 1}$; **(ж)** 0; **(з)** 1; **(и)** 1.

225. Каждая такая функция однозначно определяется количеством единиц среди аргументов, при котором её значение становится равным единице. Поэтому таких функций $n + 1$.

226. Если рассмотреть подмножества $\{x_1, \dots, x_n\}$, содержащие ровно $\lfloor n/2 \rfloor$ переменных, то все они несравнимы с помощью $\subseteq$. Их количество равно $C_n^{\lfloor n/2 \rfloor}$. Согласно задаче 216 любое составленное из них множество определяет некоторую уникальную монотонную функцию. Количество таких множеств равняется $2^{C_n^{\lfloor n/2 \rfloor}}$.

227. Индукция по количеству слагаемых в сумме: для x_i и $x_i \oplus 1 = \neg x_i$, количества нулей и единиц равны 2^{n-1}. При добавлении очередного слагаемого половина нулей превращаются в единицы, а половина единиц — в нули. Поэтому их количества остаются равными. Обратное неверно: функция $xy \oplus z$ нелинейна, хотя условию удовлетворяет.

228. Индукция по построению формулы. Для формул вида x_i единицами будут ровно половина значений. Для формул $(\Phi \to \Psi)$ и $(\Phi \vee \Psi)$ достаточно, чтобы значение Ψ было равно единице, а таких не менее половины по индукционному предположению. Обратное неверно, отрицание тоже имеет половину значений равных единице, но оно не сохраняет единицу и не может быть построено с помощью дизъюнкции и импликации.

229. Если удалить конъюнкцию, то по предыдущей задаче единицами будут не менее половины значений всех построенных функций. Но тогда конъюнкцию построить невозможно, так как у неё единицами являются только четверть значений.

230. Такие функции не должны сохранять ни ноль, ни единицу: $f(\bar{0}) = 1$, $f(\bar{1}) = 0$, из этого сразу следует, что они немонотонны. Они несамодвойственны, поэтому $f(0, \sigma, \tau) = f(1, \neg\sigma, \neg\tau) = w$ для каких-то σ и τ не равных нулю одновременно. Перебирая, находим такие попарно различные варианты: $(1wxy\,zuw0)$, $(11wx\,yw00)$, $(10wx\,yw10)$, $(100w\,w110)$, $(101w\,w010)$, $(110w\,w100)$, $(111w\,w000)$. Все эти функции нелинейны (задача 209: первая — по x_3, вторая и третья — по x_2, остальные — по x_1). Всего 56 функций.

231. Таких функций четыре: x_1, x_2, конъюнкция и дизъюнкция. Все они монотонны. Трёхместная функция $x \oplus y \oplus z$ сохраняет ноль и единицу, но не монотонна.

232. Самодвойственны четыре функции: $x_1, x_2, \neg x_1, \neg x_2$. Все они линейны. Трёхместная функция $xy \oplus yz \oplus xz$ самодвойственна, но не линейна.

233. Пусть f_m — немонотонная функция базиса. Если она одноместная, то может быть только отрицанием, поэтому базис может содержать ещё только нелинейную функцию (она же несамодвойственная, задача 232). Если f_m двухместная, то не может сохранять ноль и единицу одновременно (задача 231). Есть три варианта. Первый — f_m сохраняет ноль, но не единицу. Тогда $f_m(0,0) = f_m(1,1) = 0$, следовательно, f_m немонотонная, не сохраняющая единицу и несамодвойственная. Базис может содержать ещё только нелинейную и не сохраняющую ноль функции. Второй случай — f_m сохраняет единицу, но не ноль. Тогда $f_m(0,0) = f_m(1,1) = 1$, следовательно, f_m немонотонная, не сохраняющая ноль и несамодвойственная. Базис может содержать ещё только нелинейную и не сохраняющую единицу функции. Третий случай — f_m не сохраняет ни ноль, ни единицу. Базис может содержать ещё только нелинейную (она же несамодвойственная) функцию. В любом случае, кроме f_m есть не более двух функций. Базис из трёх функций: $\{\oplus, \wedge, 1\}$.

234. Пусть f_0 — функция базиса, не сохраняющая ноль: $f_0(\bar{0}) = 1$. Если она одноместная, то она не может сохранять единицу, так как в этом случае единственный аргумент был бы фиктивным: $f_0(0) = f_0(1) = 1$. Следовательно, $f_0(1) = 0$ и f_0 является отрицанием, поэтому базис может содержать ещё только две функции: нелинейную и несамодвойственную. Пусть f_0 имеет местность два или больше. Если $f_0(\bar{1}) = 0$, то f_0 не сохраняет единицу и немонотонна, поэтому базис может содержать ещё только нелинейную и несамодвойственную функции. Если $f_0(\bar{1}) = 1$, то f_0 несамодвойственна. Так как f_0 не может быть константой, то $f_0(\bar{a}) = 0$ для некоторого $\bar{a}$, следовательно, f_0 немонотонна. Поэтому базис может содержать ещё только нелинейную и не сохраняющую единицу функции. Базис из трёх функций: $\{\oplus, \neg, xy \oplus yz \oplus xz\}$.

Хорновские формулы и задача получения продукции

235. Принадлежит только $\mathcal{S}_1$, так как $(1 \wedge \dots \wedge 1) \to 1 = 1$. Функция не сохраняет ноль, так как $(0 \wedge \dots \wedge 0) \to 0 = 1$; не самодвойственна, так как $(0 \wedge \dots \wedge 0) \to 0 = (1 \wedge \dots \wedge 1) \to 1 = 1$; не линейна, так как функция $f(1, \dots, 1, x_n, y) = x_n \to y = x_n y \oplus x_n \oplus 1$ не линейна; не монотонна, так как $(0 \wedge \dots \wedge 0) \to 0 = 1$, $(1 \wedge \dots \wedge 1) \to 0 = 0$.

236. Только (а).

237. Пусть $I(x) = 1$ для всех переменных x.

238. $\neg x_1 \vee \dots \vee \neg x_n \vee y \equiv x_1 \wedge \dots \wedge x_n \to y$.

239. Если $\Phi = A \to x$ ложна в K, то $K(A) = 1$ и $K(x) = 0$. Из первого получаем $J_i(A) = 1$ для всех i, из второго — $J_i(x) = 0$ для некоторого i. Значит, в J_i формула Φ ложна.

240. Если $C \wedge \neg a \wedge \neg b$ — минимальный импликант формулы $\neg\Phi$, то из каждой из формул $C \wedge \neg a$ и $C \wedge \neg b$ не следует $\neg\Phi$. Следовательно, есть интерпретации I и J, в которых формула $\neg\Phi$ ложна, а $C \wedge \neg a$ и $C \wedge \neg b$ соответственно истинны. Тогда в конъюнкции интерпретаций IJ будет истинна формула $C \wedge \neg a \wedge \neg b$, а, значит, и следующая из неё формула $\neg\Phi$. Окончательно получили, что в I и J формула Φ истинна, а в IJ — ложна. Следовательно, Φ не может быть конъюнкцией хорновских формул (задача 239). Если минимальный импликант формулы $\neg\Phi$ не содержит отрицаний, то он истинен в интерпретации, где истинны все переменные, тогда истинна и сама формула $\neg\Phi$, а формула Φ ложна, что для хорновских формул невозможно (задача 237). В обратную сторону — задача 238.

241. В прямую сторону. В I_1 истинна любая хорновскся формула, так как истинно заключение импликации. Следовательно, истинна их конъюнкция и эквивалентная ей Φ. Пункт (б) следует из задачи 239. В обратную сторону. Из Φ следует Ψ по определению Ψ. Докажем, что из Ψ тоже следует Φ. Допустим, что в интерпретации I выполнена Ψ, но Φ ложна. Следовательно, в Φ есть ложная в I дизъюнкция $D = x_1 \vee \dots \vee x_n \vee \neg y_1 \vee \dots \vee \neg y_m$. Сразу отметим, что $n > 0$, иначе был бы нарушен пункт (а) условия. Из ложности D получаем, что в I значения $y_1, \dots, y_m$ равны 1, а значения $x_1, \dots, x_n$ равны 0. Все хорновские формулы $\Theta_i = y_1 \wedge \dots \wedge y_m \to x_i$, $i = 1, \dots, n$, ложны в I, следовательно, они не входят в Ψ и не следуют из Φ. Значит, существуют интерпретации J_i, в которых Φ истинна, а формулы Θ_i ложны соответственно. Следовательно, в каждой J_i истинны все $y_1, \dots, y_m$, но ложна соответствующая x_i. Тогда в пересечении $K = J_1 J_2 \dots J_n$ будут истинны все $y_1, \dots, y_m$, но ложны все $x_1, \dots, x_n$. Значит, будут ложными дизъюнкция D и вся формула Φ. Это противоречит условию (б).

242. Первая не удовлетворяет условию (а) из задачи 241, вторая — условию (б): формула $a \vee b$ истинна при $a = 1, b = 0$ и при $a = 0, b = 1$, но ложна в пересечении таких интерпретаций $a = 0, b = 0$.

243. Индукция по количеству отрицаний: если их нет — формула уже хорновская. Индукционный шаг: $\neg a \wedge A \to b \equiv a \vee (A \to b) \equiv ((A \to b) \to a) \to a \equiv (\chi \to a) \to a$, где $\chi \equiv A \to b$ — некоторая суперпозиция хорновских формул, существующая по индукционному предположению.

244. В прямую сторону: все нормальные формулы сохраняют единицу (истинно заключение импликации), поэтому их конъюнкция тоже сохраняет единицу. В обратную сторону. Если функция является константой 1, то она эквивалентна $a \to a$. В противном случае её КНФ не содержит элементарных дизъюнкций, содержащих только отрицания переменных. Значит, каждая элементарная дизъюнкция имеет вид $D \vee a \equiv \neg D \to a$, где $\neg D$ эквивалентна элементарной конъюнкции C. Следовательно, эта КНФ эквивалентна конъюнкции нормальных формул $C \to a$.

245. $x_1 \wedge \dots \wedge x_n \to y_1 \wedge \dots \wedge y_m \equiv \bigwedge\limits_i (x_1 \wedge \dots \wedge x_n \to y_i)$.

246. $N_0 = \{c, d\}$, $N_1 = \{c, d, k\}$, $N_2 = \{c, d, k, h\}$, $N_3 = \{c, d, k, h, g\}$, $N_4 = \{c, d, k, h, g, b\}$, $N_5 = \{c, d, k, h, g, b, a\}$, $N_6 = \{c, d, k, h, g, b, a, e, f\}$. $d, c \to k$; $c, d, k \to h$; $h, d, c \to g$; $d, g \to b$; $b, k \to a$; $d, g, a \to e$.

247. $C_0 = (3, 4, 2, 2, 2, 3)$, $A_0 = \{a, f\}$, $N_0 = \{a, f\}$, $C_1 = (2, 3, 1, 0, 2, 1)$, $A_1 = \{d\}$, $N_1 = \{a, f, d\}$, $C_2 = (2, 2, 1, 0, 1, 0)$, $A_2 = \{g\}$, $N_2 = \{a, f, d, g\}$, $C_3 = (2, 1, 1, 0, 0, 0)$, $A_3 = \{e\}$, $N_3 = \{a, f, d, g, e\}$, $C_4 = (2, 1, 0, 0, 0, 0)$, $A_4 = \{c\}$, $N_4 = \{a, f, d, g, e, c\}$, $C_5 = (1, 0, 0, 0, 0, 0)$, $A_5 = \{h\}$, $N_5 = \{a, f, d, g, e, c, h\}$, $C_6 = (1, 0, 0, 0, 0, 0)$, $A_5 = \varnothing$, $N_5 = \{a, f, d, g, e, c, h\}$. $f, a \to d$; $d, f, a \to g$; $g, d \to e$; $e, f \to c$; $a, c, d, g \to h$.

248. $C_0 = (4, 3, 2, 2, 2, 2)$, $A_0 = \{b, f\}$, $N_0 = \{b, f\}$, $C_1 = (3, 2, 1, 1, 1, 0)$, $A_1 = \{g\}$, $N_1 = \{b, f, g\}$, $C_2 = (3, 2, 0, 1, 1, 0)$, $A_2 = \{e\}$, $N_2 = \{b, f, g, e\}$, $C_3 = (3, 2, 0, 0, 0, 0)$, $A_3 = \{c, d\}$, $N_3 = \{b, f, g, e, c, d\}$, $C_4 = (2, 0, 0, 0, 0, 0)$, $A_4 = \{a\}$, $N_4 = \{b, f, g, e, c, d, a\}$, $C_4 = (1, 0, 0, 0, 0, 0)$, $A_4 = \varnothing$, $N_4 = \{b, f, g, e, c, d, a\}$. $b, f \to g$; $g, b \to e$; $e, f \to c$; $f, e \to d$; $b, c, d \to a$.

Логика предикатов

249. **(а)** $(\forall x)((\exists z)Q(x,z,y) \to (\forall x)(Q(z,x,y) \to (\exists y)(Q(z,y,x) \vee x \approx z)))$;
(б) $(\exists y)((\exists x)(\neg x \approx y \wedge Q(x,x,y) \wedge (\forall x)(Q(x,z,y) \to (\exists z)(Q(x,y,z) \vee Q(z,z,y))))$;
(в) $(\forall x)((\exists z)Q(z,z,x) \wedge (\exists x)Q(y,y,x)) \vee (\exists x)((\exists z)Q(x,x,z) \to (\forall x)Q(x,y,y))$;
(г) $(\exists y)Q(x,y,z) \vee (\exists x)(Q(z,x,y) \wedge (\forall y)Q(y,x,y)) \wedge (\forall z)(\exists x)Q(z,z,x)$.
250. **(а)** Возможны первые две, результат не меняется.
(б) Возможны первые четыре, результат не меняется.
(в) Невозможна только третья. Две первые и две последние: результат не меняется, $(\Phi)^y_z = (\forall x)((\exists z)Q(z,z,x) \wedge (\exists x)Q(z,z,x)) \vee (\exists x)((\exists z)Q(x,x,z) \to (\forall x)(Q(x,z,z)))$.
(г) $(\Phi)^x_z = (\exists y)Q(z,y,z) \vee (\exists x)(Q(z,x,y) \wedge (\forall y)Q(y,x,y)) \wedge (\forall z)(\exists x)Q(z,z,x)$, $(\Phi)^y_z = (\exists y)Q(x,y,z) \vee (\exists x)(Q(z,x,z) \wedge (\forall y)Q(y,x,y)) \wedge (\forall z)(\exists x)Q(z,z,x)$.
251. **(а)** $(\forall x)R(x,x)$;
(б) $\mathrm{aref} = (\forall x)R(x,x)$;
(в) $\mathrm{symm} = (\forall x)(\forall y)(R(x,y) \to R(y,x))$;
(г) $\mathrm{asymm} = (\forall x)(\forall y)(R(x,y) \wedge R(y,x) \to x \approx y)$;
(д) $\mathrm{trans} = (\forall x)(\forall y)(\forall z)(R(x,y) \wedge R(y,z) \to R(x,z))$;
(е) $\mathrm{ref} \wedge \mathrm{asymm} \wedge \mathrm{trans}$;
(ж) $\mathrm{aref} \wedge \mathrm{asymm} \wedge \mathrm{trans} \wedge (\forall x)(\forall y)(R(x,y) \vee R(y,x) \vee x \approx y)$.
252. **(а)** $(\forall y)y \leqslant x$;
(б) $\max(x) = \neg(\exists y)(x \leqslant y \wedge \neg x \approx y)$;
(в) $\max(x) \wedge (\forall y)(\max(y) \to x \approx y)$;
(г) формула $\mathrm{comp}(x,y) = x \leqslant y \vee y \leqslant x$ означает, что x и y сравнимы. Тогда $(\forall y)\,\mathrm{comp}(x,y)$;
(д) $(\forall y)(\max(y) \to \mathrm{comp}(x,y))$;
(е) $(\exists y)(\neg\,\mathrm{comp}(x,y) \wedge (\forall z)(\neg\,\mathrm{comp}(x,z) \to z \approx y))$.
253. **(а)** $M(x) \wedge (\exists z)(P(y,z) \wedge P(z,x))$;
(б) $(\exists y)(\exists z)(P(x,y) \wedge P(x,z) \wedge \neg y \approx z)$;
(в) $(\exists z)(P(x,z) \wedge P(y,z) \wedge \neg(\exists u)(P(x,u) \wedge P(y,u) \wedge \neg u \approx z))$;
(г) $(\exists z)(\exists y)(P(z,x) \wedge P(z,y) \wedge \neg y \approx x \wedge \neg M(y) \wedge \neg(\exists u)(S(y,u) \wedge M(u)))$;
(д) $M(x) \wedge M(y) \wedge (\exists u)(\exists v)(\exists w)(P(u,v) \wedge P(v,x) \wedge P(u,w) \wedge P(w,y) \wedge \neg v \approx w)$;
(е) $(\exists u)(S(x,u) \wedge \neg M(u)) \wedge (\exists y)(M(y) \wedge P(x,y) \wedge (\exists u)(S(y,u) \wedge \neg M(u)) \wedge \neg(\exists z)(\neg z \approx y \wedge M(z) \wedge P(x,z) \wedge (\exists u)(S(z,u) \wedge \neg M(u))))$.
254. **(а)** $\mathbb{R}(x) = (\exists f)V(f,x,x)$;
(б) $F(f) = \neg\mathbb{R}(f)$;
(в) $\mathrm{const}(f) = (\exists x)(\mathbb{R}(x) \wedge (\forall y)(\mathbb{R}(y) \to V(f,y,x)))$;

(г) $\mathrm{fzero}(f) = \mathrm{const}(f) \wedge D(f, f)$;
(д) $\mathrm{zero}(x) = (\exists f)(\mathrm{fzero}(f) \wedge V(f, x, x))$;
(е) $\mathrm{id}(f) = (\forall x)(\mathbb{R}(x) \to V(f, x, x))$;
(ж) $\mathrm{one}(x) = (\exists f)(\exists f')(\mathrm{id}(f) \wedge D(f', f) \wedge V(f', x, x))$;
(з) $\exp(f) = D(f, f) \wedge (\exists e)(\exists o)(\mathrm{zero}(o) \wedge \mathrm{one}(e) \wedge V(f, o, e))$;
(и) $A(x, y, z) = (\exists f)(\exists f')(\exists e)(\exists o)(D(f', f) \wedge \mathrm{zero}(o) \wedge \mathrm{one}(e) \wedge (\forall u)V(f', u, e) \wedge V(f, o, x) \wedge V(f, y, z))$;
(к) $M(x, y, z) = (\exists f)(\exists f')(\exists o)(D(f', f) \wedge \mathrm{zero}(o) \wedge (\forall u)V(f', u, x) \wedge V(f, o, o) \wedge V(f, y, z))$;
(л) $\mathrm{pos}(x) = (\exists z)M(z, z, x)$;
(м) $\mathrm{less}(x, y) = (\exists z)(\mathrm{pos}(z) \wedge A(x, z, y) \wedge \neg x \approx y)$;
(н) $\mathrm{surj}(f) = (\forall y)(\mathbb{R}(y) \to (\exists x)V(f, x, y))$;
(о) $\mathrm{bij}(f) = \mathrm{surj}(f) \wedge (\forall y)(\forall x_1)(\forall x_2)(V(f, x_1, y) \wedge V(f, x_2, y) \to x_1 \approx x_2)$;
(п) вторая производная синуса противоположна ему: $\sin(f) = (\exists f')(\exists f'')(\exists o)(\exists e)(D(f', f) \wedge D(f'', f') \wedge \mathrm{zero}(o) \wedge \mathrm{one}(e) \wedge (\forall x)(\forall y)(\forall z)(V(f, x, y) \wedge V(f'', x, z) \to A(y, z, o)) \wedge V(f, o, o) \wedge V(f', o, e))$;
(р) наименьшее положительное число, синус которого равен нулю: $\pi(x) = (\exists s)(\exists o)(\sin(s) \wedge \mathrm{zero}(o) \wedge V(s, x, o) \wedge \mathrm{less}(o, x) \wedge (\forall u)(V(s, u, o) \wedge \mathrm{less}(o, u) \to u \approx x \vee \mathrm{less}(x, u)))$;
(с) неотрицательное число, синус произведения его на π равен нулю: $\Omega(x) = \mathrm{pos}(x) \wedge (\exists o)(\exists p)(\exists u)(\exists s)(\mathrm{zero}(o) \wedge \pi(p) \wedge M(x, p, u) \wedge \sin(s) \wedge V(s, u, o))$.

255. **(а)** $\mathrm{even}(x) = (\exists u)A(u, u, x)$;
(б) $\mathrm{coprime}(x, y) = (\forall z)(\forall u)(\forall v)(M(z, u, x) \wedge M(z, v, y) \to z \approx 1)$;
(в) $\mathrm{less}(x, y) = (\exists z)(A(x, z, y) \wedge \neg z \approx 0)$; $\mathrm{between}(x, y, z) = (\mathrm{less}(x, z) \wedge \mathrm{less}(z, y)) \vee (\mathrm{less}(y, z) \wedge \mathrm{less}(z, x))$;
(г) $\text{НОД}(x, y, z) = (\exists u)(\exists v)(M(x, u, y) \wedge M(x, v, z) \wedge \mathrm{coprime}(u, v))$;
(д) $\text{НОК}(x, y, z) = (\forall u)(\neg\,\mathrm{less}(u, y) \wedge \mathrm{less}(u, z) \to (\exists v)M(u, v, x)) \wedge (\forall w)(\mathrm{less}(w, x) \to (\exists u)(\neg\,\mathrm{less}(u, y) \wedge \mathrm{less}(u, z) \wedge \neg(\exists v)M(u, v, w)))$.

256. **(а)** $\mathrm{empty}(x) = (\forall y)x \subseteq y$;
(б) $(\forall y)y \subseteq x$;
(в) $\mathrm{one}(x) = \neg\,\mathrm{empty}(x) \wedge (\forall y)(y \subseteq x \to \mathrm{empty}(y) \vee y \approx x)$;
(г) $x \subseteq y \wedge x \subseteq z \wedge (\forall u)(u \subseteq y \wedge u \subseteq z \to u \subseteq x)$;
(д) $y \subseteq x \wedge z \subseteq x \wedge (\forall u)(y \subseteq u \wedge z \subseteq u \to x \subseteq u)$;
(е) $x \subseteq y \wedge (\forall u)(\mathrm{one}(u) \wedge u \subseteq y \wedge \neg u \subseteq z \to u \subseteq x)$.

257. **(а)** $L'(x, y, z) = L(x, y, z) \vee L(y, z, x) \vee L(z, x, y)$; $\mathrm{triangle}(x, y, z) = \neg L'(x, y, z)$;
(б) $\mathrm{middle}(x, y, z) = L(y, x, z) \wedge E(y, x, x, z)$;
(в) $\mathrm{bissect}(x, y, z, u) = (\exists y')(\exists z')(\exists u')(L(x, y, y') \wedge L(x, z, z') \wedge L(x, u, u') \wedge E(x, y', x, z') \wedge E(x, y', x, u') \wedge E(u', y', y', z'))$;

(г) $\mathrm{right}(x,y,z) = (\exists u)(\mathrm{middle}(x,u,z) \wedge E(u,x,u,y))$;
(д) $\mathrm{circle}(x,y,z,u) = (\exists o)(E(o,x,o,y) \wedge E(o,y,o,z) \wedge E(o,z,o,u))$;
(е) $\mathrm{acute}(x,y,z) = (\exists x')(\exists y')(\exists z')(L(x,y',z) \wedge L(y,z',x) \wedge L(z,x',y) \wedge \mathrm{right}(x,x',y) \wedge \mathrm{right}(y,y',z) \wedge \mathrm{right}(z,z',x))$;
(ж) $\mathrm{less}(x,y,z,u) = (\exists v)(L(z,v,u) \wedge \neg v \approx u \wedge E(x,y,z,v))$;
(з) $\mathrm{inside}(x,y,z) = \mathrm{less}(y,x,y,z)$.

258. **(а)–(г)** Сигнатура: $S^{(1)}(x)$: «x — студент», $D^{(1)}(x)$: «x — дисциплина», $T^{(2)}(x,y)$: «x изучает y», $R^{(3)}(x,y,z)$: «x сдал y на z баллов», $x <^{(2)} y$: «x меньше y».
(а) $\neg(\forall x)(S(x) \to T(x, \text{анализ}) \wedge T(x, \text{история}))$;
(б) $(\exists x)(S(x) \wedge \neg(\exists b)R(x, \text{ДМ}, b) \wedge (\forall u)(\neg u \approx x \to (\exists b)R(u, \text{ДМ}, b)))$;
(в) $(\exists x)(S(x) \wedge (\forall y)R(x,y,100) \wedge (\forall u)(\neg u \approx x \to (\exists y)(\exists b)(R(u,y,b) \wedge \neg b \approx 100)))$;
(г) $(\exists b)(\exists x)(R(x, \text{ДМ}, b) \wedge (\forall x)(\forall v)(R(x, \text{Инф}, v) \to v < b))$.
(д) Сигнатура: $B^{(1)}(x)$: «x — брадобрей», $S^{(2)}(x,y)$: «x бреет y». $(\exists b)(B(b) \wedge (\forall x)(S(b,x) \to \neg B(x,x)))$.
(е) Сигнатура: $P^{(1)}(x)$: «x — политик», $T^{(1)}(x)$: «x — момент времени», $L^{(3)}(x,y,z)$: «x лжёт y в момент z». $(\exists p)(P(p) \wedge (\exists z)(\forall y)(\neg T(y) \to L(p,y,z))) \wedge (\exists p)(P(p) \wedge (\exists y)(\forall z)(T(z) \to L(p,y,z))) \wedge \neg(\exists p)(\forall y)(\forall z)(\neg T(y) \wedge T(z) \to L(p,y,z))$.

259. $(\exists x_1)\dots(\exists x_k)(\bigwedge\limits_{i \neq j} \neg x_i \approx x_j \wedge (\forall y)\bigvee\limits_i y \approx x_i)$.

260. $(\forall x_1)\dots(\forall x_n)(\exists y)(F(x_1,\dots,x_n,y) \wedge \neg(\exists z)(\neg z \approx y \wedge F(x_1,\dots,x_n,y)))$.

261. **(а)** Ложна: взять $x = 1$;
(б) истинна: каждое число либо чётно, либо нечётно;
(в) истинна: если $x \mid y$ и $y \mid z$, то $x \mid z$;
(г) ложна: утверждается, что каждое число является квадратом или кубом;
(д) истинна: если $x^2 \neq x$ (то есть x не равен нулю или единице), то разность $x^2 - x$ больше или равна двум.

262. **(а)** Истинна: каждый отрезок xy можно продолжить на такое же расстояние;
(б) ложна: не каждые три точки x, y, z лежат на одной окружности;
(в) истинна: у каждого отрезка xy есть серединный перпендикуляр uv;
(г) истинна: на некотором отрезке yz есть точка u, расстояние от которой до x не совпадает с расстоянием до x от других точек v этого отрезка;
(д) истинна: если $|xz| < |xy|$, то u — точка пересечения отрезка xy и окружности с центром x и радиусом xz.

263. **(а)** Ложна: даже если производные совпадают, исходные функции могут отличаться;

(б) истинна: для любых двух функций можно построить их композицию;
(в) ложна: не каждая функция имеет обратную;
(г) истинна: если производные функций совпадают, то функции отличаются на константу. Если эта константа ноль, то сами функции тоже совпадают;
(д) истинна, в качестве x можно взять экспоненту exp;
(е) истинна, в качестве x можно взять функцию $^1/_{\exp}$.

264. Истинны (а), (в), (г), (д), (е).

265. **(а)** Если $I((\exists x)(\Phi \vee \Psi)) = 1$, то $(I)^x_a(\Phi \vee \Psi) = 1$ для некоторого a. Следовательно, $(I)^x_a(\Phi) = 1$ или $(I)^x_a(\Psi) = 1$, поэтому $I((\exists x)\Phi) = 1$ или $I((\exists x)\Psi) = 1$, значит, $I((\exists x)\Phi \vee (\exists x)\Psi) = 1$. Если $I((\exists x)\Phi \vee (\exists x)\Psi) = 1$, то $I((\exists x)\Phi) = 1$ или $I((\exists x)\Psi) = 1$. Значит, существует a, для которого $(I)^x_a(\Phi) = 1$, или существует b, для которого $(I)^x_b(\Psi) = 1$. В первом случае получаем $(I)^x_a(\Phi \vee \Psi) = 1$ и $I((\exists x)(\Phi \vee \Psi)) = 1$, во втором — $(I)^x_b(\Phi \vee \Psi) = 1$ и $I((\exists x)(\Phi \vee \Psi)) = 1$.
(б) Если $I((\forall x)(\Phi \wedge \Psi)) = 1$, то $(I)^x_a(\Phi \wedge \Psi) = 1$ для любого a. Поэтому для любого a получаем $(I)^x_a(\Phi) = 1$ и $(I)^x_a(\Psi) = 1$, следовательно, $I((\forall x)\Phi) = 1$ и $I((\forall x)\Psi) = 1$, значит, $I((\forall x)\Phi \wedge (\forall x)\Psi) = 1$. Если $I((\forall x)\Phi \wedge (\forall x)\Psi) = 1$, то $I((\forall x)\Phi) = 1$ и $I((\forall x)\Psi) = 1$. Значит, для любого a выполнено $(I)^x_a(\Phi) = 1$ и $(I)^x_a(\Psi) = 1$, поэтому $(I)^x_a(\Phi \wedge \Psi) = 1$, и $I((\forall x)(\Phi \wedge \Psi)) = 1$.
(в) Если $x = y$, то формулы просто совпадают и эквивалентны. Пусть $x \neq y$. Если $I((\exists x)(\exists y)\Phi) = 1$, то существуют a и b такие, что $((I)^x_a)^y_b(\Phi) = 1$. Тогда $((I)^x_a)^y_b = ((I)^y_b)^x_a$, поэтому $((I)^y_b)^x_a(\Phi) = 1$, $(I)^y_b((\exists x)\Phi) = 1$, $I((\exists y)(\exists x)\Phi) = 1$.
(г) Аналогично (в).

266. **(а)** Если $I((\exists x)(x \approx y \wedge \Phi)) = 1$, то $(I)^x_a(x \approx y \wedge \Phi) = 1$ для некоторого a. Из последнего получаем, что значение y тоже равно a, а формула Φ истинна. Следовательно, $I((\Phi)^x_y) = (I)^x_a(\Phi) = 1$. Обратно: из $I((\Phi)^x_y) = 1$ получаем $(I)^x_a(\Phi) = 1$, где a — значение y. Тогда $(I)^x_a(x \approx y \wedge \Phi) = 1$ и $I((\exists x)(x \approx y \wedge \Phi)) = 1$.
(б) Если $I((\forall x)(x \approx y \to \Phi)) = 1$, то $(I)^x_b(x \approx y \to \Phi) = 1$ для всех b. В частности $(I)^x_a(x \approx y \to \Phi) = 1$, где a — значение y. В последнем случае равенство истинно, следовательно, $I((\Phi)^x_y) = (I)^x_a(\Phi) = 1$. Обратно: из $I((\Phi)^x_y) = 1$ получаем $(I)^x_a(\Phi) = 1$, где a — значение y. Тогда $(I)^x_a(x \approx y \to \Phi) = 1$. Для всех $b \neq a$ в силу ложности равенства тоже получаем $(I)^x_b(x \approx y \to \Phi) = 1$. следовательно, $I((\forall x)(x \approx y \to \Phi)) = 1$.

(в) $\Phi \Rightarrow (\exists x)\Phi$, применяем задачу 141.
(г) $(\forall x)\Phi \Rightarrow \Phi$, применяем задачу 141.

267. **(а)** Если $I((\exists x)(\Phi \wedge \Psi)) = 1$, то $(I)^x_a(\Phi \wedge \Psi) = 1$ для некоторого a. Следовательно, $(I)^x_a(\Phi) = 1$ и $(I)^x_a(\Psi) = 1$, поэтому $I((\exists x)\Phi) = 1$ и $I((\exists x)\Psi) = 1$, значит $I((\exists x)\Phi \wedge (\exists x)\Psi) = 1$.
(б) Если $I((\forall x)\Phi \vee (\forall x)\Psi) = 1$, то $I((\forall x)\Phi) = 1$ или $I((\forall x)\Psi) = 1$. Рассмотрим случай $I((\forall x)\Phi) = 1$, тогда $(I)^x_a(\Phi) = 1$ для любого a, значит, $(I)^x_a(\Phi \vee \Psi) = 1$ для любого a и $I((\forall x)(\Phi \vee \Psi)) = 1$. Аналогично в случае $I((\forall x)\Psi) = 1$.
(в) Пусть $I((\forall x)(\Phi \to \Psi)) = 1$. Из этого следует, что $(I)^x_a(\Phi \to \Psi) = 1$ для любого a. Если $I((\forall x)\Phi) = 0$, то правая формула тоже истинна. Допустим, $I((\forall x)\Phi) = 1$, то есть $(I)^x_a(\Phi) = 1$ для любого a. Но тогда $(I)^x_a(\Psi) = 1$ для любого a, поэтому $I((\forall x)\Psi) = 1$. Следовательно, $I((\forall x)\Phi \to (\forall x)\Psi) = 1$.
(г) Если $I((\exists x)(\forall y)\Phi) = 1$, то $(I)^x_a((\forall y)\Phi) = 1$ для некоторого a. Последнее означает $((I)^x_a)^y_b(\Phi) = 1$ для всех b. Далее возможны два случая: $x = y$ или $x \neq y$. Если $x = y$, то $((I)^x_a)^y_b = (I)^y_b$. Получаем $(I)^y_b(\Phi) = 1$ для всех b, в частности, при $b = a$: $(I)^x_a(\Phi) = 1$. Значит, $I((\exists x)\Phi) = 1$ и $I((\forall y)(\exists x)\Phi) = 1$. Если $x \neq y$, то $((I)^x_a)^y_b = ((I)^y_b)^x_a$. Тогда $(I)^y_b((\exists x)\Phi) = 1$ для всех b и $I((\forall y)(\exists x)\Phi) = 1$.

268. **(а)** и **(в)** $\Phi = x \approx 1$, $\Psi = x \approx 2$;
(б) $\Phi = x \approx 0$, $\Psi = (\exists y)A(y, 1, x)$;
(г) $\Phi = A(y, 1, x)$.

269. **(а)** $(\exists z)(\forall x)(\exists y)(\forall u)(Q(x, y) \wedge Q(z, u))$;
(б) $(\forall x)(\exists y)(\exists u)(\forall v)(\neg Q(x, y) \vee Q(v, u))$;
(в) $(\exists u)(\forall v)(\exists w)(\exists y)(Q(u, v) \wedge \neg Q(u, z) \wedge (\neg Q(w, x) \vee Q(y, x)))$.

270. **(а)** $(\exists x)(\Phi \to \Psi) \equiv (\exists x)(\neg\Phi \vee \Psi) \equiv (\exists x)\neg\Phi \vee (\exists x)\Psi \equiv \neg(\forall x)\Phi \vee (\exists x)\Psi \equiv (\forall x)\Phi \to (\exists x)\Psi$;
(б) $(\forall x)(\Theta \to \Phi) \equiv (\forall x)(\neg\Theta \vee \Phi) \equiv \neg\Theta \vee (\forall x)\Phi \equiv \Theta \to (\forall x)\Phi$;
(в) $(\forall x)(\Phi \to \Theta) \equiv (\forall x)(\neg\Phi \vee \Theta) \equiv (\forall x)\neg\Phi \vee \Theta \equiv \neg(\exists x)\Phi \vee \Theta \equiv (\exists x)\Phi \to \Theta$.

271. Для каждой атомной подформулы Θ введём пропозициональную переменную a_Θ. Построим Ψ заменив каждую Θ на a_Θ. Рассмотрим интерпретацию логики предикатов $I = (A, \nu, \sigma)$, в которой множество A бесконечно, а все значения предметных переменных $\sigma(x)$ попарно различны. Пусть J — произвольная интерпретация логики высказываний. Доопределим интерпретацию I до I_J, задав значения предикатных символов ν так, чтобы

$I_J(\Theta) = J(a_\Theta)$ для всех атомных подформул Θ. Противоречий возникнуть не может, так как все формулы Θ попарно различны, как и значения предметных переменных. Тогда $I_J(\Phi) = J(\Psi)$. Поэтому если формула Φ тождественно истинна, то и Ψ тождественно истинна. Обратно, пусть Φ получено из Ψ заменой каждой пропозициональной переменной a на Θ_a. Для каждой интерпретации I логики предикатов построим интерпретацию J_I логики высказываний, положив $J_I(a) = I(\Theta_a)$. Тогда $J_I(\Psi) = I(\Phi)$. Следовательно, если Ψ тождественно истинна, то и Φ тождественно истинна.

272. (а) $(\forall x)(\Phi \to \Psi) \to ((\forall x)\Phi \to (\forall x)\Psi) \equiv \neg(\forall x)(\Phi \to \Psi) \vee (\neg(\forall x)\Phi \vee (\forall x)\Psi) \equiv (\exists x)\neg(\Phi \to \Psi) \vee (\exists x)\neg\Phi \vee (\forall x)\Psi \equiv (\exists x)(\Phi \wedge \neg\Psi) \vee (\exists x)\neg\Phi \vee (\forall x)\Psi \equiv (\exists x)((\Phi \wedge \neg\Psi) \vee \neg\Phi) \vee (\forall x)\Psi \equiv (\exists x)(\neg\Psi \vee \neg\Phi) \vee (\forall x)\Psi \equiv (\exists x)\neg\Psi \vee (\exists x)\neg\Phi \vee (\forall x)\Psi \equiv \neg(\forall x)\Psi \vee (\exists x)\neg\Phi \vee (\forall x)\Psi \equiv 1$;

(б) $(\forall x)(\Phi \to \Psi) \to ((\exists x)\Phi \to (\exists x)\Psi) \equiv \neg(\forall x)(\neg\Phi \vee \Psi) \vee (\neg(\exists x)\Phi \vee (\exists x)\Psi) \equiv ((\exists x)\neg(\neg\Phi \vee \Psi) \vee (\exists x)\Psi) \vee \neg(\exists x)\Phi \equiv (\exists x)((\neg\neg\Phi \wedge \neg\Psi) \vee \Psi) \vee \neg(\exists x)\Phi \equiv (\exists x)(\Phi \vee \Psi) \vee \neg(\exists x)\Phi \equiv (\exists x)\Phi \vee (\exists x)\Psi \vee \neg(\exists x)\Phi \equiv 1$;

(в) $(\forall x)\Phi \to ((\exists x)\Psi \to (\exists x)(\Phi \wedge \Psi)) \equiv \neg(\forall x)\Phi \vee (\neg(\exists x)\Psi \vee (\exists x)(\Phi \wedge \Psi)) \equiv (\exists x)\neg\Phi \vee \neg(\exists x)\Psi \vee (\exists x)(\Phi \wedge \Psi) \equiv (\exists x)(\neg\Phi \vee (\Phi \wedge \Psi)) \vee \neg(\exists x)\Psi \equiv (\exists x)(\neg\Phi \vee \Psi) \vee \neg(\exists x)\Psi \equiv (\exists x)\neg\Phi \vee (\exists x)\Psi \vee \neg(\exists x)\Psi \equiv 1$;

(г) $((\forall x)\Phi \to (\forall x)\neg\Psi) \to ((\forall x)(\Phi \to \Psi) \to (\exists x)\neg\Phi) \equiv \neg(\neg(\forall x)\Phi \vee (\forall x)\neg\Psi) \vee (\neg(\forall x)(\neg\Phi \vee \Psi) \vee (\exists x)\neg\Phi) \equiv (\neg\neg(\forall x)\Phi \wedge \neg(\forall x)\neg\Psi) \vee ((\exists x)\neg(\neg\Phi \vee \Psi) \vee (\exists x)\neg\Phi) \equiv ((\forall x)\Phi \wedge (\exists x)\neg\neg\Psi) \vee ((\exists x)(\neg\neg\Phi \wedge \neg\Psi) \vee (\exists x)\neg\Phi) \equiv ((\forall x)\Phi \wedge (\exists x)\Psi) \vee ((\exists x)(\Phi \wedge \neg\Psi) \vee (\exists x)\neg\Phi) \equiv ((\forall x)\Phi \wedge (\exists x)\Psi) \vee (\exists x)((\Phi \wedge \neg\Psi) \vee \neg\Phi) \equiv ((\forall x)\Phi \wedge (\exists x)\Psi) \vee (\exists x)(\neg\Psi \vee \neg\Phi) \equiv ((\forall x)\Phi \wedge (\exists x)\Psi) \vee (\exists x)\neg\Psi \vee (\exists x)\neg\Phi \equiv ((\forall x)\Phi \wedge (\exists x)\Psi) \vee (\exists x)\neg\Psi \vee \neg(\forall x)\Phi \equiv (((\forall x)\Phi \wedge (\exists x)\Psi) \vee \neg(\forall x)\Phi) \vee (\exists x)\neg\Psi \equiv ((\exists x)\Psi \vee (\exists x)\neg\Psi) \vee \neg(\forall x)\Phi \equiv (\exists x)(\Psi \vee \neg\Psi) \vee \neg(\forall x)\Phi \equiv (\exists x)1 \vee \neg(\forall x)\Phi \equiv 1 \vee \neg(\forall x)\Phi \equiv 1$.

273. Пусть x_0 — элемент предметной области из первой части формулы. Согласно второй части получаем, что существуют $x_1, x_2, \ldots$ такие, что $R(x_0, x_1), R(x_1, x_2), \ldots$ Если бы среди них были равные: $x_i = x_j$, $i < j$, то, взяв наименьшее из таких i, получим противоречие. В самом деле, выполнено $R(x_{j-1}, x_j)$, то есть $R(x_{j-1}, x_i)$, поэтому $i \neq 0$. Значит, будет выполнено $R(x_{i-1}, x_i)$ и $R(x_{j-1}, x_i)$, согласно третьей части получаем $x_{i-1} = x_{j-1}$, что противоречит минимальности i. Взять натуральные числа и отношение $R(x, y)$, означающее $x + 1 = y$.

Логика предикатов и базы данных

274. (а) $\{(a_1, 7, c_1), (a_1, 5, c_2)\}$;

(б) $\{(7,\ 2), (5,\ 2), (4,\ 2), (7,\ 8), (5,\ 8), (4,\ 8), (7,\ 7), (5,\ 7), (4,\ 7), (7,\ 4), (5,\ 4), (4,\ 4), (7,\ 5), (5,\ 5), (4,\ 5)\}$;

(в) $\{(a_1,\ 7,\ c_1,\ 2),\ (a_1,\ 5,\ c_2,\ 8),\ (a_2,\ 7,\ c_1,\ 2),\ (a_1,\ 7,\ c_1,\ 4),\ (a_1,\ 5,\ c_2,\ 7),\ (a_2,\ 7,\ c_1,\ 4),\ (a_1,\ 5,\ c_2,\ 5)\}$;
(г) $\{(a_1,c_1),(a_1,c_2),(a_2,c_1),(a_3,c_3)\}$;
(д) $\{(a_1,\ 5,\ c_2,\ c_1,\ 2),\ (a_3,\ 4,\ c_3,\ c_1,\ 2),\ (a_1,\ 5,\ c_2,\ c_1,\ 4),\ (a_1,\ 7,\ c_1,\ c_2,\ 5),\ (a_2,\ 7,\ c_1,\ c_2,\ 5)\}$.

275. **(а)** выполнено. Если $\bar{a} \in (R \cup S : \Theta)$, то $\bar{a} \in (R \cup S)$ и $\bar{a}$ удовлетворяет Θ. Следовательно, $\bar{a} \in R$ и $\bar{a}$ удовлетворяет Θ или же $\bar{a} \in S$ и $\bar{a}$ удовлетворяет Θ, то есть $\bar{a} \in (R : \Theta) \cup (S : \Theta)$. Аналогично в обратную сторону.
(б)–(в) выполнены, доказательство аналогично.
(г) выполнено. Если $\bar{a} \in (R \cup S)[X]$, то $(\bar{a},\bar{b}) \in R \cup S$ для некоторого $\bar{b}$. Следовательно, $(\bar{a},\bar{b}) \in R$ или $(\bar{a},\bar{b}) \in S$, поэтому $\bar{a} \in R[X]$ или $\bar{a} \in S[X]$ и $\bar{a} \in R[X] \cup S[X]$. Если $\bar{a} \in R[X] \cup S[X]$, то $\bar{a} \in R[X]$ или $\bar{a} \in S[X]$. Значит, $(\bar{a},\bar{b}) \in R$ или $(\bar{a},\bar{c}) \in S$ для каких-то $\bar{b}$ и $\bar{c}$. Получаем $(\bar{a},\bar{b}) \in R \cup S$ или $(\bar{a},\bar{c}) \in R \cup S$ и в любом случае $\bar{a} \in (R \cup S)[X]$.
(д)–(е) могут не выполняться. Пусть $R = \{(a,b_1)\}$, $S = \{(a,b_2)\}$, где $b_1 \neq b_2$. Тогда $R[A] = S[A] = \{(a)\}$, поэтому $R[A] \cap S[A] = \{(a)\}$ и $R[A] \setminus S[A] = \varnothing$. С другой стороны $R \cap S = \varnothing$, $R \setminus S = \{(a,b_1)\}$, поэтому $(R \cap S)[A] = \varnothing$, $(R \setminus S)[A] = \{(a)\}$.

276. $A =$ Комнаты[НомерКомнаты] $\setminus$ Оборудование[НомерКомнаты],
(Сотрудники $\bowtie$ Комнаты $\bowtie$ A)[ФИО],
$(\exists n)(\exists o)(\exists d)(\exists z)(\text{Сотрудники}(n,f,o,d,z) \quad \wedge$
$\neg(\exists e)(\exists nk)(\exists ob)(\text{Комнаты}(n,e,nk) \wedge \text{Оборудование}(e,nk,ob)))$.

277. $C = R[A_1,\ldots,A_n]$, $D = (C \times S) \setminus R$, $Q = C \setminus (D[A_1,\ldots,A_n])$. Если формулы Φ_R и Φ_S имеют свободные переменные $x_1,\ldots,x_n,y_1,\ldots,y_m$ и $y_1,\ldots,y_m$ соответственно, то $\Phi_Q = (\exists y_1)\ldots(\exists y_m)\Phi_R \wedge (\forall y_1)\ldots(\forall y_m)(\Phi_S \to \Phi_R)$.

278. **(а)** `SELECT DISTINCT ФИО FROM Сотрудники WHERE Оклад > 5500;`
(б) `SELECT DISTINCT Отдел FROM Сотрудники WHERE Оклад > 8000;`
(в) `SELECT DISTINCT Должность, Оклад FROM Сотрудники;`
(г) `SELECT DISTINCT ФИО FROM Сотрудники, Комнаты WHERE Отдел = 'Торговый' AND Оклад >= 6000 AND Оклад <= 6500 AND Номер = НомерСотрудника AND Этаж != 3;`
(д) `SELECT DISTINCT НомерКомнаты FROM Комнаты EXCEPT SELECT DISTINCT НомерКомнаты FROM Сотрудники, Комнаты WHERE Оклад >= 7500 AND Номер = НомерСотрудника.`

279. Второе не выполнено.

280. **(а)** $(\forall ns_1)(\forall nk_1)(\forall ns_2)(\forall nk_2)((\exists e)\text{Комнаты}(ns_1,e,nk_1) \wedge (\exists e)\text{Комнаты}(ns_2,e,nk_2) \to ns_1 \approx ns_2 \wedge nk_1 \approx nk_2)$;

(б) $(\forall n)((\exists f)(\exists o)(\exists d)(\exists z)$Сотрудники$(n, f, o, d, z) \quad \to$ $(\exists e)(\exists nk)$Комнаты$(n, e, nk))$;
(в) $(\forall nk)((\exists n)$Комнаты$(n, 2, nk) \to 10 < nk \wedge nk < 20) \wedge$ $(\forall nk)((\exists n)$Комнаты$(n, 3, nk) \to 20 < nk)$.
Выполнено только (а).

Ориентированные графы

281. **(а)** 2^{n^2}; **(б)** 2^{n^2-n}; **(в)** $3^{n(n-1)/2} \cdot 2^n$; **(г)** $3^{n(n-1)/2}$; **(д)** $2^{n(n-1)/2}$.
282. Пути не могут иметь сколь угодно большую длину (иначе вершины начнут повторяться и будет цикл). Тогда если взять путь наибольшей длины, то его первая вершина будет истоком, а последняя — стоком.
283. Выполнено только (в): если взять путь наибольшей длины, исходящий из данной вершины, то он должен закончиться в стоке.
(а) не выполнено, достаточно взять путь из одной вершины (v), которая не является ни стоком, ни истоком.
(б) не выполнено, если граф состоит из двух рёбер, не имеющих общих вершин.
(г) не выполнено для графа в виде однонаправленной линии.
284. Если v — сток (задача 282 на стр. 88), то он получает номер n. По индукционному предположению остальные вершины можно пронумеровать $v_1, \dots, v_{n-1}$.
285. Aw — множество вершин, из которых ведут рёбра в W, $w^T A$ — множество вершин, в которые ведут рёбра из W.
286. Пусть $V = \{v_i : i < n\}$. Тогда положим $E = \{(v_i, v_j) : j \equiv i+1, \dots, i+r \pmod{n}\}$. Например, для $n = 11$ и $r = 3$ см. рис. 55.
287. Допустим, что $u \neq v$ и нет ребра (v, u). Пусть количество вершин равно n, а полустепень исхода v равна m, X — множество вершин, куда из v ведут рёбра, Y — откуда в u ведут рёбра. Тогда $|X| = m$, $|V \setminus (Y \cup \{u\})| \leqslant m$, $|Y| \geqslant n - m - 1$. Но $|X \cup Y| \leqslant n - 2$, так как ни X, ни Y не содержат ни v, ни u. Следовательно, X и Y пересекаются.
288. Для двух вершин очевидно. Пусть для турнира из n вершин есть простой путь $(v_1, \dots, v_n)$. Рассмотрим новую вершину v. Если для v есть только рёбра вида (v_i, v), то новый простой путь: $(v_1, \dots, v_n, v)$. Если есть ребро (v, v_1), то новый простой путь: $(v, v_1, \dots, v_n)$. Иначе есть ребро (v_1, v), пусть i — наименьший номер такой, что нет ребра (v_{i+1}, v). Тогда есть рёбра (v_i, v) и (v, v_{i+1}) и новый простой путь: $(v_1, \dots, v_i, v, v_{i+1}, \dots, v_n)$.

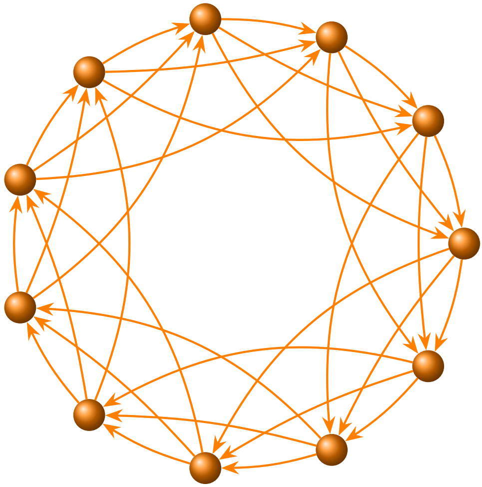


Рис. 55: Граф из задачи 286.

289. Построить граф, где вершины — команды, а стрелки направлены от выигравшей к проигравшей. Тогда этот граф — турнир, далее применяем задачу 287.

290. Пусть граф полусвязен. Рассмотрим путь $P = (v_0, \ldots, v_n)$, солджержащий наибольшее количество различных вершин. Допустим, что P не содержит вершину v. Если есть путь P_0' из v в v_0, то P можно удлинить, добавив P_0' к началу, в результате чего количество различных вершин возрастёт: добавится v. Это невозможно, следовательно, P_0' отсутствует. Аналогично, если есть путь P_n'' из v_n в v, то P можно удлинить, добавив P_n'' к концу, с тем же результатом. Это невозможно, следовательно, P_n'' отсутствует, значит существует путь P_n' из v в v_n. Рассмотрим вершину с наибольшим номером v_i, для которой нет пути P_i' из v в v_i. Как мы доказали, $i < n$. Следовательно, существует путь P_{i+1}' из v в v_{i+1}. Так как нет пути P_i' из v в v_i, то есть путь P_i'' из v_i в v. Следовательно, P можно расширить до пути $(v_0, \ldots, v_i, P_i'', v, P_{i+1}', v_{i+1}, \ldots, v_n)$, который содержит больше различных вершин, чем P, что невозможно. Значит, P содержит все вершины. Если есть путь, проходящий через все вершины, то любая пара вершин лежит на этом пути и из одной из них по этому пути можно попасть в другую.

291. **(а)** Граф с петлями у каждой вершины,
(б) полный граф, то есть со всеми возможными рёбрами.

$$A_{\mathfrak{G}} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}; \qquad A_{\mathfrak{G}^*} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Рис. 56: Матрицы смежности и достижимости.

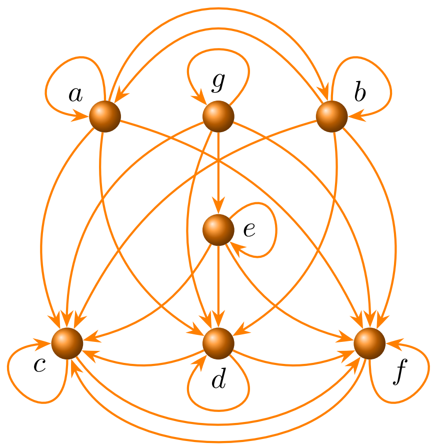


Рис. 57: Граф достижимости.

292. $A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$, $(a(b,c,a), b(a,b), c(a,d), d(b))$.

293. Рис. 56. Граф достижимости изображён на рис. 57. Базы: $\{a,g\}$, $\{b,g\}$.

294. Граф $\mathfrak{G}_1$: рис. 58; $(a(), b(), c(a,b,d), d(a,f), e(d), f(e))$. Граф $\mathfrak{G}_2$: рис. 59; $(a(c,d), b(c), c(d), d(e,f), e(c), f())$.

$$A = \begin{pmatrix} 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 1&1&0&1&0&0\\ 1&0&0&0&0&1\\ 0&0&0&1&0&0\\ 0&0&0&0&1&0 \end{pmatrix}; \qquad A^* = \begin{pmatrix} 1&0&0&0&0&0\\ 0&1&0&0&0&0\\ 1&1&1&1&1&1\\ 1&0&0&1&1&1\\ 1&0&0&1&1&1\\ 1&0&0&1&1&1 \end{pmatrix}$$

Рис. 58: Матрицы смежности и достижимости для графа $\mathfrak{G}_1$.

$$A = \begin{pmatrix} 0&0&1&1&0&0\\ 0&0&1&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&1\\ 0&0&1&0&0&0\\ 0&0&0&0&0&0 \end{pmatrix}; \qquad A^* = \begin{pmatrix} 1&0&1&1&1&1\\ 0&1&1&1&1&1\\ 0&0&1&1&1&1\\ 0&0&1&1&1&1\\ 0&0&1&1&1&1\\ 0&0&0&0&0&1 \end{pmatrix}$$

Рис. 59: Матрицы смежности и достижимости для графа $\mathfrak{G}_2$.

295. Кроме петель: из v_5 достижимы все, из v_1: v_2, v_3, v_4; из v_3: v_2; шесть новых рёбер, включая две петли.

296. Если компонента сильной связности K содержит более одного элемента: u и v, $u \neq v$, то пути из u в v и из v в u образуют цикл. Если в графе есть цикл, то его вершины попадут в одну компоненту.

297. Пусть $V = \{1, \ldots, n\}$, $E = \{(i,j) : i < j\}$. Если рёбер больше, и нет петель, то есть два ребра (u,v) и (v,u), образующие цикл.

298. Все истоки должны попасть в любую базу, так как они ниоткуда больше не достижимы. Если для любой вершины v рассмотреть самый длинный путь через неё проходящий, то он должен начинаться с истока (задача 282). Следовательно, множество истоков является порождающим.

299. Рис. 60; $\{a,f\}, \{b,e\}, \{c\}, \{d,g,h\}$.

300. Рис. 61; $\{a,d\}, \{b,c,g,h\}, \{e,f\}$.

301. Рис. 62; $\{a,c,d,g\}, \{b\}, \{e,h\}, \{f\}$.

302. $\mathfrak{G}_1$: $K_1 = \{a,d,g\}$, $K_2 = \{b,f\}$, $K_3 = \{c,e,h\}$, из K_1 строго достижимы K_2 и K_3, из K_2 — K_3. $\mathfrak{G}_2$: $K_1 = \{a\}$, $K_2 = \{c,d,f\}$, $K_3 = \{b,g,h\}$, $K_4 = \{e\}$, из K_1 строго достижимы K_2, K_3, K_4, из K_2 — K_3, K_4, из K_3 — K_4.

$$\begin{pmatrix} 0&0&0&1&0&1&0&1\\ 0&0&0&0&1&0&0&0\\ 0&1&0&0&1&0&0&0\\ 0&0&0&0&0&0&1&0\\ 0&1&0&1&0&0&0&1\\ 1&1&1&1&1&0&1&1\\ 0&0&0&0&0&0&0&1\\ 0&0&0&1&0&0&1&0 \end{pmatrix};\begin{pmatrix} 1&1&1&1&1&1&1&1\\ 0&1&0&1&1&0&1&1\\ 0&1&1&1&1&0&1&1\\ 0&0&0&1&0&0&1&1\\ 0&1&0&1&1&0&1&1\\ 1&1&1&1&1&1&1&1\\ 0&0&0&1&0&0&1&1\\ 0&0&0&1&0&0&1&1 \end{pmatrix};\begin{pmatrix} 1&0&0&0&0&1&0&0\\ 0&1&0&0&1&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&1&1\\ 0&1&0&0&1&0&0&0\\ 1&0&0&0&0&1&0&0\\ 0&0&0&1&0&0&1&1\\ 0&0&0&1&0&0&1&1 \end{pmatrix}$$

Рис. 60: Матрицы смежности, достижимости, взаимной достижимости из задачи 299.

$$\begin{pmatrix} 0&0&1&1&1&1&0&0\\ 0&0&1&0&0&0&1&1\\ 0&1&0&0&0&0&0&1\\ 1&1&0&0&0&0&1&0\\ 0&0&1&0&0&1&1&1\\ 0&0&1&0&1&0&1&0\\ 0&1&0&0&0&0&0&1\\ 0&0&0&0&0&0&1&0 \end{pmatrix};\begin{pmatrix} 1&1&1&1&1&1&1&1\\ 0&1&1&0&0&0&1&1\\ 0&1&1&0&0&0&1&1\\ 1&1&1&1&1&1&1&1\\ 0&1&1&0&1&1&1&1\\ 0&1&1&0&1&1&1&1\\ 0&1&1&0&0&0&1&1\\ 0&1&1&0&0&0&1&1 \end{pmatrix};\begin{pmatrix} 1&0&0&1&0&0&0&0\\ 0&1&1&0&0&0&1&1\\ 0&1&1&0&0&0&1&1\\ 1&0&0&1&0&0&0&0\\ 0&0&0&0&1&1&0&0\\ 0&0&0&0&1&1&0&0\\ 0&1&1&0&0&0&1&1\\ 0&1&1&0&0&0&1&1 \end{pmatrix}$$

Рис. 61: Матрицы смежности, достижимости, взаимной достижимости из задачи 300.

303. $\mathfrak{G}_1$: $K_1 = \{a, n, \ell\}$, $K_2 = \{b\}$, $K_3 = \{c\}$, $K_4 = \{d, e, f, g\}$, $K_5 = \{h, m\}$, $K_6 = \{k\}$, $K_7 = \{r\}$; $\mathfrak{G}_2$: $K_1 = \{a\}$, $K_2 = \{b\}$, $K_3 = \{c, e, g, f\}$, $K_4 = \{d\}$, $K_5 = \{k\}$, $K_6 = \{h, m\}$, $K_7 = \{n, \ell\}$; $\mathfrak{G}_3$: $K_1 = \{a\}$, $K_2 = \{b\}$, $K_3 = \{c\}$, $K_4 = \{d\}$, $K_5 = \{e, h, m, k, g\}$, $K_6 = \{f\}$, $K_7 = \{n, \ell\}$. Графы строгой достижимости представлены на рис. 63. Базы $\mathfrak{G}_1$: $\{b, r, d\}$, $\{b, r, e\}$, $\{b, r, f\}$, $\{b, r, g\}$. Базы $\mathfrak{G}_2$: $\{b, h, n\}$, $\{b, h, \ell\}$, $\{b, m, n\}$, $\{b, m, \ell\}$. Базы $\mathfrak{G}_3$: $\{b, e\}$, $\{b, h\}$, $\{b, m\}$, $\{b, k\}$, $\{b, g\}$.

304. Пусть граф сильно связный. Зафиксируем вершину v. Тогда для любой другой вершины v_i, $i = 1, \ldots, n$, существует путь P_i' из v в v_i и путь P''

$$\begin{pmatrix} 0&0&1&1&0&1&1&0\\ 0&0&0&0&1&0&0&1\\ 1&1&0&0&0&0&0&1\\ 0&1&1&0&1&0&1&1\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&1\\ 1&0&1&0&0&1&0&0\\ 0&0&0&0&1&0&0&0 \end{pmatrix};\begin{pmatrix} 1&1&1&1&1&1&1&1\\ 0&1&0&0&1&0&0&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 0&0&0&0&1&0&0&1\\ 0&0&0&0&1&1&0&1\\ 1&1&1&1&1&1&1&1\\ 0&0&0&0&1&0&0&1 \end{pmatrix};\begin{pmatrix} 1&0&1&1&0&0&1&0\\ 0&1&0&0&0&0&0&0\\ 1&0&1&1&0&0&1&0\\ 1&0&1&1&0&0&1&0\\ 0&0&0&0&1&0&0&1\\ 0&0&0&0&0&1&0&0\\ 1&0&1&1&0&0&1&0\\ 0&0&0&0&1&0&0&1 \end{pmatrix}$$

Рис. 62: Матрицы смежности, достижимости, взаимной достижимости из задачи 301.

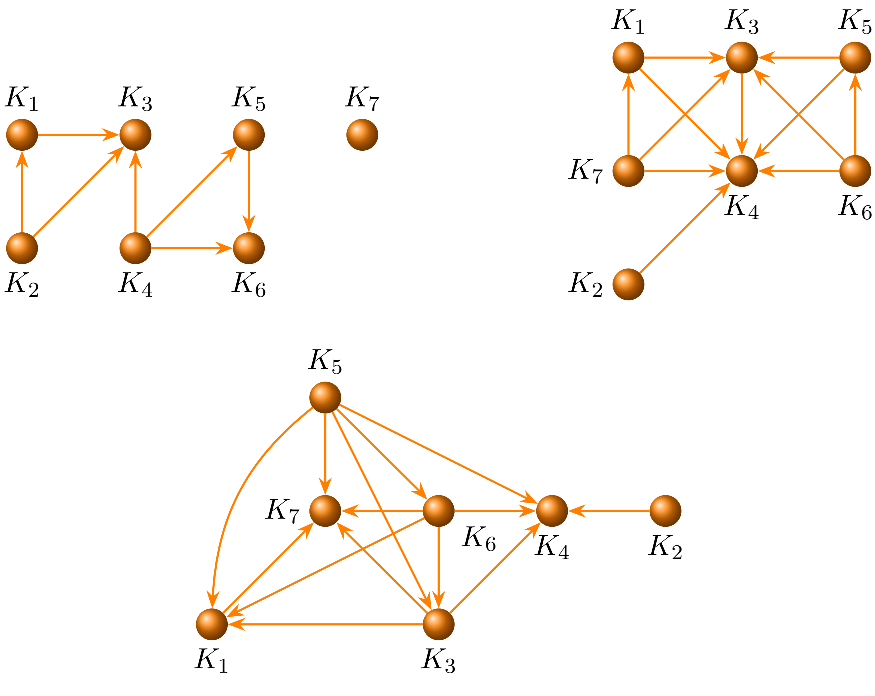


Рис. 63: Графы сильной достижимости для $\mathfrak{G}_1$, $\mathfrak{G}_2$ и $\mathfrak{G}_3$.

из v_i в v. Тогда искомым циклом будет $(P_1', P_1'', \ldots, P_n', P_n'')$. Если цикл проходит через все вершины, то, очевидно, двигаясь по нему, можно из любой вершины попасть во все остальные.

Неориентированные графы

305. Число сочетаний $C_n^2 = {}^{n(n-1)}/_2$.

306. Каждое ребро имеет два конца и добавляет 2 к общей сумме степеней вершин.

307. Чтобы сумма степеней вершин была чётной, нужно чтобы количество нечётных вершин было чётным.

308. Согласно задаче 306 удвоенное количество дорог равняется количеству городов, умноженному на 5. Следовательно, количество дорог делится на 5 и не может быть равно 77. Если взять 32 города, расположить их по кругу и соединить каждый с четырьмя ближайшими и одним противоположным, то получится 80 дорог (рис. 64).

309. С одной вершиной: один. С двумя вершинами: два. С тремя вершинами: четыре. С четырьмя вершинами: одиннадцать (по одному с 0, 1, 5 и 6 рёбрами, по два с 2 и 4 рёбрами, три с тремя рёбрами). Рис. 65.

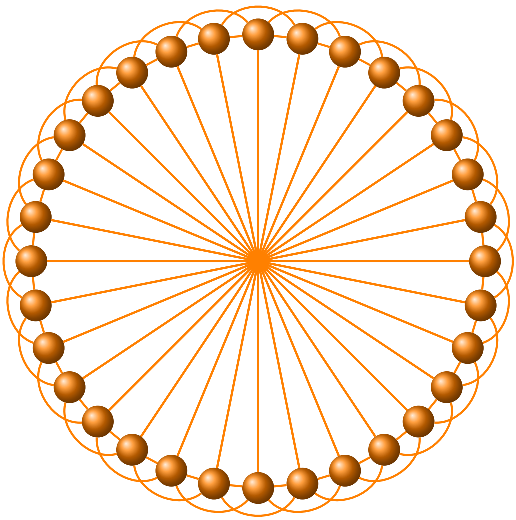


Рис. 64: 80 дорог.

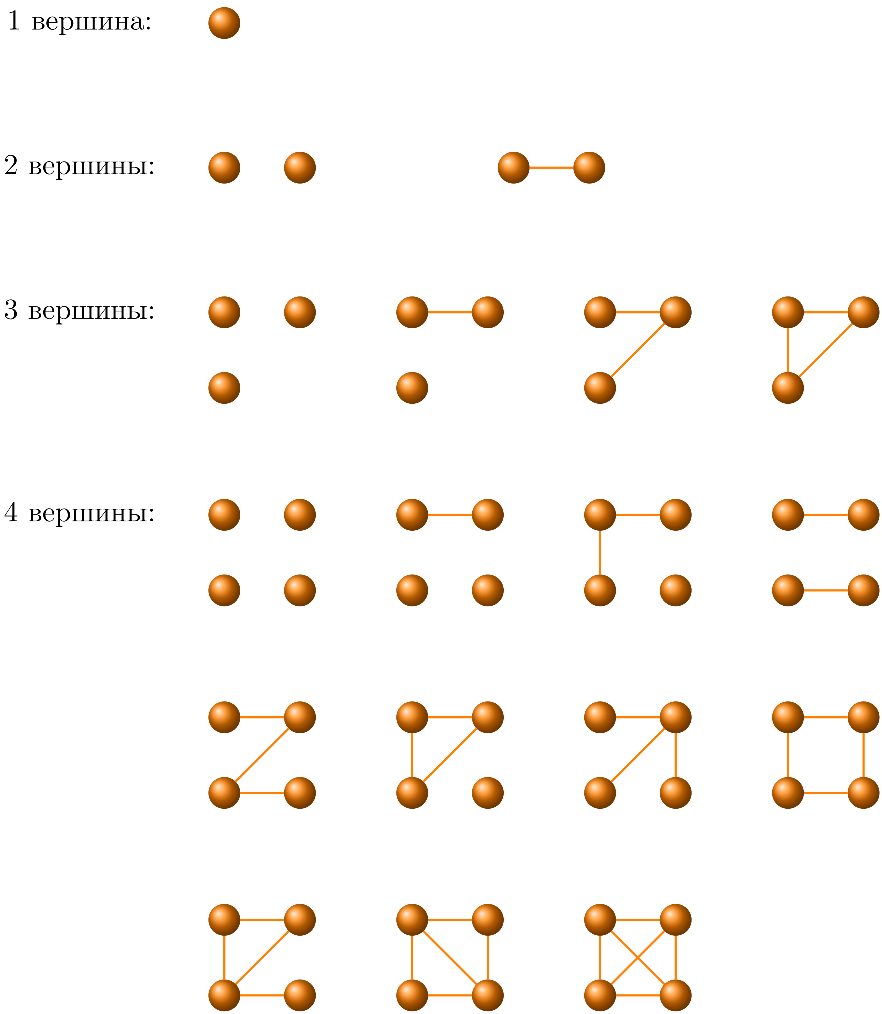


Рис. 65: Графы с не более чем четырьмя вершинами.

310. Если ребро e входит в некоторый цикл, то при удалении e все вершины цикла остаются взаимно достижимыми, поэтому связность графа не меняется, а ребро мостом быть не может. Если $e = (u, v)$ не является мостом, то при его удалении вершины u и v остаются взаимно достижимыми, то есть имеется путь P из u в v, не проходящий через e. Следовательно, имеется цикл вида (v, u, P, v), проходящий через ребро e.

311. Если в графе есть мост (u, v), то при любой ориентации этого ребра одна из вершин будет недостижима из другой, следовательно, количество компонент сильной связности будет больше единицы. Пусть теперь мостов в графе нет. Рассмотрим следующую процедуру: пока в графе есть хотя бы одно неориентированное ребро (u, v) берём простой цикл C, в который это ребро входит (задача 310) и ориентируем рёбра этого цикла, которые ещё неориентированы, в одном и том же направлении. Покажем, что после этого все вершины цикла C будут взаимно достижимы. Допустим, в цикл C входит ребро (x, y), которое ранее ориентировано в напралении противоположном выбранному. По индукционному предположению это ребро входит в некоторый цикл D, вершины которого взаимно достижимы, следовательно, вершины x и y взаимно достижимы. В конце получим, что два конца любого ребра будут взаимно достижимы, следовательно, все вершины графа взаимно достижимы.

312. Индукция по n. Для $n = 1$ оба утверждения очевидны, в случае (б) будет петля. Пусть для n доказано, докажем для $n+1$. Если убрать вершину v и k инцидентных рёбер, то образуется $\ell \leqslant k$ компонент связности, для каждой из которых выполнено $|E_i| \geqslant |V_i| - 1$. Просуммировав и добавив удалённые рёбра, получим $|E| = \sum\limits_i |E_i| + k \geqslant (\sum\limits_i |V_i| - \ell) + k \geqslant \sum\limits_i |V_i| = |V| - 1$. Если $|E| \geqslant |V|$, то либо $|E_i| \geqslant |V_i|$ для какого-то i и цикл есть в V_i по индукционному предположению, либо $\ell < k$. В последнем случае две вершины, соединённые с v, соединены между собой ещё как-то, следовательно, есть цикл.

313. Во всяком графе V из шести вершин без петель есть три попарно соединённых или три попарно не соединённых. Пусть V_a — вершины, соединённые с a, не включая a. Если в V_a есть две соединённых вершины, то они вместе с a образуют искомую тройку. Если в V_a соединённых нет, но $|V_a| \geqslant 3$, то само V_a (или его подмножество) образует искомую тройку. Если $|V_a| \leqslant 2$, то рассмотрим $V \setminus V_a$ — множество не соединённых с a вершин: $|V \setminus V_a| \geqslant 3$. Если в $V \setminus V_a$ есть две несоединённых вершины, то они вместе с a образуют искомую тройку. Иначе все в V_a соединены и оно само или его подмножество является нужной тройкой.

314. Рис. 66.

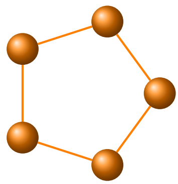


Рис. 66: Граф знакомств из задачи 314.

315. (а) Пусть $\deg a \leqslant 4$ и V'_a — вершины, не соединённые с a. Тогда $|V'_a| \geqslant 4$. Если среди V'_a есть две вершины b и c, не соединённые друг с другом, то a, b, c — требуемые вершины. Иначе все вершины в V'_a друг с другом соединены и они образуют нужное множество. (б) Все вершины имеют степень 5 или больше. Так как сумма степеней должна быть чётной, то существует вершина a степени 6 или больше. Пусть V_a — вершины, соединённые с a. Тогда $|V_a| \geqslant 6$. Согласно задаче 313 в V_a есть три вершины b, c, d или попарно соединённые, или попарно не соединённые. В первом случае нужным множеством будет a, b, c, d, во втором — b, c, d.

316. Пусть a — любая вершина, V_a — вершины, соединённые с a, V'_a — вершины, не соединённые с a. Тогда $|V_a| \geqslant 9$ или $|V'_a| \geqslant 9$. Пусть $|V'_a| \geqslant 9$. Согласно задаче 315 в подграфе с вершинами V'_a найдутся или четыре попарно соединённые вершины (тогда они и являются искомыми), или три попарно не соединённые вершины (тогда они вместе с a являются искомыми). Пусть теперь $|V_a| \geqslant 9$. Рассмотрим граф $\mathfrak{G}_a$ с множеством вершин V_a, содержащий в точности те рёбра, которых нет в исходном графе. Согласно задаче 315 в $\mathfrak{G}_a$ найдутся или четыре попарно соединённые вершины, или три попарно не соединённые вершины. В первом случае это будут четыре попарно не соединённые вершины исходного графа. Во втором — эти три вершины и вершина a будут попарно соединены в исходном графе.

317. Если при разбиении такого ребра нет, то, очевидно, граф не связен. Если граф не связен, то взять в качестве V_1 одну компоненту связности, а в качестве V_2 — остальные.

318. Для каждой компоненты связности сумма степеней всех вершин чётна, задача 306. Следовательно, если нечётных вершин две, то они принадлежат одной компоненте, то есть связаны путём.

319. Три компоненты: $\{1, 2, 4, 5, 6, 7\}$, $\{3, 9\}$ и $\{8\}$.

320. Пусть пути $(u_1, \ldots, u_n)$ и $(v_1, \ldots, v_n)$ не имеют общих вершин. Так как граф связен, то вершины этих двух путей достижимы друг из друга.

Пусть самый короткий путь соединяет u_i и v_j: $(u_i, w_1, \ldots, w_k, v_j)$. Если, например, $i \leqslant {}^n/_2$ и $j \leqslant {}^n/_2$, то путь $u_n, \ldots, u_i, w_1, \ldots, w_k, v_j, \ldots, v_n)$ будет иметь длину бóльшую n. Аналогично строится более длинный путь для других случаев i и j.

321. **(а)** Если в графе $\mathfrak{G}$ есть более одной компоненты связности: $V_1, \ldots, V_k$, $k \geqslant 2$, то в графе $\bar{\mathfrak{G}}$ все вершины V_i соединены со всеми вершинами V_j, $j \neq i$. Поэтому любые две вершины u и v соединены в $\bar{\mathfrak{G}}$ путём вида (u, v), если они находятся в разных V_i и V_j, или путём вида (u, w, v), если $u, v \in V_i$ и $w \notin V_i$.

(б) Водные маршруты образуют дополнение графа автобусных маршрутов. Следовательно, хотя бы один из этих графов связен.

322. Пусть $A = V \cup E$ и $f(u)$ — множество, состоящее из u и инцидентных u рёбер.

323. Индукция по $n = |V|$. При $n = k$ рёбер быть не может. Пусть для n доказано. Самая большая из компонент содержит не более чем $n - k + 1$ вершину. Следовательно, добавив новую вершину, мы можем добавить не более $n - k + 1$ ребро. Получаем $(n - k)(n - k + 1)/2 + n - k + 1 = (n + 1 - k)((n - k)/2 + 1) = (n + 1 - k)(n + 1 - k + 1)/2$.

324. Согласно лемме «о рукопожатиях» (задача 307), количество нечётных вершин чётно. Если количество нечётных вершин равно $2n$, то, добавив $n-1$ ребро, можно оставить две нечётные вершины (граф станет полуэйлеровым), а добавив n рёбер, убрать все нечётные вершины (граф станет эйлеровым).

325. Триангуляция — плоский граф, у которого все грани, кроме внешней, треугольные: $e + 2 = f + v$, где количество вышек совпадёт с количеством вершин, а количество граней на единицу больше количества треугольников: $f = N + 1$. Далее, $2e = 3N + K$, следовательно, ${}^{(3N + K)}/_2 + 2 = N + 1 + v$ и $v = {}^{(N + K)}/_2 + 1$.

326. $2e = 3f$, $e + 2 = f + v$, следовательно, $v = {}^f/_2 + 2$, $e = {}^{3f}/_2$. Такое возможно только для чётных $f \geqslant 4$. Построение индукцией по f: на каждом шаге вместо одной из треугольных граней «приделываем» тетраэдр.

327. При исключении изображение на плоскости можно оставить тем же, только убрать вершину. Чётность оставшихся вершин не меняется. При исключении вершины из цикла длины четыре (два цвета) получится цикл длины три (три цвета). При исключении e в графе $\{(a, b), (b, c), (c, d), (d, a), (a, e), (e, c)\}$ появится гамильтонов цикл.

328. Рис. 67 и 68. Эйлеров только октаэдр — все вершины чётные. Остальные содержат более двух нечётных вершин и не являются ни эйлеровыми, ни полуэйлеровыми. Гамильтоновы циклы выделены жирными линиями, все графы гамильтоновы. Хроматическое число у куба 2, у октаэдра и

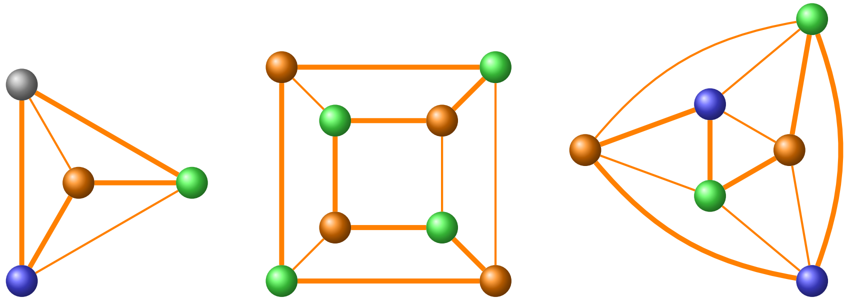


Рис. 67: Тетраэдр, куб, октаэдр.

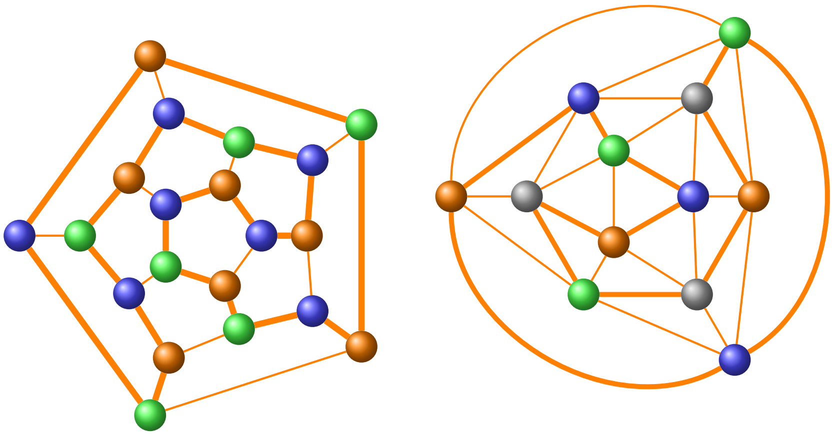


Рис. 68: Додекаэдр, икосаэдр.

додекаэдра 3, у тетраэдра и икосаэдра 4. Куб нельзя раскрасить меньше чем в два цвета. Октаэдр и додекаэдр не являются двудольными, так как есть цикл нечётной длины, поэтому их нельзя раскрасить меньше чем в три цвета. Тетраэдр и икосаэдр содержат цикл нечётной длины (его нельзя раскрасить меньше чем в три цвета), все вершины которого соединены ещё с одной, её необходимо раскрасить в четвёртый цвет.

329. Если такой многогранник существует и имеет f граней, e рёбер и v вершин, то для него выполнена формула Эйлера $f+v=e+2$ и соотношение $6f=2e$. Следовательно, $e=3f$, $f+v=3f+2$ и $v=2f+2$. С другой стороны, в каждой вершине сходится не менее трёх рёбер, поэтому $\deg w \geqslant 3$ для всех вершин w и в силу задачи 306 получаем $2e \geqslant 3v$. Таким образом, $6f = 2e \geqslant 3v = 3(2f + 2) = 6f + 6$, противоречие.

330. Пусть каждая грань имеет n сторон, а каждая вершина — степень m. Очевидно, $n, m \geqslant 3$. Тогда сумма степеней вершин равна $2e$, nf и mv одновременно, то есть $2e = nf = mv$. Умножим формулу Эйлера на nm: $nme + 2nm = nmf + nmv$, подставим в это равенство оценки $nf = mv = 2e$: $nme + 2nm = 2me + 2ne$, то есть $e = {}^{2nm}/_{(2n + 2m - nm)}$. Значит, $2n + 2m - nm > 0$, то есть $2n > nm - 2m = m(n - 2) \geqslant 3(n - 2)$ или $n < 6$. Таким образом, n может быть равно $3, 4, 5$. Для $n = 3$ получаем $e = {}^{6m}/_{(6 - m)}$, откуда $m = 3$ (тетраэдр), $m = 4$ (октаэдр) или $m = 5$ (икосаэдр). Для $n = 4$ получаем $e = {}^{8m}/_{(8 - 2m)}$, откуда $m = 3$ (куб). Для $n = 5$ получаем $e = {}^{10m}/_{(10 - 3m)}$, откуда $m = 3$ (додекаэдр).

331. Граф с рёбрами $\{(a, b), (b, c), (c, a)\}$ имеет оба цикла, $\{(a, b)\}$ — ни одного, $\{(a, b), (b, c), (c, d), (d, a), (a, c)\}$ имеет только гамильтонов цикл, $\{(a, b), (a, c), (a, d), (e, b), (e, c), (e, d), (a, e)\}$ — только эйлеров.

332. Имеется две нечётных вершины: g и m, граф полуэйлеров, но не эйлеров. При удалении ребра (g, m) граф станет эйлеровым: эйлеров цикл $(h,\ g,\ e,\ f,\ n,\ k,\ f,\ b,\ k,\ c,\ b,\ m,\ c,\ a,\ k,\ h,\ a,\ m,\ h)$.

333. Есть цикл длины три (b, f, m, b) — граф не двудольный. При удалении ребра (b, m) граф станет двудольным: доли $\{a, b, c, e, m\}$ и $\{f, g, h, k, n\}$.

334. Если n нечётно и все вершины многоугольника соединены с центром, то 4 (раскраска многоугольника потребует три цвета, центра — ещё один). Если n нечётно и не все вершины многоугольника соединены с центром, то 3 (раскраска многоугольника потребует три цвета, причём третий цвет можно использовать только один раз для вершины, не соединённой с центром. В этот же цвет можно раскрасить центр). Если n чётно и есть две вершины многоугольника, которые разделены чётным количеством вершин и соединены с центром, то 3 (есть цикл нечётной длины, граф не двудольный. Раскраска многоугольника потребует два цвета, центр — третий). В противном случае 2 (раскраска многоугольника потребует два цвета, причём все вершины, соединённые с центром, будут окрашены в один цвет, центр можно окрасить во второй).

Деревья

335. Всегда плоский. Эйлеров, только если нет рёбер. Полуэйлеров, только если граф является неветвящейся «линией». Гамильтонов путь имеется только в двух указанных выше случаях. Хроматическое число равно единице для дерева без рёбер или двум для дерева с рёбрами.

336. Просуммировать равенство $|V_i| = |E_i| - 1$ по всем деревьям $\mathfrak{T}_i$ леса: $|V| = |E| - k$.

337. Индукция по количеству вершин. Базис: $|V| = 2$ (иначе ребёр не может существовать). Тогда единственно ребро соединяет эти вершины, обе они являются висячими, следовательно, $|V_2| = 0$ и формула верна. Пусть для k вершин утверждение доказано. Рассмотрим дерево с $k+1$ вершиной. Среди них есть хотя бы одна висячая вершина u, иначе существовал бы цикл. Уберём u и инцидентное ребро (u, w), получим дерево $\mathfrak{T}'$ с k вершинами. По индукционному предположению количество висячих вершин в $\mathfrak{T}'$ равно $2 + \sum\limits_{v \in V_2'} (\deg v - 2)$. Если в дереве $\mathfrak{T}'$ вершина w висячая, то количество висячих вершин в $\mathfrak{T}$ и $\mathfrak{T}'$ совпадает. С другой стороны $\deg w = 2$, поэтому добавление слагаемого $\deg w - 2$ не меняет суммы и формула остаётся верной. Если в дереве $\mathfrak{T}'$ вершина w невисячая, то количество висячих вершин в $\mathfrak{T}$ увеличивается на единицу. Поскольку $w \in V_2'$, то слагаемое $\deg w - 2$ при переходе от $\mathfrak{T}'$ к $\mathfrak{T}$ тоже возрастает на единицу и формула снова остаётся справедливой.

338. Так как есть висячие вершины, то в графе есть и рёбра. Из предыдущей задачи получаем, что $\deg v - 2 = 0$ для любой другой вершины v, то есть их степени равны двум. Так как в дереве нет циклов, то существует хотя бы один простой путь P максимальной длины. Если предположить, что он не проходит через вершину v, то в силу связности должен существовать путь из v в ближайшую к v вершину u пути P. Но тогда или путь P не является максимальных (если u — висячая), или степень u больше двух. Полученное противоречие показывает, что путь P содержит все вершины графа.

339. Согласно задаче 312 в графе есть цикл, тогда можно выбросить любое ребро, которое в него входит, граф останется связным, и будет выполнено $|V| = |E| + 1$, следовательно, граф станет деревом.

340. **(a)** Пусть v и u — разные центры дерева, n — максимальная длина кратчайшего пути от v и u до остальных вершин. Допустим, что простой путь R из v в u имеет длину $m \geqslant 2$, а w — первая из промежуточных вершин этого пути. Пусть v' — вершина на расстоянии не меньшем $n - 1$ от v. Допустим, что простой путь P от v до v' не проходит через u. Рассмотрим простой путь Q из u в v'. Если Q не проходит через v, то есть цикл $(v, P, v', Q^{-1}, u, R^{-1}, v)$, что невозможно. С помощью Q^{-1} и R^{-1} мы обозначили пути противоположные Q и R соответственно. Следовательно, Q проходит через v, но тогда расстояние от u до v' не меньше $n - 1 + m \geqslant n + 1$, что невозможно. Итак, P проходит через u и, следовательно, через w. Мы получили, что для любой вершины u', которая находится на расстоянии $n-1$ или n от v, путь из v в v' проходит через w. Следовательно, расстояние от w до таких вершин v' не превосходит $n-1$. Расстояние от v до всех остальных вершин не превосходит $n-2$, поэтому расстояние от w до них тоже не превосходит $n-1$. Значит, максимальное расстояние от w до остальных вершин не превосходит $n-1$, что невозможно.

(б) Если есть три центра, то они попарно смежные и образуют цикл.

341. Если выбрать на каждом максимальном пути p вершину v_p, то из них можно построить множество X. Пусть X — минимальное по мощности из всех таких множеств. Если $|X|=1$, то единственная его вершина принадлежит всем максимальным путям, что и требуется. Предположим, $|X| \geqslant 2$, $w, u \in X$, $w \neq u$. Если бы всякий максимальный путь, проходящий через w, проходил и через u, то w можно было бы выбросить и получить множество меньшей мощности, следовательно, есть максимальный путь p, проходящий через w, но не через u. Аналогично есть максимальный путь q, проходящий через u, но не через w. Согласно задаче 320 максимальные пути p и q имеют общую вершину v. Аналогично предыдущему показывается, что есть максимальный путь r, проходящий через w или через u, но не через v. Рассмотрим случай, когда r проходит через w, второй случай аналогичен. Максимальные пути q и r имеют общую вершину x. Но тогда в графе есть цикл $x \xrightarrow{q} v \xrightarrow{p} w \xrightarrow{r} x$, что невозможно.

342. Обозначим с помощью $n(u)$ длину кратчайшего пути от v до u. 1) v — единственный исток: если $u \neq v$ и $(u_0 = v, u_1, \ldots, u_{n-1}, u_n = u)$ — кратчайший путь из v в u, то $n(u) > n(u_{n-1})$, поэтому ребро (u_{n-1}, u_n) ведёт из u_{n-1} в u, следовательно, u — не исток. 2) Если в $u \neq v$ ведёт два ребра (x, u) и (y, u), $x \neq y$, то $n(x) < n(u)$ и $n(y) < n(u)$, следовательно, существуют пути из v в x и y, не проходящие через u. Тогда в графе $\mathfrak{T}$ есть цикл $(\ldots, x, u, y, \ldots)$, что для дерева невозможно. 3) Если $(v = u_0, u_1, \ldots, u_{n-1}, u_n = u)$ — кратчайший путь из v в u в графе $\mathfrak{T}$, то ориентация ребёр будет именно такой: (u_i, u_{i+1}), поэтому в новом графе u достижима из v.

343. **(а)** Из задачи 337 сразу получаем, что количество висячих вершин не меньше $2 + (d-2) = d$.

(б) В задаче 342 возьмём v в качестве корня, тогда у него не менее d сыновей и, следовательно, не менее d листьев, то есть висячих вершин.

344. **(а)** Если исключить v_1, то можно построить $(n-1)^{n-3}$ неориентированных дерева. Вершину v_1 можно соединить с любой из $n-1$ оставшихся, всего $(n-1)^{n-3} \cdot (n-1) = (n-1)^{n-2}$.

(б) Если исключить v_1, то можно построить $(n-2)^{n-3}$ неориентированных дерева, в которых v_2 будет висячей. Вершину v_1 можно соединить с любой другой, кроме v_2, всего $(n-2)^{n-3} \cdot (n-2) = (n-2)^{n-2}$ вариантов.

(в) $(n-k)^{n-2}$ деревьев. Индукция по k. Базис: для $k = 0$ — теорема Кэли. Индукционный шаг. Пусть для k выделенных вершин утверждение доказано (для любого n). Если исключить v_1, то по индукционному предположению можно построить $(n-1-k)^{n-3}$ неориентированных дерева, в которых k вершин $v_2, \ldots, v_{k+1}$ будут висячими. Вершину v_1 можно соединить с $n-1-k$ вершинами: $v_{k+2}, \ldots, v_n$, всего $(n-1-k)^{n-3} \cdot (n-1-k) = (n-1-k)^{n-2}$ вариантов.

(г) Такой граф является «линией», количество «линий» равно $n!$, с учётом того, что не важно с какой стороны её рассматривать, получаем $^{n!}/_2$.

345. **(а)** 1) Единственный исток — v, так как достижимость остальных вершин означает, что в них должно входить хотя бы одно ребро; 2) в остальные вершины ведёт по одному ребру, так как это было выполнено в $\mathfrak{T}$; 3) по определению $\mathfrak{T}_v$.

(б) Если бы $\mathfrak{T}_v$ и $\mathfrak{T}_u$ пересекались и имели общую вершину w, то w достижима из v и из u: есть пути $(u = u_0, \ldots, u_n = w)$ и $(v = v_0, \ldots, v_m = w)$, Выберем среди всех таких вершин w ту, у которой сумма $n + m$ будет наименьшей, тогда $u_{n-1} \neq v_{m-1}$ и получается, что в w ведут два ребра, что для дерева невозможно.

346. $\mathfrak{G} = (\{1,2,3\}, \{(2,3),(3,2)\})$. Вершина 1 является единственным истоком, в каждую из вершин 2 и 3 входит по одному ребру.

347. Количество человек: $1 + 3 + 3 \cdot 4 + 3 \cdot 4 \cdot 5 = 76$. Количество звонков: $3 + 3 \cdot 4 + 3 \cdot 4 \cdot 5 = 75$.

348. 1) r — единственный исток, так как у любого другого u найдётся меньший его $v < u$, поэтому среди них есть максимальный w и $(w,u) \in E$. 2) Для $u \neq r$ наличие ребра вида (v,u) показано в предыдущем пункте. Если $(v,u),(w,u) \in E$, то v и w — максимальные элементы меньшие u. Они несравнимы, так как иначе один из них не был бы максимальным. Но тогда согласно (б) u несравним с u, что невозможно. 3) Индукция по количеству k элементов строго меньших u. Если $k = 0$, то $u = r$, так как иначе $r < u$ и $k > 0$, тогда r достижим из r. Если для всех $\ell < k$ утверждение доказано и u имеет k меньших его элементов, то выберем v — максимальный из них. Тогда $(v,u) \in E$, а у v количество меньших элементов равно $\ell < k$. По индукционному предположению из r в v есть путь, значит, есть путь и в u.

349. Рефлексивность: каждая вершина достижима из себя. Антисимметричность: если u достижима из v, а v — из u, то при $u \neq v$ был бы цикл, что невозможно. Транзитивность: если из u достижима v, а из v — w, то из

u достижима w. Корень является наименьшим элементом, так как из него достижимы все вершины. Если вершины u и v недостижимы друг из друга, то большие их вершины образуют деревья $\mathfrak{T}_u$ и $\mathfrak{T}_v$, которые не могут иметь общих вершин (задача 345), поэтому они тоже недостижимы друг из друга.

350. В прямую сторону. Если бы был цикл $(u_0, u_1, \ldots, u_n = u_0)$, то он не мог бы содержать корня r. Так как из r есть путь в вершины этого цикла, то хотя бы в одну из этих вершин должно входить второе ребро, что невозможно. Второе и третье условие повторяют определение дерева. В обратную сторону. Пункты 1) и 2) определения повторяют условия задачи. 3) Так как в графе нет циклов, то пути не могут иметь сколь угодно большую длину. Рассмотрим самый длинный путь, ведущий в u: $(u_0, u_1, \ldots, u_n = u)$. Тогда u_0 является истоком (иначе путь можно сделать длиннее), следовательно, $u_0 = r$ и из r достижима u.

351. В прямую сторону — индукция по глубине вершины v. Для $v = r$ есть только путь (r) так как в r рёбра не входят. Для $v \neq r$ есть вершина w и ребро (w, v). Тогда глубина w меньше глубины v, следовательно, по индукционному предположению в w из r есть только один путь p. Так как (w, v) является единственным ребром, ведущим в v, то и в вершину v из r тоже есть только один путь (p, v). В обратную сторону. Если бы в r входило ребро (u, r), то так как есть путь p из r в u, то был бы цикл (p, r), поэтому из r в себя было бы бесконечно много путей. Следовательно, это невозможно и r является истоком. Если бы существовал ещё один исток r', то не было бы пути из r в r', поэтому такое невозможно. Если бы в вершину $u \neq r$ не входило ни одного ребра, то не было бы пути из r в u. Поэтому хотя бы одно ребро в u входит. Если бы в вершину u входило два ребра (v, u) и (w, u), то так как есть пути p_1 и p_2 из r в v и w соответственно, то былы бы два пути из r в u: (p_1, u) и (p_2, u). Значит, это невозможно, и в любую вершину кроме r входит только одно ребро. Третье условие дерева непосредственно выполнено.

352. Из задачи 337 получаем $26 = 2 + (5-2) + n(5-2) + 2n(4-2)$, где n — количество вершин с четырьмя сыновьями. Следовательно, $7n = 21$, количество вершин с четырьмя сыновьями равно трём, с тремя — шести. Общее количество вершин $1 + n + 2n + 26 = 36$, рёбер — 35.

353. Пусть n — количество внутренних вершин с двумя сыновьями. Тогда из задачи 337 получаем уравнение $2 + (3-2) + (3-2)n + (5-2) \cdot 2n = 38$. Отсюда $n = 5$, общее количество вершин равняется $1 + n + 2n + 38 = 54$.

354. Индукция по k — длине $P = (v_1^{(s_1)}, \ldots, v_k^{(s_k)})$. Базис: $k = 0$, единственный вариант — пустой лес $\mathfrak{F} = ()$. Пусть для k утверждение доказано, а

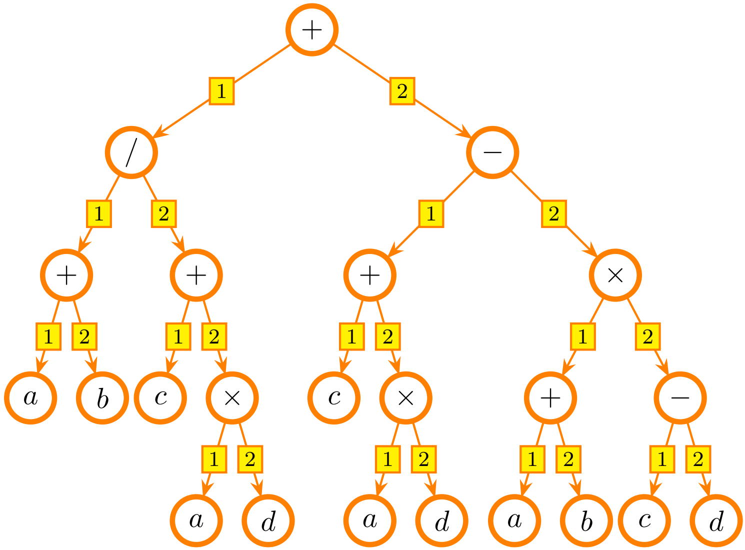


Рис. 69: Дерево из задачи 357.

последовательность имеет длину $k+1$: $P = (v_0^{(s_0)}, v_1^{(s_1)}, \ldots, v_k^{(s_k)})$. По индукционному предположению по последовательности $P' = (v_1^{(s_1)}, \ldots, v_k^{(s_k)})$ длины k можно однозначно восстановить лес $\mathfrak{F}' = (\mathfrak{T}_1, \ldots, \mathfrak{T}_m)$. Если $s_0 = 0$, то в $\mathfrak{F}$ первое дерево $\mathfrak{T}_0$ состоит из единственной вершины, поэтому $\mathfrak{F} = (\mathfrak{T}_0, \mathfrak{T}_1, \ldots, \mathfrak{T}_m)$. Если $1 \leqslant s_1 \leqslant m$, то первые s_1 деревьев $\mathfrak{F}'$ войдут в новое дерево $\mathfrak{T}_0$ с корнем v_0, поэтому $\mathfrak{F} = (\mathfrak{T}_0, \mathfrak{T}_{s_1+1}, \ldots, \mathfrak{T}_m)$. Наконец, при $s_1 > m$ указанная последовательность не может быть префиксным обходом никакого леса, так как у v_0 должно быть s_1 сыновей, которых в лесу $\mathfrak{F}'$ нет. Для суффиксного обхода нужно аналогичным образом рассматривать последнюю вершину.

355. Базис: дерево содержит одну вершину, которая является листом, вершин с двумя сыновьями нет, поэтому $0 = 1 - 1$. Шаг: пусть для деревьев $\mathfrak{T}_1$ и $\mathfrak{T}_2$ утверждение доказано: $n_1 = \ell_1 - 1$ и $n_2 = \ell_2 - 1$. Если $\mathfrak{T}$ построено из $\mathfrak{T}_1$, $\mathfrak{T}_2$ и нового корня, то $n = n_1 + n_2 + 1 = \ell_1 - 1 + \ell_2 - 1 + 1 = \ell - 1$. Если $\mathfrak{T}$ построено только из $\mathfrak{T}_1$ и нового корня, то n и ℓ остаются без изменений.

356. Количество листьев 2^h, количество вершин $2^{h+1} - 1$.

357. Дерево изображено на рис. 69, ациклический граф — на рис. 70. Сокращается 8 вершин.

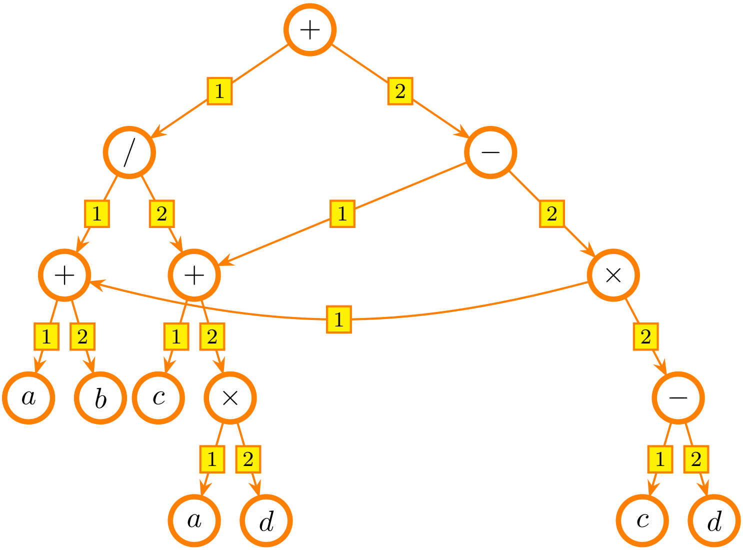


Рис. 70: Ациклический граф из задачи 357.

358. Рис. 71. PRF $= (\vee,\ \wedge,\ \vee,\ x,\ \neg,\ y,\ \neg,\ \to,\ z,\ \wedge,\ x,\ y,\ \oplus,\ \neg,\ z,\ y)$; SFF $=$ $(x,\ y,\ \neg,\ \vee,\ z,\ x,\ y,\ \wedge,\ \to,\ \neg,\ \wedge,\ z,\ \neg,\ y,\ \oplus,\ \vee)$; INF $= (x,\ \vee,\ \neg,\ y,\ \wedge,\ \neg,\ z,\ \to,$ $x,\ \wedge,\ y,\ \vee,\ \neg,\ z,\ \oplus,\ y)$.

359. PRF $= (a,\ b,\ d,\ e,\ h,\ i,\ c,\ f,\ g,\ k,\ \ell)$; SFF $= (d,\ h,\ i,\ e,\ b,\ f,\ k,\ \ell,\ g,\ c,\ a)$; INF $= (d,\ b,\ h,\ e,\ i,\ a,\ f,\ c,\ k,\ g,\ \ell)$.

360. PRF $= (1,\ 6,\ 7,\ 8,\ 2,\ 9,\ 11,\ 10,\ 12,\ 13,\ 14,\ 3,\ 5,\ 4)$; SFF $= (7,\ 8,\ 6,\ 11,\ 9,$ $12,\ 13,\ 14,\ 10,\ 5,\ 4,\ 3,\ 2,\ 1)$.

361. Рис. 72, $(x \times (z - (u/v))) + ((y - z) \times (x + y))$.

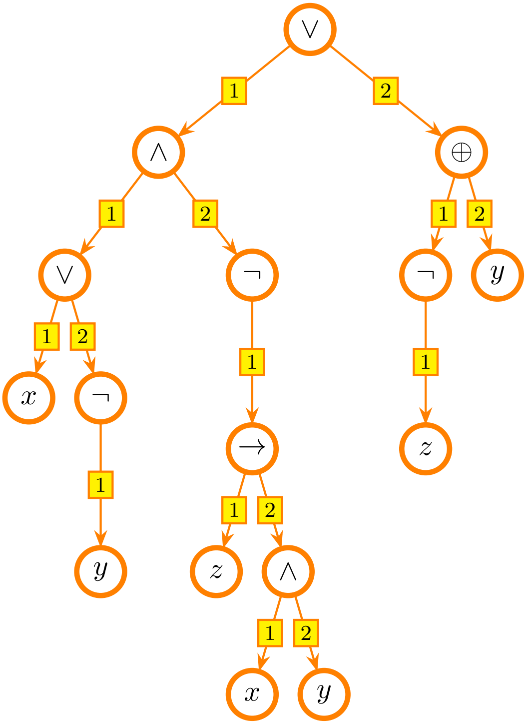


Рис. 71: Дерево для формулы из задачи 71.

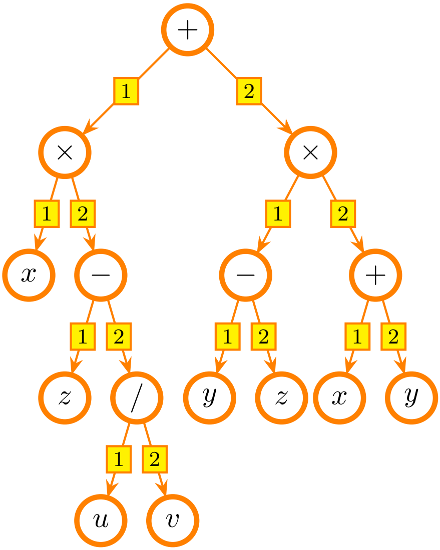


Рис. 72: Дерево для формулы из задачи 361.

Алгоритмы на графах

362. Рис. 73.
363. Рис. 74.
364. **(а)** Пусть $\mathfrak{T}$ — минимальное остовное дерево $\mathfrak{G}$. Если оно не содержит e, то является и остовным деревом $\mathfrak{G}'$, очевидно, минимальным. Если $\mathfrak{T}$ содержит e, то при его удалении оно распадётся на две компоненты связности $\mathfrak{T}_1$ и $\mathfrak{T}_2$, причём цикл C будет содержать вершины из обеих этих компонент. Выберем в цикле C ребро e', отличное от e, соединяющее эти компоненты. Тогда связность восстановится, то есть получится дерево $\mathfrak{T}'$. Так как $w(e') \leqslant w(e)$, то вес нового дерева не может увеличиться. Теперь $\mathfrak{T}'$ — минимальное остовное дерево для $\mathfrak{G}$ и $\mathfrak{G}'$.
(б) В этом случае получим $w(e') < w(e)$, поэтому $w(\mathfrak{T}') < w(\mathfrak{T})$ и $\mathfrak{T}$ не является минимальным, противоречие.
365. Если ребро d_i не входит в $\mathfrak{T}$, то оно вместе с какими-то рёбрами $e_{j_{i,k}}$ образует цикл. Согласно задаче 364, вес ребра d_i в таком цикле должен быть максимальным: $c(e_{j_{i,k}}) \leqslant c(d_i)$ для всех k. Допустим, $c(e_i) > c(d_i)$, тогда $j_{i,k} < i$ для всех k. Это означает, что оба конца каждого из рёбер d_ℓ, $\ell \leqslant i$, принадлежат одному и тому же дереву $\mathfrak{T}_{i,m_\ell}$ для какого-то m_ℓ. Следовательно, если заменить в лесу все рёбра e_j, $j < i$, на d_ℓ, $\ell \leqslant j$, то количество деревьев не уменьшится, так как никакое из рёбер d_ℓ не соединяет разные деревья $\mathfrak{T}_{i,m_\ell}$. Количество рёбер при этом возросло, количество вершин не изменилось, а количество деревьев равняется их

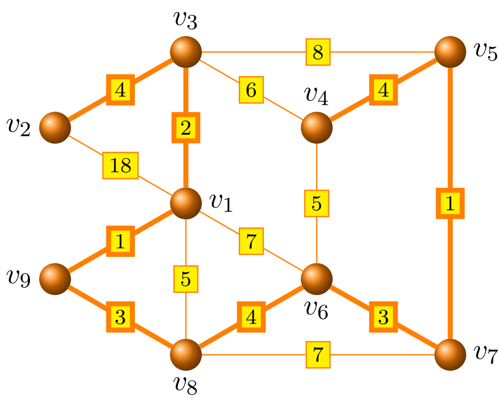


Рис. 73: Минимальное остовное дерево.

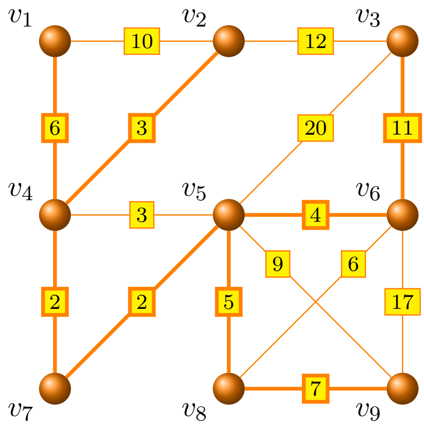


Рис. 74: Минимальное остовное дерево.

разности (задача 336). Получили противоречие, значит, предположение $c(e_i) > c(d_i)$ неверно.

366. Если ребро наибольшего веса является мостом. В этом случае оно должно попасть в любое остовное дерево, а алгоритм МинОД выберет его самым последним. Наоборот, если ребро наибольшего веса мостом не является, то в силу задачи 364 оно не войдёт в минимальное остовное дерево, поэтому дерево будет построено ранее последнего шага.

367. Поскольку алгоритм не остановится, пока в графе есть циклы, то в полученном результате циклов быть не может. При выкидывании любого ребра из цикла граф остаётся связным. Следовательно, в результате получится связный граф без циклов, то есть неориентированное дерево. В силу задачи 364 на каждом шаге построения граф имеет некоторое минимальное остовное дерево, являющееся минимальным остовным деревом исходного графа. Значит, последний граф, который как раз и является остовным деревом, будет минимальным.

368. В силу задачи 364 ни в какое остовное дерево не могут попасть рёбра, имеющие наибольший вес в некотором простом цикле и являющиеся при этом единственными (рис. 75). Такими являются (a, b) из цикла (a, b, d, c, a) и (e, h) из цикла (e, h, g, f, e). Если эти рёбра убрать, то в графе останется единственный простой цикл (b, f, e, d, b). Чтобы получить минимальное остовное дерево, из этого цикла нужно выкинуть одно из рёбер наибольшего веса: (b, f) или (d, e). Поскольку второе ребро при этом останется, то условию задачи удовлетворяют только уже найденные (a, b) и (e, h).

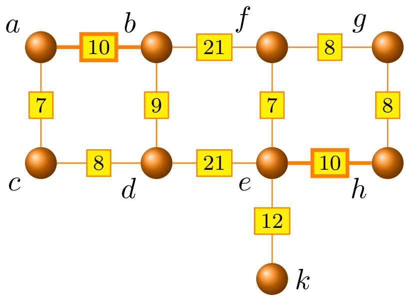


Рис. 75: Рёбра, не входящие в минимальные остовные деревья.

369. Если $\mathfrak{T}_i = (V_i, T_i)$ — дерево, полученное после i-го шага, то докажем, что $\mathfrak{T}_i$ можно расширить до минимального остовного дерева. Базис: $i = 0$ — очевидно. Шаг: пусть для i утверждение доказано и $\mathfrak{T}_i$ расширяется до минимального остовного дерева $\mathfrak{T} = (V, T)$. Допустим, на $(i+1)$-м шаге выбрано ребро (u, v), где $u \in V_i$, $v \notin V_i$. Если $(u, v) \in T$, то условие вновь выполняется. В противном случае $(u, v) \notin T$, но тогда есть путь $(u = u_0, \ldots, u_n = v)$ в $\mathfrak{T}$. Так как $u = u_0 \in V_i$ и $v = u_n \notin V_i$, то существует наименьшее j, для которого $u_j \notin V_i$. Это означает $u_{j-1} \in V_i$. Согласно пункту (б) алгоритма $w(u_{j-1}, u_j) \geqslant w(u, v)$. Рассмотрим $\mathfrak{T}' = (V, E \setminus \{(u_{j-1}, u_j)\} \cup \{(u, v)\})$. Это снова остовное дерево, так как количества вершин и рёбер не изменились и оно осталось связным. Его вес не мог увеличиться, поэтому оно минимальное и $\mathfrak{T}_{i+1}$ — поддерево $\mathfrak{T}'$.

370. Пусть $Num[u] < Num[v]$, то есть вызов Поиск (u) происходит раньше вызова Поиск (v). Докажем, что u — предок v. Рассмотрим вызов Поиск (u). Так как $v \in L_u$ и ребро (u, v) не является прямым, то вызов Поиск (v) произошёл до того, как в цикле 20–25 была рассмотрена вершина v. Следовательно, вершина v была обработана в ходе выполнения этого цикла, это и означает, что v принадлежит поддереву с корнем u, то есть является потомком u.

371. Кроме уже описанных строк для Up добавить после строки 23: if $Up[w] = Num[w]$ then print(v, w).

372. Рис. 76. Первое обратное ребро (v_8, v_1), цикл $(v_1, v_5, v_9, v_3, v_6, v_8, v_1)$. Мосты: (v_1, v_4) и (v_9, v_2).

373. Рис. 77. Мосты: (b, d), (d, e), (d, f) и (f, h).

374. Рис. 78. Первым будут найдены обратное ребро (v_4, v_1) и, соответственно, цикл (v_1, v_2, v_4, v_1). Мосты: (v_2, v_9), (v_1, v_8) и (v_6, v_5).

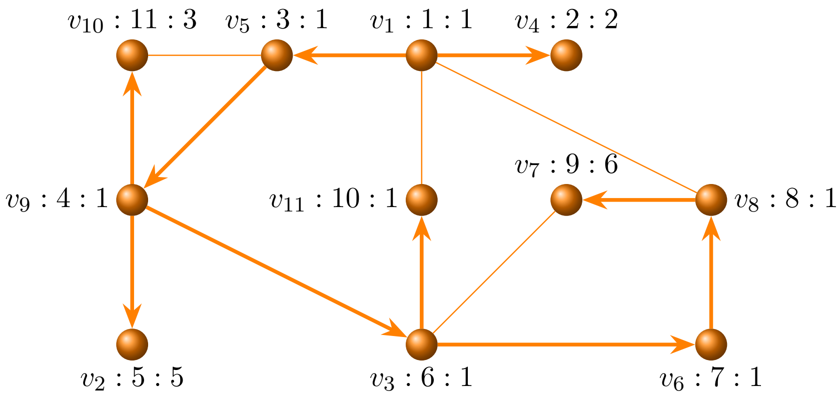


Рис. 76: Результат поиска в глубину, для каждой вершины указаны номер и значение функции Up.

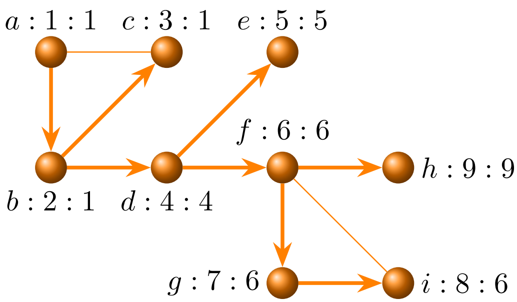


Рис. 77: Результат поиска в глубину, для каждой вершины указаны номер и значение функции Up.

375. Рис. 79. Первым будут найдены обратное ребро (h, a) и, соответственно, цикл (a, e, b, h, a). Мосты: (a, f) и (c, d).

376. Вставить: после строки 11 print(«{»), после строки 12 print(«}»), после строки 16 print(v).

377. Применимы алгоритмы Ярника-Прима-Дейкстры и алгоритм поиска в глубину, если начинать с вершины r. Алгоритм Крускала непосредственно неприменим, так как не всякий ориентированный граф без циклов является деревом.

378. Рис. 80.

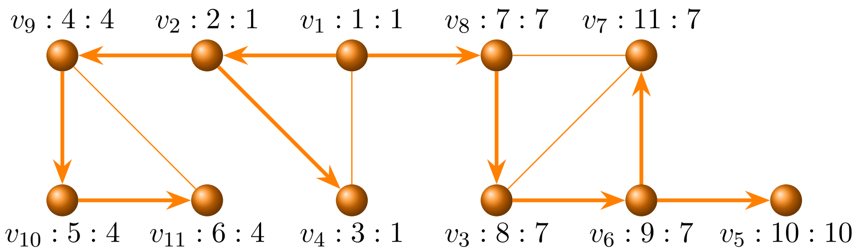


Рис. 78: Результат поиска в глубину, для каждой вершины указаны номер и значение функции Up.

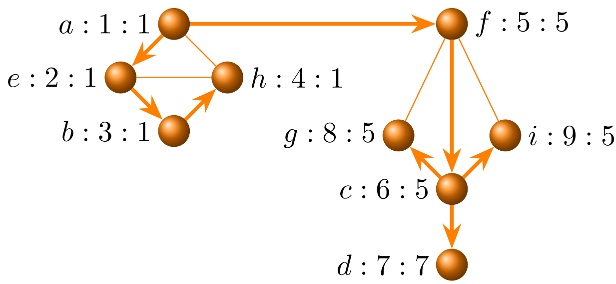


Рис. 79: Результат поиска в глубину, для каждой вершины указаны номер и значение функции Up.

379. Рис. 81. Сумма длин рёбер равна 100.
380. Рис. 82.
381. Рис. 83.
382. Допустим, что существует кратчайший путь P из исходной вершины v в вершину $u \in R$, который проходит через некоторую вершину, не принадлежащую R. Пусть u попало в R на шаге i. Выберем первую вершину $w \in P$, которая перед выполнением шага i ещё не попала в R. Тогда на шаге i получим $D[w] < D[u]$, поэтому u не может попасть в R на шаге i, противоречие.
383. В графе $\{(a,\ c;\ 2),\ (a,\ b;\ 3),\ (b,\ c;\ -2)\}$ путь (a, c), выбранный на самом первом шаге, будет длиннее, чем путь (a, b, c).
384. От одного до пяти (с учётом инициализации). Пример: граф с рёбрами $(a_i, a_j; (j-i)^2)$ для всех $1 \leqslant i < j \leqslant 6$.

№	R	w	$D[w]$	D					F				
				b	c	d	e	f	b	c	d	e	f
1	a	c	17	154	17	214	63	—	a	a	a	a	—
2	a, c	e	50	140	17	209	50	—	c	a	c	c	—
3	a, c, e	b	140	140	17	209	50	190	c	a	c	c	e
4	a, c, e, b	d	165	140	17	165	50	190	c	a	b	c	e
5	a, c, e, b, d	f	60	140	17	165	50	170	c	a	b	c	d

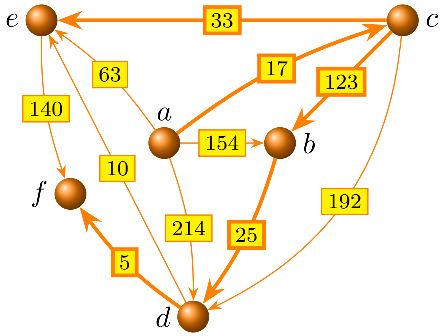


Рис. 80: Построение кратчайших путей.

385. Пункты определения: 1) в a рёбер не входит, а во все остальные вершины — входят; 2) если в v входит больше одного ребра, то есть два различных кратчайших пути, проходящих через v, тогда и до v есть не менее двух кратчайших путей; 3) по условию задачи.

№	R	w	$D[w]$	D					F				
				b	c	d	e	f	b	c	d	e	f
1	a	c	5	25	5	26	53	75	a	a	a	a	a
2	a, c	b	20	20	5	25	52	65	c	a	c	c	c
3	a, c, b	d	25	20	5	25	52	60	c	a	c	c	b
4	a, c, b, d	e	45	20	5	25	45	60	c	a	c	d	b
5	a, c, b, d, f	f	60	20	5	25	45	60	c	a	c	d	b

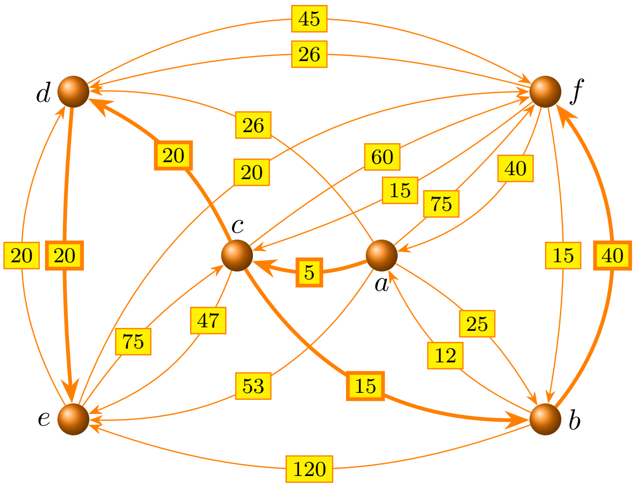


Рис. 81: Построение кратчайших путей.

№	R	w	$D[w]$	D: a	b	c	d	e	g	h	F: a	b	c	d	e	g	h
1	f	h	2	–	–	12	13	–	15	2	–	–	f	f	–	f	f
2	f, h	a	7	7	–	12	13	7	15	2	h	–	f	f	h	f	f
3	f, h, a	e	7	7	10	12	13	7	12	2	h	a	f	f	h	a	f
4	f, h, a, e	d	8	7	10	12	8	7	12	2	h	a	f	e	h	a	f
5	f, h, a, e, d	b	10	7	10	10	8	7	12	2	h	a	d	e	h	a	f
6	f, h, a, e, d, b	c	10	7	10	10	8	7	12	2	h	a	d	e	h	a	f
7	f, h, a, e, d, b, c	g	12	7	10	10	8	7	11	2	h	a	d	e	h	c	f

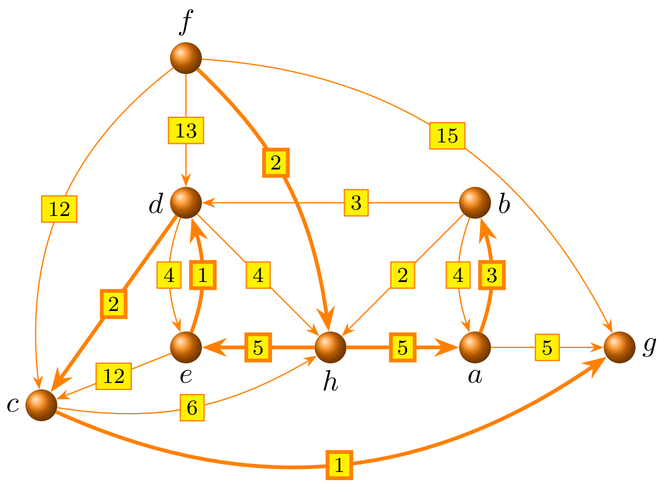


Рис. 82: Построение кратчайших путей.

№	R	w	$D[w]$	D							F						
				a	b	c	d	e	f	g	a	b	c	d	e	f	g
1	h	c	7	28	154	7	9	–	126	–	h	h	h	h	–	h	–
2	c, h	d	9	28	154	7	9	78	23	118	h	h	h	h	c	c	c
3	c, d, h	a	17	17	154	7	9	78	23	118	d	h	h	h	c	c	c
4	a, c, d, h	f	23	17	154	7	9	28	23	117	d	h	h	h	a	c	a
5	a, c, d, f, h	e	28	17	31	7	9	28	23	117	d	f	h	h	a	c	a
6	a, c, d, e, f, h	b	31	17	31	7	9	28	23	53	d	f	h	h	a	c	e
7	a, b, c, d, e, f, h	g	47	17	31	7	9	28	23	47	d	f	h	h	a	c	b

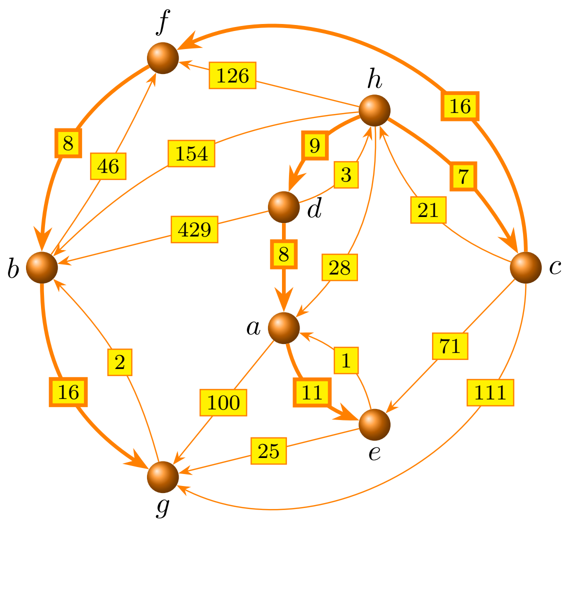


Рис. 83: Построение кратчайших путей.

Схемы из функциональных элементов

386. Рассмотрим схему с наименьшим количеством элементов, реализующую одну из этих функций. Так как функции $x_1 \oplus x_2$ и $x_1 \leftrightarrow x_2$ немонотонны, то требуется хотя бы одно отрицание. Но отрицание не может быть последним элементом, так как в противном случае предпоследний элемент реализовывал бы другую из функций с меньшей сложностью. Рассмотрим вариант, когда последним элементом является $\vee$, случай с $\wedge$ рассматривается аналогично. Следовательно, вычисляемая функция имеет вид $\Phi \vee \Psi$. Функции $x_1 \oplus x_2$ и $x_1 \leftrightarrow x_2$ не могут быть представлены в виде $\Theta \vee x_i$ или $\Theta \vee \neg x_i$, поэтому формулы Φ и Ψ не могут быть переменной или её отрицанием. Значит, для вычисления значений каждой из этих формул требуется бинарная операция. При вычислении значений Φ и Ψ нельзя использовать одну и ту же бинарную операцию с одними и теми же аргументами, так как суммарно получатся формулы вида $\Theta \vee \Theta$ или $\Theta \vee \neg\Theta$. Следовательно, всего требуется не менее четырёх элементов: отрицание, внешний $\vee$ и по одной бинарной операции для Φ и Ψ.

387. $f_v(x_1, x_2, x_3) = ((x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_3)) \vee (x_2 \wedge x_3) \equiv (x_1 \wedge \neg x_2) \vee x_3$. Схема на рис. 84.

388. $f_v(x_1, x_2, x_3, x_4) = ((x_1 \wedge x_3) \vee ((x_1 \vee x_2) \wedge x_4)) \wedge \neg(x_3 \vee x_4)$. Функция тождественно равна нулю.

389. Построить для f ДНФ, использовать одну переменную для отрицаний входов, вторую — для вычисления элементарных конъюнкций, третью — для вычисления ДНФ. Можно обойтись двумя переменными: использовать x для вычисления элементарных конъюнкций, y — для вычисления ДНФ. Для каждой элементарной конъюнкции $\neg a_1 \wedge \cdots \wedge \neg a_n \wedge b_1 \wedge \cdots \wedge b_m$ построить следующую программу: $x \leftarrow a_1$, $x \leftarrow x \vee a_2$, $\ldots$, $x \leftarrow x \vee a_n$, $x \leftarrow \neg x$, $x \leftarrow x \wedge b_1$, $\ldots$, $x \leftarrow x \vee b_m$, $y \leftarrow y \vee x$.

390. Очевидно, что программами в базисе $\mathcal{B}_0$, содержащими только один оператор, эта функция не вычисляется. Пусть в программе, вычисляющей

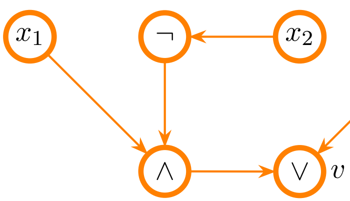


Рис. 84: Схема $\mathfrak{S}$ из задачи 387.

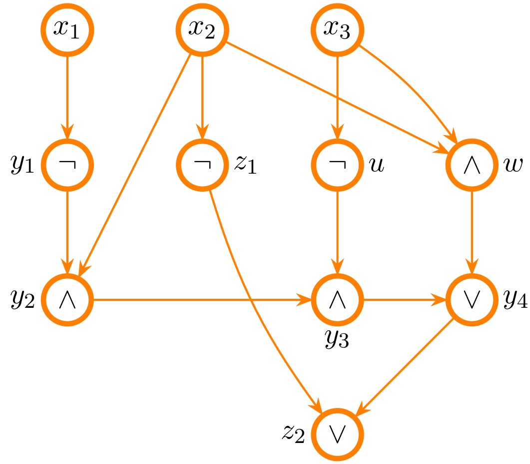


Рис. 85: Схема $\mathfrak{S}_\Pi$ из задачи 392.

функцию $\oplus$, присаивания выполняются только переменной y. Рассмотрим самую короткую из таких программ. Тогда все присваивания, кроме самого первого, должны иметь вид (а) $y \leftarrow \neg y$, (б) $y \leftarrow y \wedge x_i$, (в) $y \leftarrow y \vee x_i$, так как любые другие делают ненужным или начало программы (если правая часть присваивания не содержит y), или сам оператор (например, $y \leftarrow y \wedge y$). Такую программу можно было бы сократить. Если последний оператор имеет вид (б) или (в), то вычисляемая функция имеет вид $f'(x_1, x_2) \wedge x_i$ или $f'(x_1, x_2) \vee x_i$, она не совпадает с $\oplus$. Если последний оператор имеет вид (а), то предпоследний не может быть отрицанием, иначе два отрицания подряд можно было бы сократить. Следовательно, вычисляемая функция имеет вид $f'(x_1, x_2) \wedge \neg x_i$ или $f'(x_1, x_2) \vee \neg x_i$, такая функция тоже не совпадает с $\oplus$.

391. Вторая и третья.

392. Рис. 85. Глубина 5. Вычисляется функция $f(x_1, x_2, x_3) = (\neg x_1 \wedge x_2 \wedge \neg x_3) \vee (x_2 \wedge x_3) \vee \neg x_2 \equiv (\neg x_1 \wedge \neg x_3) \vee x_3 \vee \neg x_2 \equiv \neg x_1 \vee x_3 \vee \neg x_2 \equiv \neg(x_1 \wedge x_2) \vee x_3$. Более короткая программа: $y \leftarrow x_1 \wedge x_2$; $z \leftarrow \neg y$; $z \leftarrow z \vee x_3$. Более короткая схема приведена на рис. 86.

393. Так как базис $\mathcal{B}$ позволяет получить константу 0, то существует формула Φ_0 с переменной x_1 в этом базисе, значение которой всегда равно нулю. Значит, существует линейная программа Π_0, вычисляющая константу 0 в переменной y, причём в Π_0 каждая переменная не используется до того, как ей присвоено значение. Будем считать, что Π и Π_0 не имеют

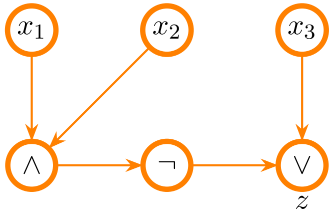


Рис. 86: Оптимизированная схема из задачи 392.

общих переменных. Преобразуем Π в Π', для каждой невходной переменной x все её вхождения до первого присваивания ей заменим на y: если $\Pi = (\Pi_1, x \leftarrow e, \Pi_2)$, где Π_1 не содержит присваиваний переменной x, то $\Pi' = ((\Pi_1)_y^x, x \leftarrow (e)_y^x, \Pi_2)$. Пусть $\Pi'' = \Pi_0\Pi'$. Очевидно, что Π'' вычисляет в переменных те же функции, что и Π, но теперь каждое использование невходной переменной происходит только после явного присваивания ей значения. Далее применяем индукцию по длине Π''. Если Π'' пусто, то вычисляются только значения входных переменных и схема состоит из одних входов. Если $\Pi'' = (\Pi''', z \leftarrow f(z_1, \ldots, z_k))$, то по индукционному предположению для Π''' уже построена схема $\mathfrak{S}_{\Pi'''}$, в вершинах v_{z_i} которой вычисляются те же функции, что и в переменных z_i программы Π'''. Тогда, чтобы построить $\mathfrak{S}_{\Pi''}$, нужно к $\mathfrak{S}_{\Pi'''}$ добавить новую вершину v_z, помеченную f, и рёбра из всех v_{z_i} в v_z.

394. Индукция по сложности схемы $\mathfrak{S}$, которая реализует булеву функцию f в базисе $\mathcal{B}_1$. Базис: если сложность равна нулю, то вершин-констант нет, поэтому схема уже в базисе $\mathcal{B}_0$, сложность не меняется. Индукционный шаг: допустим, что для схем сложности менее k утверждение доказано. Если вершин-констант нет, то схема остаётся той же самой, как и сложность. Пусть есть вершина-константа v. Она не может быть выходной, так как функция f константой не является. Пусть из v есть рёбра в вершины u_i. Тогда исключаем из $\mathfrak{S}$ вершину v, а с вершинами u_i производим следующие изменения. (а) Если u_i — отрицание, то меняем её на константу $\neg v$. (б) Если u_i — конъюнкция, $v = 0$, то меняем u_i на 0 и убираем второе входящее в u_i ребро. (в) Если u_i — конъюнкция, $v = 1$, то исключаем u_i, а второе входящее в u_i ребро перенаправляем во все вершины, куда вели рёбра из u_i. (г) Если u_i — дизъюнкция, $v = 1$, то меняем u_i на 1 и убираем второе входящее в u_i ребро. (д) Если u_i — дизъюнкция, $v = 0$, то исключаем u_i, а второе входящее в u_i ребро перенаправляем во все вершины, куда вели рёбра из u_i. В случаях (а), (б), (г) вершина u_i не может быть выходной,

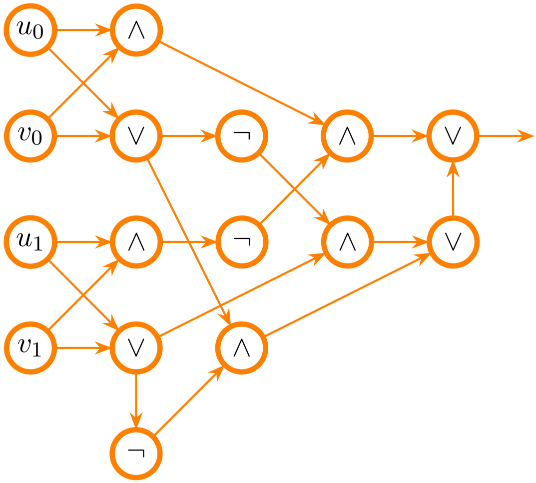


Рис. 87: Схема из задачи 396.

так как в ней вычисляется константа. В случаях (в) и (д), когда вершина u_i является выходной, мы делаем выходной вторую вершину, откуда в u_i вело ребро. Таким образом, получается схема меньшей k сложности. По индукционному предположению она эквивалентна некоторой схеме, которая не содержит констант и имеет не бóльшую сложность.

395. Для n-разрядных двоичных чисел выполнено $d = d + 2^n$ (так как в числе 2^n младшие n разрядов нулевые). Следовательно, $d = a - b = a + (2^n - b) = a + 1 + (2^n - 1 - b)$. Число $2^n - 1 - b$ получается просто инвертированием всех двоичных разрядов b: $2^n - 1 - b = \bar{b}$, следовательно, $d = a + \bar{b} + 1$. Для нахождения первой суммы достаточно использовать n инверторов $\neg$ и сумматор SUM_n. Для второй нужен инвертор $\neg$ (для младшего разряда) и $n - 1$ сумматор по модулю два $\mathfrak{S}_\oplus$ для остальных (суммируем текущий разряд с переносом из предыдущего). Таким образом, требуется n инверторов сложности 1, один сумматор SUM_n сложности $9n - 5$, ещё один инвертор сложности 1 и $n - 1$ схема $\mathfrak{S}_\oplus$ сложности 4. Всего получается $n + (9n - 5) + 1 + 4(n - 1) = 14n - 8$.

396. Рис. 87.

397. Рис. 88 и 89.

398. Функция $\mathrm{odd}(x_1, \ldots, x_n)$ принимает значение 1 тогда и только тогда, когда истинно одно из $\lfloor n/2 \rfloor$ условий для нечётных $i \leqslant n$: среди $x_1, \ldots, x_n$ есть i единиц, но нет $i + 1$ единицы. Каждое из условий требует не более

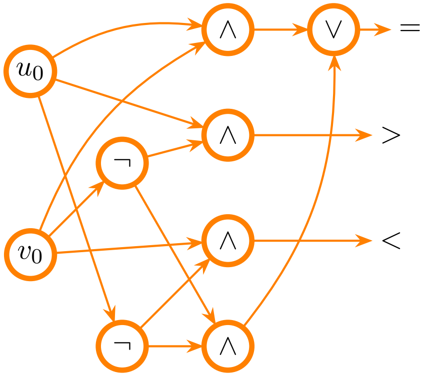


Рис. 88: Схема $\mathfrak{C}_1$ из задачи 397.

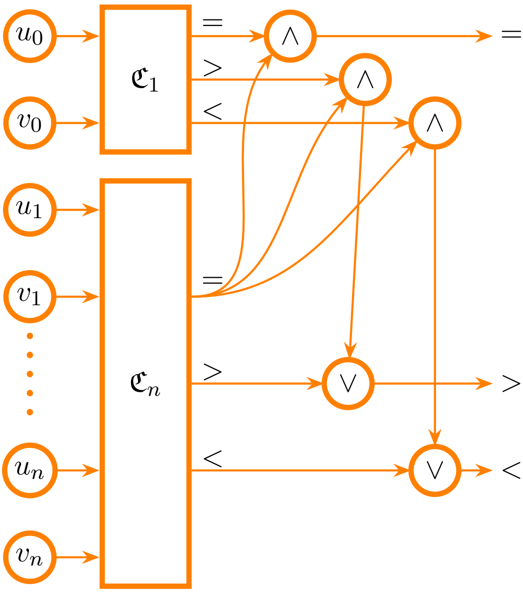


Рис. 89: Схемы $\mathfrak{C}_{n+1}$ из задачи 397.

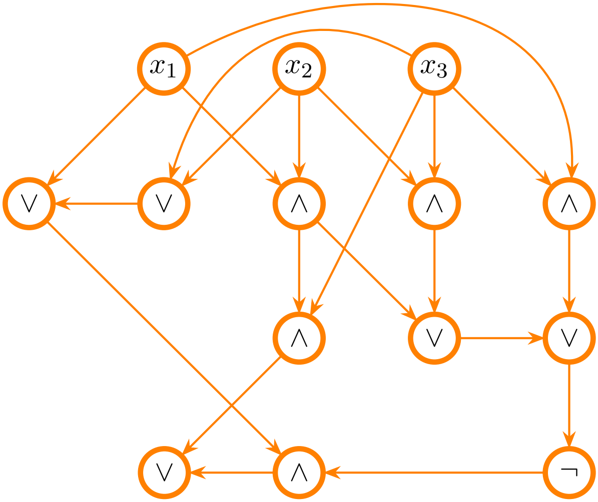


Рис. 90: Схема $\mathfrak{S}_{\mathrm{odd}}$ для трёх переменных.

одного отрицания, при $i = n$ отрицаний не требуется. Например, для $n = 3$: $\mathrm{odd}(x_1, \ldots, x_3) \equiv (x_1 \vee x_2 \vee x_3) \wedge \neg(x_1 \wedge x_2 \vee x_1 \wedge x_3 \vee x_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge x_3)$, рис. 90.

399. Базис. При $k = 0$ функция является монотонной, поэтому 1 не может меняться на 0 нигде, $2^{1-1} - 1 = 0$. Индукционный шаг. Допустим, что для k утверждение доказано. Рассмотрим вершину v, реализующую отрицание, которая находится ближе всего ко входам, то есть в неё нет путей из других вершин-отрицаний. Тогда на вход ей подаётся значение некоторой монотонной функции g. Эта функция g может сменить своё значение только один раз с 0 на 1. Разобьём последовательность s на две: $s_1 = (v_j, v_{j+1}, \ldots, v_{j+m'})$ и $s_0 = (v_{j+m'+1}, v_{j+m'+2}, \ldots, v_{j+m})$ в соответствии со значением функции $\neg g$. Для s_1 можно считать, что в схеме $\mathfrak{S}$ вместо вершины v стоит константа 1, для s_0 — константа 0. В любом случае количество отрицаний будет равно k, поэтому можно использовать индукционное предположение: и в s_1, и в s_0 существует не больше $2^k - 1$ позиций, где 1 меняется на 0. Тогда во всей последовательности s их не больше $2(2^k - 1) + 1 = 2^{k+1} - 1$.

400. Базис индукции: $f_0(x_1) = x_1$. Индукционный шаг. Допустим, функция f_k с $2^{k+1} - 1$ аргументами построена и при этом сохраняет ноль и единицу. Пусть $f_{k+1}(\bar{x}) = f_k(\bar{y})$, где $y_i = (x_i \wedge \neg x_{2^{k+1}}) \vee (x_{i+2^{k+1}} \wedge x_{2^{k+1}})$

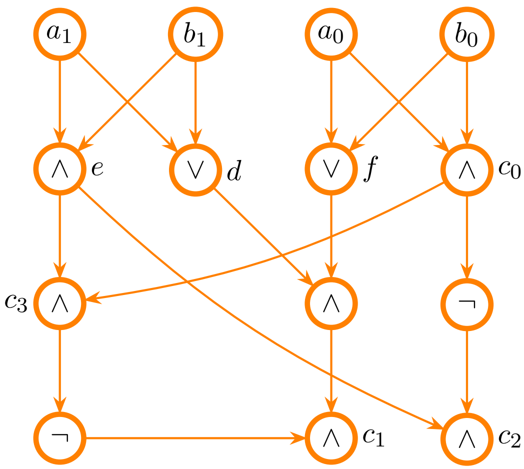


Рис. 91: Схема из задачи 401.

для $i < 2^{k+1}$. Тогда $(y_1, \ldots, y_{2^{k+1}-1}) = (x_1, \ldots, x_{2^{k+1}-1})$ при $x_{2^{k+1}} = 0$ и $(y_1, \ldots, y_{2^{k+1}-1}) = (x_{2^{k+1}+1}, \ldots, x_{2^{k+1}+2^{k+1}-1})$ при $x_{2^{k+1}} = 1$. По индукционному предположению в каждой из последовательностей $(v_0, \ldots, v_{2^{k+1}-1})$ и $(v_{2^{k+1}}, \ldots, v_{2^{k+2}-1})$ существует $2^k - 1$ позиция, где 1 меняется на 0. Во всей последовательности $(v_0, \ldots, v_{2^{k+2}-1})$ их будет $2^{k+1} - 1$. Для построения схемы для f_{k+1} потребуется только одно новое отрицание — для $x_{2^{k+1}}$.

401. Реализовать следующую схему: $c_0 = a_0 \wedge b_0$, $e = a_1 \wedge b_1$, $c_3 = e \wedge c_0$, $c_2 = e \wedge \neg c_0$, $d = a_1 \vee b_1$, $f = a_0 \vee b_0$, $c_1 = d \wedge f \wedge \neg c_3$. Рис. 91.

402. Пусть входы x_1, x_2, x_3, y — голоса «за» трёх обычных членов и председателя соответственно. Реализовать схему для формулы $(y \wedge (x_1 \vee (x_2 \vee x_3))) \vee ((x_1 \wedge x_2) \wedge x_3)$. Рис. 92.

403. Реализовать схему для формулы $(y \wedge z) \vee ((y \vee z) \wedge ((x_1 \wedge (x_2 \vee x_3)) \vee (x_2 \wedge x_3)))$. Рис. 93. Сложность 8, глубина 5.

404. Рис. 94.

405. **(а)** Рис. 95, $L(\mathfrak{L}_n) = 3 + 4(n-1) = 4n - 1$, $D(\mathfrak{L}_1) = 2$, $D(\mathfrak{L}_n) = 3$ при $n > 1$.

(б) Рис. 96, $L(\mathfrak{R}_n) = L(\mathfrak{L}_n) + 1 = 4n$, $D(\mathfrak{R}_n) = 3$.

(в) Рис. 97, $L(\mathfrak{C}_n) = 1 + L(\mathfrak{L}_2) + \cdots + L(\mathfrak{L}_n) = 1 + 4(2 + \cdots + n) - (n-1) = 4(n(n+1)/2 - 1) - n + 2 = 2n(n+1) - n - 2 = 2n^2 + n - 2$, $D(\mathfrak{C}_1) = 1$, $D(\mathfrak{C}_n) = 3(n-1)$ при $n > 1$.

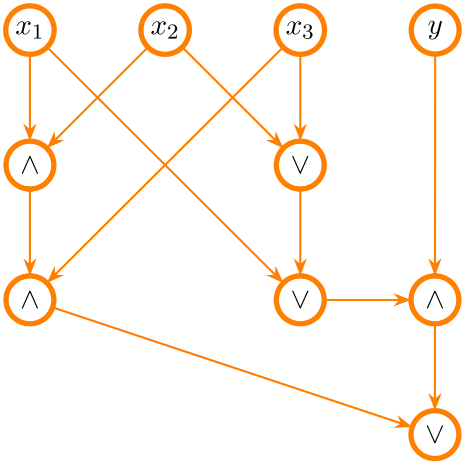


Рис. 92: Схема из задачи 402.

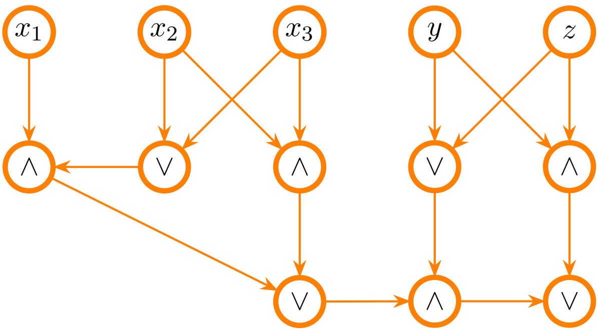


Рис. 93: Схема из задачи 403.

406. **(а)** Рис. 98, $L(\mathfrak{F}_n) = 1+7+11+16(n-3) = 16n-29$, $D(\mathfrak{F}_n) = 3(n-1) = 3n-3$.
(б) Рис. 99, $L(\mathfrak{G}_n) = 1+7+12(n-2)+1 = 12n-16$, $D(\mathfrak{G}_n) = 3(n-1)+1 = 3n-2$.
(в) Рис. 100, $L(\mathfrak{T}_k^n) = 2n^2+n-2+n-k = 2n^2+2n-(k+2)$, $D(\mathfrak{T}_k^n) = 1+3(n-k)+(n-k) = 4(n-k)+1$ при $k>0$.
407. Рис. 101, $L=29$, $D=8$.
408. Рис. 102, $L=12$, $D=6$.

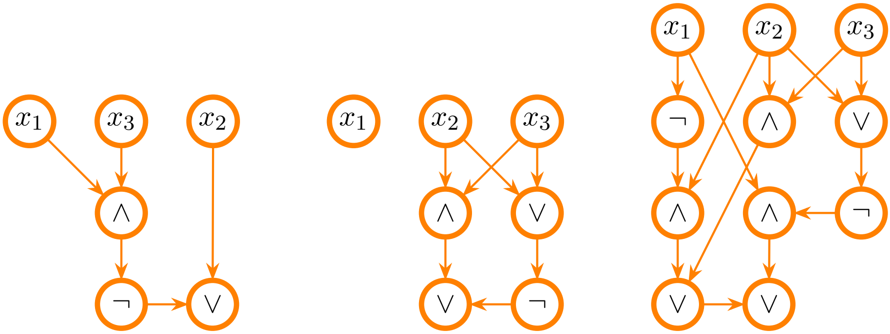


Рис. 94: Схемы из задачи 404.

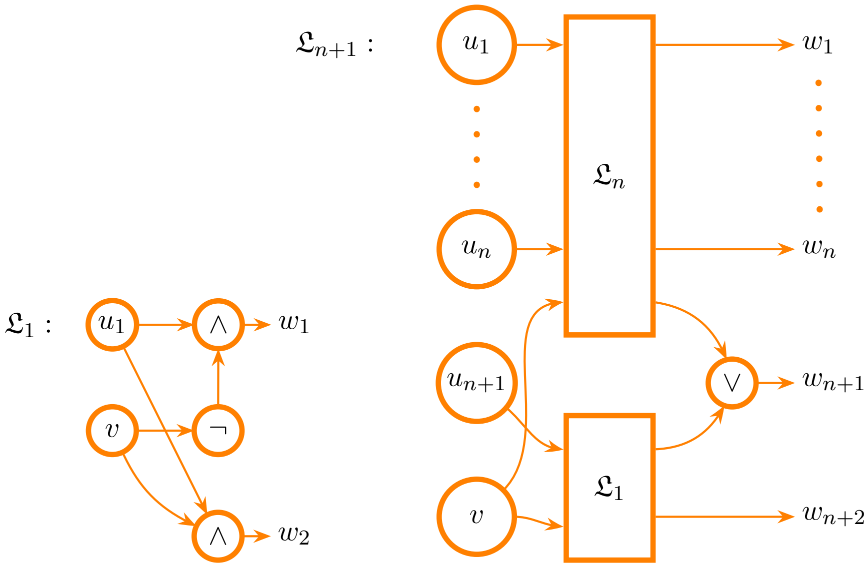


Рис. 95: Схемы $\mathfrak{L}_n$ из задачи 405.

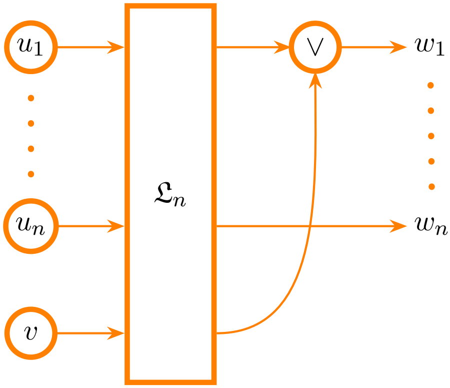


Рис. 96: Схема $\mathfrak{R}_n$ из задачи 405.

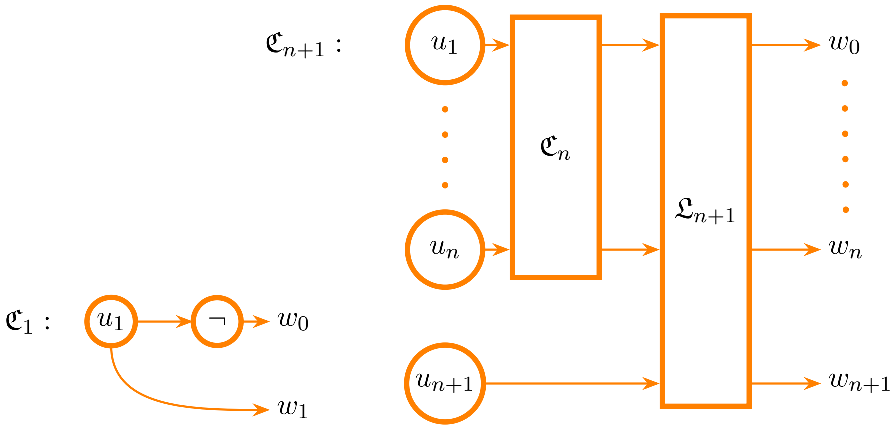


Рис. 97: Схемы $\mathfrak{C}_n$ из задачи 405.

409. Рис. 103, $L = 14$, $D = 8$.
410. Рис. 104.

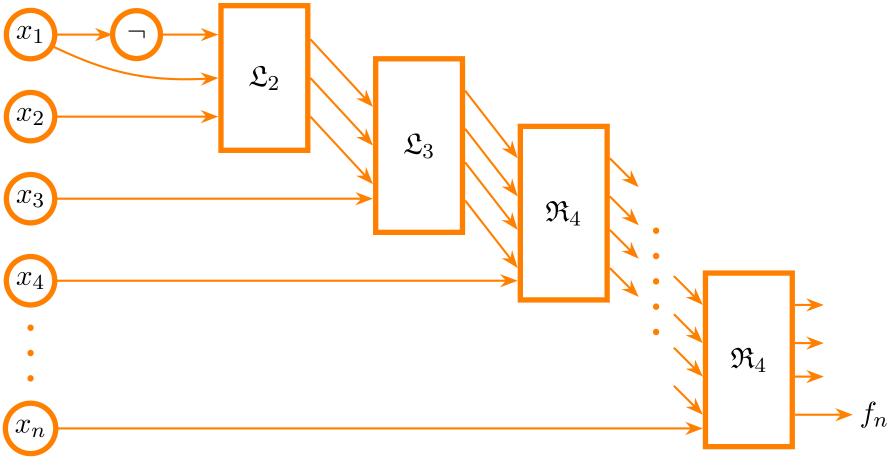


Рис. 98: Схема для функции f_n из задачи 406.

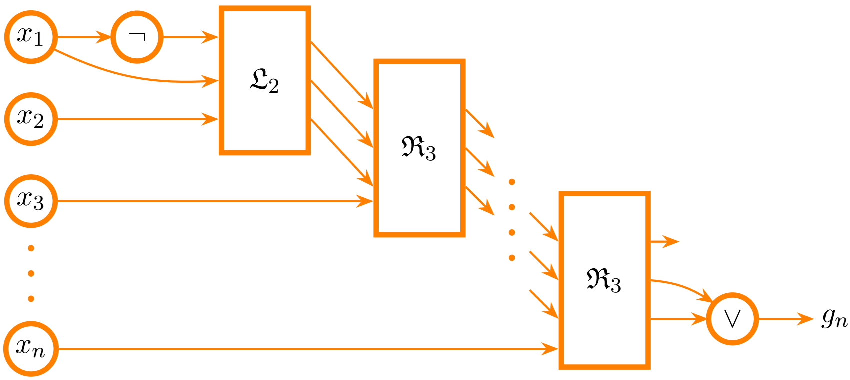


Рис. 99: Схема для функции g_n из задачи 406.

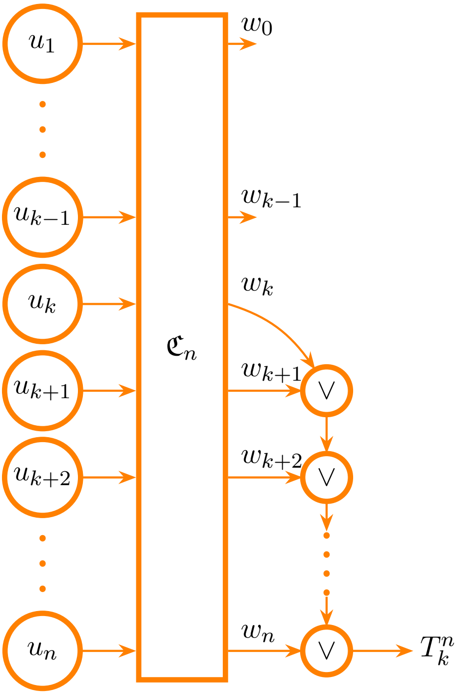


Рис. 100: Схема для функции T_k^n из задачи 406.

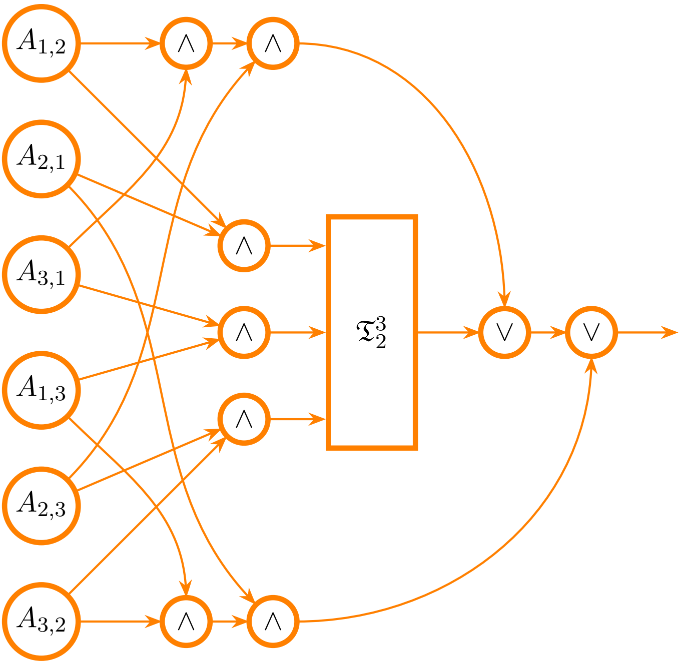


Рис. 101: Схема для функции из задачи 407.

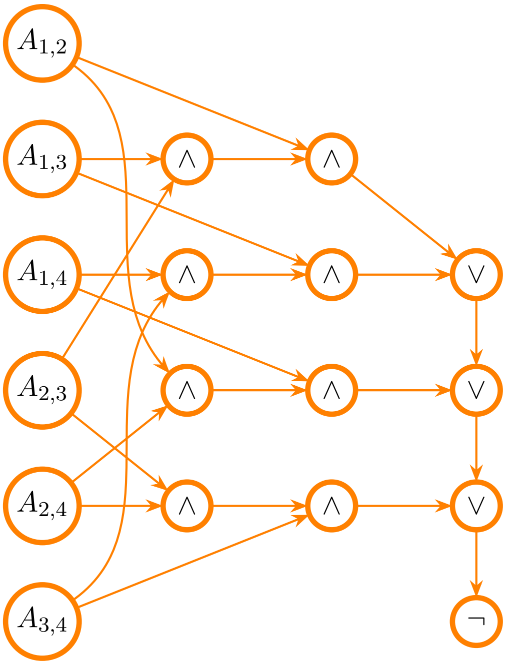


Рис. 102: Схема для функции из задачи 408.

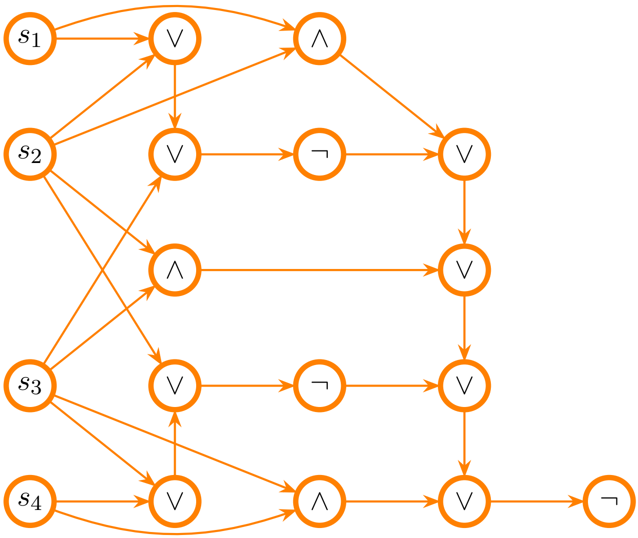


Рис. 103: Схема для функции из задачи 409.

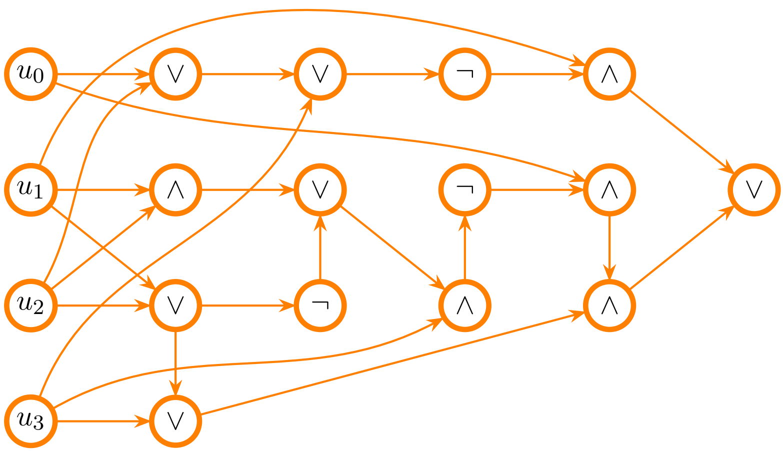


Рис. 104: Схема для функции из задачи 410.

Упорядоченные бинарные диаграммы решений

411. Полное БДР содержит 2^{i-1} вершин для переменной x_i. При слиянии останется 2 вершины для x_n, 4 для x_{n-1} и т. д., 2^{i+1} для x_{n-i}. Всего $1+2\cdot 2+2\cdot 2^2+\cdots \leqslant 2^{n/2+2}-1$.

412. **(а)** Может быть самое большое две вершины x_2 (0- и 1-сыновья x_1) и самое большое две вершины x_3 (0- и 1-рёбра должны вести в 0 и 1 или наоборот). Следовательно, может быть максимум семь вершин.

(б) Как указано в пункте (а), в УБДР должны быть две вершины $x_2^{(0)}$ и $x_2^{(1)}$ — 0- и 1-сыновья x_1 соответственно, а также две вершины $x_3^{(0,1)}$ и $x_3^{(1,0)}$. Следовательно, УБДР отличаются только рёбрами, исходящими из x_2. Для вершины $x_2^{(0)}$ есть $A_4^2 = 12$ вариантов 0- и 1-сыновей, рассмотрим их все и возможности для $x_2^{(1)}$ (обе вершины x_3 должны быть достижимы). (i) $(0,1)$ и $(1,0)$: $(x_3^{(0,1)}, x_3^{(1,0)})$, $(x_3^{(1,0)}, x_3^{(0,1)})$. (ii) $(0, x_3^{(0,1)})$, $(1, x_3^{(0,1)})$, $(x_3^{(0,1)}, 0)$, $(x_3^{(0,1)}, 1)$: варианты, содержащие $x_3^{(1,0)}$, $(0, x_3^{(1,0)})$, $(1, x_3^{(1,0)})$, $(x_3^{(1,0)}, 0)$, $(x_3^{(1,0)}, 1)$, $(x_3^{(0,1)}, x_3^{(1,0)})$, $(x_3^{(1,0)}, x_3^{(0,1)})$. (iii) $(0, x_3^{(1,0)})$, $(1, x_3^{(1,0)})$, $(x_3^{(1,0)}, 0)$, $(x_3^{(1,0)}, 1)$: варианты, содержащие $x_3^{(0,1)}$, аналогично. (iv) $(x_3^{(0,1)}, x_3^{(1,0)})$ и $(x_3^{(1,0)}, x_3^{(0,1)})$: все варианты, кроме совпадающих. Всего $2\cdot 2+4\cdot 6+4\cdot 6+2\cdot 11 = 74$ варианта.

413. Поменять все метки (и рёбер, и вершин) 0 и 1 на противоположные.

414. Обратная индукция по глубине вершин. Для стоков 0 и 1 берём $\Pi_0 = y \leftarrow 0$ и $\Pi_1 = y \leftarrow 1$ соответственно. Если вершина w с переменной x имеет 0-сына w_0 и 1-сына w_1, для которых уже построены программы Π_{w_0} и Π_{w_1}, то $\Pi_w = \textsf{if } x \textsf{ then } \Pi_{w_1} \textsf{ else } \Pi_{w_0}$. Рис. 105.

415. Пронумеруем вершины $\mathfrak{D}$ согласно задаче 284. Тогда для каждой внутренней вершины v_i с меткой x и 0- и 1-сыновьями v_j и v_k соответственно строим оператор α_i $\textsf{if } x \textsf{ then } \alpha_j \textsf{ else } \alpha_k$. По стокам v_i и v_j с результатом 0 и 1 соответственно строим операторы $\alpha_i\ y \leftarrow 0;\ \alpha_k$ и $\alpha_j\ y \leftarrow 1;\ \alpha_k$, где α_k — заключительная метка.

416. Индукцией по высоте вершин v_i строим линейные программы, вычисляющие соответствующие функции f_i в переменной v_i. Для стока r_0: $r_0 \leftarrow \neg x, r_0 \leftarrow r_0 \wedge x$, где x — любая переменная. Аналогично для стока r_1: $r_1 \leftarrow \neg x$, $r_1 \leftarrow r_1 \vee x$. Пусть уже построена линейная программа Π_h, которая вычисляет функции для всех вершин v высоты не превосходящей h. Тогда для $h+1$ строим программу Π_{h+1}, добавляя к концу Π_h новые операторы следующим образом. Если вершина v_i высоты $h+1$ помечена переменной x и имеет 0-сына v_j и 1-сына v_k, то их высота не больше h. Следовательно, программа Π_h уже вычисляет значение соответствующих функций f_j и

```
if y then
    if x then  u ← 0
    else  u ← 1
else
    if z then  u ← 1
    else  u ← 0
```

```
if x1 then
    if x2 then
        if x4 then  u ← 1
        else  u ← 0
    else
        if x3 then
            u ← 1
        else
            if x4 then  u ← 1
            else  u ← 0
else
    if x2 then
        if x4 then  u ← 0
        else  u ← 1
    else
        if x3 then
            u ← 1
        else
            if x4 then  u ← 1
            else  u ← 0
```

Рис. 105: Ветвящиеся программы из задачи 414.

f_k в переменных v_j и v_k. Тогда для вычисления f_i в переменной v_i мы добавляем $v' \leftarrow \neg x$, $v' \leftarrow v' \wedge v_k$, $v_i \leftarrow x \wedge v_j$, $v_i \leftarrow v_i \vee v'$. Очевидно, что длина такой линейной программы не превосходит $4n$.

417. Рис. 106.

418. Рис. 107, $L = 2n - 1$.

419. **(а)** Рис. 108: 10 вершин, на каждом шаге вычисляем конъюнкцию x_{2i-1} и x_{2i}, а затем — сложение по модулю два с предыдущим результатом. Всего по одной вершине для x_1 и x_2, по две для остальных.

(б) Всего 17 вершин: для x_6 — две вершины (функции x_6 и $x_6 \oplus 1$), для x_4 — четыре (функции x_4, $x_4 \oplus 1$, $x_4 \oplus x_6$, $x_4 \oplus x_6 \oplus 1$), для x_2 — четыре (функции x_2, $x_2 \oplus x_4$, $x_2 \oplus x_6$, $x_2 \oplus x_4 \oplus x_6$). Вершины x_1, x_3, x_5 образуют полное дерево из 7 вершин.

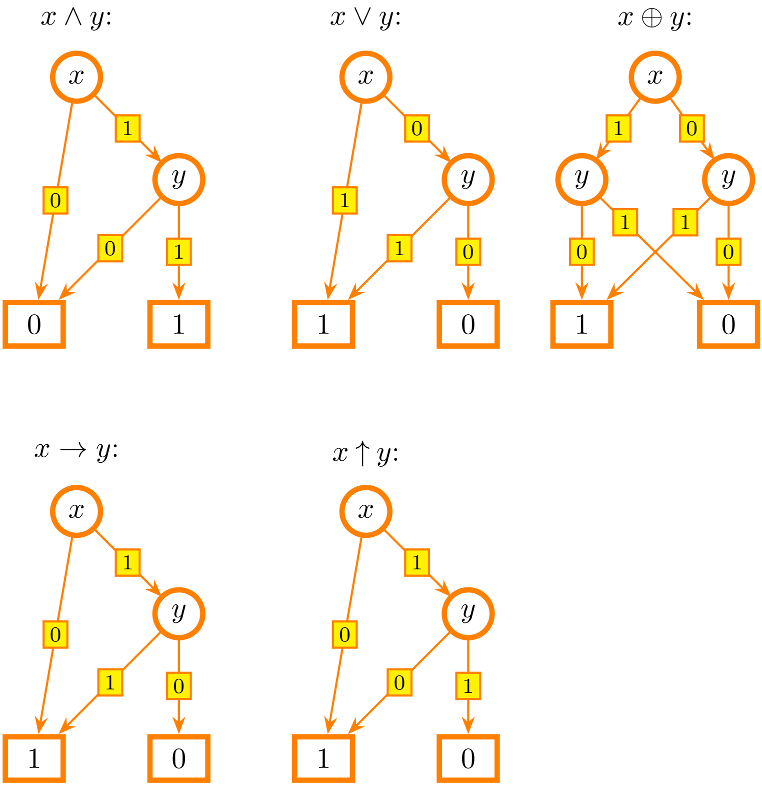


Рис. 106: УБДР для основных булевых функций.

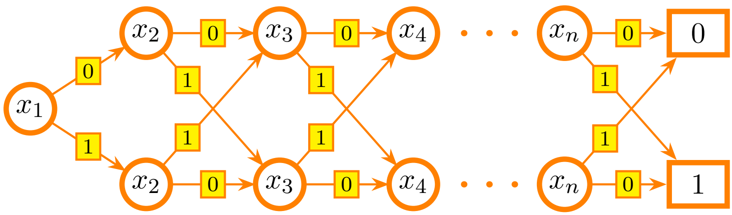


Рис. 107: УБДР для функции odd.

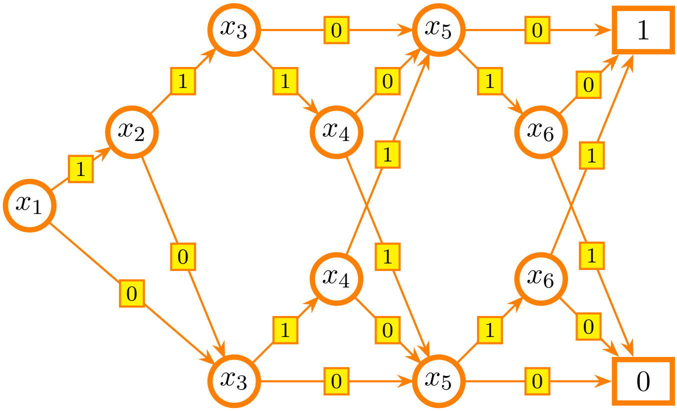


Рис. 108: УБДР для функции f.

420. Рис. 109. Функция $(x_1 \to x_3) \oplus x_2$. Значения функции при лексикографическом упорядочении значений аргументов: (1100 0110).

421. Базис: если $n = 1$, то, очевидно, будет только одна внутренняя вершина x_1. Допустим, что для любой совершенной ДНФ Φ_n с n переменными утверждение доказано. Допустим, что первой переменной будет x_i. Представим совершенную ДНФ Φ_{n+1} в виде $\Psi = (x_i \wedge \Phi'_n) \vee (\neg x_i \wedge \Phi''_n)$, где Φ'_n и Φ''_n — совершенные ДНФ с n переменными $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{n+1}$. По индукционному предположению для Φ'_n и Φ''_n можно построить УБДР $\mathfrak{D}'$ и $\mathfrak{D}''$ соответственно, суммарное количество внутренних вершин в которых будет не больше, чем сумма их длин. Тогда УБДР для Ψ строится добавлением нового истока x_i, который имеет 0- и 1-сыновьями соответственно истоки УБДР $\mathfrak{D}''$ и $\mathfrak{D}'$. Количество внутренних вершин при этом возрастёт на единицу. В то же время длина Φ_{n+1} превосходит сумму длин Φ'_n и Φ''_n тоже как минимум на единицу.

422. **(а)** Рис. 110.

(б) Возьмём УБДР наименьшего размера. Покажем, что от истока до уровня x_n УБДР является полным бинарным деревом. В самом деле, ни одну переменную x_i нельзя пропустить, так как значения функции на наборах $(\sigma_1, \ldots, \sigma_{i-1}, 0, \sigma_{i+1}, \ldots, \sigma_n, \neg\sigma_n, \ldots, \neg\sigma_{i+1}, 1, \neg\sigma_{i-1}, \ldots, \neg\sigma_1)$ и $(\sigma_1, \ldots, \sigma_{i-1}, 1, \sigma_{i+1}, \ldots, \sigma_n, \neg\sigma_n, \ldots, \neg\sigma_{i+1}, 1, \neg\sigma_{i-1}, \ldots, \neg\sigma_1)$ различны. Аналогично показывается, что два разных пути не могут вести в одну и ту же вершину x_j: если $(\sigma_1, \ldots, \sigma_n)$ и $(\tau_1, \ldots, \tau_n)$ — два разных пути, то значения функции на $(\sigma_1, \ldots, \sigma_n, \neg\sigma_n, \ldots, \neg\sigma_1)$ и $(\tau_1, \ldots, \tau_n, \neg\sigma_n, \ldots, \neg\sigma_1)$

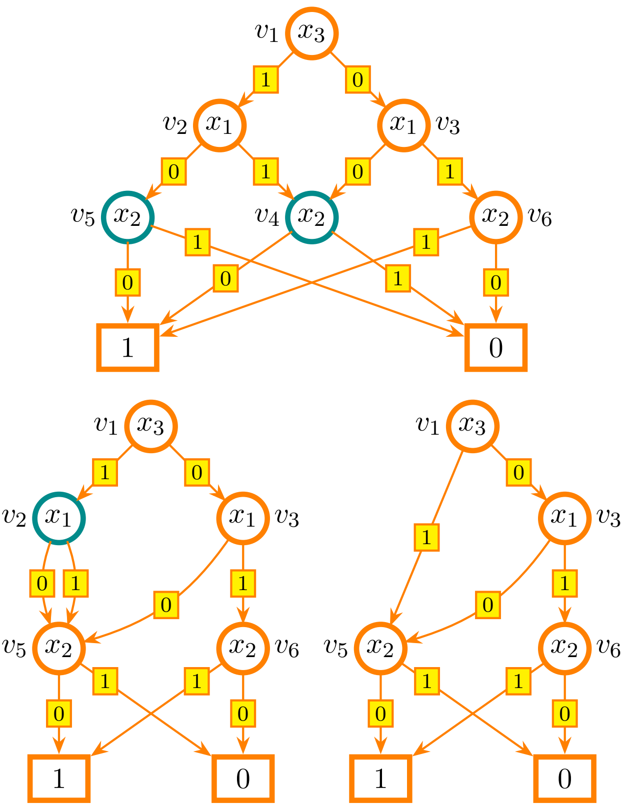


Рис. 109: Сокращение УБДР.

различны. По той же причине, что и ранее, из каждой вершины x_i есть путь в какую-ту вершину y_j для любых i и j: значения функции на наборах $(\sigma_1, \ldots, \sigma_n, \neg\sigma_n, \ldots, \neg\sigma_{j+1}, 1, \neg\sigma_{j-1}, \ldots, \neg\sigma_1)$ и $(\sigma_1, \ldots, \sigma_n, \neg\sigma_n, \ldots, \neg\sigma_{j+1}, 0, \neg\sigma_{j-1}, \ldots, \neg\sigma_1)$ различны. Далее заметим, что количество вершин y_i равняется количеству вершин x_i, то есть 2^{i-1}. В самом деле, если их меньше, то два разных пути до x_i могут быть продолжены до одной и той же вершины y_i: $(\sigma_1, \ldots, \sigma_{i-1}, \sigma_i, \sigma_{i+1}, \ldots, \sigma_n, \sigma'_n, \ldots, \sigma'_{i+1})$ и $(\tau_1, \ldots, \tau_{i-1}, \tau_i, \tau_{i+1}, \ldots, \tau_n, \tau'_n, \ldots, \tau'_{i+1})$. Заметим, что можно полагать $\sigma_n \wedge \sigma'_n = \cdots = \sigma_{i+1} \wedge \sigma'_{i+1} = \tau_n \wedge \tau'_n = \cdots = \tau_{i+1} \wedge \tau'_{i+1} = 0$ (иначе значение функции уже известно и нет необходимости проверять значение y_i). Но тогда значение функции будет различным, если эти наборы продолжить

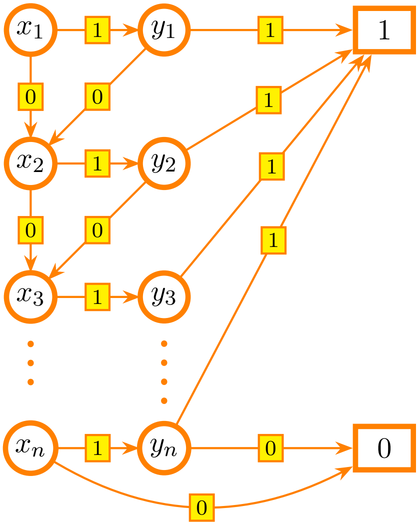


Рис. 110: УБДР для функции p_n.

так: $(\neg\sigma_i, \ldots, \neg\sigma_1)$, первый должен привести к 0, второй — к 1. Таким образом, УБДР содержит не менее $2 \cdot (2^n - 1) = 2^{n+1} - 2$ вершины. УБДР нужного размера выглядит так. Истоком является $x_1^{()}$. У вершины $x_i^{(u)}$ сыновья: 0-сын $x_{i+1}^{(u,0)}$ и 1-сын $x_{i+1}^{(u,1)}$ для $i < n$. У вершины $x_n^{(u,1,0,\ldots,0)}$ сыновья: 0-сын $y_{n-1-m}^{(u)}$ и 1-сын $y_n^{(u)}$. У вершины $y_i^{(u,1,0,\ldots,0)}$ сыновья: 0-сын $y_{i-1-m}^{(u)}$ и 1-сын 1 для $i > 1$. С помощью m в обоих случаях обозначено количество завершающих нулей в последовательности $(u, 1, 0, \ldots, 0)$. Если эта последовательность не содержит единиц, то соответствующий 0-сын — это 0. У вершины $y_1^{()}$ сыновья: 0-сын 0 и 1-сын 1.

423. **(а)** Удалить все вершины v, помеченные x_i, если было ребро $(u, v; \tau)$, то заменить его ребром $(u, w; \tau)$, где w — σ-сын вершины v. Если v была истоком диаграммы $\mathfrak{D}$, то сделать истоком $\mathfrak{D}'$ её σ-сына. Удалить недостижимые из истока $\mathfrak{D}'$ вершины. Полученная УБДР может не быть сокращённой: если в диаграмме $\mathfrak{D}_3$ на рис. 22 заменить значение переменной x_3 на 0, то оба сына v_7 будут равны v_3.

(б) Рис. 111.

424. **(а)** Рис. 112. **(б)** Рис. 113. **(в)** Нет.

(г) Когда мы доходим до вершины с переменной x_i, нам нужно знать, какое

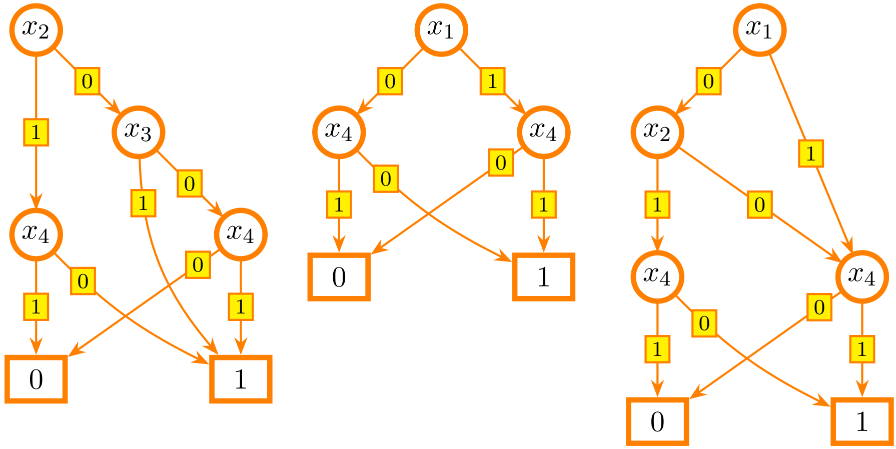


Рис. 111: УБДР для функций из задачи 423.

количество предыдущих переменных имеет значение 1. Таким образом, для x_1 нужна одна вершина, для x_2 — две, для x_3 — три, ..., для x_k — k вершин. Наоборот, значение переменной x_n требуется анализировать, только если среди предыдущих переменных в точности $k-1$ единица (одна вершина), значение x_{n-1} — если $k-1$ или $k-2$ единицы (две вершины), ..., значение x_{n-k+1} — от 0 до $k-1$ единиц. Таким образом, «параллелограмм», как в пункте (б), является минимальной УБДР с количеством внутренних вершин $k(n-k+1)$.

425. Рис. 114, 115 и 116 на стр. 305.

426. Рис. 117 и 118, $L_{f_n} = 4(n-3)+6 = 4n-6$, $L_{g_n} = 3(n-2)+3 = 3n-3$.

427. Рис. 119, $L = 14$.

428. Рис. 120, $L = 15$.

429. Рис. 121, $L = 8$.

430. Рис. 122, $L = 8$.

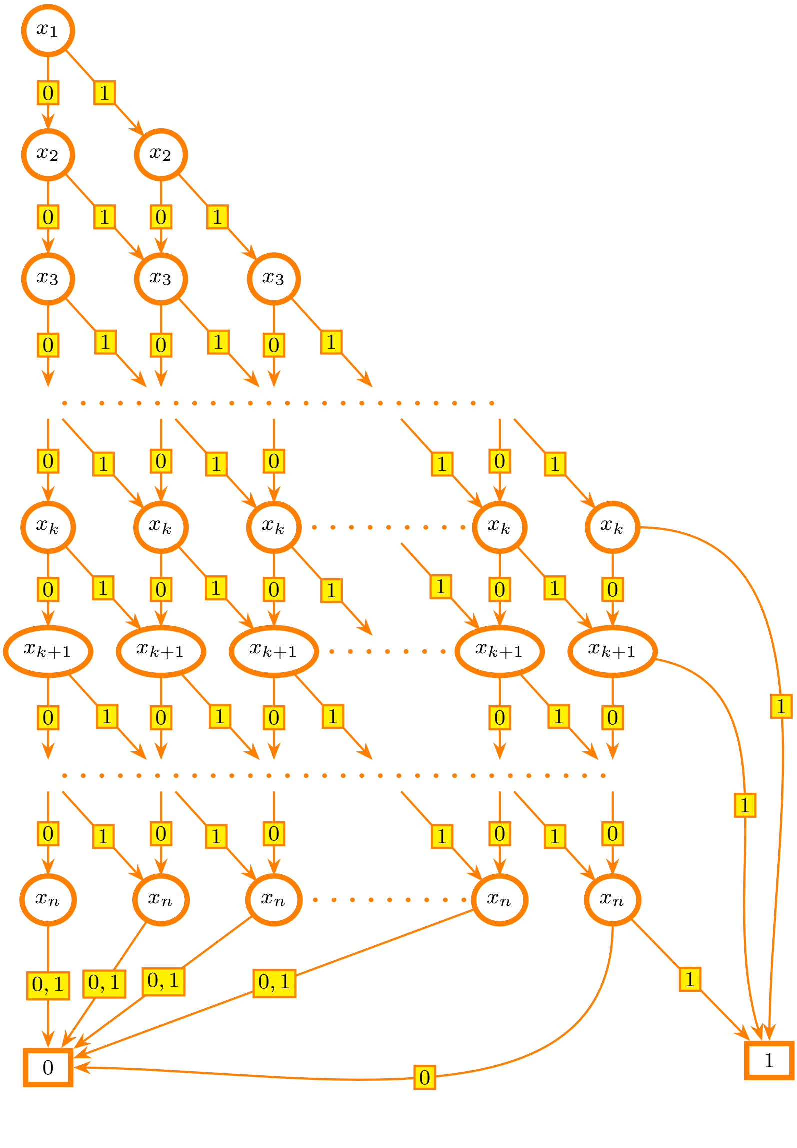


Рис. 112: УБДР для функции T_k^n из задачи 424.

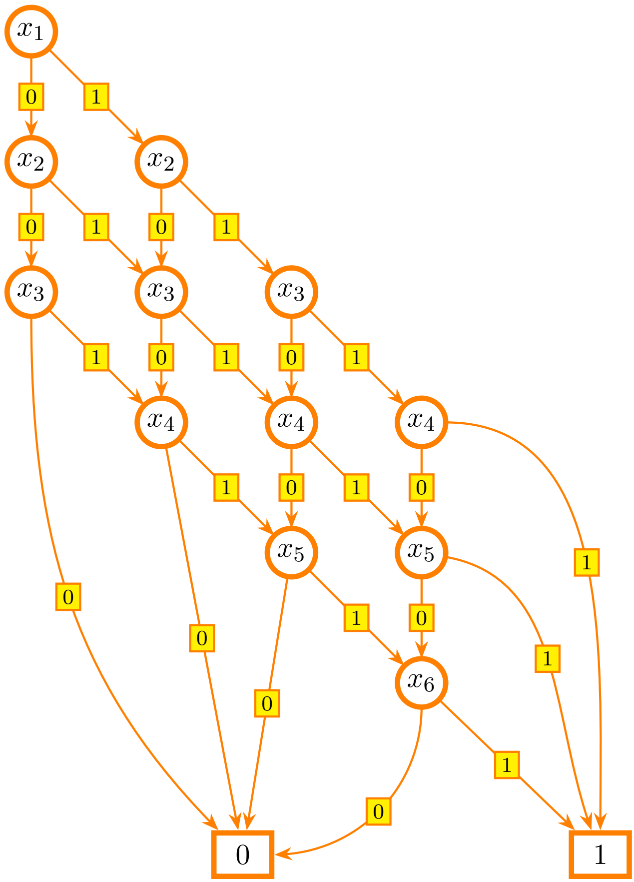


Рис. 113: Минимальная УБДР для функции T_4^6 из задачи 424.

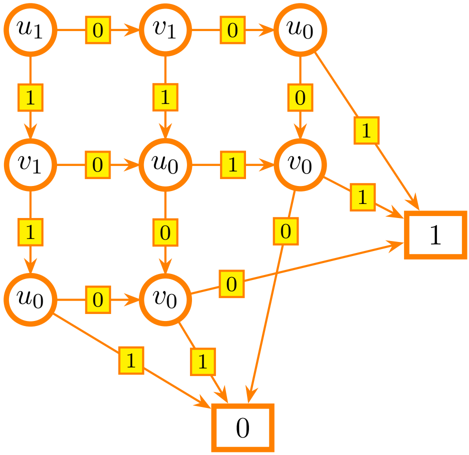


Рис. 114: УБДР для задачи 425.

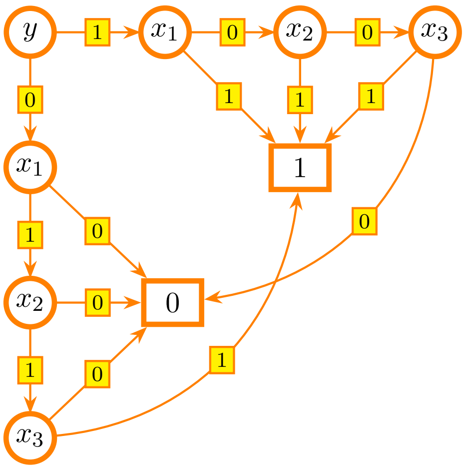


Рис. 115: УБДР для задачи 425.

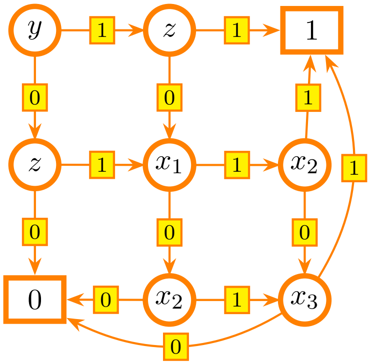


Рис. 116: УБДР для задачи 425.

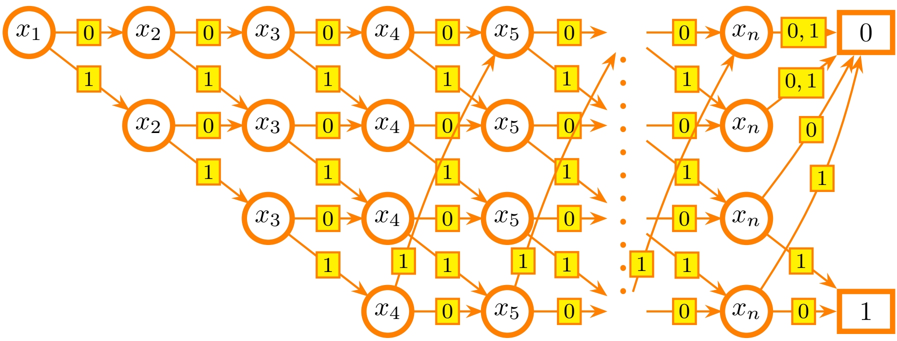


Рис. 117: УБДР для функции f_n из задачи 426.

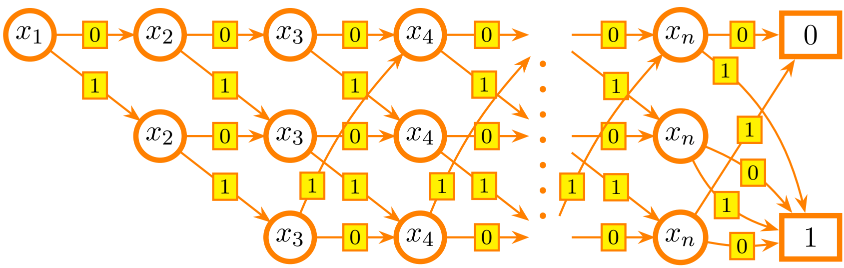


Рис. 118: УБДР для функции g_n из задачи 426.

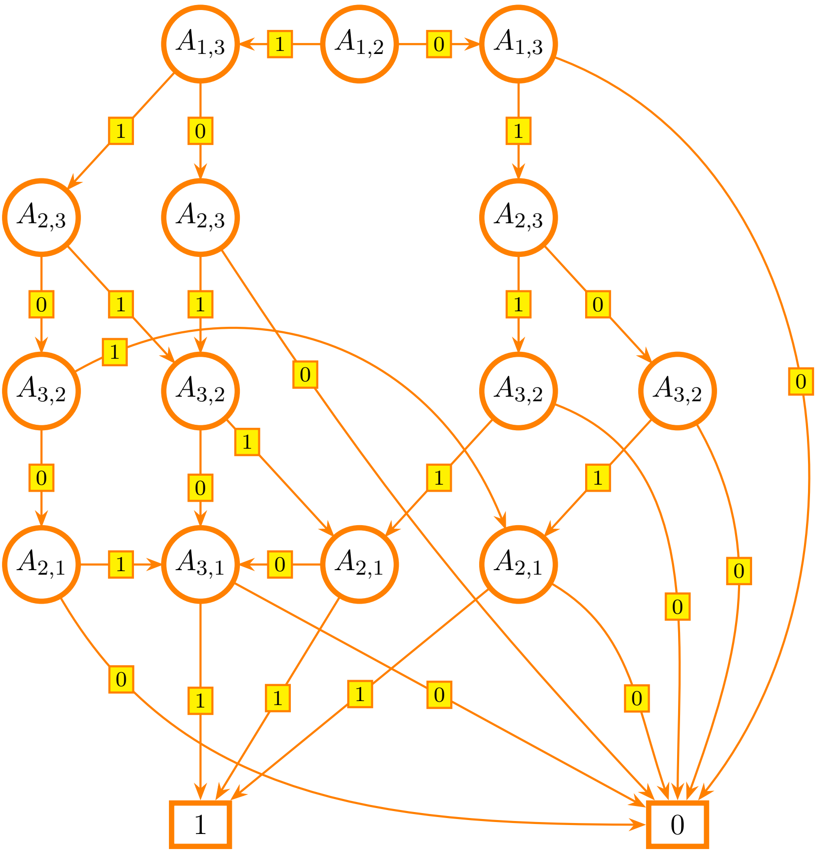


Рис. 119: УБДР для определения сильной связности графа, задача 427.

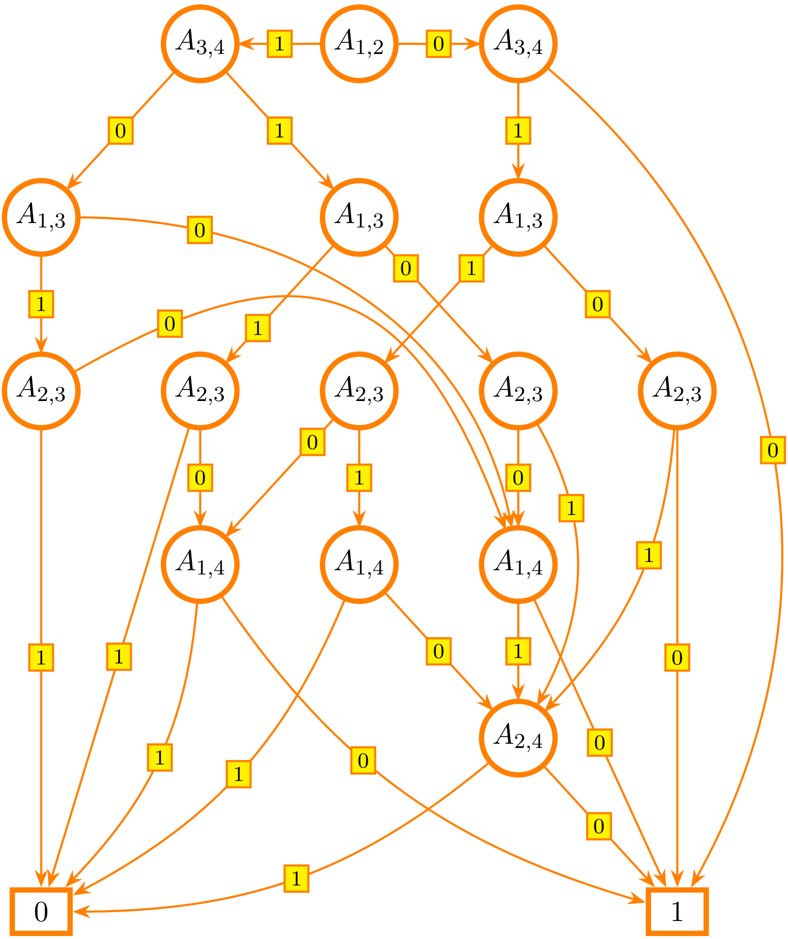


Рис. 120: УБДР для определения двудольности графа, задача 428.

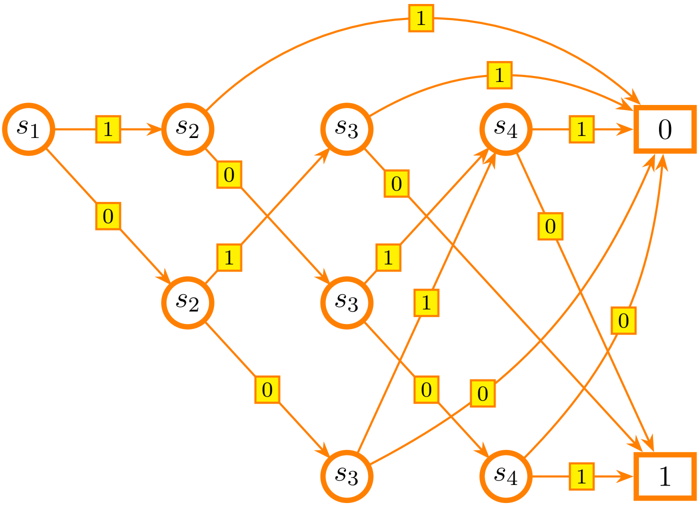


Рис. 121: УБДР для задачи 429.

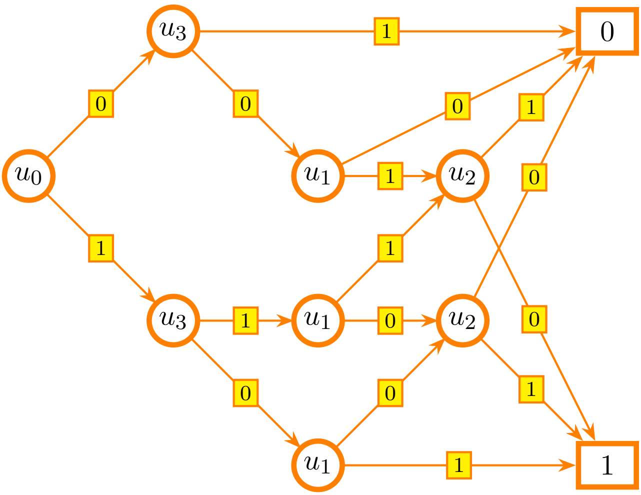


Рис. 122: УБДР для задачи 430.

Конечные автоматы: преобразователи и распознаватели

431. **(а)** ТАРТАР; **(б)** 100001.

432. $c = 3$, в двоичной записи 11.

433. Входной алфавит: $1, 2, 10$ — приём монет. Выходной алфавит: $0, 1, 2, 3, 4$ — показания табло, 5 — выдача одной монеты 5 рублей. $q_0, 1 \to q_1, 1$; $q_0, 2 \to q_2, 2$; $q_0, 10 \to q_0, 055$; $q_1, 1 \to q_2, 2$; $q_1, 2 \to q_3, 3$; $q_1, 10 \to q_1, 155$; $q_2, 1 \to q_3, 3$; $q_2, 2 \to q_4, 4$; $q_2, 10 \to q_2, 255$; $q_3, 1 \to q_4, 4$; $q_3, 2 \to q_0, 05$; $q_3, 10 \to q_0, 355$; $q_4, 1 \to q_0, 05$; $q_4, 2 \to q_1, 15$; $q_4, 10 \to q_0, 455$.

434. Входной алфавит: $1, 2, 5$ — приём монет, b — нажатие на кнопку. Выходной алфавит: r и g — красная и зелёная лампочки, c — кофе, $1, 2, 5$ — выдача монет. $q_0, 1 \to q_1, r$; $q_0, 2 \to q_2, r$; $q_0, 5 \to q_5, r$; $q_0, b \to q_0, r$; $q_1, 1 \to q_2, r$; $q_1, 2 \to q_3, r$; $q_1, 5 \to q_6, r$; $q_1, b \to q_1, r$; $q_2, 1 \to q_3, r$; $q_2, 2 \to q_4, r$; $q_2, 5 \to q_7, r$; $q_2, b \to q_2, r$; $q_3, 1 \to q_4, r$; $q_3, 2 \to q_5, r$; $q_3, 5 \to q_8, g$; $q_3, b \to q_3, r$; $q_4, 1 \to q_5, r$; $q_4, 2 \to q_6, r$; $q_4, 5 \to q_9, g$; $q_4, b \to q_4, r$; $q_5, 1 \to q_6, r$; $q_5, 2 \to q_7, r$; $q_5, 5 \to q_{10}, g$; $q_5, b \to q_5, r$; $q_6, 1 \to q_7, r$; $q_6, 2 \to q_8, g$; $q_6, 5 \to q_{11}, g$; $q_6, b \to q_6, r$; $q_7, 1 \to q_8, g$; $q_7, 2 \to q_9, g$; $q_7, 5 \to q_{12}, g$; $q_7, b \to q_7, r$; $q_8, 1 \to q_8, g1$; $q_8, 2 \to q_8, g2$; $q_8, 5 \to q_8, g5$; $q_8, b \to q_0, cr$; $q_9, 1 \to q_9, g1$; $q_9, 2 \to q_9, g2$; $q_9, 5 \to q_9, g5$; $q_9, b \to q_0, c1r$; $q_{10}, 1 \to q_{10}, g1$; $q_{10}, 2 \to q_{10}, g2$; $q_{10}, 5 \to q_{10}, g5$; $q_{10}, b \to q_0, c2r$; $q_{11}, 1 \to q_{11}, g1$; $q_{11}, 2 \to q_{11}, g2$; $q_{11}, 5 \to q_{11}, g5$; $q_{11}, b \to q_0, c12r$; $q_{12}, 1 \to q_{12}, g1$; $q_{12}, 2 \to q_{12}, g2$; $q_{12}, 5 \to q_{12}, g5$; $q_{12}, b \to q_0, c22r$.

435. Входной алфавит: $1, 2$ — нажатие на кнопки. Выходной алфавит: n, h, m, s — режим, H, M, S — увеличение соответствующего числа. Программа: $q_0, 1 \to q_1, h$; $q_0, 2 \to q_0, n$; $q_1, 1 \to q_2, m$; $q_1, 2 \to q_1, hH$; $q_2, 1 \to q_3, s$; $q_2, 2 \to q_2, mM$; $q_3, 1 \to q_0, n$; $q_3, 2 \to q_3, sS$. Рис. 123.

436. $q_0, 0 \to q_0, 0$; $q_0, 1 \to q_1, 1$; $q_0, \Lambda \to q_0, \varepsilon$; $q_1, 0 \to q_0, 1$; $q_1, 1 \to q_2, 0$; $q_1, \Lambda \to q_0, 1$; $q_2, 0 \to q_1, 0$; $q_2, 1 \to q_2, 1$; $q_2, \Lambda \to q_0, 01$.

437. $q_0, 0 \to q_0, 0$; $q_0, 1 \to q_1, 0$; $q_0, 2 \to q_2, 0$; $q_0, 3 \to q_0, 1$; $q_0, 4 \to q_1, 1$; $q_0, 5 \to q_2, 1$; $q_0, 6 \to q_0, 2$; $q_0, 7 \to q_1, 2$; $q_0, 8 \to q_2, 2$; $q_0, 9 \to q_0, 3$; $q_1, 0 \to q_1, 3$; $q_1, 1 \to q_2, 3$; $q_1, 2 \to q_0, 4$; $q_1, 3 \to q_1, 4$; $q_1, 4 \to q_2, 4$; $q_1, 5 \to q_0, 5$; $q_1, 6 \to q_1, 5$; $q_1, 7 \to q_2, 5$; $q_1, 8 \to q_0, 6$; $q_1, 9 \to q_1, 6$; $q_2, 0 \to q_2, 6$; $q_2, 1 \to q_0, 7$; $q_2, 2 \to q_1, 7$; $q_2, 3 \to q_2, 7$; $q_2, 4 \to q_0, 8$; $q_2, 5 \to q_1, 8$; $q_2, 6 \to q_2, 8$; $q_2, 7 \to q_0, 9$; $q_2, 8 \to q_1, 9$; $q_2, 9 \to q_2, 9$.

438. Если множество принимающих состояний F пусто, то распознаётся пустой язык. Если $F = Q$, то распознаётся язык Σ^*. Пусть $F = \{q_0\}$. Программа автомата состоит из команд $q_0, 0 \to q_a$; $q_0, 1 \to q_b$; $q_1, 0 \to q_c$; $q_1, 1 \to q_d$. Для всевозможных наборов $abcd$ получаем следующие распознаваемые языки: 0000, 0001, 0010, 0011 — Σ^*; 0100 — все, кроме слов, заканчивающихся на нечётное количество единиц; 0101 — все, кроме слов, заканчивающихся на 1; 0110 — слова с чётным количеством единиц; 0111 —

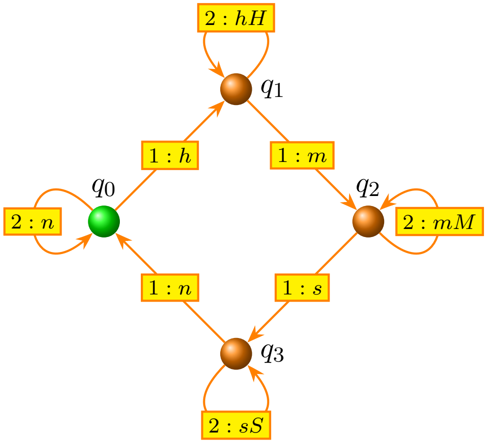


Рис. 123: Преобразователь из задачи 435.

слова, состоящие только из нулей; 1000 — все, кроме слов, заканчивающихся на нечётное количество нулей; 1001 — слова с чётным количеством нулей; 1010 — все, кроме слов, заканчивающихся на 0; 1011 — слова, состоящие только из единиц; 1100 — слова чётной длины; 1101 — слова, состоящие только из чётного числа нулей, и слова, содержащие единицы и заканчивающихся на нечётное количество нулей; 1110 — слова, состоящие только из чётного числа единиц, и слова, содержащие нули и заканчивающихся на нечётное количество единиц; 1111 — $\{\varepsilon\}$. Для $F = \{q_1\}$ языки будут дополнениями языков из предыдущего случая.

439. v и u, автомат распознаёт слова, которые начинаются с одной из букв, заканчиваются другой, и при этом вторая буква нигде не повторяется два или более раза подряд.

440. Только $\mathfrak{M}_4$.

441. Индукцией по длине прочитанного слова w доказать следующее утверждение: в состоянии q_3 автомат находится, если w оканчивается на aba; в состоянии q_2 автомат находится, если w оканчивается на ab; в состоянии q_1 автомат находится, если w оканчивается на a, но не на aba; в состоянии q_0 автомат находится во всех остальных случаях.

442. Пусть автомат распознаёт слово w, то есть переходит по его прочтении в состояние q_2. Из любого состояния по слову aa автомат попадёт в q_4 и не может вернуться в q_2, следовательно, w не содержит aa. Попасть в q_2 автомат может только из q_1 по букве a, а в состояние q_1 — только по букве

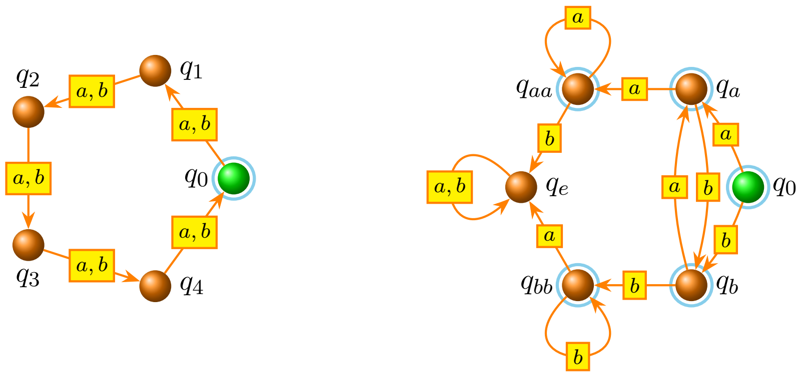


Рис. 124: Детерминированные автоматы из задачи 444, (а)–(б).

b. Следовательно, последними буквами в w являются ba, что и требуется. Наоборот, если w не содержит aa, то автомат не попадёт в состояние q_4. Из любого другого состояния по слову ba автомат попадёт в q_2, то есть примет слово.

443. Базис: $w = \varepsilon$, тогда $(q, wx) = (q, x)$. Поскольку входное слово не изменилось, то количество шагов равно нулю, следовательно, $p = q$, путь (q) несёт пустое слово. Наоборот: если путь несёт пустое слово, он не содержит ни одного ребра, следовательно, $p = q$. Предположим, что для всех слов, длина которых меньше w, утверждение доказано и $|w| > 0$. Тогда $w = w'a$. По индукционному предположению $(q, w'ax) \vdash^*_{\mathfrak{M}} (r, ax)$ тогда и только тогда, когда когда в $\mathfrak{D}_{\mathfrak{M}}$ есть путь из q в w, несущий w'. Если $(q, wx) \vdash^*_{\mathfrak{M}} (p, x)$, то $(r, ax) \vdash_{\mathfrak{M}} (p, x)$, что означает наличие команды $r, a \to p$ в программе $\mathfrak{M}$. Следовательно, в $\mathfrak{D}_{\mathfrak{M}}$ есть ребро (r, p), помеченное a. Объединяя путь из q в r, несущий w', с этим ребром, получим путь из q в p, несущий w. Наоборот, если в $\mathfrak{D}_{\mathfrak{M}}$ есть путь из q в p, несущий w, то есть путь из q в r, несущий w', и ребро (r, p), помеченное a. Последнее означает команду $q, a \to p$, поэтому $(r, ax) \vdash_{\mathfrak{M}} (p, x)$, и $(q, wx) \vdash^*_{\mathfrak{M}} (p, x)$.

444. **(а)–(б)** Рис. 124. **(в)–(г)** Рис. 125.

445. Рис. 126, $*$ означает любые варианты.

446. **(а)** Состояния: q_v, где v — слово длины 4; программа: $q_{zv}, x \to q_{vx}$, где $|v| = 3$; начальное состояние: q_{0000}; принимающие — состояния вида q_{1xyz}. Здесь $x, y, z \in \{0, 1\}$.

(б) Состояния: q_e и всевозможные q_v^w, где v — слово длины не больше 2, w — слово длины 2; программа: множество всевозможных команд $q_e, x \to q_e$,

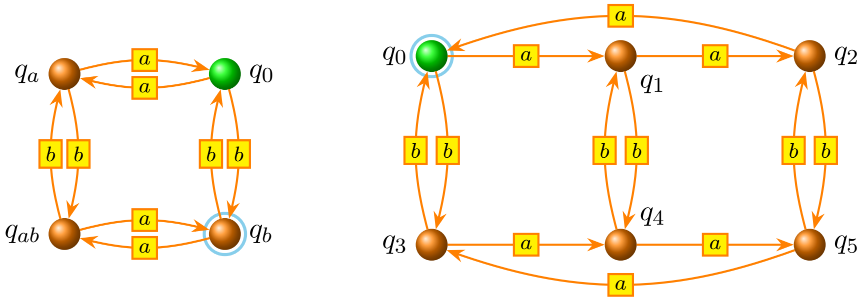


Рис. 125: Детерминированные автоматы из задачи 444, (в)–(г).

$q_v^w, x \to q_{vx}^w$, если $|v| < 2$, $q_{00}^{zu}, 1 \to q_e$, $q_{01}^{zu}, 0 \to q_{10}^{z1}$, $q_{11}^{zu}, 1 \to q_{11}^{1u}$ и $q_{xy}^{zu}, a \to q_{ya}^{zu}$ во всех других случаях; начальное состояние: q_ε^{00}; принимающие — состояния вида q_{xy}^{11}. Здесь $x, y, z, u, a \in \{0, 1\}$.

(в) Состояния: q_v, где v — слово длины не больше 2; программа: $q_\varepsilon, x \to q_x$, $q_x, y \to q_{xy}$, и $q_{zx}, y \to q_{xy}$; начальное состояние: q_ε; принимающие — состояния q_{10}, q_{11}, q_{01}. Здесь $x, y, z \in \{0, 1\}$.

447. Рис. 127.

448. **(а)** Рис. 128, сверху.

(б) Рис. 128, снизу. Множествами принимающих состояний будут $F_1 = \{q_{00}$, q_{01}, q_{02}, q_{03}, q_{10}, q_{11}, q_{12}, q_{20}, q_{21}, $q_{22}\}$, $F_2 = \{q_{00}, q_{01}, q_{02}\}$, $F_3 = \{q_{03}\}$.

449. $\mathfrak{M}_1$ и $\mathfrak{M}_3$. Автомат $\mathfrak{M}_2$ не распознаёт слово 00001.

450. **(а)** После первого этапа: $P_1 = \{q_0,\ a \to q_1;\ q_0,\ b \to q_2;\ q_1,\ b \to q_2;\ q_2,\ b \to q_2\}$; $F_1 = \{q_0, q_1, q_2\}$. Фактически автомат уже детерминированный, добавляем состояние $\varnothing$ для отсутствующих команд: $P_2 = \{q_0,\ a \to q_1;\ q_0,\ b \to q_2;\ q_1,\ a \to \varnothing;\ q_1,\ b \to q_2;\ q_2,\ a \to \varnothing;\ q_2,\ b \to q_2;\ \varnothing,\ a \to \varnothing;\ \varnothing,\ b \to \varnothing\}$; $F_2 = \{q_0, q_1, q_2\}$.

(б) После первого этапа: $P_1 = \{q_0,\ a \to q_1;\ q_0,\ a \to q_2;\ q_1,\ b \to q_0;\ q_1,\ b \to q_2;\ q_2,\ b \to q_0\}$; $F = \{q_1, q_2\}$. После детерминизации: $P_2 = \{q_0,\ a \to q_{12};\ q_0,\ b \to \varnothing;\ q_{12},\ a \to \varnothing;\ q_{12},\ b \to q_{02};\ q_{02},\ a \to q_{12};\ q_{02},\ b \to q_0;\ \varnothing,\ a \to \varnothing;\ \varnothing,\ b \to \varnothing\}$; $F = \{q_{12}, q_{02}\}$.

451. **(а)** w_2 и w_3. **(б)** Рис. 129.

(в) Непустые слова, которые не заканчиваются одиночной буквой b.

(г) Рис. 130.

452. **(а)** v, x, y, z.

(б) Имеются следующие пути из q_0 в q_5: $q_0 \xrightarrow{1} q_5$; $q_0 \xrightarrow{1} q_1 \xrightarrow{\varepsilon} q_2 \xrightarrow{0} q_5$;

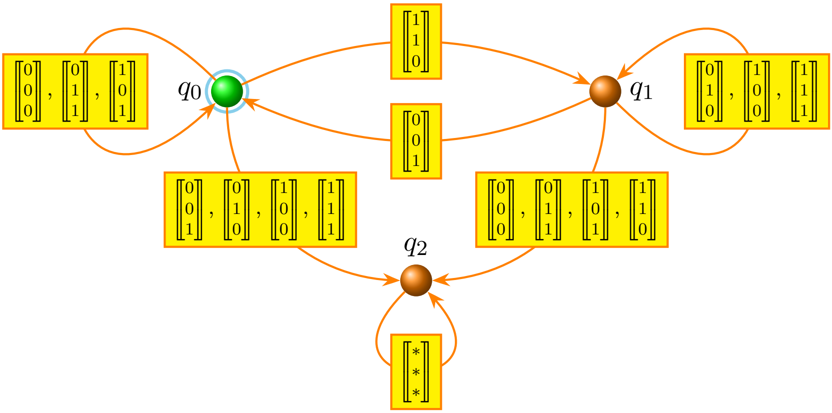


Рис. 126: Автомат из задачи 445.

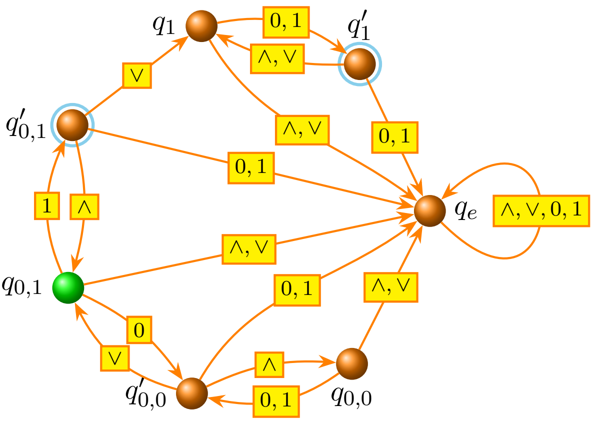


Рис. 127: Автомат из задачи 447.

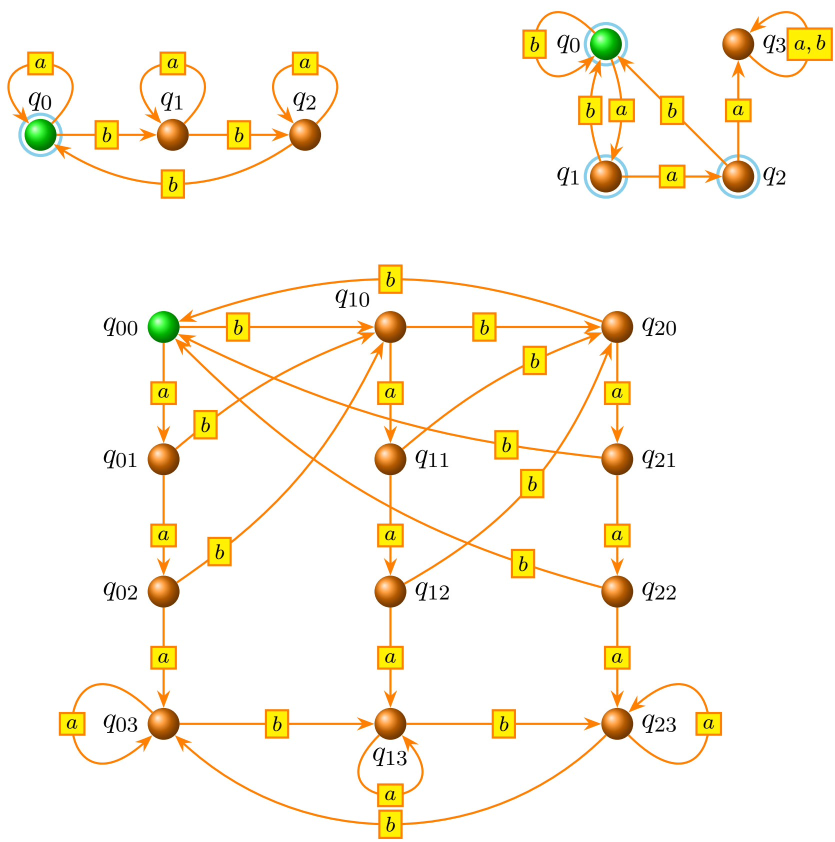


Рис. 128: Автоматы из задачи 448.

$q_0 \xrightarrow{1} q_1 \xrightarrow{\varepsilon} q_2 \xrightarrow{1} q_4 \xrightarrow{0} q_4 \xrightarrow{0} \ldots \xrightarrow{0} q_4 \xrightarrow{0} q_5$; $q_0 \xrightarrow{1} q_1 \xrightarrow{\varepsilon} q_2 \xrightarrow{1} q_4 \xrightarrow{0} q_4 \xrightarrow{0} \ldots \xrightarrow{0} q_4 \xrightarrow{\varepsilon} q_5$; $q_0 \xrightarrow{1} q_1 \xrightarrow{\varepsilon} q_3 \xrightarrow{0} q_3 \xrightarrow{0} \ldots \xrightarrow{0} q_3 \xrightarrow{\varepsilon} q_4 \xrightarrow{0} q_4 \xrightarrow{0} \ldots \xrightarrow{0} q_4 \xrightarrow{0} q_5$; $q_0 \xrightarrow{1} q_1 \xrightarrow{\varepsilon} q_3 \xrightarrow{0} q_3 \xrightarrow{0} \ldots \xrightarrow{0} q_3 \xrightarrow{\varepsilon} q_4 \xrightarrow{0} q_4 \xrightarrow{0} \ldots \xrightarrow{0} q_4 \xrightarrow{\varepsilon} q_5$. Все они несут слова из языка L. Обратно: для слов 10^i есть путь $q_0 \xrightarrow{1} q_1 \xrightarrow{\varepsilon} q_3 \xrightarrow{0} q_3 \xrightarrow{0} \ldots \xrightarrow{0} q_3 \xrightarrow{\varepsilon} q_4 \xrightarrow{\varepsilon} q_5$; для слов 110^i есть путь $q_0 \xrightarrow{1} q_1 \xrightarrow{\varepsilon} q_2 \xrightarrow{1} q_4 \xrightarrow{0} \ldots \xrightarrow{0} q_4 \xrightarrow{\varepsilon} q_5$.

(в) Рис. 131.

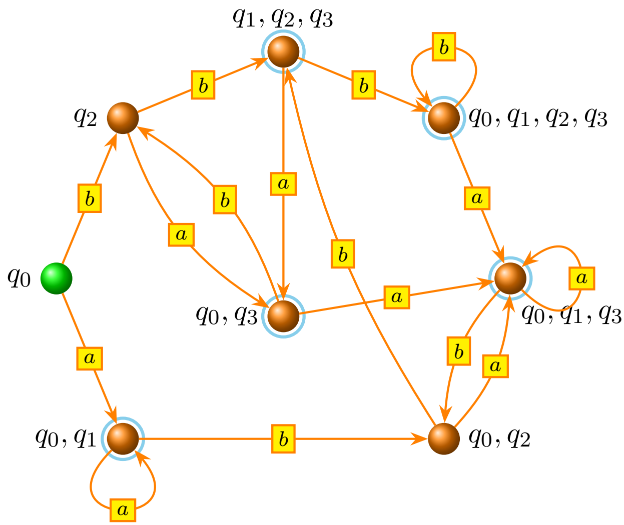


Рис. 129: Детерминированный автомат из задачи 451.

453. После первого этапа состояния p_1 и q_6 исчезнут, принимающими будут q_0, p_3, q_8. После детерминизации $\{q_0\}, 0 \to \{p_2, s_2, q_7, q_8\}$; $\{q_0\}, 1 \to \{p_3\}$; $\{p_3\}, 0 \to \{p_2, s_2, q_7, q_8\}$; $\{p_3\}, 1 \to \{p_3\}$; $\{p_2, s_2, q_7, q_8\}, 0 \to \{s_3, q_8\}$; $\{p_2, s_2, q_7, q_8\}, 1 \to \{p_3\}$; $\{s_3, q_8\}, 0 \to \varnothing$; $\{s_3, q_8\}, 1 \to \{p_3\}$; $\varnothing, 0 \to \varnothing$; $\varnothing, 1 \to \varnothing$; принимающие — все, кроме $\varnothing$.

454. (а) $\mathfrak{N} = (\{q_0, \ldots, q_k\}, \Sigma, P, q_0, \{q_1, \ldots, q_k\})$, где программа P состоит из команд $q_0, \varepsilon \to q_i$; $q_i, a_j \to q_i$ для всех $i, j = 1, \ldots, k$, $i \neq j$.

(б) Пусть детерминированный автомат $\mathfrak{M}$ с менее чем 2^k состояниями распознаёт язык L. Для каждого $U \subseteq \Sigma$ зафиксируем одно слово w_U, содержащее в точности буквы из U. Существует ровно 2^k слов w_U. Следовательно, автомат $\mathfrak{M}$ при работе на двух разных словах w_{U_1} и w_{U_2} перейдёт в одно и то же состояние q. Так как U_1 и U_2 различны, то существует символ из Σ, входящий в одно из этих множеств и не входящий в другое. Пусть, например, $a_i \in U_1$ и $a_i \notin U_2$. Обозначим $v = a_1 \ldots a_{i-1} a_{i+1} \ldots a_k$. Тогда $w_{U_1} v \notin L$, так как слово $w_{U_1} v$ содержит все символы из Σ, но $w_{U_2} v \in L$, так как слово $w_{U_2} v$ не содержит a_i. Но в силу детерминированности автомат $\mathfrak{M}$ из q по слову v должен перейти в какое-то одно состояние q_f. Если q_f — принимающее, то автомат примет $w_{U_1} v$, а если нет, то не примет $w_{U_2} v$. И то, и другое означает, что автомат $\mathfrak{M}$ не распознаёт язык L.

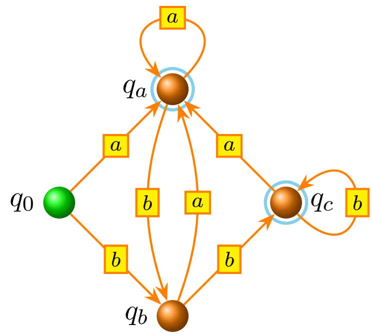


Рис. 130: Упрощённый вариант автомата из задачи 451.

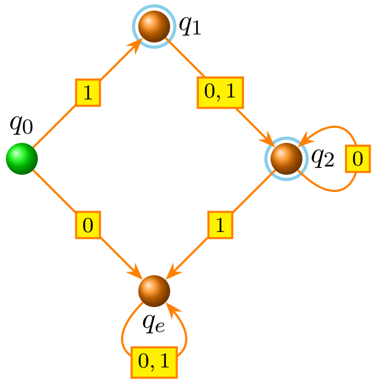


Рис. 131: Детерминированный автомат из задачи 452.

455. **(а)** Если $R = \varnothing$, то $w_R = bc$. Если $R = \{q_i\}$, то $w_R = a^i$. Иначе пусть $R = \{q_{i_0}, q_{i_1}, \ldots, q_{i_k}\}$, где $i_0 < i_1 < \cdots < i_k$. Тогда $w_R = c(ab)^{i_k - i_{k-1}} c(ab)^{i_{k-1} - i_{k-2}} c \ldots c(ab)^{i_2 - i_1} c(ab)^{i_1 - i_0 - 1} a^{i_0}$.

(б) Очевидно, что для любого $T \subseteq Q$ выполнено $(T, a^i) \vdash^*_{\mathfrak{M}} (T', \varepsilon)$, где $T' = \{q_{(i+j) \bmod n} : q_j \in T\}$. Если $q_i \in R \setminus S$, то по слову a^{n-i} автомат $\mathfrak{M}$ перейдёт из R в состояние R', содержащее q_0, а из состояния S — в S', которое q_0 не содержит.

(в) Допустим, что существует детерминированный автомат $\mathfrak{M}'$ эквивалентный $\mathfrak{N}$ (и, следовательно, $\mathfrak{M}$), который имеет меньше 2^n состояний. Тогда, согласно пункту (а), существуют $R_1, R_2 \subseteq Q$, $R_1 \neq R_2$, и слова w_1 и w_2 такие, что автомат $\mathfrak{M}$ переходит по ним из начального в состояния R_1 и

R_2 соответственно, а автомат $\mathfrak{M}'$ — в одно и то же состояние t. Согласно пункту (б) существует слово $w_{1,2}$, которое только одно из состояний R_1 или R_2 переводит в принимающее. Следовательно, одно из слов $w_1w_{1,2}$ или $w_2w_{1,2}$ автомат $\mathfrak{M}$ примет, а второе — нет. Но автомат $\mathfrak{M}'$ по слову $w_{1,2}$ может перейти из t только в одно состояние. Следовательно, автомат $\mathfrak{M}'$ оба слова $w_1w_{1,2}$ и $w_2w_{1,2}$ примет или нет одновременно. Значит, автомат $\mathfrak{M}'$ не эквивалентен $\mathfrak{M}$ и, следовательно, $\mathfrak{N}$.

Регулярные языки и конечные автоматы

456. **(а)** $\{a, ab, abb, aa, aba, abba, abbb, aab, abab, abbab, aba, abba, abbba\}$;
(б) $\{a, b, aa, ab, ba, aba, abb, aabb, abba, abbb, ababb, abbba, abbabb\}$;
(в) $\{\varepsilon, a, b, ab, aba, aa, ba, abaa, bb, abb, abab, aab, bab, abaab, bba, abba, ababa\}$.

457. Только (в).

458. (а) и (е).

459. (б), (г) и (д).

460. $\mathfrak{N}_1$ и $\mathfrak{N}_3$, $\mathfrak{N}_2$ отвергает слово 1.

461. $\mathfrak{N}_5$ и $\mathfrak{N}_6$, $\mathfrak{N}_4$ отвергает слово 0.

462. Если слово удовлетворяет регулярному выражению, то оно не может содержать трёх нулей подряд: после двух будет либо единица, либо конец слова. Если в слове нет трёх нулей подряд, то можно разбить его на фрагменты, каждый из которых состоит из нулей и заканчивается (кроме, может быть, последнего) единицей. Тогда каждый такой фрагмент имеет вид 1, 01 или 001, а последний — 0, 00 или пустой. Следовательно, такое слово описывается регулярным выражением $(1 + 01 + 001)^*(0 + 00 + \varepsilon)$.

463. **(а)** Слова, которые оканчиваются нулём, до него стоит сколько-то (может быть, ноль) единиц, а до них — сколько-то (может быть, ноль) нулей;
(б) слова, которые начинаются и оканчиваются одним нулём, а между ними стоит сколько-то (может быть, ноль) единиц;
(в) все слова, начинающиеся и оканчивающиеся нулями, в том числе и просто 0;
(г) слова с чётным количеством и нулей, и единиц.

464. **(а)** следует из коммутативности объединения множеств;
(б) следует из ассоциативности объединения множеств;
(в) следует из ассоциативности конкатенации слов;
(г) если $w \in L((r^*)^*)$, то $w = v_1 \ldots v_n$, где $v_i \in L(r^*)$. В свою очередь $v_i = u_{i1} \ldots u_{im_i}$, где $u_{ij} \in L(r)$. Тогда $w = u_{11} \ldots u_{nm_n}$, что и означает

$w \in L(r^*)$. Обратное включение непосредственно следует из $L \subseteq L^*$ для любого языка L;

(д) если $w \in L((r+p)q)$, то $w = uv$, где $u \in L(r+p)$, $v \in L(q)$. В свою очередь $u \in L(r+p)$ означает, что $u \in L(r)$ или $u \in L(p)$. В первом случае $w = uv \in L(r)L(q) = L(rq) \subseteq L(rq) \cup L(pq) = L(rq+pq)$, во втором — $w = uv \in L(p)L(q) = L(pq) \subseteq L(rq) \cup L(pq) = L(rq+pq)$. Обратно, если $w \in L(rq+pq)$, то $w \in L(rq)$ или $w \in L(pq)$. В первом случае получаем $w = uv$, где $u \in L(r)$, $v \in L(q)$. Тогда $u \in L(r) \cup L(p) = L(r+p)$. Следовательно, $w = uv \in L(r+p)L(q) = L((r+p)q)$. Аналогично во втором случае.

465. Будем для удобства с помощью p, q и r обозначать и произвольные слова, удовлетворяющие выражениям p, q и $p+q$ соответственно. Скобки в словах расставлены для удобства, сами они в слова не входят.

(а) Если $w \in L(p^*(p+q)^*)$, то

$$w = (p\ldots p)(r\ldots r) = (p\ldots p)(p\ldots pqp\ldots pq\ldots p\ldots qp\ldots p) = \\ = p\ldots pp\ldots p(qp\ldots p)(qp\ldots p)\ldots(qp\ldots p),$$

что удовлетворяет выражению $p^*(qp^*)^*$. Если $w \in L(p^*(qp^*)^*)$, то

$$w = (p\ldots p)(qp\ldots p)(qp\ldots p)\ldots(qp\ldots p) = \\ = p\ldots p(qp\ldots p)(qp\ldots p)\ldots(qp\ldots p),$$

следовательно, $w \in L((p+qp^*)^*)$. Если $w \in L((p+qp^*)^*)$, то $w \in L((p+q)^*)$, так как последний язык содержит все возможные комбинации p и q. Если $w \in L((p+q)^*)$, то $w \in L(p^*(p+q)^*)$, так как в качестве p^* можно взять пустое слово.

(б) Если $w \in L(p(qp)^*)$, то $w = p(qp)(qp)\ldots(qp) = (pq)(pq)\ldots(pq)p$, что соответствует $(pq)^*p$. Аналогично в обратную сторону.

(в) Так как итерация содержит пустое слово, то, используя (б), получаем $L((p^*q^*)^*) \subseteq L((p^*q^*)^*p^*) \subseteq L(p^*(q^*p^*)^*) \subseteq L(q^*p^*(q^*p^*)^*) \subseteq L((q^*p^*)^*)$. Аналогично в обратную сторону.

(г) $(pq)^+(q^*p^* + q^*) \equiv (pq)^*(pq)(q^*p^* + q^*) \equiv (pq)^*p(qq^*p^* + qq^*) \equiv (pq)^*p(q^+p^* + q^+) \equiv (pq)^*pq^+(p^* + \varepsilon) \equiv (pq)^*pq^+p^*$.

466. **(а)** $\varepsilon + 000^*$; **(б)** $(0+1)(00)^*$; **(в)** 0^*1; **(г)** 0^*01^*.

467. $(aa)^* + a(aa)^*(ba(aa)^*)^*b(aa)^* + a(aa)^* + a(aa)^*(ba(aa)^*)^*ba(aa)^* \equiv (aa)^*(aa)^* + (aa)^*a(ba(aa)^*)^*b(aa)^* + a(aa)^*(aa)^* + (aa)^*a(ba(aa)^*)^*ba(aa)^* \equiv (aa)^*(aa)^* + (aa)^*a(ba(aa)^*)^*b(aa)^* + (aa)^*a(aa)^* + (aa)^*a(ba(aa)^*)^*ba(aa)^* \equiv (aa)^*(\varepsilon + a(ba(aa)^*)^*b + a + a(ba(aa)^*)^*ba)(aa)^* \equiv (aa)^*(\varepsilon + a + a(ba(aa)^*)^*b(\varepsilon + a))(aa)^* \equiv (aa)^*((\varepsilon + a(ba(aa)^*)^*b)(\varepsilon + a))(aa)^* \equiv (aa)^*((\varepsilon +$

$a(b(aa)^*a)^*b)(\varepsilon+a))(aa)^* \equiv (aa)^*(\varepsilon+(ab(aa)^*)^*ab)(\varepsilon+a)(aa)^* \equiv (aa)^*(\varepsilon+(ab(aa)^*)^*ab)((aa)^*+a(aa)^*) \equiv (aa)^*(\varepsilon+(ab(aa)^*)^*ab)((aa)^*+(aa)^*a) \equiv (aa)^*(\varepsilon+(ab(aa)^*)^*ab)(aa)^*(\varepsilon+a) \equiv (aa)^*((aa)^*+(ab(aa)^*)^*ab(aa)^*)(\varepsilon+a) \equiv (aa)^*((aa)^*+(ab(aa)^*)^+)(\varepsilon+a) \equiv ((aa)^*(aa)^*+(aa)^*(ab(aa)^*)^+)(\varepsilon+a) \equiv ((aa)^*+(aa)^*(ab(aa)^*)^+)(\varepsilon+a) \equiv (aa)^*(\varepsilon+(ab(aa)^*)^+)(\varepsilon+a) \equiv (aa)^*(ab(aa)^*)^*(\varepsilon+a) \equiv (aa+ab)^*(\varepsilon+a)$.

468. Индукция по построению r. Базисные случаи 1)–3) не содержат $+$ изначально. Пусть для r' и r'' уже построены эквивалентные $s' = s'_1+\dots+s'_n$ и $s'' = s''_1+\dots+s''_m$, соответственно. Тогда для $r'+r''$ нужным будет $s'+s''$. Для $r'r''$ получаем $r'r'' \equiv s's'' \equiv (s'_1+\dots+s'_n)(s''_1+\dots+s''_m) \equiv s'_1s''_1+\dots+s'_ns''_m$. Для $(r')^*$ имеем $(r')^* \equiv (s')^* \equiv (s'_1+\dots+s'_n)^* \equiv ((s'_1)^*\cdots(s'_n)^*)^*$.

469. **(а)** Разобьём слово на двухсимвольные фрагменты и последний символ, тогда количество 00 и 11 может быть произвольным, а количество 01 и 10 чётно, если последний символ 0, и нечётно, если последний символ 1. В первом случае получаем $(00+11+(01+10)(00+11)^*(01+10))^*0$, во втором — $(00+11+(01+10)(00+11)^*(01+10))^*(01+10)(00+11)^*1$. Общее выражение: $(00+11+(01+10)(00+11)^*(01+10))^*(0+(01+10)(00+11)^*1)$;

(б) $(0+1)^*(001+110)(0+1)^*$;

(в) $(0+1)^*00(0+1)^*$;

(г) 1 может стоять после 0 только в самом конце слова: $1^*0^*(\varepsilon+1)$.

470. Индукция по n: для языка $\{w^{m_1}\}$ длина регулярного выражения r_1, построенного простой конкатенацией всех букв, равна $4km_1-3 = 4km_1+4\cdot 1-7$ (km_1 букв и по km_1-1 амперсандов, открывающих и закрывающих скобок). Пусть для языка $\{w^{m_2-m_1},\dots,w^{m_n-m_1}\}$ построено регулярное выражение r' длины $4k(m_n-m_1)+4(n-1)-7$. Тогда для языка $\{w^{m_1},\dots,w^{m_n}\}$ таким выражением будет $r = (r_1 \,\&\, (\varepsilon+r'))$. Длина r равна $4km_1-3+4k(m_n-m_1)+4(n-1)-7+7 = 4km_n+4n-7$.

471. $$\left(\begin{bmatrix}0\\0\\0\end{bmatrix}+\begin{bmatrix}0\\1\\1\end{bmatrix}+\begin{bmatrix}1\\0\\1\end{bmatrix}+\begin{bmatrix}1\\1\\0\end{bmatrix}\left(\begin{bmatrix}1\\1\\1\end{bmatrix}+\begin{bmatrix}0\\1\\0\end{bmatrix}+\begin{bmatrix}1\\0\\0\end{bmatrix}\right)^*\begin{bmatrix}0\\0\\1\end{bmatrix}\right)^*.$$

472. $((0+1)(\wedge(0+1))^*\vee)^*(1(\wedge 1)^*)(\vee(0+1)(\wedge(0+1))^*)^*$.

473. Рис. 132–137.

474. **(а)** $r = aabb$; **(б)** $r = (abb)^*$;

(в) $r = (ab+bb)^*((aa)^*+b^*)$;

(г) $r = (b^*+(ab)^*)((aa)^*+bb^*)$.

475. **(а)** $r = (a+b(\varepsilon+a(ba)^*a^+)b^+a)(ab^*a)^*(\varepsilon+b(a+b)^*)$;

(б) $r = a(ba)^*(\varepsilon+(a+bb)(a+b)^*)+b((\varepsilon+b)ab)^*(\varepsilon+b+bb(a+b)^*)$;

(в) $r = \varnothing$;

(г) $r = (ab)^*+(ba)^*b+(a+b)^*(a+aa(ba)^*b)$.

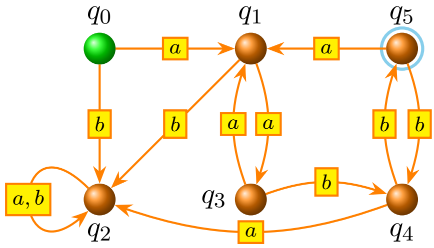


Рис. 132: Автомат из задачи 473, (а).

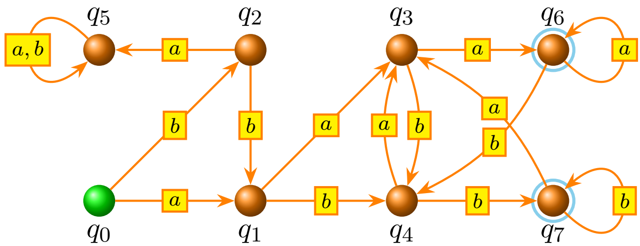


Рис. 133: Автомат из задачи 473, (б).

476. (а) $r = a + b$; **(б)** $r = a + b$; **(в)** $r = b(\varepsilon + a)$; **(г)** $r = b(aa)^*$.

477. (а) $(aa + ba + bb)^*(ab + aab + bab)$;
(б) $(b + bb + aab)(aab + abab)^*a$;
(в) $(aba + b + a + aaa)(bba + baba + bb + ba + baaa)^*$;
(г) $((bba + bb + a)(ba)^*b)^*(bb + (bba + bb + a)(ba)^*(b + ba))$.

478. $(c + (a + b + c)(b + (a(b + c)^*)^{n-2}a))^*$.

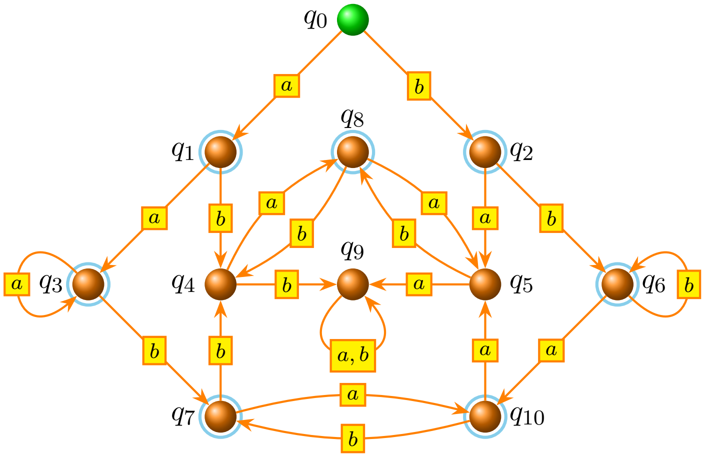


Рис. 134: Автомат из задачи 473, (в).

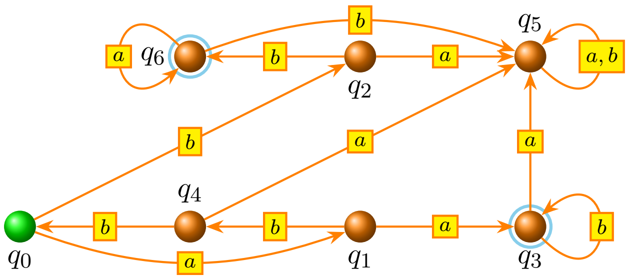


Рис. 135: Автомат из задачи 473, (г).

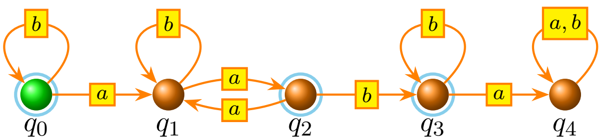


Рис. 136: Автомат из задачи 473, (д).

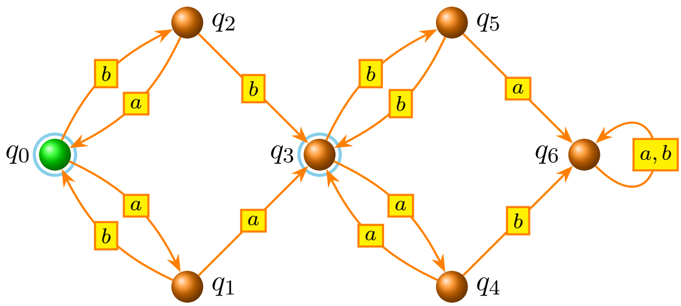


Рис. 137: Автомат из задачи 473, (е).

Кодирования

479. **(а)** Способы разбиения: $ab|aa|ba|c$; множество кодов: $\{22021\}$.
(б) Способы разбиения: $ba|c|b|ba|c|c|ab$; множество кодов: $\{2112122\}$.
(в) Способы разбиения: $aa|b|c|ab|c|c|aa|b$, $aa|b|c|abc|c|aa|b$; множество кодов: $\{012201, 011101\}$.
(г) Способы разбиения: $abc|bb|aa|b|c$, $abc|b|b|aa|b|c$, $abc|b|ba|abc$, $abc|b|ba|ab|c$, $ab|c|bb|aa|b|c$, $ab|c|b|b|aa|b|c$, $ab|c|b|ba|abc$, $ab|c|b|ba|ab|c$; множество кодов: $\{11201, 111101, 1112111, 1112122, 22201, 221101, 2212111, 2212122\}$.

480. Индукция по n. При $n = 0$ имеем $0^n = \varepsilon$ и $C[\varepsilon] = \{\varepsilon\}$ — один способ, $1 = F_{0+1}$. При $n = 1$ имеем $0^1 = 0$ и $C[0] = \{a\}$ — один способ, $1 = F_{1+1}$. Если для всех $k < n$ утверждение доказано и $n \geqslant 2$, то слово 0^n можно закодировать так: $C[0^n] = C[0]C[0^{n-1}] \cup C[00]C[0^{n-2}] = aC[0^{n-1}] \cup bC[0^{n-2}]$. По индукционному предположению имеем $|C[0^{n-1}]| = F_n$, $|C[0^{n-2}]| = F_{n-1}$. Тогда $|C[0^n]| = F_n + F_{n-1} = F_{n+1}$, что и требуется.

481. Только (б).

482. Только (в).

483. Только (г).

484. **(а)** $(a+b)^*$; **(б)** $(b+ab^*aaa)^*$; **(в)** $(aa+aaba)^*$; **(г)** $(aa)^*+a(aa)^*b(a+b)^*$.

485. **(а)** $(a + ab + bb)^*$; **(б)** $(a + ab)^*$; **(в)** $(abab + bb)^*$; **(г)** $(bbaba^*bba)^*$.

486. $\mathfrak{M}_1$ и $\mathfrak{M}_2$, автомат $\mathfrak{M}_3$ принимает пустое слово.

487. **(а)** Рис. 138.
(б) Рис. 139, состояния q_1, q_2 и связанные с ними недостижимы из начального состояния q_0, поэтому не изображены.

488. Рис. 140.

489. Рис. 141.

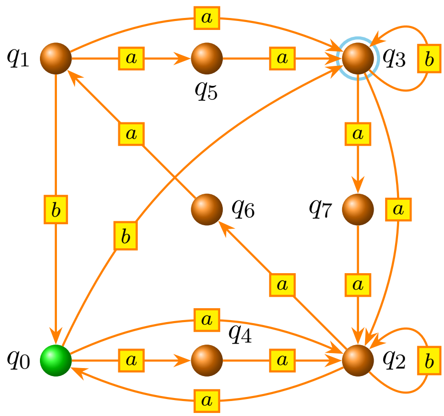


Рис. 138: Автомат из задачи 487, (а).

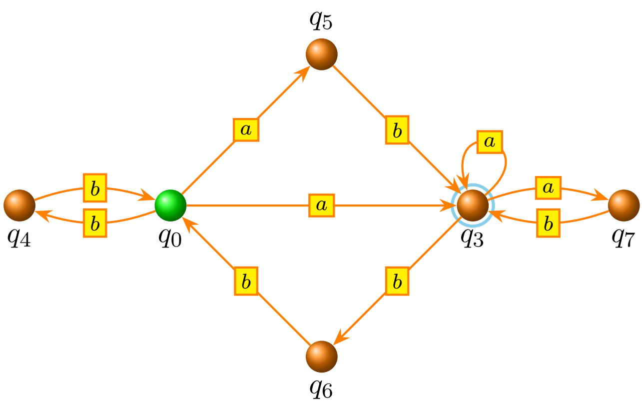


Рис. 139: Автомат из задачи 487, (б).

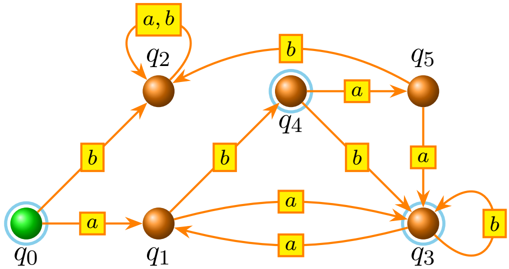


Рис. 140: Автомат из задачи 488.

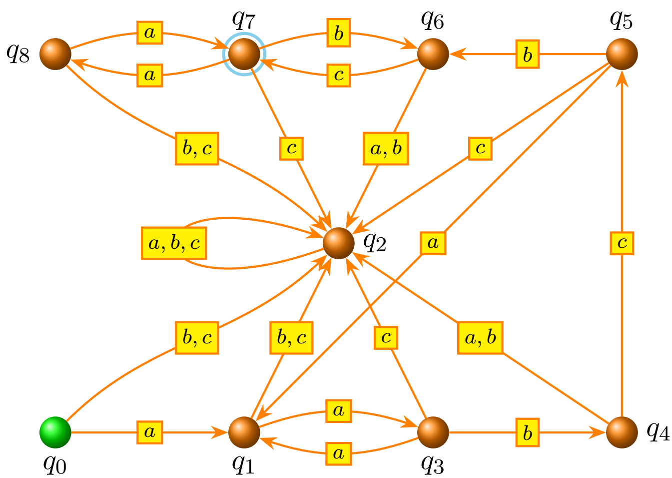


Рис. 141: Автомат из задачи 489.

490. Рис. 142.
491. Рис. 143.
492. Рис. 144.
493. Рис. 145.
494. Пусть схема C состоит из пар $c_i = (u_i, v_i)$, $u_i \in \Sigma^*$, $v_i \in \Omega^*$, $i = 1, \dots, n$. Возьмём алфавит $\Xi = \{c_1, \dots, c_n\}$ и определим $\psi(c_i) = u_i$, $\varphi(c_i) = v_i$. Получаем $w \in C(L)$ тогда и только тогда, когда $w = v_{i_1} \dots v_{i_k}$ для $u_{i_1} \dots u_{i_k} \in L$

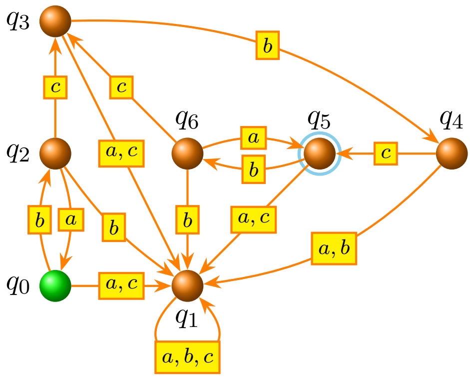


Рис. 142: Автомат из задачи 490.

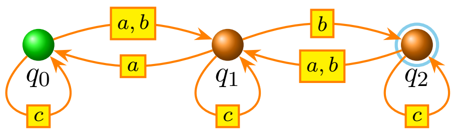


Рис. 143: Автомат из задачи 491.

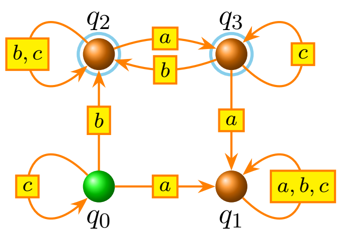


Рис. 144: Автомат из задачи 492.

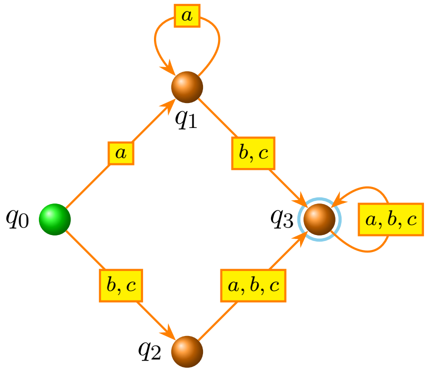


Рис. 145: Автомат из задачи 493.

тогда и только тогда, когда $w = \varphi(c_{i_1} \dots c_{i_k})$ для $\psi(c_{i_1} \dots c_{i_k}) \in L$ тогда и только тогда, когда $w \in \varphi(\psi^{-1}(L))$.

495. Новый автомат получается добавлением команд вида $q, a \to q$ для всех состояний и всех $a \in \Sigma \setminus \Omega$. Доказать автоматность нового языка можно также с помощью схемы кодирования $D = C^{-1} = \{(a, a) : a \in \Omega\} \cup \{(\varepsilon, a) : a \in \Sigma \setminus \Omega\}$, тогда $Z_\Sigma(L) = D(L)$.

496. Пусть h — наибольшая длина кодового слова. Рассмотрим полное дерево с корнем r высоты h, в котором все исходящие из любой вершины (кроме листьев) рёбра помечены в точности элементами Σ. Если для кодового слова $C(a)$ путь из корня, несущий слово $C(a) = ub$, ведёт в вершину v, то единственное входящее в v ребро с меткой b заменяем ребром с меткой $b : a$, ведущим в r. В силу беспрефиксности такое преобразование не может быть проделано для вершин, лежащих одна под другой. Если после этого останутся достижимые из корня листья v, то единственное входящее в v ребро с меткой b заменяем ребром с той же меткой, ведущим в r. Исключив не достижимые из r вершины, получим диаграмму преобразователя.

497. Графы кодирования изображены на рис. 146. Тривиальные петли не показаны.

498. Графы кодирования изображены на рис. 147. Тривиальные петли не показаны. Для второй схемы — можно, для первой — нельзя: слово aa можно получить из слов 00 и 11.

499. Рис. 148. Точка обозначена нулём, тире — единицей. Тривиальные петли не показаны.

500. Рис. 149. Тривиальные петли не показаны. Например, слово 01100100

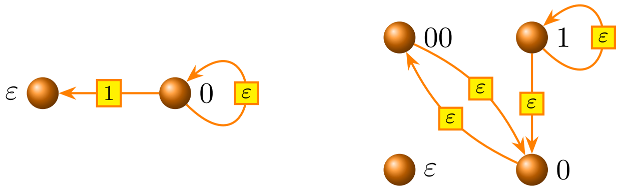


Рис. 146: Графы кодирования из задачи 497.

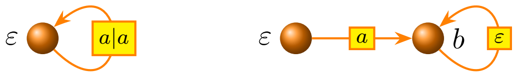


Рис. 147: Графы кодирования из задачи 498.

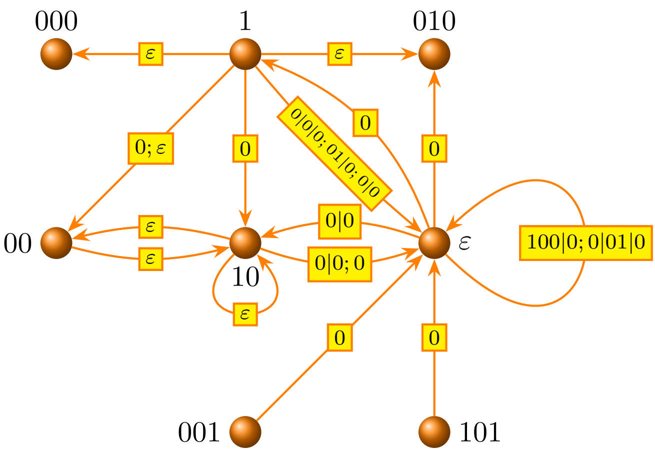


Рис. 148: Граф кодирования из задачи 499.

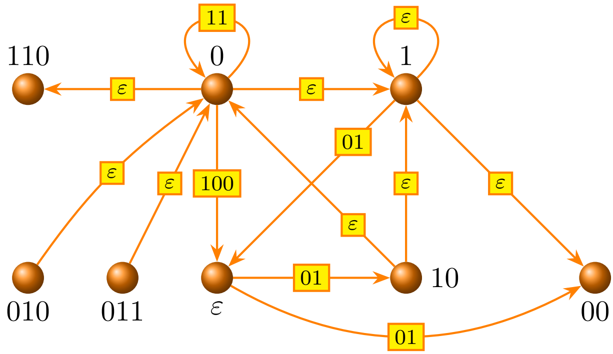


Рис. 149: Граф кодирования из задачи 500.

можно декодировать abb и ce.

501. Рис. 150. Тривиальные петли не показаны. Например, слово 00100010100 можно декодировать $fbaf$ и ddb.

502. Чтобы из вершины ε исходило ребро, не являющееся тривиальной петлёй, нужно, чтобы одно кодовое слово было префиксом другого.

503. **(а)** $C(\text{К}) = 01$; $C(\text{А}) = 00$; $C(\text{Р}) = 110$; $C(\text{Т}) = 111$; $C(\text{И}) = 100$; $C(\text{Ц}) = 101$; длина кода 24, при равномерном кодировании для каждого символа нужно три знака: $2^2 < 6 \leqslant 2^3$, длина будет равна 30, что на 6 символов длиннее.

(б) $C(\text{П}) = 01$; $C(\text{А}) = 000$; $C(\text{Р}) = 001$; $C(\text{Л}) = 11$; $C(\text{Е}) = 101$; $C(\text{И}) = 1001$; $C(\text{Д}) = 1000$; длина кода 38, при равномерном кодировании для каждого символа нужно три знака $2^2 < 7 \leqslant 2^3$, длина будет равна 42, что на 4 символа длиннее.

(в) $C(\text{Т}) = 01$; $C(\text{Е}) = 100$; $C(\text{Л}) = 1110$; $C(\text{А}) = 00$; $C(\text{П}) = 101$; $C(\text{Р}) = 110$; $C(\text{У}) = 1111$; длина кода 38, при равномерном кодировании для каждого символа нужно три знака $2^2 < 7 \leqslant 2^3$, длина будет равна 42, что на 4 символа длиннее.

(г) $C(\text{И}) = 11$; $C(\text{Н}) = 101$; $C(\text{Д}) = 100$; $C(\text{В}) = 0010$; $C(\text{У}) = 0011$; $C(\text{А}) = 0000$; $C(\text{Л}) = 0001$; $C(\text{Ь}) = 011$; $C(\text{О}) = 0100$; $C(\text{С}) = 01010$; $C(\text{Т}) = 01011$; длина кода 54, при равномерном кодировании для каждого символа нужно четыре знака $2^3 < 11 \leqslant 2^4$, длина будет равна 64, что на 10 символов длиннее.

(д) $C(\text{П}) = 001$; $C(\text{Е}) = 01$; $C(\text{Р}) = 11$; $C(\text{А}) = 0000$; $C(\text{С}) = 0001$; $C(\text{Д}) = 1000$; $C(\text{Л}) = 1001$; $C(\text{Н}) = 1011$; $C(\text{И}) = 1010$; длина кода 48, при равномерном кодировании для каждого символа нужно четыре знака

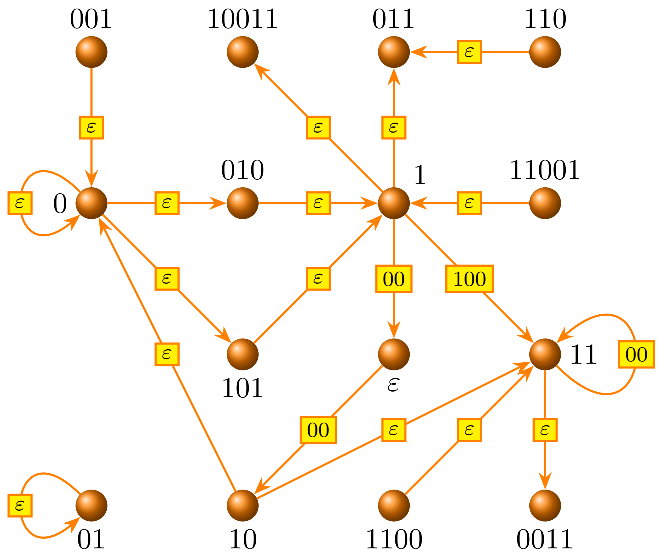


Рис. 150: Граф кодирования из задачи 501.

$2^3 < 9 \leqslant 2^4$, длина будет равна 68, что на 20 символов длиннее.
(е) $C(\text{О}) = 1$; $C(\text{Б}) = 0101$; $C(\text{Р}) = 01000$; $C(\text{Н}) = 001$; $C(\text{С}) = 011$; $C(\text{П}) = 01001$; $C(\text{Т}) = 0000$; $C(\text{Ь}) = 0001$; длина кода 48, при равномерном кодировании для каждого символа нужно четыре знака $2^2 < 8 \leqslant 2^3$, длина будет равна 54, что на 6 символов длиннее.
(ж) $C(\text{С}) = 010$; $C(\text{Т}) = 000$; $C(\text{Р}) = 011$; $C(\text{О}) = 001$; $C(\text{Н}) = 100$; $C(\text{Г}) = 1010$; $C(\text{И}) = 1011$; $C(\text{Л}) = 1100$; $C(\text{Ц}) = 1101$; $C(\text{Е}) = 1110$; $C(\text{У}) = 1111$; длина кода 60, при равномерном кодировании для каждого символа нужно четыре знака $2^3 < 11 \leqslant 2^4$, длина будет равна 72, что на 12 символов длиннее.
(з) $C(\text{Т}) = 00$; $C(\text{А}) = 01$; $C(\text{Р}) = 100$; $C(\text{О}) = 101$; $C(\text{Б}) = 110$; $C(\text{Л}) = 111$; длина кода 31, при равномерном кодировании для каждого символа нужно четыре знака $2^2 < 6 \leqslant 2^3$, длина будет равна 39, что на 8 символов длиннее. Оптимальный код не единственен, но длина минимального кода постоянна.
504. (а) $q_0, 0 \to q^0, \varepsilon$; $q_0, 1 \to q^1, \varepsilon$; $q_0, \Lambda \to q_0, \varepsilon$; $q^0, 0 \to q_0, \text{А}$; $q^0, 1 \to q_0, \text{К}$; $q^0, \Lambda \to q_0, \varepsilon$; $q^1, 0 \to q^{10}, \varepsilon$; $q^1, 1 \to q^{11}, \varepsilon$; $q^1, \Lambda \to q_0, \varepsilon$; $q^{10}, 0 \to q_0, \text{И}$; $q^{10}, 1 \to q_0, \text{Ц}$; $q^{10}, \Lambda \to q_0, \varepsilon$; $q^{11}, 0 \to q_0, \text{Р}$; $q^{11}, 1 \to q_0, \text{Т}$; $q^{11}, \Lambda \to q_0, \varepsilon$;

(б) $q_0, 0 \to q^0, \varepsilon$; $q_0, 1 \to q^1, \varepsilon$; $q_0, \Lambda \to q_0, \varepsilon$; $q^0, 0 \to q^{00}, \varepsilon$; $q^0, 1 \to q_0$, П; $q^0, \Lambda \to q_0, \varepsilon$; $q^1, 0 \to q^{10}, \varepsilon$; $q^1, 1 \to q_0$, Л; $q^1, \Lambda \to q_0, \varepsilon$; $q^{00}, 0 \to q_0$, А; $q^{00}, 1 \to q_0$, Р; $q^{00}, \Lambda \to q_0, \varepsilon$; $q^{10}, 0 \to q^{100}, \varepsilon$; $q^{10}, 1 \to q_0$, Е; $q^{10}, \Lambda \to q_0, \varepsilon$; $q^{100}, 0 \to q_0$, Д; $q^{100}, 1 \to q_0$, И; $q^{100}, \Lambda \to q_0, \varepsilon$.
Для остальных пунктов аналогично.

505. Базис: две буквы, два кодовых слова длины один, $2^{-1} + 2^{-1} = 1$. Индукционный шаг: допустим, что для k букв доказано. Для $k+1$ буквы кодовые слова будут получены заменой одного кодового слова w длины ℓ на два кодовых слова $w0$ и $w1$ длины $\ell + 1$ каждое. Сумма при этом не изменится: $2^{-\ell} = 2^{-(\ell+1)} + 2^{-(\ell+1)}$.

506. Если неравенство Крафта-МакМиллана строгое и ℓ — наибольшая длина кодового слова, то разность между единицей и суммой будет не меньше $2^{-\ell}$, так как наименьший общий знаменатель суммируемых дробей равен 2^ℓ. Следовательно, хотя бы одно кодовое слово длины ℓ можно заменить словом длины $\ell - 1$, а исходный гомоморфизм оптимальным не будет. Если выходной алфавит содержит $k > 2$ символов, то для любого гомоморфизма из двухбуквенного алфавита мы получим $k^{-\ell_1} + k^{-\ell_2} \leqslant {}^2/_k < 1$.

507. **(а)** Пусть C — оптимальный гомоморфизм, $C(a_i) = w$ — самое короткое из кодовых слов, $C(a_j) = v$ — самое длинное. Если $|v| > |w| + 2$, то построим новый гомоморфизм, в котором $D(a_i) = w0$, $D(a_j) = w1$ и $D(a) = C(a)$ для всех остальных символов a. Так как частоты a_i и a_j одинаковы и равны какому-то c, то $c|w| + c|v| = c(|w|+1) + c(|v|-1) > 2c(|w|+1)$, поэтому C не мог быть оптимальным.

(б) Используем индукцию по количеству пар (w, v) кодовых слов вида $C(a_i)$, для которых выполнено $|v| = |w| + 2$. Базис: таких пар нет, тогда условие уже выполнено. Индукционный шаг. Пусть C — оптимальный двоичный гомоморфизм, $C(a_i) = w$, $C(a_j) = v$, $|v| = |w| + 2$. Построим новый гомоморфизм ψ как в пункте (а). Так как частоты a_i и a_j одинаковы и равны какому-то c, то $c|w| + c|v| = c(|w|+1) + c(|v|-1) = 2c(|w|+1)$. Следовательно, новый гомоморфизм тоже оптимальный. Теперь количество пар стало меньше и необходимый гомоморфизм существует по индукционному предположению.

(в) При построении гомоморфизма D в пункте (а) сумма в неравенстве Крафта-МакМиллана строго возрастает: $2^{-|w|} + 2^{-|v|} = 2 \cdot 2^{-(|w|+1)} + 2^{-|v|} < 2 \cdot 2^{-(|w|+1)}$. Следовательно, гомоморфизм C неоптимальный (задача 506).

508. Из неравенства Крафта-МакМиллана получаем $2^{-\ell_g} + 2^{-\ell_h} \leqslant 1 - 2^{-5} - 2^{-1} - 2^{-3} - 2^{-3} - 2^{-4} - 2^{-8} = {}^{39}/_{256}$. Функция $L(\ell_g, \ell_h) = {}^{\ell_g}/_8 + {}^{\ell_h}/_5$ должна иметь минимальное значение. Так как $\ell_g \geqslant -\log_2(\frac{39}{256} - 2^{-\ell_h})$, то $L(\ell_g, \ell_h) \geqslant M(\ell_h) = (-\log_2(\frac{39}{256} - 2^{-\ell_h}))/8 + {}^{\ell_h}/5$. Производная равна $M'(\ell_h) = \dfrac{1}{5} -$

$\dfrac{32}{39 \cdot 2^{\ell_h} - 256}$, она монотонно возрастает, поэтому обращается в ноль не более чем в одной точке. Получаем $M'(3) < 0$, $M'(4) > 0$, следовательно, оптимальное значение ℓ_h лежит на отрезке $[3; 4]$. Для $\ell_h = 3$ получаем $\ell_g = 6$, для $\ell_h = 4$ получаем $\ell_g = 4$. В первом случае $L(\ell_g, \ell_h) = {}^{27}/{20}$, во втором $L(\ell_g, \ell_h) = {}^{13}/{10} < {}^{27}/{20}$. Таким образом, оптимально $\ell_h = 4$, $\ell_g = 4$.

509. Из задачи 507 получаем, что длины кодовых слов $C(a)$, $C(b)$, $C(c)$ отличаются не более чем на единицу. Следовательно, эти кодовые слова должны иметь длину 2 или более, в противном случае получим $2^{-1} + 2^{-2} + 2^{-2} + \cdots > 1$. Таким образом, в качестве первых трёх кодовых слов можно взять 00, 01, 10. Оставшиеся пять кодовых слов должны начинаться с 11 и тоже должны иметь длины, отличающиеся не более чем на единицу: 1100, 1101, 1110, 11110, 11111.

510. Пусть $C = \{(u_i, v_i) : i = 1, \ldots, n\}$. Тогда $C[x] \neq \varnothing$ тогда и только тогда, когда $x \in L((u_1 + \cdots + u_n)^*)$. Следовательно, $C^*(L) = C(L \cap L((u_1 + \cdots + u_n)^*)) \cup (L \setminus L((u_1 + \cdots + u_n)^*))$. Все языки в правой части равенства являются автоматными, следовательно, $C^*(L)$ тоже автоматный.

Замкнутость класса автоматных языков. Неавтоматные языки

511. Новый автомат получается из старого $\mathfrak{M} = (Q, \Sigma, P, q_0, F)$ обращением всех рёбер в противоположную сторону, добавлением нового начального состояния q_0', команд $q_0', \varepsilon \to p$ для всех $p \in F$. Единственным принимающим состоянием будет q_0.

512. Пусть детерминированный конечный автомат $\mathfrak{M}$ распознаёт язык L.

(а) Объявить в новом автомате принимающими все состояния, из которых были достижимы принимающие состояния старого автомата.

(б) Создать новое начальное состояние q_0' и добавить команды $q_0', \varepsilon \to p$ для всех p, достижимых из старого начального состояния.

(в) Сначала применить (а), потом (б).

(г) Оставить принимающими только те состояния, из которых принимающие не являются строго достижимыми (то есть из них за 1 или более шагов нельзя попасть в принимающее).

(д) Удалить все команды вида $q, a \to p$ для принимающих q.

(е) Взять две копии автомата: $\mathfrak{M}^1$ и $\mathfrak{M}^2$. Начальное состояние взять из $\mathfrak{M}^1$, принимающие — из $\mathfrak{M}^2$. Вставить всевозможные команды вида $q_1, \varepsilon \to p_2$, если в $\mathfrak{M}$ есть путь из q в p.

(ж) Для каждого состояния q автомата $\mathfrak{M}$ создать автомат $\mathfrak{M}_q$ из двух копий автомата $\mathfrak{M}$: $\mathfrak{M}_q^1$ и $\mathfrak{M}_q^2$, соединив принимающие состояния $\mathfrak{M}_q^1$ с начальным состоянием $\mathfrak{M}_q^2$ с помощью пустых переходов. Весь автомат содержит новое начальное состояние q_0, из которого ведут пустые переходы в состояния q автоматов $\mathfrak{M}_q^1$. Принимающими являются состояния q из $\mathfrak{M}_q^2$.

513. С автоматом $\mathfrak{M}$, распознающем язык L сделать такое преобразование. Вместо каждой команды $c = q, a_i \to p$ создать уникальную копию $\mathfrak{M}_i^c = (Q_i^c, \Delta, P_i^c, q_{0i}^c, F_i^c)$ автомата $\mathfrak{M}_i$ и добавить команды $q, \varepsilon \to q_{0i}^c$ и $q_f, \varepsilon \to p$ для всех $q_f \in F_i^c$.

514. Изменить детерминированный конечный автомат $\mathfrak{M}$, распознающий L_1, следующим образом. Новыми принимающими состояниями будут такие q, что $\mathfrak{M}$ переходит из q в некоторое старое принимающее состояние, прочитав какое-то слово из L_2.

515. Пусть языки L_1 и L_2 распознаются автоматами $\mathfrak{M}_1 = (Q_1, \Sigma, P_1, q_1^0, F_1)$ и $\mathfrak{M}_2 = (Q_2, \Sigma, P_2, q_2^0, F_2)$ соответственно. Состояния нового автомата $\mathfrak{N}$ — пары (q_1, q_2), где $q_1 \in Q_1$, $q_2 \in Q_2$. Если $\mathfrak{M}_1$ и $\mathfrak{M}_2$ содержали команды $q_1, x \to p_1$ и $q_2, x \to p_2$, то $\mathfrak{N}$ содержит $(q_1, q''), x \to (p_1, q'')$ и $(q', q_2), x \to (q', p_2)$ для всех $q' \in Q_1$ и $q'' \in Q_2$. Начальным в $\mathfrak{N}$ является состояние (q_1^0, q_2^0). Принимающими в $\mathfrak{N}$ являются состояния (q_1, q_2), где $q_1 \in F_1$ и $q_2 \in F_2$.

516. (б), (г), (д).

517. (б), (д).

518. (б), (в), (е), (з).

519. Предположим, что язык L автоматный, тогда возьмём для него константу k из леммы о разрастании.

(а) Возьмём слово $w = a^{k+3}b^k \in L$, разобьём на три части: $\alpha = a^{k+3}$, $u = b^k$, $\beta = \varepsilon$. Тогда существует разбиение $b^k = xyz$, для которого $1 \leqslant |y| \leqslant k$. Поскольку x, y и z состоят только из букв b, то $x = b^p$, $y = b^q$, $z = b^{k-p-q}$ для каких-то p и q, причём $1 \leqslant q \leqslant k$. Согласно лемме о разрастании $w_0 = \alpha x y^0 z \beta = a^{k+3}b^p b^{k-p-q}\varepsilon = a^{k+3}b^{k-q} \in L$. Тогда должно выполняться $k + 3 = k - q + 3$, откуда $q = 0$, что противоречит $q \geqslant 1$. Противоречие доказывает, что язык неавтоматный.

(б) Возьмём слово $w = a^k c b^{3k+1} \in L$, разобьём на три части: $\alpha = \varepsilon$, $u = a^k$, $\beta = cb^{3k+1}$. Тогда существует разбиение $a^k = xyz$, для которого $1 \leqslant |y| \leqslant k$. Поскольку x, y и z состоят только из букв a, то $x = a^p$, $y = a^q$, $z = a^{k-p-q}$ для каких-то p и q, причём $1 \leqslant q \leqslant k$. Согласно лемме о разрастании $w_2 = \alpha x y^2 z \beta = \varepsilon a^p a^{2q} a^{k-p-q} c b^{3k+1} = a^{k+q} c b^{3k+1} \in L$. Тогда должно выполняться $3k + 1 > 3(k + q) = 3k + 3q$, откуда $3q < 1$, что противоречит $q \geqslant 1$. Противоречие доказывает, что язык неавтоматный.

(в) Возьмём слово $w = a^2 b^k acab^k a^2 \in L$, разобьём на три части: $\alpha = a^2$, $u = b^k$, $\beta = acab^k a^2$. Тогда существует разбиение $b^k = xyz$, для которого

$1 \leqslant |y| \leqslant k$. Поскольку x, y и z состоят только из букв b, то $x = b^p$, $y = b^q$, $z = b^{k-p-q}$ для каких-то p и q, причём $1 \leqslant q \leqslant k$. Согласно лемме о разрастании $w_0 = \alpha x y^0 z \beta = a^2 b^p b^{k-p-q} acab^k a^2 = a^2 b^{k-q} acab^k a^2 \in L$. Тогда должно выполняться $(a^2 b^{k-q} a)^{-1} = ab^k a^2$, что неверно, так как количество букв b в словах различается. Противоречие доказывает, что язык неавтоматный.

(г) Возьмём слово $w = a^{2^k} \in L$, разобьём на три части: $\alpha = \varepsilon$, $u = a^{2^k}$, $\beta = \varepsilon$. Тогда существует разбиение $a^{2^k} = xyz$, для которого $1 \leqslant |y| \leqslant k$. Поскольку x, y и z состоят только из букв a, то $x = a^p$, $y = a^q$, $z = a^{2^k - p - q}$ для каких-то p и q, причём $1 \leqslant q \leqslant k$. Согласно лемме о разрастании $w_2 = \alpha x y^2 z \beta = \varepsilon a^p a^{2q} a^{2^k - p - q} \varepsilon = a^{2^k + q} \in L$. Так как $q \geqslant 1$, то $2^k + q > 2^k$. С другой стороны, так как $a^{2^k + q} \in L$, то $2^k + q$ должно быть степенью двойки. Наименьшая степень двойки, превосходящая 2^k, — это 2^{k+1}. Следовательно, $2^k + q \geqslant 2^{k+1}$, то есть $q \geqslant 2^k > k$, что противоречит $q \leqslant k$. Противоречие доказывает, что язык неавтоматный.

(д) Возьмём слово $w = a^k c^k \in L$, разобьём на три части: $\alpha = \varepsilon$, $u = a^k$, $\beta = c^k$. Тогда существует разбиение $a^k = xyz$, для которого $1 \leqslant |y| \leqslant k$. Поскольку x, y и z состоят только из букв a, то $x = a^p$, $y = a^q$, $z = a^{k-p-q}$ для каких-то p и q, причём $1 \leqslant q \leqslant k$. Согласно лемме о разрастании $w_2 = \alpha x y^2 z \beta = \varepsilon a^p a^{2q} a^{k-p-q} c^k = a^{k+q} c^k \in L$. Это означает, что $k + q = k$ и $q = 0$, что противоречит $1 \leqslant q$. Противоречие доказывает, что язык неавтоматный.

520. L_0 будет, остальные — нет.

(а) Рис. 151.

(б) Допустим, язык L_1 автоматный, n — константа из леммы о разрастании. ДНФ $x_n \vee \neg x_n$ тождественно истинна, поэтому $a^n \vee \neg a^n \in L_1$. Разобьём слово: $\alpha = \varepsilon$, $u = a^n$, $\beta = \vee \neg a^n$. По лемме о разрастании существует разбиение $u = xyz$ и все слова $w_i = \alpha x y^i z \beta$ принадлежат L_1. Так как u состоит только из букв a, то $x = a^p$, $y = a^q$, $z = a^{n-p-q}$ для каких-то p и q, где $1 \leqslant q \leqslant n$. Тогда $w_i = a^{n+(i-1)q} \vee \neg a^n$. В частности, $w_0 = a^{n-q} \vee \neg a^n \notin L_1$, так как ДНФ $x_{n-q} \vee \neg x_n$ не является тождественно истинной.

(в) Аналогично для ДНФ $x_n \wedge \neg x_n$.

(г) Допустим, язык L_3 автоматный, n — константа из леммы о разрастании. ДНФ $x_1 \wedge \dots \wedge x_n \wedge \neg x_{n+1}$ выполнима, поэтому $a \wedge \dots \wedge a^n \wedge \neg a^{n+1} \in L_3$. Разобьём слово: $\alpha = a \wedge \dots \wedge a^n \wedge \neg$, $u = a^{n+1}$, $\beta = \varepsilon$. По лемме о разрастании существует разбиение $u = xyz$ и все слова $w_i = \alpha x y^i z \beta$ принадлежат L_3. Так как u состоит только из букв a, то $x = a^p$, $y = a^q$, $z = a^{n+1-p-q}$ для каких-то p и q, где $1 \leqslant q \leqslant n$. Тогда $w_i = a \wedge \dots \wedge a^n \wedge \neg a^{n+1+(i-1)q}$. В частности, $w_0 = a \wedge \dots \wedge a^n \wedge \neg a^{n+1-q} \notin L_3$, так как элементарная конъюнкция содержит пару x_{n+1-q} и $\neg x_{n+1-q}$ (из неравенства $1 \leqslant q \leqslant n$ следует $1 \leqslant n+1-q \leqslant n$), поэтому невыполнима.

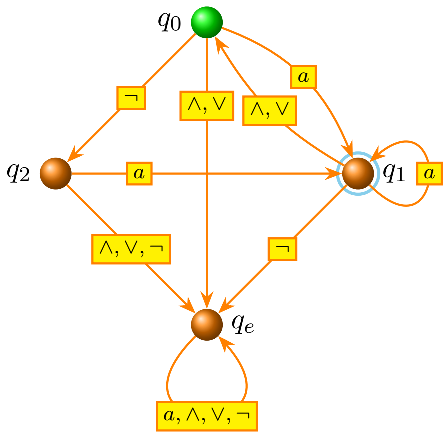


Рис. 151: Автомат из задачи 520, (а).

521. Из определения легко по индукции получается, что в λ-выражении количества открывающих и закрывающих скобок совпадают, так как они всегда добавляются парами. Допустим, что множество λ-выражений является автоматным языком, пусть тогда k — константа из леммы о разрастании для него. Возьмём такое λ-выражение: $(x(x(\ldots(xx)\ldots)))$, в котором k открывающих и k закрывающих скобок. Разобьём его на три части: $\alpha = (x(x(\ldots(xx$, $u =)^k$, $\beta = \varepsilon$. Далее как в задаче 519, пункт (д).

522. Допустим, что такой язык автоматный, а k — константа из леммы о разрастании для него. Рассмотрим такое слово: $\left[\!\!\left[\begin{matrix}0\\0\\0\end{matrix}\right]\!\!\right]^{k+1} \left[\!\!\left[\begin{matrix}1\\1\\0\end{matrix}\right]\!\!\right] \left[\!\!\left[\begin{matrix}0\\0\\0\end{matrix}\right]\!\!\right]^{k} \left[\!\!\left[\begin{matrix}0\\0\\1\end{matrix}\right]\!\!\right] \in L$, здесь в двоичном виде записано произведение $2^{k+1} \cdot 2^{k+1} = 2^{2k+2}$. Разобьём его на следующие части: $\alpha = \left[\!\!\left[\begin{matrix}0\\0\\0\end{matrix}\right]\!\!\right]^{k+1} \left[\!\!\left[\begin{matrix}1\\1\\0\end{matrix}\right]\!\!\right]$, $u = \left[\!\!\left[\begin{matrix}0\\0\\0\end{matrix}\right]\!\!\right]^{k}$, $\beta = \left[\!\!\left[\begin{matrix}0\\0\\1\end{matrix}\right]\!\!\right]$. Согласно лемме о разрастании u может быть представлено в виде xyz, где $x = \left[\!\!\left[\begin{matrix}0\\0\\0\end{matrix}\right]\!\!\right]^{p}$, $y = \left[\!\!\left[\begin{matrix}0\\0\\0\end{matrix}\right]\!\!\right]^{q}$, $z = \left[\!\!\left[\begin{matrix}0\\0\\0\end{matrix}\right]\!\!\right]^{k-p-q}$, причём $1 \leqslant q \leqslant k$. По лемме получаем $w_0 = \alpha x z \beta = \left[\!\!\left[\begin{matrix}0\\0\\0\end{matrix}\right]\!\!\right]^{k+1} \left[\!\!\left[\begin{matrix}1\\1\\0\end{matrix}\right]\!\!\right] \left[\!\!\left[\begin{matrix}0\\0\\0\end{matrix}\right]\!\!\right]^{p} \left[\!\!\left[\begin{matrix}0\\0\\0\end{matrix}\right]\!\!\right]^{k-p-q} \left[\!\!\left[\begin{matrix}0\\0\\1\end{matrix}\right]\!\!\right] = \left[\!\!\left[\begin{matrix}0\\0\\0\end{matrix}\right]\!\!\right]^{k+1} \left[\!\!\left[\begin{matrix}1\\1\\0\end{matrix}\right]\!\!\right] \left[\!\!\left[\begin{matrix}0\\0\\0\end{matrix}\right]\!\!\right]^{k-q} \left[\!\!\left[\begin{matrix}0\\0\\1\end{matrix}\right]\!\!\right] \in L$. Но теперь записано произведение $2^{k+1} \cdot 2^{k+1} = 2^{2k+2-q} \neq 2^{2k+2}$, то есть ошибочный результат. Тогда слово не должно принадлежать L. Противоречие показывает, что предположение неверно, язык неавтоматный.

523. Так как язык, содержащий все корректные строки, является автоматным, то для него существует константа n из леммы о разрастании. Предположим, что указанной константы k не существует: для любого k найдётся набор аргументов, для которого будет выполнено $f(x_1, \ldots, x_m) > k \max\{x_1, \ldots, x_m\}$. Возьмём $k = 2^{n+1}$. Тогда двоичная запись результата будет как минимум на $n+1$ символ длиннее, чем двоичная запись аргументов:

$$w = \underbrace{\begin{bmatrix} a_{1,1} \\ \vdots \\ a_{1,m} \\ b_1 \end{bmatrix} \ldots \begin{bmatrix} a_{t,1} \\ \vdots \\ a_{t,m} \\ b_t \end{bmatrix}}_{\alpha} \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ b_{t+1} \end{bmatrix} \ldots \begin{bmatrix} 0 \\ \vdots \\ 0 \\ b_{t+s} \end{bmatrix}}_{u} \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}}_{\beta},$$

где $s \geqslant n$. Разобьём слово w, как указано. Тогда $|u| \geqslant n$, поэтому применима лемма о разрастании: $u = xyz$ и все слова $w_i = \alpha x y^i z \beta$ корректны. Но слова w_i описывают значения функции f на одних и тех же аргументах: к верхним m этажам добавляются только старшие нули, которые не влияют на записанное число. В то же время числа, записанные на нижнем этаже в словах w_i, попарно различны — в каждом из них $t+s+1+(i-1)q$ разрядов соответственно, где $q > 0$ — длина слова y. Такого быть не может, так как функция на одном наборе аргументов принимает одно значение. Значит, предположение неверно, константа k существует.

524. Так как язык, содержащий все корректные строки, является автоматным, то для него существует константа n из леммы о разрастании. Предположим, что указанной константы k не существует: для любого $k > 0$ найдётся x и $f(x) < kx$. Возьмём $k = 2^{-(n+1)}$. Тогда двоичная запись результата будет как минимум на $n+1$ символ короче, чем двоичная запись аргумента:

$$w = \underbrace{\begin{bmatrix} a_1 \\ b_1 \end{bmatrix} \ldots \begin{bmatrix} a_t \\ b_t \end{bmatrix}}_{\alpha} \underbrace{\begin{bmatrix} a_{t+1} \\ 0 \end{bmatrix} \ldots \begin{bmatrix} a_{t+s} \\ 0 \end{bmatrix}}_{u} \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\beta},$$

где $s \geqslant n$. Разобьём слово w, как указано. Тогда $|u| \geqslant n$, поэтому применима лемма о разрастании: $u = xyz$ и все слова $w_i = \alpha x y^i z \beta$ корректны. Но слова w_i описывают значения функции f на сколь угодно больших аргументах: в каждом из них $t+s+1+(i-1)q$ разрядов соответственно, где $q > 0$ — длина слова y. В то же время значение функции, записанное в каждом из слов w_i, будет одним и тем же: нижний этаж только дополняется незначащими нулями. Следовательно, существует неограниченно

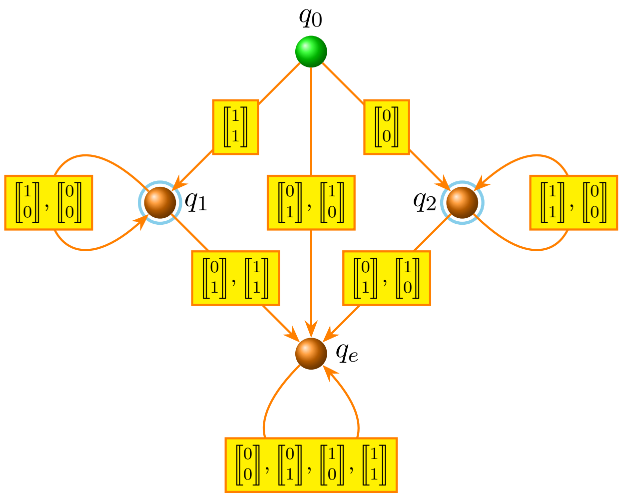


Рис. 152: Автомат из задачи 525.

возрастающая последовательность аргументов x_i таких, что $f(x_i) = y$. Но функция f монотонно не убывает, поэтому для любого $x \in [x_i; x_{i+1}]$ получаем $y = f(x_i) \leqslant f(x) \leqslant f(x_{i+1}) = y$, откуда $f(x) = y$ для всех достаточно больших x. Следовательно, функция f ограничена.

525. Пусть $f(x) = x$ для чётных x и $f(x) = 1$ для нечётных. Эта функция неограничена. Для неё нельзя найти константу $k > 0$, чтобы было выполнено $f(x) \geqslant kx$ для всех x, поскольку для нечётных x получим $k \leqslant \frac{f(x)}{x} = \frac{1}{x}$. Легко построить автомат, проверяющий корректность f (рис. 152).

526. Если бы язык L был автоматным, то и его дополнение $\bar{L}$ тоже было бы автоматным языком. Неавтоматность $\bar{L}$ доказывается так же как в задаче 519 пункт (д), достаточно взять слово $a^k b^k \in \bar{L}$, где k — константа из леммы о разрастании.

527. (а) $L_1 = L(a^2 b^+ a^2)$.

(б) Допустим, L_2 является автоматным. Тогда существует константа n из леммы о разрастании. Пусть $w = a^2 b^n a^4 b^n a^2 \in L_2$, $\alpha = a^2$, $u = b^n$, $\beta = a^4 b^n a^2$. По лемме $u = xyz$, где $1 \leqslant m \leqslant n$, $|y| = m$. Тогда $w_0 = a^2 b^{n-m} a^4 b^n a^2 \notin L_2$, что противоречит лемме.

(в) $L_3 = L(a^2 b^+ a^2 b^2 a^+ b^2)$.

(г) $L_4 = L((abab)^+)$.

(д) Допустим, L_5 является автоматным. Тогда существует константа n из леммы о разрастании. Пусть $w = (ab)^{n+1}ba(ab)^{n+1} \in L_5$, $\alpha = \varepsilon$, $u = (ab)^n$, $\beta = abba(ab)^{n+1}$. По лемме $u = xyz$, где $1 \leqslant m \leqslant n$, $|y| = m$. Если m нечётно, то в слове w_0 будут более двух блоков одинаковых букв подряд, поэтому $w_0 \notin L_5$. Если m чётно, то $w_0 = (ab)^{n+1-m/2}ba(ab)^{n+1} \notin L_5$, что противоречит лемме.

(е) $L_6 = L(a^*)$.

(ж) $L_7 = L(a^*)$.

(з) Допустим, L_8 является автоматным. Тогда существует константа n из леммы о разрастании. Пусть $w = a^{(n+1)^2+(n+1)} \in L_8$, $\alpha = \varepsilon$, $u = a^n$, $\beta = a^{(n+1)^2+1}$. По лемме $u = xyz$, где $1 \leqslant m \leqslant n$, $|y| = m$. Тогда $w_0 = a^{(n+1)^2+(n+1)-m}$. Получаем $(n+1)^2+(n+1)-m < (n+1)^2+(n+1)$, но в то же время $(n+1)^2+(n+1)-m = n^2+n+(2n+2-m) > n^2+n$. Следовательно, не существует натурального k, для которого $k^2+k = (n+1)^2+(n+1)-m$ и $w_0 \notin L_8$, что противоречит лемме.

528. Пусть $C = \{(a,\varepsilon),(b,aa),(b,bb)\}$. Рассмотрим слово $x = a^{2n}b^{2m}$. Тогда $C[x]$ состоит из слов длины $4m$, поэтому $x \in C[x]$ только при $n = m$. Это условие же является и достаточным. Если бы язык L был автоматным, то автоматным был бы и язык $L \cap L((aa)^*(bb)^*)$. Но последний язык, состоящий из всех слов вида $a^{2n}b^{2n}$, автоматным не является.

529. Нет. Если $L = (ab)^*$, то пересечением $\mathrm{COMM}(L)$ с автоматным языком a^*b^* будет язык из слов вида a^nb^n, который автоматным не является. Следовательно, язык $\mathrm{COMM}(L)$ неавтоматный.

530. Пусть $u_1,\dots,u_k$ — слова короче n из языка L. Для всякого $m = 0,\dots,n!-1$ выберем $v_m = a^{t_m} \in L$ такое слово (если они вообще есть), что t_m — наименьшее, для которого $t_m \geqslant n$ и $t_m \bmod n! = m$. Тогда L задаётся регулярным выражением $e = (u_1+\dots+u_k)+(v_{m_1}+\dots+v_{m_s})(a^{n!})^*$. Докажем, что $L = L(e)$. Если $w \in L$ и его длина меньше n, то w совпадает с одним из u_i. Иначе пусть $m = |w| \bmod n!$. Тогда v_m существует, $|v_m| \leqslant |w|$ и $|w|-|v_m| = cn!$ для некоторого натурального c. Получаем $w = a^{|w|} = a^{|v_m|}a^{|w|-|v_m|} = a^{|v_m|}a^{cn!} \in L(e)$. Пусть теперь $w \in L(e)$. Если w совпадает с одним из u_i, то $w \in L$ по определению u_i. Иначе w имеет вид $v_m a^{cn!}$ для некоторых m и c. Так как $|v_m| \geqslant n$, то к слову v_m можно применить лемму о разрастании, получим, что все слова вида $v_{m,i} = a^{t_m+(i-1)q}$ принадлежат L для некоторого $q = 1,\dots,n$ и всех натуральных i. Следовательно, $w = a^{t_m+cn!} = a^{t_m+\frac{cn!}{q}\cdot q} = v_{m,\frac{cn!}{q}+1} \in L$. Нужно только заметить: из $q \leqslant n$ вытекает, что $n!$ делится на q.

531. В указанных двух случаях конечный автомат для проверки строится тривиально.

(а) $q_0, \left[\!\!\left[\begin{matrix} a \\ C(a)\end{matrix}\right]\!\!\right] \to q_0$; $q_0, \left[\!\!\left[\begin{matrix} a \\ b\end{matrix}\right]\!\!\right] \to q_e$; $q_e, \left[\!\!\left[\begin{matrix} a \\ c\end{matrix}\right]\!\!\right] \to q_e$ для любых $a \in \Sigma$, $b, c \in \Omega$,

$b \neq C(a)$.

(б) $q_0, \begin{bmatrix} a \\ \Lambda \end{bmatrix} \to q_0$; $q_0, \begin{bmatrix} a \\ b \end{bmatrix} \to q_e$; $q_e, \begin{bmatrix} a \\ c \end{bmatrix} \to q_e$ для любых $a \in \Sigma$, $b \in \Omega$, $c \in \Omega \cup \{\Lambda\}$. Для обоих автоматов принимающее состояние — q_0. Пусть C не удовлетворяет ни одному из указанных условий. Докажем от противного, что соответствующий язык L не будет автоматным. Допустим, что язык L автоматный и n — константа из леммы о разрастании для него. Сначала рассмотрим случай $C(a) = v$ для некоторого символа $a \in \Sigma$, где $|v| \geqslant 2$. Тогда $C(a^k) = v^k$ для любых k. Рассмотрим слово

$$w = \underbrace{\begin{bmatrix} a \\ b_1 \end{bmatrix} \dots \begin{bmatrix} a \\ b_n \end{bmatrix}}_{\alpha} \underbrace{\begin{bmatrix} \Lambda \\ c_1 \end{bmatrix} \dots \begin{bmatrix} \Lambda \\ c_m \end{bmatrix}}_{u},$$

соответствующее условию $C(a^n) = v^n$. Здесь $v^k = b_1 \dots b_n c_1 \dots c_m$, $m \geqslant n$. Так как $|u| \geqslant n$, то существует разбиение $u = xyz$ и все слова $w_i = \alpha x y^i z$ тоже принадлежат L. Но это невозможно, так как верхнее слово a^n остаётся неизменным (оно целиком находится внутри α), а его гомоморфный образ внизу меняется. Теперь рассмотрим случай $C(a) = c$, $C(b) = \varepsilon$ для некоторых символов $a, b \in \Sigma$, $c \in \Omega$. Тогда $C(b^k a^m) = c^m$ для любых k, m. Рассмотрим слово

$$w = \underbrace{\begin{bmatrix} b \\ c \end{bmatrix}^n}_{\alpha} \underbrace{\begin{bmatrix} a \\ \Lambda \end{bmatrix}^n}_{u},$$

соответствующее условию $C(b^n a^n) = c^n$. Аналогично рассуждая, получим, что язык L будет содержать все слова вида

$$w = \begin{bmatrix} b \\ c \end{bmatrix}^n \begin{bmatrix} a \\ \Lambda \end{bmatrix}^{n+(i-1)q},$$

где $q = |y|$. Но это невозможно, так как количество a сверху не совпадает с количеством c снизу при $i \neq 1$.

532. Пусть $\Sigma = \{a\}$ и $L_n = \{a^n\}$ состоит из одного слова. Тогда распознающий L_n НКА имеет программу из команд вида $q_i, a \to q_{i+1}$, $i = 0, \dots, n-1$, то есть имеет $n+1$ состояние. Принимающим в нём будет только q_n. Если бы существовал автомат с n состояниями, то при работе на a^n он два раза побывал бы в одном и том же состоянии, поэтому фрагмент между этими состояниями можно было бы неограниченно интерировать и получить бесконечно много слов, которые автомат тоже принимал бы. Для ДКА нужно будет одно лишнее состояние после q_n.

533. (а) Если язык в односимвольном алфавите содержит больше чем n слов, то в нём есть слово длины n или больше. Как и в предыдущей задаче, если автомат с n состояниями принимает слово длины n или больше, то он принимает бесконечно много слов.

(б) В алфавите из k символов существует $1 + k + k^2 + \cdots + k^{n-1} = \dfrac{k^n - 1}{k - 1}$ слов длины меньше n. Поэтому если слов в языке больше, то среди них есть слово длины n или больше. Дальнейшие рассуждения аналогичны пункту (а).

534. Пусть n — количество состояний преобразователя $\mathfrak{M}$, m — наибольшая длина слова w в его командах $q, a \to p, w$. Тогда при чтении фрагмента u слова w преобразователь как минимум два раза побывает в одном и том же состоянии. Разделим фрагмент u соответствующим образом: $(q_0, \alpha xyz\beta, \varepsilon) \vdash^* (q', xyz\beta, X') \vdash^* (q, yz\beta, X'X'') \vdash^* (q, z\beta, X'X''Y) \vdash^* (q'', \beta, X'X''YZ') \vdash^* (q_f, \varepsilon, X'X''YZ'Z'')$. Тогда $|X'| \leqslant m|\alpha|$, $|X''| \leqslant m|x|$, $|Y| \leqslant m|y|$, $|Z'| \leqslant m|z|$, $|Z''| \leqslant m|\beta|$. Таким образом, в силу детерминированности получаем $(q, y^i z\beta, X'X'') \vdash^* (q, z\beta, X'X''Y^i)$ для любого i, поэтому $f(\alpha xy^i z\beta) = XY^iZ$ для $X = X'X''$, $Z = Z'Z''$. Кроме того $|X| \leqslant m|\alpha x|$ и $|Z| \leqslant m|z\beta|$.

535. Согласно задаче 534 длины аргумента и результата функции растут в арифметической прогрессии. Если длина двоичной записи растёт в арифметической прогрессии (добавляется k двоичных разрядов), то длина унарной записи будут расти в геометрической (длиннее в 2^k раз). Поэтому такие функции конечными преобразователями не вычисляются.

536. Допустим, что такой преобразоватль $\mathfrak{M}$ существует и n и m — константы из задачи 534. Рассмотрим $\mathfrak{M}(a^n b a^{nm}) = a^{nm} b a^n$. Пусть $\alpha = \varepsilon$, $u = a^n$, $\beta = ba^{nm}$, тогда $x = a^p$, $y = a^q$, $z = a^{n-p-q}$ для каких-то p, q, $1 \leqslant q \leqslant n$. Следовательно, $a^{nm} b a^n = XYZ$ и $|XY| \leqslant m(p+q) \leqslant mn$. Поэтому $X = a^P$, $Y = a^Q$ и получаем $\mathfrak{M}(a^{n-q} b a^{nm}) = a^{nm-Q} b a^n$, что невозможно.

537. Пусть C — гомоморфизм, заданный так: $C(a) = 1$, $C(b) = 10$, $C(c) = 00$. Граф кодирования изображён на рис. 153, гомоморфизм разнозначен. Тогда $C^{-1}(10^k 1) = ac^{k/2}a$ для чётных k, $C^{-1}(10^k 1) = bc^{(k-1)/2}a$ для нечётных. Допустим, что кодирование C^{-1} выполняется некоторым конечным преобразователем $\mathfrak{M}$ с n состояниями. Тогда при чтении фрагмента 0^n слова $10^n 0^m 1$ преобразователь как минимум два раза побывает в одном и том же состоянии: $(q_0, 10^{n+m}1, \varepsilon) \vdash^* (q, 0^{k+\ell+m}1, X) \vdash^* (q, 0^{\ell+m}1, XY) \vdash^* (q_f, \varepsilon, XYZ)$ для некоторых $k > 0$ и ℓ. Если $Y = \varepsilon$, то для любого натурального i мы получим $(q, 0^{ik+\ell+m}1, X) \vdash^* (q, 0^{\ell+m}1, X)$. Таким образом, $C^{-1}(10^{n+m+ik}1) = XZ$ для любого натурального i, что невозможно. Если $Y \neq \varepsilon$, то XY содержит

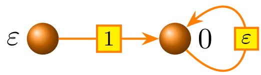


Рис. 153: Граф кодирования из задачи 537.

первую букву результата (a или b), которая от m не зависит в силу детерминированности. Но для $m = 0$ и $m = 1$ эта буква должна быть разной. Снова получили противоречие. Следовательно, конечный преобразователь $\mathfrak{M}$ существовать не может.

Программная вычислимость

538. (г), (д), (е).
539. $\Pi(\sigma) = \{(x, 6); (y, 6); (z, 7)\}$.
540. $\Pi(\sigma) = \{(x, 4); (y, 5); (z, 4); (u, 1); (v, 0)\}$.
541. $\Pi(\sigma) = \{(x, 6); (y, 5); (z, 0); (u, 7); (v, 6)\}$.
542. Рис. 154
543. Первая и вторая.
544. Вторая и третья.
545. Вторая и третья.
546. **(а)** задача 543;
(б)–(г) рис. 155;
(д) задача 545;
(е) задача 544;

$z_0 \leftarrow 0;\ z_1 \leftarrow (z_0 < y);$
$\mathsf{while}\ z_1\ \mathsf{do}$
$\quad x \leftarrow s(x);\ z_0 \leftarrow s(z_0);$
$\quad z_1 \leftarrow (z_0 < y);$
$\mathsf{end};$

Рис. 154: Программа Π_+ из задачи 542.

```
u ← x;  v ← x;
while v do
    x ← w;
    w ← s(w);
    v ← (w < u);
end;
```

```
u ← y;
while u do
    x ← dec(x);
    u ← dec(u);
end;
```

```
u ← x;  x ← 0;
x ← s(x);
while u do
    x ← xu;
    u ← dec(u);
end;
```

Рис. 155: Программы (б), (в), (г) из задачи 546.

```
u ← x;  x ← 0;
x ← s(x);  v ← y;
while v do
    x ← xu;
    v ← dec(v);
end;
```

```
u ← s(x);  w ← x;
v ← 0;  v ← s(v);
while w do
    x ← z;  z ← s(z);
    v ← vy;  w ← (v < u);
end;
```

Рис. 156: Программы (ж)–(з) из задачи 546.

(ж)–(з) рис. 156;
(и)–(к) рис. 157.
547. Рис. 158.
548. Использовать программы из задачи 546.
549. (б) Для $x > 0$ получаем $B(x, y) = (x-1) + \cdots + (x-1) + B(x, 0) = y(x-1) + B(x, 0)$, $B(x, 0) = B(x-1, 1) + 1 = x - 1 + B(x-1, 0) + 1 = x + B(x-1, 0)$, следовательно, $B(x, 0) = x + (x-1) + \cdots + 2 + 1 = \frac{(x+1)x}{2}$ и $B(x, y) = y(x-1) + \frac{(x+1)x}{2}$, при $x > 0$.
(в) При $x > 1$ получаем $C(x, 0) = C(x-1, 1) + 1 = C(x-1, 0) \dot{-} (x-2) + 1$, следовательно, $C(0, 0) = 2$, $C(1, 0) = 4$, $C(2, 0) = 5$, $C(3, 0) = 5$, $C(4, 0) = 4$, $C(5, 0) = 2$, $C(x, 0) = 1$ при $x > 5$. При $y > 0$ получаем $C(1, y) = 4$, $C(2, y) = 5 \dot{-} y$, $C(3, y) = 5 \dot{-} 2y$, $C(4, 1) = 1$, $C(x, y) = 0$ для остальных случаев при $x > 0$.
(д) $E(1, 0) = 2$, $E(x, y) = 0$ во всех остальных случаях при $x > 0$;
(ж) Для $y \geqslant 2$, $x > 0$ получаем $H(x, y) = (y-2) + (y-4) + \cdots + (y \bmod 2) + H(x, y \bmod 2) = (y - \lfloor y/2 \rfloor - 1) \lfloor y/2 \rfloor + H(x, y \bmod 2)$. Для $y < 2$, $x > 1$ получаем $H(x, y) = H(x-2, y) + 3$, следовательно, $H(x, y) = {}^{3x}/_{2} + y$ для

```
if y then
    u ← s(x);  u ← u ∸ y;
    while u do
        x ← x ∸ y;
        u ← u ∸ y;
    end;
else
    x ← 0;
end;
```

```
u ← x;
y ← x;
while y do
    v ← mod(u, y);
    if v then
        x ← dec(x);
    end;
    y ← dec(y);
end;
```

Рис. 157: Программы (и)–(к) из задачи 546.

```
1    f ← 0;   2
2    n ← s(f);   3
3    if i then 4 else 7
4    n ← n + f;   5
5    f ← n ∸ f;   6
6    i ← dec(i);   3
```

Рис. 158: Числа Фибоначчи.

чётных x и $H(x, y) = {}^{3(x-1)}/_2 + 3 - 2y$ для нечётных x. Итого при $x > 0$ получаем $H(x, y) = {}^{y^2}/_4 - {}^{y}/_2 + {}^{3x}/_2$ для чётных x и y; $H(x, y) = {}^{(y-1)^2}/_4 + {}^{3x}/_2 + 1$ для чётных x и нечётных y; $H(x, y) = {}^{y^2}/_4 - {}^{y}/_2 + {}^{3(x-1)}/_2 + 3$ для нечётных x и чётных y; $H(x, y) = {}^{(y-1)^2}/_4 + {}^{3(x-1)}/_2 + 1$ для нечётных x и y.

(а), рис. 159. **(г)**, **(е)** рис. 160.

550. Увеличивать переменную x до тех пор, пока она не станет равна c. Если за это время она когда-либо была равна b, то условие выполнено. Рис. 161.

551. Полное ветвление if x then Π_1 else Π_2 end; можно заменить на $z \leftarrow 0$; $z \leftarrow s(z)$; if x then Π_1 $z \leftarrow 0$; end; if z then Π_2 end; , где z — новая переменная. Неполное ветвление if x then Π_1 end; можно заменить на $z \leftarrow x$; while z do Π_1 $z \leftarrow 0$; end; , где z — новая переменная.

552. Последовательно перебираем все натуральные числа $i = 0, 1, 2, \ldots$, для каждого вычисляем $f(i)$, как только будет $f(i) = x$, то $i = f^{-1}(x)$.

```
u ← (y < x);
if u then
    a ← s(x ∸ y); a ← ⌊a/2⌋ ; a ← pow(2, a);
else
    a ← y ∸ x; a ← s(a);
end;
u ← xy;
while u do
    x ← dec(x); y ← dec(y);
    u ← xy; a ← a + u;
end;
```

Рис. 159: Программа (а) из задачи 549.

```
if y then
    while x do
        x ← dec(x);
        d ← d + pow(x, dec(y));
    end;
    d ← d + y;
else
    d ← x;
end;
```

```
g ← 3;
if x then
    u ← x;
    while u do
        g ← g + ⌊log₂ u⌋ ;
        u ← dec(u);
    end;
    while y do
        y ← dec(y);
        g ← g + y · (dec(x))²;
    end;
end;
```

Рис. 160: Программы (г), (е) из задачи 549.

```
x ← 0;  a ← 0;  y ← eq(x, c);
while y do
    y ← eq(x, b);
    if y then
        a ← s(a);
    end;
    x ← s(x);  y ← eq(x, c);
end;
```

Рис. 161: Программа из задачи 550.

553. **(а)** $A_m^n \cdot m$, $m \sum\limits_{n=0}^{m} A_m^n = mA_m \approx em \cdot m!$ (см. задачу 67).

(б)–(в) С помощью вложенных полных ветвлений проверяем все переменные x_i. Если x_i не равно нулю, то каждую переменную x_j увеличиваем на $2^{i(n+1)+j}$ соответственно. Пусть $\bar{x}'$ и $\bar{x}''$ — два варианта входных переменных. При этом первый аргумент, которым они отличаются, является k-м по счёту, в наборе $\bar{x}'$ — это переменная x_p, а в наборе $\bar{x}'$ — это переменная x_q, причём можно считать $p < q$. Рассмотрим вычисление функций на наборе $(0, \dots, 0, 1, 0, \dots, 0)$, где единица стоит на k-м месте. В первом случае переменная x_p получит значение $1 + 2^{p(n+1)+p}$, а любая другая переменная x_i — значение $2^{p(n+1)+i}$. Во втором случае переменная x_q получит значение $1 + 2^{q(n+1)+q}$, а любая другая переменная x_j — значение $2^{q(n+1)+j}$. Очевидно, что среди этих значений нет равных, так как $q(n+1)+j-(p(n+1)+i) = (q-p)(n+1)+(j-i) \geqslant n+1+(1-n) = 2$. Если показатели отличаются не меньше чем на два, то сами степени отличаются как минимум в четыре раза. Поэтому все функции, которые получаются в первом и во втором случае, попарно различны.

554. При наличии одной переменной $x \leftarrow (x < x)$; эквивалентно $x \leftarrow 0$;, а $x \leftarrow x$; ничего не меняет, поэтому такие операторы можно исключить. Ветвление имеет смысл только как самый первый оператор, поскольку после $x \leftarrow s(x)$; или $x \leftarrow 0$; его результат заранее известен. Таким образом, имеются следующие три варианта программ: (а) Π_{+m} — увеличение x на $m \leqslant n$: $\Phi(x) = x + m$; (б) Π_m — присваивание $x \leftarrow 0$; и увеличение на $m < n$: $\Phi(x) = m$; (в) Π_{-} — зацикливание. Всего $2n + 2$ варианта. Кроме того, возможна проверка в начале, больше аргумент нуля или нет, а дальше — комбинация (а), (б) и (в) в зависимости от результата сравнения. Рассмотрим все варианты. (а, а) — Π_{+m} и Π_{+k} для $m \neq k$, $m, k \leqslant n-1$,

n^2-n вариантов. Одна из программ яляется частью другой. (а, б) и (б, а) — Π_{+m} и Π_k, $m,k \leqslant n-2$, n^2-2n+1 вариант. Программы имеют общую часть $\Pi_{+\min\{m,k\}}$. (б, б) — Π_m и Π_k для $m \neq k$, $m,k \leqslant n-3$, n^2-5n+6 вариантов. Программы имеют общую часть $\Pi_{+\min\{m,k\}}$. (а, в) и (в, а) — n вариантов. (б, в) и (в, б) — $n-1$ вариант. Всего $2n+2+n^2-n+2(n^2-2n+1)+n^2-5n+6+2n+2(n-1)=4n^2-4n+8$.

555. После оператора $\alpha\ x \leftarrow s(x);\ \beta$ ветвление β if x then γ else δ предопределено, поэтому можно сразу написать $\alpha\ x \leftarrow s(x);\ \gamma$. Если после оператора $\alpha\ x \leftarrow s(x);\ \beta$ идёт $\beta\ x \leftarrow \mathrm{dec}(x);\ \gamma$, то суммарно ничего не меняется, поэтому вместо перехода к α можно сразу переходить к γ. Исходя из вышесказанного делаем вывод, что присваивания $x \leftarrow s(x)$; могут встречаться только в самом конце, а начало состоит из декрементов и ветвлений. После ветки else декременты и ветвления смысла уже не имеют, так как значение переменной известно: ноль. Поэтому можно считать, что после else всегда начинаются присваивания $x \leftarrow x+1$;. Таким образом, программа устроена так: сначала идут несколько ветвлений и декрементов, затем (возможно) начинается цикл, состоящий из декрементов и ветвлений, который продолжается, пока x не станет равным нулю. Следовательно, вычисляемая функция имеет такой вид: на некотором начальном отрезке она задана таблично: $f(0)=a_0, \ldots, f(k)=a_k$, а для $x>k$ либо $f(x)=x+c$ для некоторой константы c, либо $f(x)$ задана таблично в зависимости от $x \bmod c$ для некоторой константы c.

556. Если программа содержит ℓ меток, то она имеет $\ell 2^n$ всевозможных конфигураций, так как переменные могут принимать только два значения. Следовательно, если вычисление имеет длину больше $\ell 2^n$, то оно уже содержит две одинаковые конфигурации, поэтому программа зацикливается. Значит, чтобы найти Φ, нужно перебрать все возможные наборы аргументов (их не более 2^n), для каждого из них построить вычисление длины не больше $\ell 2^n+1$, если до этого момента программа остановится, то результат будет значением функции Φ, а иначе результат не определён.

557. Если программа не содержит циклов, то каждый оператор присваивания выполняется не более одного раза. Следовательно, значение любой переменной будет находиться в промежутке $[a, a+n]$, где a — значение какой-либо входной переменной или ноль, а n — общее количество операторов присваивания в программе. Если $x,y>n$, то $x+y>\max\{x,y\}+n$ и $xy>\max\{x,y\}+n$, поэтому программа не может вычислять сумму или произведение. Если $x=2n+3$ и $y=n+2$, то $x-y=n+1>0+n$ и $x-y=n+1<\min\{x,y\}$, поэтому программа не может вычислять разность.

Если $x = (n+2)(n+1)$ и $y = n+2$, то $\lfloor x/y \rfloor = n+1$, поэтому программа не может вычислять частное по тем же причинам, что и разность в предыдущем случае.

Частично рекурсивные функции

558. $F(x,y) = f(h(x), g(h(y),y), y) = (2x^2)(2(2y^2)+y)+y = 8x^2y^2+2x^2y+y.$

559. **(а)** $f(x) = 0+(2\cdot 0+1)+(2\cdot 1+1)+\cdots+(2\cdot(x-1)+1) = x^2$;
(б) $f(x) = 0+(2\cdot 0+2)+(2\cdot 1+2)+\cdots+(2\cdot(x-1)+2) = x^2+x$;
(в) $f(x) = 0+(3\cdot 0\cdot 1+1)+(3\cdot 1\cdot 2+1)+\cdots+(3\cdot(x-1)\cdot x+1) = x^3$.

560. Индукция по x. При $x=0$: $f(x) = 0 = 3\cdot 2^{0+1}-4\cdot 0-6$. Если для x доказано, то для $x+1$ выполнено $f(x+1) = s(x)+s(x)+x+x+f(x)+f(x) = 4x+2+2(3\cdot 2^{x+1}-4x-6) = 4x+2+6\cdot 2^{x+1}-8x-12 = 6\cdot 2^{x+1}-4x-10 = 3\cdot 2^{(x+1)+1}-4(x+1)-6$.

561. **(а)** $F(0)=1$, $F(1) = \lfloor 2^1/1^2 \rfloor = 2$, $F(2) = \lfloor 2^2/2^2 \rfloor = 1, \ldots, F(5)=2$;
(б) $F(0)=1$, $F(1) = \lfloor 2^1/1 \rfloor = 2$, $F(2) = \lfloor 2^2/2 \rfloor = 2, \ldots, F(5)=2$;
(в) $F(0)=2$, $F(1) = 3\cdot 2 \dot{-} (2\cdot 0+1) = 5$, $F(2) = 3\cdot 5 \dot{-} (2\cdot 1+1) = 12$, $F(3) = 3\cdot 12 \dot{-} (2\cdot 2+1) = 31$, $F(4) = 3\cdot 31 \dot{-} (2\cdot 3+1) = 86$, $F(5) = 3\cdot 86 \dot{-} (2\cdot 4+1) = 249$.

562. **(а)** $\mathrm{pow} = \mathrm{Pr}[1, \mathrm{id}_1^3\cdot \mathrm{id}_3^3]$;
(б) $\mathrm{fact} = \mathrm{Pr}[1, s(\mathrm{id}_1^2)\cdot \mathrm{id}_2^2]$;
(в) $\mathrm{if} = \mathrm{test}(\mathrm{id}_1^3)\cdot \mathrm{id}_2^3 + \mathrm{not}(\mathrm{id}_1^3)\cdot \mathrm{id}_3^3$;
(г) индукция по n. Для $n=1$: $\min^{(1)} = \mathrm{id}_1^1$, для $n+1$: $\min^{(n+1)} = \mathrm{if}(\mathrm{less}(\mathrm{id}_1^{n+1},\ \mathrm{id}_{n+1}^{n+1}),\ \min^{(n)}(\mathrm{id}_1^{n+1},\ \ldots,\ \mathrm{id}_n^{n+1}),\ \min^{(n)}(\mathrm{id}_2^{n+1},\ \ldots,\ \mathrm{id}_{n+1}^{n+1}))$;
(д) аналогично (г);
(е) $\mathrm{diff} = (\mathrm{id}_1^2 \dot{-} \mathrm{id}_2^2) + (\mathrm{id}_2^2 \dot{-} \mathrm{id}_1^2)$;
(ж) $\mathrm{mod} = \mathrm{Pr}[0, \mathrm{less}(s(\mathrm{id}_3^3), \mathrm{id}_1^3)\cdot s(\mathrm{id}_3^3)](\mathrm{id}_2^2, \mathrm{id}_1^2)$;
(з) $\mathrm{div} = \mathrm{Pr}[0, \mathrm{id}_3^3 + \mathrm{not}(\mathrm{mod}(s(\mathrm{id}_2^3), \mathrm{id}_1^3))](\mathrm{id}_2^2, \mathrm{id}_1^2)$.

563. Суперпозиция: $g = f(\mathrm{id}_{i_1}^n, \ldots, \mathrm{id}_{i_n}^n)$.

564. $\mu_{\lim} f = \mathrm{Pr}[0,\ \mathrm{if}(\mathrm{less}(\mathrm{id}_{n+2}^{n+2},\ \mathrm{id}_{n+1}^{n+2}),\ \mathrm{id}_{n+2}^{n+2},\ \mathrm{id}_{n+1}^{n+2} + \mathrm{test}(f(\mathrm{id}_1^{n+2},\ \ldots,\ \mathrm{id}_{n+1}^{n+2}))))]$.

565. $f = \mathrm{if}(\mathrm{less}(\mathrm{id}_n^n,\ b),\ a,\ g(\mathrm{id}_1^n,\ \ldots,\ \mathrm{id}_{n-1}^n,\ \mathrm{id}_n^n \dot{-} b))$.

566. **(а)** 0, 0; **(б)** 3, 5; **(в)** 4, 5; **(г)** 3, 4;
(д) 2, 4. $F(x) = \lfloor\sqrt{x}\rfloor$ в случае (г).

567. Только (д).

568. (а) и (г).

569. **(а)** $\mathrm{rt} = \mu\,\mathrm{leq}(\mathrm{pow}(s(\mathrm{id}_3^3), \mathrm{id}_1^3), \mathrm{id}_2^3)$;
(б) $\log = \mu\,\mathrm{leq}(\mathrm{pow}(\mathrm{id}_1^3, s(\mathrm{id}_3^3)), \mathrm{id}_2^3)$;

(в) $p = \mathrm{less}(1, \mathrm{id}_1^1) \cdot \mathrm{eq}(2, \tau(\mathrm{id}_1^1))$, где τ — количество делителей числа (задача 568);
(г) $\mathrm{pn} = \mathrm{Pr}[2, \mu(\mathrm{leq}(\mathrm{id}_3^3, \mathrm{id}_2^3) + \mathrm{not}(p(\mathrm{id}_3^3)))]$;
(д) $\sigma = \sum\limits_{x<\mathrm{id}_1^1} \mathrm{not}(\mathrm{mod}(\mathrm{id}_1^1, s(x))) \cdot s(x)$;
(е) $\mathrm{digit} = \mathrm{mod}(\lfloor \mathrm{id}_1^3 \,/\, \mathrm{pow}(\mathrm{id}_2^3, \mathrm{id}_3^3) \rfloor, \mathrm{id}_2^3)$;
(ж) $\text{НОД} = \mathrm{id}_1^2 - \mu(\mathrm{mod}(\mathrm{id}_1^3, (\mathrm{id}_1^3 - \mathrm{id}_3^3)) + \mathrm{mod}(\mathrm{id}_2^3, (\mathrm{id}_1^3 - \mathrm{id}_3^3)))$.
570. $l = \mu\,\mathrm{eq}(\mathrm{id}_1^2, \mathrm{pow}(2, \mathrm{id}_2^2))$.
571. $d^{(3)} = |\mathrm{id}_1^3 - 100|^2 + |\mathrm{id}_2^3 - 100|^2$ (квадрат расстояния до центра круга, последний аргумент фиктивный); $p = \mu\,\mathrm{leq}(d, 2500)$; $r = p + \left\lfloor \sqrt{d(\mathrm{id}_1^2, \mathrm{id}_2^2, 0)} \right\rfloor \dot{-} 50$.
572. $g'(\bar{x}, y) = \prod\limits_{z<y} g(\bar{x}, z)$; $g''(\bar{x}, y) = \mathrm{test}(g'(\bar{x}, y)) \cdot \mathrm{not}(g(\bar{x}, y))$. Теперь $g''(\bar{x}, y)$ равно единице для наименьшего y, для которого $g(\bar{x}, y) = 0$, и равно нулю для всех остальных y. Следовательно, $h(\bar{x}) = \sum\limits_{y<f(\bar{x})} y \cdot g''(\bar{x}, y)$. Все использованные конструкции не требуют применения минимизации.
573. Если f' получена из f заменой значений на b_i в точках a_i: $f'(a_i) = b_i$, для $i = 1, \ldots, k$, то $f'(x) = \mathrm{if}(\mathrm{eq}(x,\, a_1),\, b_1,\, \mathrm{if}(\mathrm{eq}(x,\, a_2),\, b_2,\, \mathrm{if}(\ldots \mathrm{if}(\mathrm{eq}(x,\, a_k),\, b_k,\, f(x)) \ldots)))$.
574. Базис: для функций 0 и id_m^n это очевидно. Индукционный шаг: пусть для функций $g, h, h_1, \ldots, h_m$ это доказано. Для суперпозиции $f = g(h_1, \ldots, h_m)$ выполнено $f(\bar{x}) = g(h_1(\bar{x}), \ldots, h_m(\bar{x}))$. По индукционному предположению для h_i выполнено $y_i = h_i(\bar{x}) < 2$, поэтому $f(\bar{x}) = g(\bar{y}) < 2$ по индукционному предположению для g. Для примитивной рекурсии $f = \mathrm{Pr}[g, h]$ выполнено $f(\bar{x}, 0) = g(\bar{x}) < 2$ по индукционному предположению для g. Далее по индукционному предположению для h получаем $f(\bar{x}, 1) = h(\bar{x}, 0, f(\bar{x}, 0)) < 2$.
575. $f^{-1} = \mu(\mathrm{not}(\mathrm{eq}(f(\mathrm{id}_2^2), \mathrm{id}_1^2)))$.
576. $f = \mathrm{if}(\mathrm{less}(\mathrm{id}_{n+2}^{n+2},\, \mathrm{id}_{n+1}^{n+2}),\, 0,\, \sum\limits_{x<s(\mathrm{id}_{n+2}^{n+2})} g(\mathrm{id}_1^{n+2},\, \ldots,\, \mathrm{id}_n^{n+2},\, \mathrm{id}_{n+1}^{n+2} + x))$.
577. $F = \mu(f(\mathrm{id}_1^{n+1},\, \ldots,\, \mathrm{id}_n^{n+1},\, \mathrm{id}_{n+1}^{n+1}) \cdot \mathrm{leq}(g(\mathrm{id}_1^{n+1},\, \ldots,\, \mathrm{id}_n^{n+1},\, \mathrm{id}_{n+1}^{n+1}),\, h(\mathrm{id}_1^{n+1},\, \ldots,\, \mathrm{id}_n^{n+1},\, \mathrm{id}_{n+1}^{n+1})))$.
578. **(а)** Имеется $n+1$ пара (x, y), у которой $x + y = n$. Поэтому количество пар (x, y), у которых $x+y < x_0+y_0$, равно сумме арифметической прогрессии $1 + 2 + 3 + \cdots + x_0 + y_0 = (x_0 + y_0)(x_0 + y_0 + 1)/2$. Среди пар (x, y), у которых $x + y = x_0 + y_0$, пара (x_0, y_0) будет x_0-я по счёту, следовательно, $\langle x_0, y_0 \rangle = (x_0 + y_0)(x_0 + y_0 + 1)/2 + x_0$.

(б) $f(z) = \lfloor(\sqrt{1+8z}-1)/2\rfloor$, $\langle z\rangle_1 = z - f(z)(f(z)+1)/2$, $\langle z\rangle_2 = f(z) - \langle z\rangle_1$.
(в) $\langle,\rangle$ — суперпозиция сложения, умножения, целочисленного деления и констант. $\langle\rangle_1$ и $\langle\rangle_2$ — суперпозиция сложения, вычитания, умножения, целочисленного деления, констант и извлечения квадратного корня.

579. $g = \Pr[\langle 0,1\rangle, \langle\langle \mathrm{id}_2^2\rangle_2, \langle \mathrm{id}_2^2\rangle_1 + \langle \mathrm{id}_2^2\rangle_2\rangle]$, $F(x) = \langle g(x)\rangle_1$.

580. $c = \mathrm{dec}(\mathrm{pow}(2, \mathrm{id}_1^2)\cdot s(\mathrm{id}_2^2 + \mathrm{id}_2^2))$;
$c_1 = \mu\,\mathrm{not}(\mathrm{mod}(\lfloor s(\mathrm{id}_1^2)/\mathrm{pow}(2, \mathrm{id}_2^2)\rfloor, 2))$; $c_2 = \lfloor \mathrm{dec}(\lfloor s/\mathrm{pow}(2, c_1)\rfloor)/2\rfloor$.

581. **(а)** Пусть $K(x_1,\ldots,x_n) = K(y_1,\ldots,y_m) = z$. Тогда $n = k_2(z) = m$. Далее используем индукцию по n. При $n = 0$ доказывать нечего. Если для n доказано, то для $n+1$ получаем $x_{n+1} = k_2(k_1(z)) = y_{n+1}$, $k_1(K(x_1,\ldots,x_n)) = k_1(k_1(z)) = k_1(K(y_1,\ldots,y_n))$. Так как $k_2(K(x_1,\ldots,x_n)) = n = k_2(K(y_1,\ldots,y_n))$, то $K(x_1,\ldots,x_n) = K(y_1,\ldots,y_n)$. По индукционному предположению получаем $x_1 = y_1$, $\ldots$, $x_n = y_n$. Докажем сюръективность. Пусть $k_2(y) = n$. Тогда y является кодом набора, состоящего из n из элементов: $k_1^n(y)$, $k_2(k_1^{n-1}(y))$, $k_2(k_1^{n-2}(y))$, $\ldots$, $k_2(k_1^1(y))$. С помощью k_1^i обозначено i-кратное применение функции k_1.
(б) По определению.
(в) Пусть $f^{(2)} = \Pr[\mathrm{id}_1^1, k_1(\mathrm{id}_3^3)]$. Тогда $K' = \mathrm{less}(\mathrm{id}_2^2, k_2(\mathrm{id}_1^2))\cdot k_2(f(\mathrm{id}_1^2, k_2(\mathrm{id}_1^2) \dot- \mathrm{id}_2^2))$.

582. **(а)** $100111011_2 = 315$; $110101100_2 = 428$.
(б) $169 = 10101001_2$, $(1,0,1,1,1)$; $19783 = 100110101000111_2$, $(3,0,0,1,1,2,0,1)$.
(в) Функция $z(x,y)$ возвращает количество нулей среди y младших разрядов числа x: $z = \Pr[0, \mathrm{id}_3^3 + \mathrm{not}(\mathrm{digit}(\mathrm{id}_1^3, 2, \mathrm{id}_2^3))]$. Функция $p(x,y)$ возвращает номер разряда, на котором стоит y-й ноль: $p = \mu\,\mathrm{less}(z(\mathrm{id}_1^3, \mathrm{id}_3^3), \mathrm{id}_2^3)$. Тогда $k = \mu\,\mathrm{digit}(\mathrm{id}_1^3, 2, s(p(\mathrm{id}_1^3, \mathrm{id}_2^3)) + \mathrm{id}_3^3)$.

583. **(а)** $A(1,y) = y+2$, $A(2,y) = 2y+3$.
(б) Базис: $A(3,0) = A(2,1) = 5 = 2^{0+3}-3$. Индукционный шаг: $A(3,y+1) = A(2, A(3,y)) = 2\cdot A(3,y)+3 = 2\cdot(2^{y+3}-3)+3 = 2\cdot 2^{y+3} - 2\cdot 3 + 3 = 2^{(y+1)+3}-3$.
(в) Для удобства обозначим $k_{21}(u) = k_2(k_1(u))$ — последний элемент последовательности, $k_{211}(u) = k_2(k_1(k_1(u)))$ — предпоследний элемент последовательности, $k_{111}(u) = k_1(k_1(k_1(u)))$ — начало последовательности. Функция $f(u) = \mathrm{if}(\mathrm{not}(k_{211}(u))$, $k(k(k_{111}(u)$, $s(k_{21}(u)))$, $\mathrm{dec}(k_2(u)))$, $\mathrm{if}(\mathrm{not}(k_{21}(u))$, $k(k(k(k_{111}(u)$, $\mathrm{dec}(k_{211}(u)))$, $1)$, $k_2(u))$, $k(k(k(k(k_{111}(u)$, $\mathrm{dec}(k_{211}(u)))$, $k_{211}(u))$, $\mathrm{dec}(k_{21}(u)))k_2(u))))$ выполняет преобразование последнего $A(x_n, z)$ по определению функции Аккермана; функция $g = \Pr[k(k(\mathrm{id}_1^2, \mathrm{id}_2^2), 2), f(\mathrm{id}_4^4)]$ выполняет заданное количество преобразований; функция $h = \mu\,\mathrm{dec}(k_2(g))$ находит шаг, когда $n = 0$. Тогда $A(x,y) = k_1(g(x,y,h(x,y)))$.

Машины Тьюринга

584. На ленте записано 1011, начиная с нулевой ячейки, головка в ячейке -1, 9 шагов.
585. $\mathfrak{M}_1$ и $\mathfrak{M}_3$.
586. (б) и (в).
587. $q_0, a \to q_1, a', +1$; $q_0, b \to q_1, a', +1$; $q_0, b' \to q_6, b, 0$; $q_1, a \to q_1, a, +1$; $q_1, b \to q_1, b, +1$; $q_1, \Lambda \to q_2, \Lambda, -1$; $q_1, b' \to q_2, b, -1$; $q_2, a \to q_3, b', -1$; $q_2, b \to q_3, b', -1$; $q_2, a' \to q_4, a, -1$; $q_3, a \to q_3, a, -1$; $q_3, b \to q_3, b, -1$; $q_3, a' \to q_0, a, +1$; $q_4, a \to q_4, a, -1$; $q_4, \Lambda \to q_5, \Lambda, +1$; $q_5, a \to q_6, \Lambda, 0$. Время: $O((n+k)^2)$.

588. $q_0, 0 \to q_1, 0, +1$; $q_0, 1 \to q_7, 1, +1$; $q_0, \Lambda \to q_7, \Lambda, 0$; $q_1, 0 \to q_1, 0, +1$; $q_1, 1 \to q_2, 1, +1$; $q_1, \Lambda \to q_8, \Lambda, -1$; $q_2, 0 \to q_7, 0, +1$; $q_2, 1 \to q_2, 1, +1$; $q_2, \Lambda \to q_3, \Lambda, -1$; $q_3, 0 \to q_9, 0, +1$; $q_3, 1 \to q_4, \Lambda, -1$; $q_3, \Lambda \to q_7, \Lambda, -1$; $q_4, 0 \to q_4, 0, -1$; $q_4, 1 \to q_4, 1, -1$; $q_4, \Lambda \to q_5, \Lambda, +1$; $q_5, 0 \to q_6, \Lambda, +1$; $q_5, 1 \to q_7, 1, +1$; $q_5, \Lambda \to q_7, \Lambda, -1$; $q_6, 0 \to q_6, 0, +1$; $q_6, 1 \to q_6, 1, +1$; $q_6, \Lambda \to q_3, \Lambda, -1$; $q_7, 0 \to q_7, 0, +1$; $q_7, 1 \to q_7, 1, +1$; $q_7, \Lambda \to q_8, \Lambda, -1$; $q_8, 0 \to q_8, \Lambda, -1$; $q_8, 1 \to q_8, \Lambda, -1$; $q_8, \Lambda \to q_{10}, 1, 0$; $q_9, 0 \to q_9, \Lambda, -1$; $q_9, 1 \to q_9, \Lambda, -1$; $q_9, \Lambda \to q_{10}, 0, 0$. Время работы: $O((n+k)^2)$.

589. q_0, $a \to q_0$, a, $+1$; q_0, $b \to q_1$, b, $+1$; q_0, $\Lambda \to q_4$, b, -1; q_1, $a \to q_2$, b, -1; q_1, $b \to q_1$, b, $+1$; q_1, $\Lambda \to q_4$, b, -1; q_2, $a \to q_3$, a, $+1$; q_2, $b \to q_2$, b, -1; q_2, $\Lambda \to q_3$, Λ, $+1$; q_3, $b \to q_0$, a, $+1$; q_4, $a \to q_5$, a, $+1$; q_4, $b \to q_4$, b, -1; q_4, $\Lambda \to q_5$, Λ, $+1$; q_5, $b \to q_5$, Λ, 0. Каждая буква a требует для перемещения не более чем $2m$ шагов (m влево и m вправо). Следовательно, общее количество шагов $O(nm)$.

590. Далее x — любой символ $\Sigma \cup \{\Lambda\}$, y — любой символ Σ: $q_0, x \to q_0, x, +1$; $q_0, * \to q_1, *, -1$; $q_1, y \to q_5, y, +1$; $q_1, \Lambda \to q_2, \Lambda, -1$; $q_2, y \to q_3, y, -1$; $q_2, \Lambda \to q_2, \Lambda, -1$; $q_3, y \to q_3, y, -1$; $q_3, \Lambda \to q_4^{\Lambda}, \Lambda, +1$; $q_4^x, y \to q_4^y, x, +1$; $q_4^x, \Lambda \to q_0, x, 0$; $q_5, * \to q_5, \Lambda, 0$.

591. Копировать символы последнего слова по одному, разделяя исходной слово и копию пробелом и штрихуя уже скопированные символы. В конце штрихи убрать. q_0, $\alpha \to q_1^{\alpha}$, α', $+1$; q_0, $\Lambda \to q_6$, Λ, $+1$; q_1^{β}, $\alpha \to q_1^{\beta}$, α, $+1$; q_1^{β}, $\Lambda \to q_2^{\beta}$, Λ, $+1$; q_2^{β}, $\alpha \to q_0$, α, 0; q_2^{β}, $\alpha' \to q_5^{\beta}$, α', $+1$; q_2^{β}, $\Lambda \to q_3$, β', -1; q_3, $\alpha' \to q_3$, α', -1; q_3, $\Lambda \to q_4$, Λ, -1; q_4, $\alpha \to q_4$, α, -1; q_4, $\alpha' \to q_0$, α, $+1$; q_5^{β}, $\alpha' \to q_5^{\beta}$, α', $+1$; q_5^{β}, $\Lambda \to q_3$, β', -1; q_6, $\alpha' \to q_6$, α, $+1$; q_6, $\Lambda \to q_7$, Λ, -1; q_7, $\alpha \to q_7$, α, -1; q_7, $\Lambda \to q_8$, Λ, -1; q_7, $\alpha' \to q_7$, α, -1; q_8, $\alpha \to q_7$, α, -1; q_8, $\alpha' \to q_7$, α, -1.

592. В решении задачи 591 копировать символы не по одному, а сразу по $2k$ штук. Тогда перенос каждого блока символов будет занимать порядка

$2n+2k$ шагов, а количество блоков не больше ${}^{n}/_{2k}$. Поэтому общее время работы будет $2n + ({}^{n}/_{2k}) \cdot (2n+2k) = n^2/k + O(n)$.

593. Сравнивать символы в словах по одному слева направо, штрихуя уже рассмотренные. Как только обнаружатся неравные символы или одно из слов закончится, стереть всё на ленте и записать ответ. Далее α, β — любые символы из множества $\{a,b,c\}$: q_0, $\alpha \to q_1^\alpha$, α', $+1$; q_0, $\Lambda \to q_5$, Λ, $+1$; q_1^α, $\beta \to q_1^\alpha$, β, $+1$; q_1^α, $\Lambda \to q_2^\alpha$, Λ, $+1$; q_2^α, $\beta' \to q_2^\alpha$, β', $+1$; q_2^α, $\alpha \to q_3$, α', -1; q_2^a, $b \to q_6^a$, b, $+1$; q_2^a, $c \to q_6^a$, c, $+1$; q_2^b, $a \to q_6^c$, a, $+1$; q_2^b, $c \to q_6^a$, c, $+1$; q_2^c, $a \to q_6^c$, a, $+1$; q_2^c, $b \to q_6^c$, b, $+1$; q_2^α, $\Lambda \to q_7^c$, Λ, -1; q_3, $\alpha' \to q_3$, α', -1; q_3, $\Lambda \to q_4$, Λ, -1; q_4, $\alpha \to q_4$, α, -1; q_4, $\alpha' \to q_0$, α', $+1$; q_5, $\alpha' \to q_5$, α', $+1$; q_5, $\alpha \to q_6^a$, α, $+1$; q_5, $\Lambda \to q_7^b$, Λ, -1; q_6^α, $\beta \to q_6^\alpha$, β, $+1$; q_6^α, $\Lambda \to q_7^\alpha$, Λ, -1; q_7^α, $\beta \to q_7^\alpha$, Λ, -1; q_7^α, $\Lambda \to q_8^\alpha$, Λ, -1; q_8^α, $\beta \to q_8^\alpha$, Λ, -1; q_8^α, $\beta' \to q_8^\alpha$, Λ, -1; q_8^α, $\Lambda \to q_9^\alpha$, Λ, $+1$; q_9^α, $\Lambda \to q_9^\alpha$, α, 0.

594. **(а)** $q_0, \alpha \to q^\alpha, \Lambda, +1$; $q^\alpha, \beta \to q^\alpha, \beta, +1$; $q^\alpha, \Lambda \to q_1, \alpha, 0$ для всех $\alpha, \beta \in \Sigma$.

(б) Отметить первый символ и двигать головку вперёд, пока не найдутся два пробела подряд: $q_0, \delta \to q_1, \delta', +1$; $q_1, \delta \to q_1, \delta, +1$; $q_1, \Lambda \to q_2, \Lambda, +1$; $q_2, \delta \to q_1, \delta, +1$ для всех непустых $\delta \in \Sigma$. Поставить $*$: $q_2, \Lambda \to q_3, \Lambda, -1$; $q_3, \Lambda \to q_4, *, -1$. После этого переносить символы по одному из начала (помеченного штрихом) за звёздочку: $q_4, \beta \to q_4, \beta, -1$; $q_4, \delta' \to q^\delta, \Lambda, +1$; $q_4, \Lambda' \to q_5, \Lambda, +1$; $q^\delta, \beta \to q_1^\delta, \beta', +1$; $q_1^\delta, \gamma \to q_1^\delta, \gamma, +1$; $q_1^\delta, * \to q_2^\delta, *, +1$; $q_2^\delta, \alpha \to q_2^\delta, \alpha, +1$; $q_2^\delta, \Lambda \to q_4, \delta, -1$; $q_5, \gamma \to q_5, \gamma, +1$; $q_5, * \to q_6, \Lambda, 0$ для всех $\alpha, \beta, \gamma, \delta \in \Sigma \cup \{*, \Lambda\}$, $\gamma \neq *$, $\alpha, \delta \neq \Lambda$.

(в) Заменяем пробелы между словами на звёздочки. С помощью палочки разделяем исходные данные и результаты. По одному символу очередного слова переносим в конец и сдвигаем всё, что правее на один символ влево. $q_0, \alpha \to q_0, \alpha, +1$; $q_0, \Lambda \to q_1, *, +1$; $q_1, \alpha \to q_0, \alpha, +1$; $q_1, \Lambda \to q_2, \Lambda, -1$; $q_2, * \to q_3, \Lambda, -1$; $q_3, \alpha \to q_3, \alpha, -1$; $q_3, * \to q_4, |, +1$; $q_4, \alpha \to q_4, \alpha, +1$; $q_4, \Lambda \to q_5^*, \Lambda, -1$; $q_5^\beta, \gamma \to q_5^\gamma, \beta, -1$; $q_5^\beta, | \to q_6^|, \beta, -1$; $q_6^|, * \to q_7^*, |, +1$; $q_6^|, \Lambda \to q_8, \Lambda, +1$; $q_6^|, \alpha \to q_6^\alpha, |, -1$; $q_6^\beta, \alpha \to q_6^\alpha, \beta, -1$; $q_6^\beta, * \to q_7^\beta, *, +1$; $q_6^\beta, \Lambda \to q_7^\beta, \Lambda, +1$; $q_7^\delta, \zeta \to q_7^\delta, \zeta, +1$; $q_7^\delta, \Lambda \to q_5^\delta, \Lambda, -1$; $q_8, \alpha \to q_8, \alpha, +1$; $q_8, * \to q_8, \Lambda, +1$. Здесь $\alpha \in \Sigma$, $\beta, \gamma \in \Sigma \cup \{*\}$, $\delta, \zeta \in \Sigma \cup \{*, |\}$.

(г) Ставим в конце слова звёздочку и по одной переносим буквы за звёздочку, начиная с первой, записывая буквы каждый раз по две штуки: $q_0, \alpha \to q_0, \alpha, +1$; $q_0, \Lambda \to q_1, *, -1$; $q_1, \alpha \to q_1, \alpha, -1$; $q_1, \Lambda \to q_2, \Lambda, +1$; $q_2, \gamma \to q^\gamma, \Lambda, +1$; $q_2, * \to q_f, \Lambda, 0$; $q^\alpha, \beta \to q^\alpha, \beta, +1$; $q^\alpha, \Lambda \to q_1^\alpha, \alpha, +1$; $q_1^\alpha, \Lambda \to q_1, \alpha, -1$ для всех $\alpha, \beta, \gamma \in \Sigma \cup \{*, \Lambda\}$, $\alpha, \beta \neq \Lambda$, $\gamma \neq *$.

(д) Аналогично предыдущему, только вместо двух a пишем одну, вместо двух b пишем bc; а c не переносим: $q^a, \Lambda \to q_1, a, -1$; $q^b, \Lambda \to q_1^b, b, +1$;

$q_1^b, \Lambda \to q_1, c, -1$; $q_2, c \to q_2, \Lambda, +1$.

(е) Заменяем a на чётных позициях звёздочками, затем удаляем звёздочки, пока они есть. $q_0, \alpha \to q_1, \alpha, +1$; $q_0, a \to q_1, a', +1$; $q_0, \Lambda \to q_3, \Lambda, -1$; $q_1, \alpha \to q_0, \alpha, +1$; $q_1, a \to q_2, a, +1$; $q_1, \Lambda \to q_3, \Lambda, -1$; $q_2, \beta \to q_1, \beta, +1$; $q_2, \Lambda \to q_3, \Lambda, -1$; $q_3, a' \to q_4, *, -1$; $q_3, a \to q_4, a, -1$; $q_3, \alpha \to q_3, \alpha, -1$; $q_3, \Lambda \to q_5^\Lambda, \Lambda, +1$; $q_4, a' \to q_4, a, -1$; $q_4, a \to q_4, a, -1$; $q_4, \alpha \to q_3, \alpha, -1$; $q_4, \Lambda \to q_5^\Lambda, \Lambda, +1$; $q_5^\gamma, \beta \to q_5^\beta, \gamma, +1$; $q_5^\gamma, \Lambda \to q_6, \gamma, 0$; $q_5^\gamma, * \to q_3, \gamma, -1$ для всех $\alpha \in \{b, c\}$, $\beta \in \Sigma$, $\gamma \in \Sigma \cup \{\Lambda\}$.

(ж) Дойти до пробела между словами, поставить звёздочку, по одной удалять буквы a из левого и правого слова, пока в одном из них букв a не останется. Проверить, что и в другом не осталось. $q_0, \alpha \to q_0, \alpha, +1$; $q_0, \Lambda \to q_1, *, +1$; $q_1, \gamma \to q_1, \gamma, +1$; $q_1, a \to q_2, b, -1$; $q_1, \Lambda \to q_3, \Lambda, -1$; $q_2, \gamma \to q_2, \gamma, -1$; $q_2, a \to q_1, \gamma, +1$; $q_2, \Lambda \to q_5, \gamma, +1$; $q_3, \gamma \to q_3, \Lambda, -1$; $q_3, \Lambda \to q_f, 1, 0$; $q_3, a \to q_4, \Lambda, -1$; $q_4, \beta \to q_4, \Lambda, -1$; $q_4, \Lambda \to q_f, 0, 0$; $q_5, \beta \to q_5, \Lambda, +1$; $q_5, \Lambda \to q_f, 0, 0$ для всех $\alpha \in \Sigma$, $\beta \in \{a, b, c, *\}$, $\gamma \in \{b, c, *\}$.

(з) Удалять буквы с концов, одновременно проверяя, совпадают ли они. $q_0, \Lambda \to q_f, 1, 0$; $q_0, \alpha \to q^\alpha, \Lambda, +1$; $q^\alpha, \beta \to q^\alpha, \beta, +1$; $q^\alpha, \Lambda \to q_1^\alpha, \Lambda, -1$; $q_1^\alpha, \alpha \to q_1, \Lambda, -1$; $q_1^\alpha, \Lambda \to q_f, 1, 0$; $q_1^\alpha, \beta \to q_2, \Lambda, -1$ при $\beta \neq \alpha$; $q_1, \alpha \to q_1, \alpha, -1$; $q_1, \Lambda \to q_0, \Lambda, +1$; $q_2, \alpha \to q_2, \Lambda, -1$; $q_2, \Lambda \to q_f, 0, 0$ для всех $\alpha, \beta \in \Sigma$.

(и) Выполнять ряд следующих шагов: последовательно проходить по слову и заменять на b буквы a, стоящие после b или c, и считать, сколько последовательностей a закончилось во время этого прохода. Если их больше одной, то ответ 1. Если букв a больше нет, то ответ 0.

(к) Сначала убирать (заменять на звёздочки, например) из слова по одной букве a, одной b и одной c на каждом шаге. Если b или c закончились раньше a, то ответ 0. Иначе нужно проверить, что букв c осталось в два раза больше, чем b: для этого на каждом шаге убираем одну букву b и две буквы c. Если обе буквы закончились одновременно, то ответ 1, иначе 0.

595. Вычитаем единицы из двоичной записи и добавляем к унарной. $q_0, \Lambda \to q_f, |, 0$; $q_0, \alpha \to q_1, \alpha, -1$; $q_1, \Lambda \to q_2, |, +1$; $q_2, \beta \to q_2, \beta, +1$; $q_2, \Lambda \to q_3, \Lambda, -1$; $q_3, 1 \to q_4, 0, -1$; $q_3, 0 \to q_3, 1, -1$; $q_3, | \to q_5, |, +1$; $q_4, \beta \to q_4, y, -1$; $q_4, \Lambda \to q_2, |, +1$; $q_5, \alpha \to q_5, \Lambda, +1$ для всех $\alpha \in \{0, 1\}$, $\beta \in \{0, 1, |\}$.

596. q_0, $|$ $\to$ q_0, $|$, $+1$; q_0, $\Lambda \to q_1$, Λ, $+1$; q_1, $| \to q_0$, $|$, $+1$; q_1, $\Lambda \to q_2$, Λ, -1; q_2, $| \to q_3$, Λ, -1; q_2, $\Lambda \to q_2$, Λ, -1; q_3, $| \to q_4$, Λ, -1; q_3, $\Lambda \to q_5$, Λ, -1; q_4, $| \to q_4$, $|$, -1; q_4, $\Lambda \to q_5$, $|$, -1; q_5, $| \to q_5$, $|$, -1; q_5, $\Lambda \to q_6$, $*$, -1; q_6, $| \to q_6$, $|$, -1; q_6, $* \to q_6$, $*$, -1; q_6, $\Lambda \to q_7$, Λ, $+1$; q_7, $| \to q_8$, Λ, $+1$; q_7, $* \to q_{11}$, Λ, $+1$; q_8, $| \to q_8$, $|$, $+1$; q_8, $* \to q_8$, $*$, $+1$; q_8, $\Lambda \to q_9$, Λ, -1; q_9, $| \to q_6$, Λ, -1; q_9, $* \to q_{10}$, Λ, -1; q_{10}, $| \to q_{10}$, Λ, -1; q_{10}, $\Lambda \to q_{15}$, 1, 0;

q_{11}, $| \to q_{12}$, Λ, $+1$; q_{11}, $\Lambda \to q_{15}$, 1, 0; q_{12}, $| \to q_{12}$, Λ, $+1$; q_{12}, $\Lambda \to q_{15}$, 0, 0. Время: $O((k+m+n)^2)$.

597. **(а)** Записать | на место пустого символа между словами, стереть две палочки в конце.

(б) Стереть по одной палочке из второго слова, затем из первого. После того, как во втором слове стёрта последняя палочка, первое слово является разностью.

(в) Стереть самую первую палочку первого слова и записать новое слово | (ноль) правее второго. Далее выполнять такую последовательность, пока первое слово не опустеет: убрать одну палочку из первого слова, скопировать второе слово в конец, убрать одну палочку, соединить его с третьим. После этого третье слово является произведением.

(г) Стирать по одной палочке из каждого слова, если первое опустело раньше, то ответ 1, иначе 0.

(д) Стереть самую первую палочку первого слова и записать новое слово || (один) правее второго. Далее выполнять такую последовательность, пока первое слово не опустеет: убрать одну палочку из первого слова, скопировать второе слово в конец, поставить перед третьим разделитель $\#'$, запустить машину (в), чтобы найти произведение, стереть разделитель. После этого третье слово является степенью.

(е) Записать новое слово | (ноль) правее исходного. Далее выполнять такую последовательность. Скопировать в конец первое слово и два раза второе, поставить перед четвёртым разделитель $\#'$, запустить машину (в), чтобы найти квадрат, перенести разделитель перед третьим словом, запустить машину (г), чтобы сравнить исходное слово с квадратом. Если ответ 0, то добавить палочку ко второму слову и начать всё заново. Иначе — стереть одну палочку из второго слова, это и будет результат.

(ж) Стереть самую первую палочку первого слова и записать новое слово | (ноль) правее исходного. Далее выполнять такую последовательность, пока в первом слове не останется только одна палочка: стереть вторую половину первого слова, добавить палочку ко второму. После этого второе слово является логарифмом.

(з) Вставить новое слово | (ноль) между исходными. Далее выполнять такую последовательность, пока в третьем слове не останется только одна палочка: добавить палочку ко второму слову (сдвинув ленту вправо), скопировать первое слово в конец, вставить разделитель $\#'$ после второго, запустить машину (б), стереть разделитель. После этого из второго слова убрать одну палочку, оно и будет частным.

(и) Создать копии исходных слов, отделив их разделителем $\#'$, запустить машину (з), передвинуть разделитель после первого слова, запустить машину (в), убрать разделитель, запустить машину (б).

(к) Стереть всё, кроме m-го слова и передвинуть его к началу.

598. (а) Запускаем $\mathfrak{M}_1$, после её остановки запускаем $\mathfrak{M}_2$ и т. д. Общее время равно $t_1(x) + t_2(f_1(x)) + \dots + t_n(f_{n-1}(\dots f_1(x)\dots))$.

(б) Преобразуем машины $\mathfrak{M}_i$ в соответствующие односторонние машины $\mathfrak{N}_i$, с использованием «многоэтажных символов», например. Записываем x_1 после x_n, запускаем $\mathfrak{N}_1$, переносим результат $f_1(x_1)$ на место x_1. Записываем x_2 после x_n, запускаем $\mathfrak{N}_2$, переносим результат $f_2(x_2)$ на место x_2 и т. д. Общее время на копирование: $2|x_1ax_2a\dots ax_n|^2 + 2|f_1(x_1)af_2(x_2)a\dots af_n(x_n)|^2$. Время работы машины $\mathfrak{N}_i$ тоже равно $t_i(x)$. Поэтому общее время работы равно $2|x_1ax_2a\dots ax_n|^2 + 2|f_1(x_1)af_2(x_2)a\dots af_n(x_n)|^2 + t_1(x_1) + \dots + t_n(x_n)$.

(в) По аналогии преобразуем машину $\mathfrak{M}_1$ в одностороннюю машину $\mathfrak{N}_1$. Копируем вход, разделяя оригинал и копию символом #, запускаем $\mathfrak{N}_1$. Если результат непуст, то стираем # и всё справа от него и запускаем $\mathfrak{M}_2$. В противном случае — стираем всё справа от # и запускаем $\mathfrak{M}_3$. Время копирования: $2|x|^2$. Поэтому общее время работы равно $2|x|^2 + t_1(x) + \max\{t_2(x), t_3(x)\}$.

(г) По аналогии преобразуем машину $\mathfrak{M}_1$ в одностороннюю машину $\mathfrak{N}_1$. Делаем копию входа справа, разделяя оригинал и копию символом #, запускаем $\mathfrak{N}_1$ на копии. Если результат непуст, то стираем # и всё справа от него, запускаем $\mathfrak{M}_2$ и всё повторяем сначала. В противном случае — стираем # и останавливаемся. Время одного копирования не превосходит $2L(x)^2$. Поэтому общее время работы равно $q(2L(x)^2 + T_1(x) + T_2(x)) + 2L(x)^2 + T_1(x)$.

599. (а) $x^2 + 2x$; **(б)** $2x^2 + 2x$; **(в)** $2x^2 + x$.

600. По строкам: 2^x, 3^x, x^x, x^2.

601. Применить методы комбинации машин, описанные в задаче 598, к машинам из задачи 597.

602. (а) Создать две копии каждого состояния q^+ и q^-, в которых запоминать направление последнего сдвига: заменить команды $q, a \to p, b, +1$ на $q^x, a \to p^+, b, +1$, команды $q, a \to p, b, -1$ на $q^x, a \to p^-, b, -1$, команды $q, a \to p, b, 0$ на $q^x, a \to p^x, b, 0$. Тогда возврат происходит командами $q^+, a \to p, b, -1$ и $q^-, a \to p, b, +1$.

(б) Отметить в начале работы нулевую ячейку, например, звёздочкой: $q_0', a \to q_0, a^*, 0$, сохранять эту звёздочку до конца работы. Для возврата в нулевую ячейку двигать головку влево и вправо со всё увеличивающейся амплитудой, пока звёздочка не будет найдена.

(в) Реализовать пункт (б), для «отражения» отметить текущую ячейку, например, штрихом, затем использовать метод из пункта (б) для поиска звёздочки, когда будет понятно, с какой стороны от звёздочки находится головка, отсчитать от звёздочки в противоположную сторону столько же ячеек, сколько находится от звёздочки до штриха.

(г) Аналогично (в), только в конце отсчитывать ячейки от штриха.

(д) Реализовать пометку штрихами записываемых символов: каждую команду $q, a \to p, b, s$ заменяем на две $q, a \to p, b', s$ и $q, a' \to p, b', s$. Для реализации поиска a следует отметить текущую ячейку, далее искать справа букву a или a', пока не встретится фрагмент $\Lambda\Lambda$. Если буква найдена, то стереть метку со старой ячейки и перейти в новую. Если не найдена — вернуться в старую ячейку.

(е) Реализовать пометку штрихами как в пункте (д). Для вставки сдвинуть все заштрихованные ячейки на одну позицию вправо, записать в освободившуюся ячейку b'.

(ж) Расширить алфавит символами вида a^b, где $a, b \in \Sigma$, b будет обозначать начальный символ. Команды $q, a \to p, b, s$ заменяем на $q, a \to p, b^a, s$ и $q, a^c \to p, b^c, s$. Для восстановления выполняется команда $q, a^b \to p, b, s$.

603. Запоминаем символ в состоянии и стираем его, идём направо до конца, записываем символ, идём налево до второго Λ и записываем символ назад: $q_0, \alpha \to q_0, \alpha, +1$; $q_0, \Lambda \to q_1, \Lambda, +1$; $q_1, \alpha \to q_0, \alpha, +1$; $q_1, \Lambda \to q_2^{\Lambda}, \Lambda, -1$; $q_2^{\beta}, \alpha \to q_2^{\beta}, \alpha, -1$; $q_2^{\beta}, \Lambda \to q_3^{\beta}, \Lambda, -1$; $q_3^{\beta}, \alpha \to q_3^{\beta}, \alpha, -1$; $q_3^{\beta}, \Lambda \to q_4, \beta, +1$; $q_4, \alpha \to q_5^{\alpha}, \Lambda, +1$; $q_5^{\beta}, \alpha \to q_5^{\beta}, \alpha, +1$; $q_5^{\beta}, \Lambda \to q_6^{\beta}, \Lambda, +1$; $q_6^{\beta}, \alpha \to q_6^{\beta}, \alpha, +1$; $q_6^{\beta}, \Lambda \to q_2^{\beta}, \beta, -1$. Здесь α — любой непустой символ, β — любой символ.

604. **(а)** По очереди смещать один символ с левой стороны на одну позицию влево и стирать один символ справа.

(б) По очереди переносить разряды второго слова, добавляя их к первому, стирать перенесённые разряды второго слова, переносить просуммированные разряды первого слова вправо, за второе слово.

605. В начале работы раздвинуть ячейки, вставив на чётных позициях пустой символ. Далее вернуться в первую ячейку и перейти в состояние q_0^+. Для всех состояний добавляем команду $q^-, \# \to q^+, \#, +1$ (нулевая ячейка имеет чётный номер, поэтому там оказываемся в «отрицательных» состояниях). Вместо команды $q, a \to p, b, 0$ вставляем $q^+, a \to p^+, b, 0$; $q^-, a \to p^-, b, 0$. Вместо команды $q, a \to p, b, +1$ вставляем $q^+, a \to p_1^+, b, +1$; $p_1^+, c \to p^+, c, +1$; $q^-, a \to p_1^-, b, -1$; $p_1^-, c \to p^-, c, -1$ для всех символов $c \neq \#$. Вместо команды $q, a \to p, b, -1$ вставляем $q^+, a \to p_2^+, b, -1$; $p_2^+, c \to p^+, c, -1$; $p_2^+, \# \to p_2^-, \#, +1$; $q^-, a \to p_2^-, b, +1$; $p_2^-, c \to p^-, c, +1$ для всех символов $c \neq \#$. В конце нужно «сжать» ленту, убрав пустые символы в чётных ячейках.

606. Считаем, что $\mathfrak{M}$ имеет стандартную заключительную конфигурацию, $a_0 = \Lambda$ и $a_1 = |$. Работа новой машины $\mathfrak{N}$ происходит в три этапа.

На первом машина $\mathfrak{N}$ вставляет между любыми двумя символами входа блок Λ^{k-1}. На втором происходит моделирование работы старой машины. Сначала для каждой пары $q \in Q$, $a_i \in \Sigma$ в программу $\mathfrak{N}$ добавляются команды (с помощью α обозначен любой символ: Λ или $|$, i, j принимают всевозможные значения от 0 до $k-1$, u принимает всевозможные значения от 0 до k) $q, \alpha \to q_0^{0,0}, \alpha, 0$; $q_0^{i,j}, | \to q_0^{i+1,j+1}, |, +1$; $q_0^{i,j}, \Lambda \to q_0^{i,j+1}, \Lambda, +1$; $q_0^{u,k}, \alpha \to q_1^{u,k-1}, \alpha, -1$; $q_1^{u,j+1}, \alpha \to q^{u,j}, \Lambda, -1$; $q_1^{u,0}, \alpha \to q_2^{u,0}, \Lambda, 0$. Теперь u — номер символа, который видела головка старой машины, а все k ячеек, соответствующих символу старой машины, содержат Λ. Если команды вида $q, a_u \to \ldots$ в программе машины $\mathfrak{M}$ не было, то происходит переход к третьему этапу. Пусть в программе $\mathfrak{M}$ есть команда $q, a_u \to p, a_v, s$. Тогда в программу $\mathfrak{N}$ вставляем команды $q_2^{u,0}, \Lambda \to q_3^{v,0}, \Lambda, 0$; $q_3^{i+1,j}, \Lambda \to q_3^{i,j+1}, |, +1$. Если $s = +1$, то добавляем $q_3^{0,j}, \Lambda \to q_3^{0,j+1}, \Lambda, +1$; $q_3^{0,k}, \alpha \to p, \alpha, 0$. Если $s = 0$, то добавляем $q_3^{0,j+1}, \alpha \to q_3^{0,j}, \alpha, -1$; $q_3^{0,0}, \alpha \to p, \alpha, 0$. Если $s = -1$, то добавляем $q_3^{0,j+1}, \alpha \to q_3^{0,j}, \alpha, -1$; $q_3^{0,0}, \alpha \to q_4^k, \alpha, 0$; $q_4^{j+1}, \alpha \to q_4^j, \alpha, -1$; $q_4^0, \alpha \to p, \alpha, 0$. На третьем этапе машина удаляет между любыми двумя символами блок Λ^{k-1}, чтобы получить такой же результат, как и исходная машина.

607. Использовать многоэтажные символы вида $\left[\!\left[\begin{matrix} a_1^{\varepsilon_1} \\ \ldots \\ a_k^{\varepsilon_k} \end{matrix}\right]\!\right]$ Здесь a_i — символ, записанный на i-й ленте, а ε_i равно 1, если головка на i-й ленте находится в данной ячейке, или 0 в противном случае. Выполнение каждой команды k-ленточной машины заключается в следующем: сначала пройти по всей ленте, запомнив символы, наблюдаемые каждой из головок, после этого становится известно, какую команду k-ленточной машины нужно выполнить, затем нужно снова пройти по всей ленте, заменив символы в ячейках, где стояли головки и сдвинув эти головки в нужном направлении. Таким образом, каждый шаг работы k-ленточной машины требует $O(L)$ шагов работы обычной машины, где L — количество занятых ячеек. Следовательно, общее время работы обычной машины будет $O(TL)$, где T — время работы k-ленточной машины. Если k-ленточная машины просматривает весь вход, то $L \leqslant T$ и время работы обычной машины будет $O(T^2)$.

608. Сначала сдвигать синхронно обе головки вправо, копируя содержимое первой ленты на вторую. Затем переместить головку на второй ленте к началу скопированного слова. Наконец, сдвигать синхронно первую головку влево, вторую — вправо, проверяя, что символы одинаковы и стирая содержимое первой ленты. Если все символы совпали, записать на первой ленте 1, иначе — 0.

Тезис Тьюринга-Чёрча и неразрешимые проблемы

609. $(|Q| \cdot |\Sigma| \cdot 3 + 1)^{|Q| \cdot |\Sigma|}$.

610. **(а)** $\mathrm{bb}(1) = 1$. Если программа не содержит ни одной команды, то на пустой ленте не будет написано ни одного символа. Возможны шесть команд: $q_0, \Lambda \to q_0, |, \pm 1$; $q_0, \Lambda \to q_0, |, 0$; $q_0, \Lambda \to q_0, \Lambda, s$; $q_0, | \to \ldots$. В случае последней команды машина сразу остановится, не написав ни одной палочки. Для первой и третьей команды машина зациклится.

(б) $q_0, \Lambda \to q_1, |, +1$; $q_1, \Lambda \to q_0, |, -1$; $q_0, | \to q_1, |, -1$.

611. $A \leqslant_m A$ посредством тождественной функции id_1^1, если $A \leqslant_m B$ и $B \leqslant_m C$ посредством f и g соответственно, то $A \leqslant_m C$ посредством $f \circ g$.

612. Свести HALT к TEST следующим образом. Построить $\mathfrak{N}$, добавив в программу машины $\mathfrak{M}$ новое состояние q_f и всевозможные команды $q, a \to q_f, a, 0$, если команд вида $q, a \to \ldots$ не было. Тогда $(\pi(\mathfrak{M}), x) \in \mathrm{HALT}$ тогда и только тогда, когда $(\pi(\mathfrak{N}), x, q_f) \in \mathrm{TEST}$.

613. Свести TOTAL к ZERO с помощью такой модификации программы: построить машину со стандартной заключительной конфигурацией, а после остановки стереть весь вход.

614. Модифицировать машину как в предыдущей задаче, а в качестве второй — взять машину, просто стирающую весь вход.

615. Модифицировать машину $\mathfrak{N}$, для которой требуется определить самоприменимость, следующим образом. Новая машина $\mathfrak{M}_{\mathfrak{N}}$ на пустом слове возвращает результат a. В противном случае она стирает вход, затем записывает на ленту $\pi(\mathfrak{N})$, запускает $\mathfrak{N}'$ (со стандартной заключительной конфигурацией) и, наконец, стирает весь результат. Тогда, если $\mathfrak{N}$ не самоприменима, то единственный результат, который может вернуть $\mathfrak{M}_{\mathfrak{N}}$ — это a. В противном случае $\mathfrak{M}_{\mathfrak{N}}$ может вернуть два разных результата: a и пустое слово ε.

616. Модифицировать машину $\mathfrak{N}$, для которой требуется определить самоприменимость, следующим образом. Новая машина $\mathfrak{M}_{\mathfrak{N}}$ для $x \leqslant 1$ возвращает сам x. В противном случае она дописывает справа на ленту $\#\pi(\mathfrak{N})$, запускает $\mathfrak{N}'$ (одностороннюю со стандартной заключительной конфигурацией) и, наконец, стирает всё содержимое ленты и записывает 0. Тогда, если $\mathfrak{N}$ не самоприменима, то $\mathfrak{M}_{\mathfrak{N}}(x)$ определена только для $x = 0, 1$ и возрастает. В противном случае $\mathfrak{M}_{\mathfrak{N}}(2) < \mathfrak{M}_{\mathfrak{N}}(1)$.

617. **(а)** $I_{A \cap B}(x) = I_A(x) \cdot I_B(x)$,

(б) $I_{A \cup B}(x) = \mathrm{test}(I_A(x) + I_B(x))$,

(в) $I_{A \times B}(x, y) = I_A(x) \cdot I_B(y)$.

618. Для сложения: для выяснения $x \in A + B$ перебрать все $y = 0, \ldots, x$ и проверить $y \in A$, $x - y \in B$, если хотя бы в одном случае выполнено, ответ «да». Для умножения: $0 \in A \cdot B$ тогда и только тогда, когда $0 \in A$ или $0 \in B$. Для выяснения $x \in A \cdot B$ при $x > 0$ перебрать все $y = 1, \ldots, x$ и проверить $y \in A$, y делит x и $x/y \in B$, если хотя бы в одном случае выполнено, ответ «да».

619. Для конкатенации — аналогично предыдущей задаче. При проверке $x \in L^*$ в случае $x = \varepsilon$ ответ — «да». Если $x \neq \varepsilon$, то перебирать все y — непустые префиксы x — и проверять $y \in L$ и $z \in L^*$ рекурсивно ($x = yz$).

620. $F(x) = \mathrm{if}(I_A(x), g(x), h(x))$.

621. Воспользоваться универсальной машиной Тьюринга, отсчитывать количество проделанных шагов.

622. Спроектировать ограниченную проблему остановки в HALT, убрав t.

623. В силу конечности количества машин Тьюринга с не более чем n состояниями (задача 609) лишь для конечного количества слов w может быть выполнено $\mathrm{arc}(w) \leqslant n$. Пусть машина $\mathfrak{M}_k$ записывает 1 и вызывает k раз машину, вычисляющую функцию 2^x (все числа в унарной записи). Тогда результатом будет число $h(k) = 2^{2^{\cdots^{2}}}$ в унарной записи (k возведений в степень). Эта машина имеет $a + bk$ состояний для некоторых констант a и b. При достаточно больших k оно будет больше количества машин с не более чем $a + bk$ состояниями. Следовательно, машины с не более чем $a + bk$ состояниями не могут порождать все числа от 0 до $h(k)$ и есть число $x < h(k)$, для порождения которого требуется более $a + bk$ состояний. Получаем $\mathrm{arc}(|^{x+1}) > a + bk \geqslant \mathrm{arc}(|^{h(k)+1})$.

624. Предположим, что существует неограниченная функция f такая, что $f(x) \leqslant \mathrm{arc}(x)$ для всех x. Пусть машина $\mathfrak{M}_f$ по входу n перебирает каким-либо образом все слова, пока не будет найдено слово w такое, что $f(w) > n$. Это слово и будет результатом работы $\mathfrak{M}_f$ на n. Пусть машина $\mathfrak{M}_k$ записывает на ленту 1, затем k раз вызывает машину, удваивающую число, и наконец, запускает машину $\mathfrak{M}_f$. Перед запуском $\mathfrak{M}_f$ на ленте записано число 2^k. Следовательно, результатом $\mathfrak{M}_k$ будет некоторое слово w такое, что $f(w) \geqslant 2^k$. С другой стороны, каждая машина $\mathfrak{M}_k$ имеет $a + bk$ состояний для некоторых констант a и b, не зависящих от k. Так как функция 2^k растёт быстрее $a+bk$, то найдётся такое k_0, что $2^{k_0} > a+bk_0$. Но тогда получаем противоречие: результатом $\mathfrak{M}_k$ будет слово w, для которого $\mathrm{arc}(w) \geqslant f(w) \geqslant 2^k > a + bk$, но сама машина $\mathfrak{M}_{k_0}$ имеет $a + bk_0$ состояний, поэтому должно быть $\mathrm{arc}(w) \leqslant a + bk_0$.

625. Если бы $T(n)$ была вычислима, то функция «усердного бобра» $\mathrm{bb}(n)$ тоже была бы вычислима. Чтобы вычислить $\mathrm{bb}(n)$, нужно последовательно перебирать все машины Тьюринга с n состояниями (их конечно много, задача 621), запускать каждую из них на пустой ленте при помощи универсальной машины и выполнять не более $T(n)$ шагов. Если за это время машина не остановилась, то она никогда не остановится, следовательно, её потенциал равен нулю. Иначе подсчитываем количество занятых ячеек — потенциал машины.

626. С помощью универсальной машины Тьюринга моделируем работу $\mathfrak{M}$ на входе x, параллельно запоминая размер рабочей области. Если он превысил $S^*(x)$, значит машина зациклилась и ответ отрицательный. Количество конфигураций машины с рабочей областью размера $S^*(x)$ или меньше не превосходит $|Q| \cdot S^*(x) \cdot |\Sigma|^{S^*(x)}$. Следовательно, если за такое количество шагов машина не остановилась, то одна и та же конфигурация встретилась как минимум два раза, то есть машина зациклилась и ответ снова отрицательный. Если остановилась, то ответ, очевидно, положительный.

627. Если бы функция $S(\pi(\mathfrak{M}), x)$ была вычислима, то методом, описанным в предыдущей задаче, можно было бы решить проблему остановки HALT, что невозможно.

628. Если $A \vee B \leqslant_m C$ посредством о. р. ф. h, то $A \leqslant_m C$ посредством о. р. ф. $f(x) = h(2x)$: $x \in A \iff 2x \in A \vee B \iff h(2x) \in C$. Аналогично $B \leqslant_m C$ посредством о. р. ф. $g(x) = h(2x + 1)$. Если $A \leqslant_m C$ и $B \leqslant_m C$ с помощью о. р. ф. f и g соответственно, то $A \vee B \leqslant_m C$ с помощью о. р. ф. $h(x) = \mathrm{if}(\mathrm{mod}(x, 2), g((x-1)/2), f(x/2))$.

629. **(а)** Пусть $a \in A$, тогда $f(x) = \mathrm{if}(I_A(x), x, a)$.

(б) Если B разрешимо и A содержит a, то $A \cap B$ перечисляется с помощью о. р. ф. $g(x) = \mathrm{if}(I_B(f(x)), f(x), a)$.

(в) Множество $A_1 \cup A_2$ перечисляется с помощью о. р. ф. $g(x) = \mathrm{if}(\mathrm{mod}\,(x, 2), f_1((x-1)/2), f_2(x/2))$.

(г) $g = \mu\,\mathrm{eq}(\mathrm{id}_1^2, f(\mathrm{id}_2^2))$.

630. Если бы TOTAL перечислялось о. р. ф. f, то вычисление $\mathfrak{U}(f(x), x) \,\&\, a$ было бы реализуемо на машине Тьюринга $\mathfrak{M}$: продублировать x, с помощью односторонней машины вычислить $f(x)$ для правой копии, поменять местами x и $f(x)$, запустить универсальную машину $\mathfrak{U}$, после остановки приписать к результату символ a. Так как $f(x) \in$ TOTAL, то описанный алгоритм всегда останавливается, следовательно, $\pi(\mathfrak{M}) \in$ TOTAL и $f(x_0) = \pi(\mathfrak{M})$ для некоторого x_0. Тогда результатом $\mathfrak{M}(x_0)$ должно быть $\mathfrak{M}(x_0) \,\&\, a$, противоречие.

631. Допустим, что f можно дополнить до о. р. ф. g. Тогда $g(x) = \mathfrak{U}(c, x)$ для некоторого c. Тогда $\mathfrak{U}(c, c)$ определено, и, следовательно, $f(c)$ определено. Получаем $f(c) = \mathfrak{U}(c, c) + 1 = g(c) + 1$. Но так как $f(c)$ определено, то должно быть $f(c) = g(c)$, то есть $g(c) = g(c) + 1$, противоречие.

Указатель терминов

Литература

[1] Гаврилов Г. П. Задачи и упражнения по дискретной математике / Г. П. Гаврилов, А. А. Сапоженко. — 3-е изд., перераб. — М. : Физматлит, 2009. — 416 с.

[2] Дехтярь М. И. Лекции по дискретной математике. — М. : Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007. — 259 с.

[3] Дехтярь М. И. Лекции по дискретной математике : учебник / М. И. Дехтярь, С. М. Дудаков, Б. Н. Карлов. — Изд. 2-е, перераб. и доп. — Тверь : ТвГУ, 2019. — 512 с.

[4] Дехтярь М. И. Задачник по дискретной математике : учебное пособие / М. И. Дехтярь, Б. Н. Карлов. — Тверь : ТвГУ, 2013 — 168 с.

[5] Дудаков С. М. Математическое введение в информатику / С. М. Дудаков, Б. Н. Карлов. — Изд. 3-е, испр. и доп. — Тверь : ТвГУ, 2020. — 320 с.

[6] Ерусалимский Я. М. Дискретная математика : Теория, задачи, приложения. — Изд. 6-е. — М. : Вузовская книга, 2004. — 265 с.

[7] Кристофидес Н. Теория графов : алгоритмический подход. — М. : Мир, 1978. — 432 с.

[8] Лавров И. А. Задачи по теории множеств, математической логике и теории алгоритмов : учебник / И. А. Лавров, Л. Л. Максимова. — Изд. 5-е, испр. — М. : ФИЗМАТЛИТ, 2002. — 256 с.

[9] Мельников О. И. Занимательные задачи по теории графов. Учебно-методическое пособие. — Минск: НТООО «ТетраСистемс», 2001. — 144 с.

[10] Новиков Ф. А. Дискретная математика : для бакалавров и магистров. — Изд. 2-е. — М. [и др.] : Питер, 2013. — 399 с.

[11] Тишин В. В. Дискретная математика в примерах и задачах. — Изд. 2-е, испр. — СПб.: БХВ-Петербург, 2016. — 336 с.: ил.

[12] Трахтенброт Б. А. Сложность алгоритмов и вычислений. — Новосибирск : Изд. НГУ, 1967. — 258 с.

[13] Эвнин А. Ю. Задачник по дискретной математике. Более 400 задач с подробными решениями : учебное пособие. — Изд. 6-е, испр. и доп. — М.: Ленанд, 2016. — 272 с.

[14] Яблонский С. В. Введение в дискретную математику. — Изд. 5-е, стер. — М. : Высшая школа, 2008. — 384 с.

Учебное издание

$\boxed{\text{Дехтярь Михаил Иосифович}}$
Дудаков Сергей Михайлович
Карлов Борис Николаевич

Задачник по дискретной математике

Учебное пособие

Отпечатано с авторских оригиналов

Подписано в печать 18.08.2021
Усл. п. л. 21,39.
Тираж 10 экз. [электр.].
Заказ № 250 от 18.08.2021

Тверской государственный университет
Издательство Тверского государственного университета
Адрес: 170100, г. Тверь, Студенческий пер., 12, корпус Б
Тел.: (4822) 35–60–63