

ЭКВИВАЛЕНТНОСТЬ РЕСУРСОВ В СЕТЯХ ПЕТРИ



V. A. Bashkin, I. A. Lomazova

Resource equivalence in Petri nets

Moscow
Scientific World
2008

В. А. Башкин, И. А. Ломазова

Эквивалентность ресурсов в сетях Петри

Москва
Научный мир
2008

УДК 681.3

ББК 22.18

Б33

Б33 Башкин, В. А. Эквивалентность ресурсов в сетях Петри / В. А. Башкин,

И. А. Ломазова. М.: Научный мир, 2008. —208 с. **ISBN 978-5-91522-002-6**

В книге рассмотрены отношения эквивалентности на множестве ресурсов сети Петри. Представлено понятие подобия, соответствующее возможности замены в системе одного набора фишек (ресурса) на другой без ущерба для её наблюдаемого поведения в смысле бисимулярности. Вводятся и исследуются отношения эквивалентности, порождаемые подобием ресурсов. Приводятся алгоритмы построения множеств пар подобных ресурсов для обыкновенных сетей Петри и некоторых формализмов на их основе: сетей с невидимыми переходами, раскрашенных сетей и вложенных сетей Петри.

Книга предназначена для научных работников, преподавателей, аспирантов и студентов, интересующихся формальными моделями параллельных и распределенных систем.

Илл. 53. Библ. 74 назв.

*Публикуется при финансовой поддержке Российского
фонда фундаментальных исследований
(проект 07-01-07038)*

Bashkin, V. A. Resourceequivalencein Petrines / V. A. Bashkin, I. A. Lomazova.—Moscow:Scientific World,2008. —208 p.

Resources are defined as submultisets of Petri net markings. Two resources are called similar if replacing one of them by another in any marking doesn't change the Petri net's behavior (modulo bisimulation). The book presents a set of equality relations induced by resource similarity and some algorithms finding similar resources in ordinary Petri nets and other formal models: nets with silent moves, high-level and nested Petri nets.

The book is designed for scientists, professors and students interested in formal models of concurrent and distributed systems.

Pict. 53. Bibl. 74 titles.

ISBN 978-5-91522-002-6

Оглавление

Предисловие	6
Глава 1 Предварительные сведения	
1.1 Множества и отношения	11
1.2 Мультимножества	13
1.3 Системы помеченных переходов	15
1.4 Бисимуляции	19
1.5 Сети Петри	25
Глава 2 Подобие ресурсов	
2.1 Конечное представление отношений	35
2.1.1 Базисы отношений	35
2.1.2 Конечность базиса AT -замыкания	44
2.1.3 Свойства основного базиса	49
2.2 Подобие ресурсов	56
2.2.1 Определение	56
2.2.2 Свойства	60
2.2.3 Неразрешимость	65
Глава 3 Сужения подобия ресурсов	
3.1 Ограниченное подобие ресурсов	69
3.1.1 Определение и основные свойства	69

3.1.2	Ограниченное подобие как способ приближения полного подобия	71
3.2	Бисимуляция ресурсов	73
3.2.1	Определение	73
3.2.2	Слабое свойство переноса	78
3.2.3	Проверка бисимулярности отношения	80
3.2.4	Построение аппроксимации максимальной бисимуляции ресурсов	83
3.2.5	Гипотеза о неразрешимости	87
Глава 4 Расширения подобия ресурсов		
4.1	Условное подобие ресурсов	89
4.1.1	Определение условного подобия	89
4.1.2	Свойства условного подобия ресурсов	91
4.1.3	Полулинейность множества пар подобных ресурсов	93
4.2	Расслоенное подобие ресурсов	100
4.2.1	Расслоенная бисимуляция разметок	100
4.2.2	Определение расслоенного подобия ресурсов	101
4.2.3	Свойства расслоенного подобия ресурсов	103
4.3	Подобие обобщенных ресурсов	107
4.3.1	Обобщенные ресурсы сети Петри	108
4.3.2	Свойства подобия обобщенных ресурсов	111
4.3.3	Материальные и инструментальные ресурсы	116
Глава 5 Другие классы формальных моделей		
5.1	Сети Петри с невидимыми переходами	118
5.1.1	Определение сетей Петри с невидимыми переходами	119
5.1.2	Подобие и бисимуляция ресурсов в сетях с невидимыми переходами	121
5.1.3	Насыщенные сети Петри	124
5.1.4	τp -бисимуляция ресурсов	128
5.1.5	Алгоритм построения аппроксимации	134

5.2	Сети Петри высокого уровня	137
5.2.1	Раскрашенные сети Петри	139
5.2.2	Элементарные ресурсы	143
5.2.3	Подобие и бисимуляция ресурсов в раскрашенных сетях	144
5.2.4	Алгоритм построения аппроксимации	149
5.3	Вложенные сети Петри	152
5.3.1	Определение вложенных сетей Петри	152
5.3.2	Объектные ресурсы	162
5.3.3	Системные ресурсы	165
5.3.4	Системно-автономные ресурсы	174
5.3.5	Рекурсивные вложенные сети Петри	177
 Глава 6 Применение эквивалентностей ресурсов		
6.1	Редукция (оптимизация) модели	184
6.2	Адаптивное управление процессом	194
6.2.1	Управление “без потерь” на основе подобия обобщенных ресурсов	194
6.2.2	Управление в условиях ограниченного времени на основе расслоенного подобия	197
 Список литературы		198

Предисловие

В настоящее время большой и устойчивый интерес проявляется к средствам моделирования и анализа сложных параллельных и распределенных систем. Такими системами являются, например, вычислительные машины и комплексы с параллельной и распределенной архитектурой, параллельные программы и алгоритмы, протоколы взаимодействия (коммуникационные, верифицирующие), модели технологических и бизнес-процессов. Это обусловлено в первую очередь высоким риском возникновения ошибок на стадии проектирования таких систем и чрезвычайно высокой ценой проявления этих ошибок на стадии эксплуатации.

Для решения задач анализа и верификации в теории параллельных и распределенных вычислений в настоящее время предлагаются различные способы моделирования реальных систем. К числу наиболее известных формализмов можно отнести конечные автоматы, алгебры процессов, CCS Р. Милнера, языки трасс, а также различные их модификации, в том числе с добавлением конструкций времени и вероятности.

Сети Петри [60] давно зарекомендовали себя как удобный и наглядный, и в то же время математически строгий формализм для моделирования и анализа распределенных систем. Важным его достоинством наряду с простотой и наглядностью является разрешимость многих поведенческих свойств (см. обзоры [39, 40, 62, 64]). Сети Петри позволяют с достаточной степенью детали-

зации моделировать вычислительные процессы, протоколы взаимодействия, а также процессы управления в параллельных и распределенных системах. В них имеются простые конструкции для описания структур параллелизма: последовательная композиция, выбор, параллельное слияние.

Среди отечественных исследований по сетям Петри и спецификации и анализу распределенных систем следует отметить работы Н.А. Анисимова, О.Л. Бандман, И.Б. Вирбицкайте, В.В. Воеводина, Н.В. Евтушенко, Ю.Г. Карпова, В.Е. Котова, В.А. Непомнящего, Р.Л. Смелянского, В.А. Соколова, Л.А. Черкасовой.

Существует большое число классов формальных моделей, построенных на основе сетей Петри. Некоторые являются расширением класса обыкновенных сетей Петри и по выразительности приближаются к машинам Тьюринга, некоторые — сужением (иногда до конечных систем). Существуют сети Петри с ингибиторными дугами, с невидимыми переходами, сети со временем и вероятностью, объектно-ориентированные сети Петри [41, 64].

Образовался довольно обширный набор классов формальных систем с общим названием “сети Петри высокого уровня”, в которых тем или иным способом вводятся конструкции модульности и иерархичности [41, 42, 53, 66, 67].

В последнее время большое внимание уделяется разработке формализмов для представления при помощи сетей Петри мультиагентных систем со сложной динамической структурой (вплоть до рекурсивности). К формализмам такого рода можно отнести объектные сети Р. Фалька [70], схемы рекурсивно-параллельных программ О.Б. Кушнаренок и В.А. Соколова [23], рекурсивные сети С. Хаддада и Д. Пватрено [43]. В работах И.А. Ломазовой [14, 16] были определены вложенные сети Петри и рекурсивные вложенные сети Петри. В таких сетях некоторые фишки в свою очередь могут быть сетями Петри, что позволяет естественным образом моделировать динамические объекты в системе.

Понятие эквивалентности поведений — важнейшее понятие те-

ории формальных систем. Поведенческие эквивалентности позволяют сравнивать параллельные и распределенные системы с учетом тех или иных аспектов их функционирования, а также абстрагироваться от излишней информации. Эквивалентностные отношения используются также для сохраняющей поведение редукции систем и в процессе верификации, когда сравнивается ожидаемое и реальное поведения систем. Эквивалентные преобразования занимают важное место в теории автоматов, теории схем программ (см., например, работу Р.И. Подловченко [20]).

Бисимуляционная эквивалентность [59, 57] — фундаментальное понятие в теории параллельных и распределенных систем. Бисимуляция обладает четкой математической трактовкой и более тонко отслеживает ветвления в дереве срабатываний системы по сравнению с языковой эквивалентностью. Два состояния системы бисимулярны, если внешний наблюдатель по наблюдаемому поведению системы не может определить, с какого из этих двух состояний она начала работу.

Отношение бисимуляции может быть использовано для определения эквивалентности различных моделей (например, для выявления соответствия системы её спецификации). Кроме того, выявление сходных структур в множестве состояний позволяет существенно упрощать систему без изменения ее наблюдаемого поведения (бисимуляционная редукция). Проблема поиска эквивалентных состояний также важна для поддержки методологии адаптивного управления системой, согласно которой структура системы может изменяться непосредственно в ходе ее функционирования, например, в ответ на изменения внешних условий или же при возникновении каких-то внутрисистемных событий (болезнь сотрудника, отказ оборудования, внедрение новых элементов системы и т.п.).

Бисимуляционная эквивалентность изучалась для различных классов формальных моделей [37, 28, 46, 49, 48, 58, 68]. Был получен ряд результатов по ее разрешимости. В частности, бисимуля-

ция разрешима для всех классов моделей с конечным множеством состояний и неразрешима для многих классов с бесконечным множеством состояний.

В случае сетей Петри текущее состояние задается с помощью разметки — мультимножества фишек (маркеров) в позициях сети. Известно, что бисимуляция разметок в сетях Петри неразрешима [48, 51], поэтому важное значение приобретает задача построения более сильных эквивалентностей на множестве состояний сети Петри, сохраняющих бисимулярность и в то же время поддающихся анализу (эффективно вычислимым). Примером таких отношений могут служить введенные Ф. Шнобеленом, С. Аутоном и Н.С. Сидоровой отношения бисимуляции позиций и корректного слияния позиций [26, 21, 65].

В данной монографии представлено понятие подобия ресурсов сети Петри. Ресурсом называется мультимножество фишек, находящихся в позициях сети (то есть некая часть разметки). Два ресурса подобны, если, заменив в любой разметке сети один из них на другой, мы получим то же самое наблюдаемое поведение сети. Подобие ресурсов обладает естественной интерпретацией и позволяет выразить ряд важных свойств системы, в частности, эквивалентность двух различных ресурсов, избыточность ресурса, эквивалентность двух различных действий при условии наличия дополнительного ресурса и т.п. Нахождение подобных ресурсов может оказаться очень полезным для понимания характера моделируемого процесса и оптимизации ресурсных затрат.

Отношение подобия ресурсов сильнее отношения бисимуляции разметок, поэтому для него выполняется ряд конструктивных свойств. В частности, подобие обладает конечным базисом и может быть эффективно аппроксимировано.

В книге также рассмотрен ряд отношений, являющихся дальнейшим развитием понятия подобия ресурсов. В частности, это сужения подобия ресурсов, такие, как ограниченное подобие ресурсов и бисимуляция ресурсов; а также расширения подобия:

условное подобие, расслоенное подобие и подобие обобщенных ресурсов. Все эти отношения обладают естественной интерпретацией и могут быть представлены при помощи конечных базисов.

Полученные результаты рассмотрены также в применении к некоторым важным формализмам, основанным на обыкновенных сетях Петри: сетям с невидимыми переходами, сетям высокого уровня, вложенным сетям. Для всех этих формальных моделей конструктивные свойства подобия ресурсов сохраняются, хотя и в разной степени.

Глава 1

Предварительные сведения

1.1 Множества и отношения

Через Nat обозначим множество неотрицательных целых чисел.

Обозначение 1.1. Пусть X и Y — два множества. Будем использовать обозначения:

- $X \subseteq Y$, если X является подмножеством Y ;
- $X \subset Y$, если X является собственным подмножеством Y ;
- $|X|$ — мощность множества X ;
- $X \cup Y$ — объединение множеств X и Y ;
- $X \cap Y$ — пересечение множеств X и Y ;
- $X \times Y = \{(x, y) \mid x \in X \wedge y \in Y\}$ — декартово произведение множеств X и Y .

Определение 1.1. Пусть X и Y — некоторые множества. Отношение R между множествами X и Y — это подмножество множества $X \times Y$.

Будем говорить, что отношение R определено на множестве X , если R — это отношение между X и X .

Обозначение 1.2. Пусть X — некоторое множество, R — отношение, определенное на X . Будем использовать обозначения:

- $Id(X) = \{(x, x) \mid x \in X\}$ — отношение идентичности на X ;
- $R^{-1} = \{(y, x) \mid (x, y) \in R\}$ — отношение, обратное R ;
- $R_1 \circ R_2 = \{(x, y) \mid \exists z : (x, z) \in R_2 \wedge (z, y) \in R_1\}$ — композиция отношений R_1 и R_2 .

Отношение R на множестве X есть отношение эквивалентности, если оно рефлексивно, симметрично и транзитивно.

Обозначение 1.3. Для стандартных отношений на X будем использовать следующие обозначения:

- R^k , где $k \in Nat$, задается рекурсивным определением:
 $R^0 = Id(X)$,
 для $k \geq 1$, $R^k = R^{k-1} \circ R$.
- $R^+ = R^1 \cup R^2 \cup \dots$ — транзитивное замыкание R ;
- $R^* = Id(X) \cup R^+$ — рефлексивное транзитивное замыкание;
- $\tilde{R}$ — рефлексивно-симметрично-транзитивное замыкание R .
 Очевидно, что это — наименьшее отношение эквивалентности на X , содержащее R .

Определение 1.2. Пусть A — некоторое множество. Конечной последовательностью на A будем называть отображение множества $\{1, 2, \dots, n\}$ в A . Отображение $\varepsilon : \emptyset \rightarrow A$ будем называть пустой последовательностью.

Пусть $\sigma : 1, \dots, n \rightarrow A : a_1 \dots a_n$ — конечная последовательность, тогда ее длину n будем обозначать $|\sigma|$. Длина пустой последовательности ϵ равна 0.

Проекцией последовательности σ на множество $A' \subseteq A$ будем называть ее подпоследовательность $s|_{A'}$, состоящую из тех и только тех элементов последовательности, которые входят в A' .

1.2 Мультимножества

Мультимножество является естественным обобщением множества. В мультимножестве один объект может находиться в нескольких экземплярах.

Пусть X — непустое множество.

Определение 1.3. *Мультимножеством* M над множеством X называется функция $M : X \rightarrow \text{Nat}$.

Мощность мультимножества $|M| = \sum_{x \in X} M(x)$.

Числа $\{M(x) \mid x \in X\}$ называются коэффициентами мультимножества, коэффициент $M(x)$ определяет число экземпляров элемента x в M . Если $\forall x \in X M(x) \leq 1$, то M является обычным множеством.

Мультимножество M *конечно*, если конечно множество

$$\{x \in X \mid M(x) > 0\}.$$

Множество всех конечных мультимножеств над данным множеством X обозначается как $\mathcal{M}(X)$.

Операции и отношения теории множеств естественно расширяются на конечные мультимножества.

Определение 1.4. Пусть $M_1, M_2, M_3 \in \mathcal{M}(X)$. Полагаем:

- $M_1 = M_2 \Leftrightarrow \forall x \in X M_1(x) = M_2(x)$ — отношение равенства;

- $M_1 \subseteq M_2 \Leftrightarrow \forall x \in X \ M_1(x) \leq M_2(x)$ — отношение включения;
- $M_1 \subset M_2 \Leftrightarrow M_1 \subseteq M_2 \wedge \exists x \in X \ M_1(x) < M_2(x)$ — отношение строгого включения;
- $M_1 = M_2 + M_3 \Leftrightarrow \forall x \in X \ M_1(x) = M_2(x) + M_3(x)$ — операция сложения двух мультимножеств;
- $M_1 = M_2 \cap M_3 \Leftrightarrow \forall x \in X \ M_1(x) = \min(M_2(x), M_3(x))$ — операция пересечения двух мультимножеств;
- $M_1 = M_2 - M_3 \Leftrightarrow \forall x \in X \ M_1(x) = M_2(x) \ominus M_3(x)$ — разность двух мультимножеств (где $\ominus$ — вычитание до нуля);
- $M_1 = kM_2, k \in \text{Nat} \Leftrightarrow \forall x \in X \ M_1(x) = kM_2(x)$ — операция умножения мультимножества на скаляр.

Пример 1.1. Рассмотрим M_1 и M_2 — мультимножества над множеством $X = \{a, b, c\}$, такие, что $M_1 = \{a, a, b\}$, $M_2 = \{b, c\}$.

Выполняется:

$$M_1(a) = 2, \ M_1(b) = 1, \ M_1(c) = 0;$$

$$M_1 \not\subseteq M_2, \ M_1 \not\supset M_2;$$

$$M_1 + M_2 = \{a, a, b, b, c\};$$

$$M_1 \cap M_2 = \{b\};$$

$$M_1 - M_2 = \{a, a\};$$

$$2M_1 = \{a, a, a, a, b, b\}.$$

Одним из способов записи мультимножеств являются вектора над Nat . При этом различным координатам вектора сопоставляются различные элементы X .

Пример 1.2. Мультимножества M_1 и M_2 из примера 1.1 могут быть записаны как

$$M_1 = (2, 1, 0), \quad M_2 = (0, 1, 1).$$

Утверждение 1.1. Множество $\mathcal{M}(X)$ мультимножеств над X изоморфно множеству $\text{Nat}^{|X|}$ векторов длины $|X|$ с целочисленными неотрицательными коэффициентами.

Доказательство. Непосредственно из определений. $\square$

1.3 Системы помеченных переходов

Системы помеченных переходов — одна из наиболее распространенных моделей для описания поведения систем.

Система переходов — это помеченный ориентированный граф, описывающий все возможные состояния моделируемой системы. При этом одна из вершин графа соответствует начальному состоянию системы, а дуги — возможным переходам системы из одного состояния в другое.

Определение 1.5. Система помеченных переходов есть набор $LTS = (S, Act, \rightarrow, s_0)$, где

- S — множество состояний с элементами $s_0, s_1, s_2, \dots$;
- Act — некоторый алфавит (множество имен действий);
- $\rightarrow \subseteq (S \times Act \times S)$ — отношение переходов между состояниями (с пометками из Act);
- $s_0 \in S$ — выделенное состояние, называемое начальным состоянием системы переходов.

Переход (s, a, s') из $\rightarrow$ обычно обозначается как $s \xrightarrow{a} s'$, что означает, что переход с меткой a переводит систему из состояния

s в состояние s' . Состояние s' в этом случае называется *последующим* для s , а состояние s — *предыдущим* для s' . Состояния, не имеющие последующих состояний называются *финальными*. Если некоторый переход переводит состояние s в состояние s' , то пишем $s \rightarrow s'$. Через $Succ(s)$ будем обозначать множество последующих состояний для s , через $Pred(s)$ — множество его предыдущих состояний. Мы рассматриваем только системы переходов с *конечным ветвлением* (finitely branching), то есть такие, в которых для любого s множество $Succ(s)$ конечно.

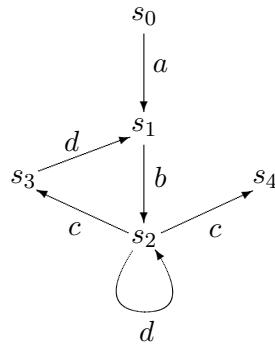
Бесконечность (в общем случае) множества S позволяет использовать помеченные системы переходов для моделирования любых классов систем с конечным ветвлением (даже универсальных, таких, как машины Тьюринга). Фактически, LTS — это формализованный способ записи всех возможных вариантов функционирования системы. Простота записи и универсальность моделирования делает системы помеченных переходов базовым языком для представления различных поведенческих свойств.

Определение 1.6. *Последовательное исполнение* для LTS есть конечная или бесконечная цепочка переходов $s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \dots$, где s_0 — начальное состояние системы. Каждому исполнению LTS соответствует некоторая строка в алфавите Act , составленная из меток сработавших переходов, называемая *распознанной строкой* или *трассой* LTS .

Запись $s \xrightarrow{*} s'$ означает, что имеется конечная последовательность переходов, переводящая состояние s в состояние s' .

Графически система переходов LTS изображается как помеченный ориентированный граф, в котором вершинами являются элементы множества состояний S , а дуги определяются отношением переходов так, что дуга, помеченная a , соединяет вершину s с вершиной s' в том и только том случае, когда $s \xrightarrow{a} s'$.

Каждому последовательному исполнению в LTS соответствует ориентированный путь с началом в вершине s_0 в графе LTS .

Рис. 1.1: Пример LTS

Одним из важнейших инструментов анализа динамики функционирования систем переходов является эквивалентность языков (трасс) [47].

Определение 1.7. Языком системы помеченных переходов LTS (языком, *распознаваемым* системой LTS) называется множество строк из алфавита Act , соответствующих всем последовательным исполнениям для LTS .

Языковая эквивалентность учитывает так называемую *интерливинговую* (последовательную) операционную семантику. Другими словами, анализируются “моментальные проекции” поведения системы, представляющие строки в некотором алфавите, без учета ветвлений.

Пример 1.3. Рассмотрим системы переходов, изображенные на рисунке 1.2. Здесь моделируются автоматы, продающие кофе и чай. Множество Act соответствует действиям автомата, действия клиента выражаются в выборе возможных сценариев действия автомата (выборе пути в графе).

В первом автомате клиент сначала опускает монету (действие “монета” — получение автоматом монеты), а затем выбирает либо кофе, либо чай — автомат отвечает действиями “кофе” или “чай”

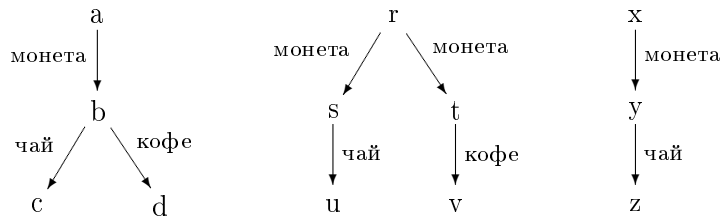


Рис. 1.2: Системы помеченных переходов

соответственно. Во втором автомате выбор кофе/чай происходит уже в самом начале, до опускания монеты. Далее от клиента уже ничего не зависит — автомат отрабатывает либо программу “монета кофе”, либо “монета чай”, не задавая дополнительных вопросов.

Языком и для первой и для второй системы является множество

$$\{\text{монета}, \text{монета чай}, \text{монета кофе}\},$$

то есть эти системы эквивалентны с точки зрения языковой эквивалентности. В то же время очевидно, что системы все-таки существенно различны. Если в первой после срабатывания “монета” можно выбрать и “чай”, и “кофе”, то во второй реальный выбор уже произошел, и изменить что-либо клиент уже не в состоянии.

Операционная семантика, учитывающая ветвления, называется семантикой *ветвящегося времени* (branching time). Если в последовательной семантике система полностью описывается распознаваемым языком (множеством трасс), то в семантике ветвящегося времени рассматривается полный граф срабатываний.

Для анализа динамики функционирования систем в семантике ветвящегося времени используется понятие бисимуляций.

1.4 Бисимуляции

Понятие бисимуляционной эквивалентности было введено Р. Милнером [57] и Д. Парком [59] и является в настоящее время классическим средством анализа систем, в том числе и систем с бесконечным числом состояний. Интуитивно, две системы бисимуляционно-эквивалентны, если они могут имитировать друг друга.

Бисимуляцию определяют с помощью так называемого свойства переноса:

Определение 1.8. Пусть $R \subseteq S \times S$ — отношение на множестве состояний системы помеченных переходов. Отношение R обладает *свойством переноса*, если для любой пары $(s, t) \in R$ и любого перехода $s \xrightarrow{a} s'$ найдется имитирующий переход $t \xrightarrow{a} t'$, такой что $(s', t') \in R$.

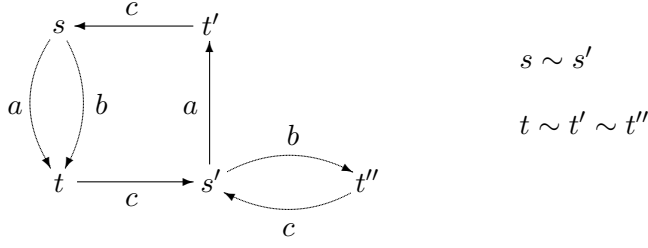
Свойство переноса можно проиллюстрировать диаграммой:

$$\begin{array}{ccc} s & R & t \\ a \downarrow & & \downarrow (\exists) a \\ s' & R & t' \end{array}$$

Определение 1.9. Отношение $R \subseteq S \times S$ на множестве состояний системы помеченных переходов называется *отношением бисимуляции*, если R и R^{-1} обладают свойством переноса.

Простейшими примерами бисимуляций являются отношение идентичности $Id(S)$ и пустое отношение.

Определение 1.10. Состояния s и t системы помеченных переходов называются *бисимуляционно эквивалентными* (или *бисимулярными*), что обозначается как $s \sim t$, если существует отношение бисимуляции R , такое, что $(s, t) \in R$.

Рис. 1.3: Бисимулярные состояния в *LTS*.

Пример бисимулярных состояний приведен на рисунке 1.3. У изображенной системы помеченных переходов бисимулярны состояния s и s' , а также t , t' и t'' .

Кроме свойства переноса, существует ещё один классический способ определения бисиммуляции — с использованием так называемых бисимуляционных игр.

В качестве “игровой доски” берется система помеченных переходов, в которой выделяются два состояния — E_0 и F_0 (в этом случае для игры используется обозначение $G(E_0, F_0)$). В игре участвуют два игрока, Алиса (подразумевается “Attacker”) и Боб (“Bisimulator”), которые являются наблюдателями, выбирающими переходы системы. Алиса стремится доказать, что состояния E_0 и F_0 в некотором смысле “различны”, Боб — что они в том же самом смысле “эквивалентны”. Результатом игры является конечная или бесконечная последовательность вида

$$(E_0, F_0), (E_1, F_1), \dots, (E_k, F_k), \dots, \text{ где } E_i, F_i \in S,$$

причем каждая следующая пара последовательности получена из предыдущей по правилу:

1. Алиса выбирает срабатывание (дугу в графе *LTS*) $E_i \xrightarrow{a} E_{i+1}$ или $F_i \xrightarrow{a} F_{i+1}$.

2. Боб выбирает некоторое имитирующее срабатывание $F_i \xrightarrow{a} F_{i+1}$ или $E_i \xrightarrow{a} E_{i+1}$ (обязательно с той же меткой a , что и у срабатывания, выбранного Алисой).

Алиса выигрывает игру, если на некотором ходе Боб не может ответить на ее ход своим (имитирующих переходов не осталось); у Боба два варианта выигрыша:

1. Алиса попала в тупик и не может сделать ни одного хода (из вершин E_i и F_i не ведет ни одной дуги).
2. Игра бесконечна, то есть последовательность ходов не закончилась ни победой Алисы, ни первым вариантом победы Боба.

Защищая свой тезис, Алиса должна выбирать срабатывания таким образом, чтобы у Боба не осталось возможности найти имитирующий переход. Боб, в свою очередь, должен отвечать на ходы Алисы так, чтобы требуемая эквивалентность состояний поддерживалась (или чтобы Алиса первой попала в тупик).

Стратегия игрока — это набор правил, которые позволяют определить следующий ход в зависимости от того, что произошло перед этим. Стратегия называется выигрывающей, если игрок побеждает в каждой игре при использовании этой стратегии (независимо от действий противника).

Определение 1.11. Состояния E_0 и F_0 называются *бисимулярными*, если у Боба существует выигрывающая стратегия для игры $G(E_0, F_0)$.

Пример 1.4. Рассмотрим игру $G(s, s')$ в системе переходов, изображенной на рисунке 1.3. У Боба существует очевидная выигрывающая стратегия — отвечать на ходы Алисы ходами с той же меткой срабатывания. В состояниях s и s' возможны срабатывания с метками a и b (причем только по одному на каждую метку!), в состояниях t , t' и t'' — только единственное срабатывание

с меткой s . Кроме того, как легко заметить, при любом последовательном исполнении состояния видов s^* и t^* чередуются.

Утверждение 1.2. *Определения бисимулярности 1.10 и 1.11 эквивалентны.*

Основные свойства бисимуляции:

Утверждение 1.3. *Пусть $R, R_1, R_2 \subseteq S \times S$ — отношения бисимуляции на множестве состояний системы помеченных переходов. Тогда:*

1. *Отношение $Id(S)$ есть бисимуляция.*
2. *Отношение R^{-1} есть бисимуляция.*
3. *Отношение $R_1 \cup R_2$ есть бисимуляция.*
4. *Отношение $\tilde{R}$ есть бисимуляция.*
5. *Отношение $R_2 \circ R_1$ есть бисимуляция.*
6. *Отношение $\sim =_{def} \bigcup \{R \mid R \subseteq S \times S \text{ — бисимуляция}\}$ есть наибольшая бисимуляция на множестве S (относительно вложения).*
7. *Отношение $\sim$ является отношением эквивалентности на множестве S .*

Доказательство. Свойства 1, 2 и 3 непосредственно следуют из определения бисимуляции.

Свойство 5: Покажем, что $R_2 \circ R_1$ обладает свойством переноса, и, следовательно, является бисимуляцией.

Пусть $(s_1, s_3) \in R_2 \circ R_1$, тогда существует состояние s_2 такое, что выполняется $(s_1, s_2) \in R_1$ и $(s_2, s_3) \in R_2$.

Рассмотрим шаг $s_1 \xrightarrow{a} s'_1$. Тогда существует шаг $s_2 \xrightarrow{a} s'_2$, такой что $(s'_1, s'_2) \in R_1$, и, следовательно, существует шаг $s_2 \xrightarrow{a} s'_2$, такой

что $(s'_1, s'_2) \in R_2$, то есть $(s'_1, s'_3) \in R_2 \circ R_1$. Аналогично можно показать, что отношение $R_2 \circ R_1$ обладает свойством переноса в обратном направлении. Таким образом, $R_2 \circ R_1$ — бисимуляция.

Свойство 4: Поскольку свойства 2 и 3 могут быть обобщены для объединения (композиции) бесконечного числа множеств, а

$$\tilde{R} = (R \cup R^{-1})^* = Id(S) \cup (R \cup R^{-1}) \cup (R \cup R^{-1})^2 \cup \dots \cup (R \cup R^{-1})^k \cup \dots,$$

то очевидно, что $\tilde{R}$ — бисимуляция.

Свойство 6 является очевидным следствием свойства 3, а свойство 7 — следствием свойств 3 и 4. $\square$

Бисимуляция — достаточно тонкая эквивалентность на множестве состояний, адекватно отражающая свойства системы в семантике ветвящегося времени. Однако, в силу своей универсальности, для многих классов систем отношение бисимуляции неразрешимо, то есть не существует алгоритма, отвечающего на вопрос, являются ли данные два состояния бисимулярными или нет.

Бисимуляция разрешима для всех классов систем с конечным числом состояний (конечных автоматов), так как в них для проверки бисимулярности достаточно просто перебрать множество состояний. Бисимуляция разрешима также для таких классов моделей с бесконечным множеством состояний, как:

- базовые параллельные процессы (BPP, Basic Parallel Processes) [37, 38, 45],
- базовые алгебры процессов (BPA, Basic Process Algebra) [28, 36],
- нормированные алгебры процессов (normed PA, normed Process Algebra) [46],
- автоматы с одним счетчиком (one-counter machines) [49],
- нормированные магазинные автоматы (normed PDA, normed Pushdown Automata) [68].

Бисимуляция неразрешима для следующих классов моделей (упорядочены по возрастанию выразительной мощности и в обратном хронологическом порядке по времени доказательства неразрешимости бисимуляции):

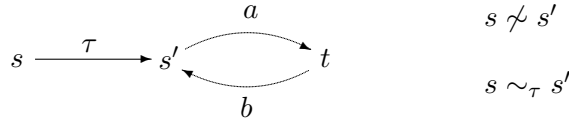
- автоматы мультимножеств (MSA, Multiset Automata) [58],
- помеченные сети Петри (labelled PN, labelled Petri Nets) [48],
- универсальные модели (машины Минского, машины Тьюринга, сети Петри с ингибиторными дугами, ...).

Класс MSA является подклассом помеченных сетей Петри и совпадает с классом параллельных магазинных автоматов (PPDA, Parallel Pushdown Automata).

Понятие слабой бисимуляции (τ -бисимуляции) было предложено Милнером для CCS (Calculus of Communicating Systems) [57]. Оно основывается на идее абстрагирования, позволяющего скрывать некоторые действия действия, которые называются τ -действиями. Таким образом, две системы считаются эквивалентными в слабом смысле, если они неразличимы для наблюдателя, который не видит τ -действия программы.

Добавим в алфавит имен срабатываний Act специальный символ τ , обозначающий внутренние срабатывания системы, невидимые для внешнего наблюдателя. Рассмотрим $LTS = (S, Act \cup \{\tau\}, \rightarrow, s_0)$ — систему помеченных переходов, в которой некоторые переходы помечены τ . Отношение τ -бисимуляции вводится следующим образом:

Определение 1.12. Пусть $R \subseteq S \times S$ — отношение на множестве состояний системы помеченных переходов LTS . Отношение R обладает *свойством τ -переноса*, если для любой пары $(s, t) \in R$ и любого перехода $s \rightarrow s'$ с меткой a найдется последовательность переходов $\sigma \in (\rightarrow)^*$, такая, что $t \xrightarrow{\sigma} t'$, $(s', t') \in R$, и, обозначив через $l(\sigma)$ последовательность меток переходов из σ , имеем $(l(\sigma))|_{Act} = a|_{Act}$.

Рис. 1.4: τ -бисимулярные состояния в LTS .

Определение 1.13. Отношение $R \subseteq S \times S$ на множестве состояний системы помеченных переходов называется *отношением τ -бисимуляции*, если R и R^{-1} обладают свойством τ -переноса.

Если состояния s и t системы помеченных переходов τ -бисимулярны, то это обозначается как $s \sim_\tau t$.

Отношение τ -бисимуляции слабее отношения обычной бисимуляции (см. пример на рисунке 1.4). Поэтому из неразрешимости бисимуляции следует и неразрешимость τ -бисимуляции. Однако обратное утверждение в общем случае неверно — существуют классы систем, для которых бисимуляция разрешима, а τ -бисимуляция — нет.

1.5 Сети Петри

Определение 1.14. *Обыкновенной сетью Петри* (ordinary Petri Net) называется набор $N = (P, T, F)$, где

P — конечное множество позиций;

T — конечное множество переходов, $P \cap T = \emptyset$;

$F : (P \times T) \cup (T \times P) \rightarrow \text{Nat}$ — функция инцидентности.

Графически сеть Петри изображается как двудольный ориентированный граф. Вершины-позиции изображаются кружками и характеризуют локальные состояния сети, вершины-переходы изображаются прямоугольниками и соответствуют действиям моделируемой системы. Дуги в графе соответствуют элементам F .

Определение 1.15. Пусть $N = (P, T, F)$ — обыкновенная сеть Петри. *Разметкой* (состоянием) сети N называется функция вида $M : P \rightarrow \text{Nat}$, сопоставляющая каждой позиции сети некоторое натуральное число или ноль.

Разметка может рассматриваться как мультимножество над множеством позиций сети.

Графически разметка изображается при помощи маркеров (называемых “фишками”) — черных точек внутри позиций. При разметке M в каждую позицию p помещается ровно $M(p)$ фишек. Если не хватает места на рисунке, то вместо точек рисуется число $M(p)$.

Определение 1.16. *Маркированной (размеченной) сетью Петри* называется пара (N, M_0) — сеть Петри N вместе с некоторой выделенной разметкой M_0 , называемой начальной разметкой.

Определим поведение сети Петри.

Определение 1.17. Пусть $N = (P, T, F)$ — обыкновенная сеть Петри.

- Для перехода $t \in T$ через $\bullet t$ и $t^\bullet$ обозначим мультимножества его входных и выходных позиций, такие, что

$$\forall p \in P \quad \bullet t(p) =_{\text{def}} F(p, t), \quad t^\bullet(p) =_{\text{def}} F(t, p).$$

- Переход $t \in T$ *готов к срабатыванию* при разметке M , если $\bullet t \subseteq M$ (все входные позиции содержат достаточное количество фишек).
- Готовый к срабатыванию переход t может *сработать*, порождая новую разметку $M' =_{\text{def}} M - \bullet t + t^\bullet$ (используется обозначение $M \xrightarrow{t} M'$).

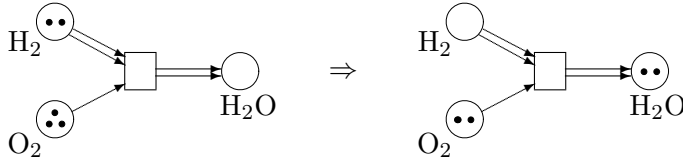


Рис. 1.5: Сеть Петри, моделирующая химическую реакцию

Фишки, находящиеся в той или иной позиции, моделируют наличие в системе того или иного ресурса, используемого или порождаемого при срабатывании переходов. Например, в сети на рисунке 1.5 фишки изображают молекулы водорода, кислорода и воды до и после химической реакции синтеза воды. Сама реакция моделируется переходом.

Определение 1.18. Пусть (N, M_0) — маркированная сеть Петри. Разметка M сети N называется *достижимой*, если существует последовательность переходов $\sigma \in T^*$, переводящая сеть из начального состояния M_0 в состояние M :

$$M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} \dots \xrightarrow{t_k} M, \quad t_1.t_2.\dots.t_k = \sigma,$$

что обозначается как $M_0 \xrightarrow{\sigma} M$ или просто как $M_0 \rightarrow M$.

Множество всех достижимых разметок сети обозначается как $R(N, M_0)$

Определение 1.19. Пусть (N, M_0) — маркированная сеть Петри. Позиция $p \in P$ называется *ограниченной*, если $\exists n \in \mathbb{N}$, такое, что $\forall M \in R(N, M_0)$ выполняется $M(p) \leq n$.

Маркированная сеть Петри называется *ограниченной*, если все её позиции ограничены. Очевидно, что множество состояний ограниченной сети Петри конечно.

На рисунках 1.6–1.8 приведены примеры того, как при помощи обыкновенных сетей Петри можно моделировать некоторые элементы реальных систем.

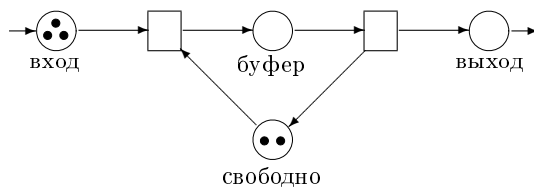


Рис. 1.6: Буфер объема 2

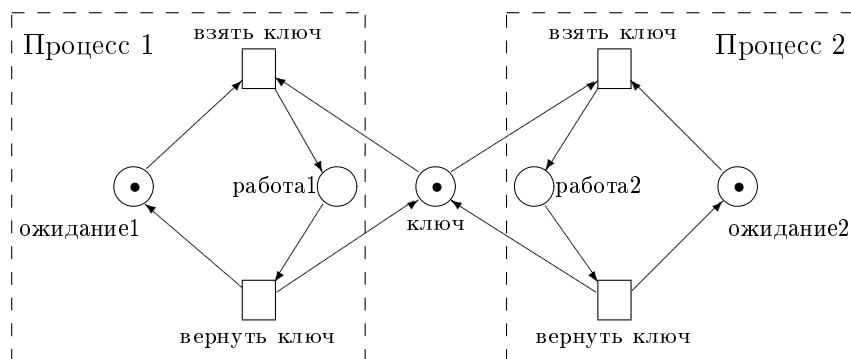


Рис. 1.7: Семафор (разделенный доступ к памяти)

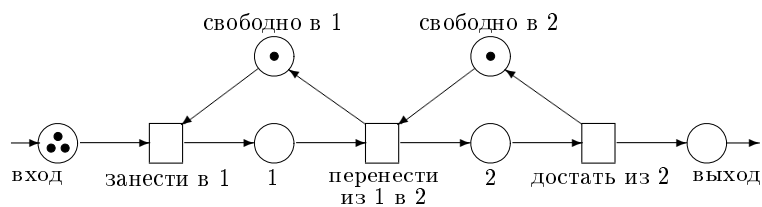


Рис. 1.8: Очередь (FIFO) из двух ячеек

На рисунке 1.6 изображен буфер ограниченной емкости. Независимо от поведения сети, количество фишек в позиции *буфер* не превысит двух. При этом количество фишек в служебной позиции *свободно* показывает, сколько еще “места” осталось в буфере. Предложенная структура сети универсальна — мы можем изменять моделируемую емкость буфера, просто изменяя количество фишек в начальной разметке позиции *свободно*. В начальном состоянии буфер пуст, на входе имеются три фишки “данных”.

Сеть 1.7 моделирует систему разделения доступа для двух различных процессов к общей ячейке памяти. Чтобы исключить одновременный доступ процессов к памяти (чтение и запись в данном случае не различаются), использовано классическое семафорное решение. Имеется один общий ключ (моделируется фишкой в позиции *ключ*), который дает право на доступ. Пока работающий процесс не вернул его на место, ожидающий процесс не может перейти в рабочее состояние. В начальном состоянии оба процесса находятся в стадии ожидания доступа.

На рисунке 1.8 показан элемент памяти FIFO (очередь), состоящий из двух ячеек (позиции 1 и 2). Позиция 1 моделирует первую ячейку FIFO (куда поступают данные), позиция 2 — последнюю (откуда они забираются). В начальном состоянии системы обе ячейки свободны, на входе имеются три фишки “данных”.

В последние годы всё более активно развивается ещё одна важная область применения сетей Петри — моделирование потоков работ (workflow). Потоки работ используются для формализации управления всевозможными технологическими процессами, бизнес-процессами, web-сервисами, распределенными вычислениями и т.д. В потоках работ возможны циклы, также возможно распараллеливание и синхронизация. Однако есть и ограничения, в частности, существуют одно начальное и одно конечное состояния, запрещены тупики.

Далее мы приводим определение специального подкласса сетей Петри, который используется при моделировании потоков ра-

бот (workflow) — это так называемые WF-сети [2].

Определение 1.20. Пусть $N = (P, T, F)$ — обыкновенная сеть Петри. Сеть N называется *WF-сетью* (сетью потока работ), если

1. в множестве P имеются две специальные позиции i и o , такие, что $\bullet i = o^\bullet = \emptyset$;
2. любой элемент множества $P \cup T$ лежит на пути из i в o .

Позиция i называется *начальной*, а позиция o — *финальной* позицией сети N . Начальная разметка сети потока работ состоит из одной фишки в позиции i .

Правильное завершение процесса, моделируемого сетью, гарантируется выполнением следующего свойства:

Определение 1.21. WF-сеть N называется *бездефектной*, если для любой достижимой разметки $M \in R(N, i)$ выполняется

1. $o \in R(N, M)$;
2. $o + M' \in R(N, M) \Rightarrow M' = \emptyset$.

Другими словами, из любого достижимого состояния бездефектной сети достижимо финальное состояние, при этом в финальном состоянии не может остаться никаких “лишних” фишек. Свойство бездефектности может быть эффективно проверено [2].

Легко заметить, что бездефектные сети потоков работ (WF-сети) являются подклассом ограниченных сетей Петри.

На рисунке 1.9 изображена сеть Петри, моделирующая процесс обработки страховых требований (этот пример взят из книги [2]). Прежде всего, требование регистрируется (задача *регистрация*), затем параллельно выполняются две задачи: клиенту посылается анкета (задача *послать_анкету*) и производится оценка требования (задача *оценка*). Если анкета возвращается в течение двух недель, то выполняется задача *обработка_анкеты*. Если в

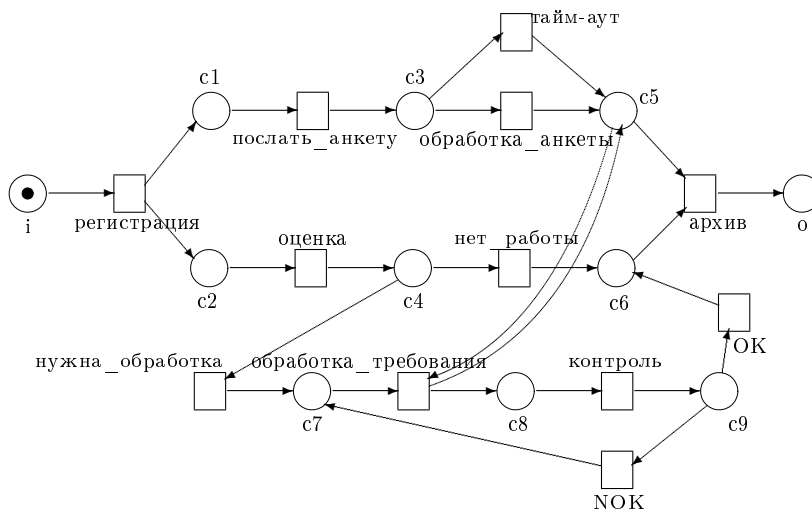


Рис. 1.9: Модель потока работ (workflow) бизнес-процесса обработки страховых требований

течение двух недель анкета не возвращена, то результат анкетирования игнорируется (задача *тайм-аут*). На основании результата оценки требование либо рассматривается, либо нет. Реальная работа с требованием (задача *обработка_требования*) откладывается до того момента, когда будет выполнено условие *c5*, т.е. либо анкета обработана, либо истекло время ее ожидания. Обработка требования контролируется с помощью задачи *контроль*. В конце концов, выполняется задача *архив*.

Для анализа поведения систем, моделируемых сетями Петри, необходимо формально сопоставить реальные объекты (действия) элементам модели. Срабатывания переходов в сети Петри соответствуют различным наблюдаемым событиям в моделируемой системе. Чтобы идентифицировать их, переходы помечаются метками из *Act*.

Определение 1.22. Помеченной сетью Петри называется набор $N = (P, T, F, l)$, где (P, T, F) — сеть Петри, $l : T \rightarrow Act$ — помечающая функция.

Внешний наблюдатель видит не сам переход t , а метку срабатывания $l(t)$, которой он помечен. Если два перехода помечены одной и той же меткой, то их срабатывания считаются идентичными. Таким образом, внешний наблюдатель рассматривает систему как “черный ящик”, порождающий те или иные события. При этом внутренняя структура состояний для него недоступна.

Введем определение отношения бисимуляции для обыкновенных сетей Петри. Так как состоянием сети Петри является её разметка, бисимуляция определяется на множестве разметок сети, то есть на множестве всех мультимножеств над множеством позиций.

Определение 1.23. Пусть $N = (P, T, F, l)$ — помеченная сеть Петри. Скажем, что отношение $R \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$ обладает *свойством переноса*, если для любой пары разметок $(M_1, M_2) \in R$ и для любого перехода $t \in T$, такого что $M_1 \xrightarrow{t} M'_1$, найдется имитирующий переход $u \in T$, такой что $l(t) = l(u)$, $M_2 \xrightarrow{u} M'_2$ и $(M'_1, M'_2) \in R$.

Свойство переноса может быть проиллюстрировано при помощи следующей диаграммы:

$$\begin{array}{ccc} M_1 & \sim & M_2 \\ \downarrow t & & \downarrow (\exists)u, \ l(u) = l(t) \\ M'_1 & \sim & M'_2 \end{array}$$

Определение 1.24. Если отношения R и R^{-1} обладают свойством переноса, то R называется *бисимуляцией разметок*.

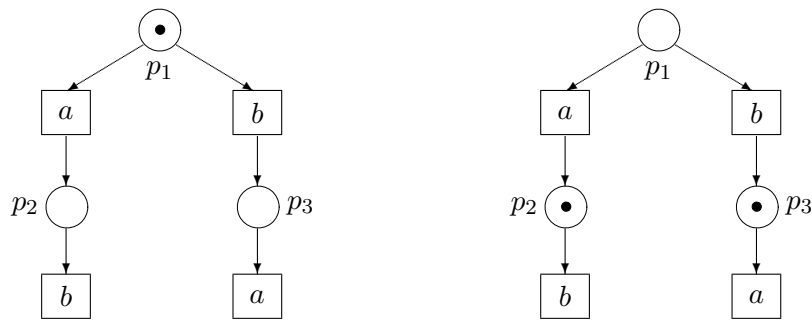


Рис. 1.10: Бисимулярные разметки в сети Петри

На рисунке 1.10 приведен пример бисимулярных разметок в сети Петри. Выполняется $(1, 0, 0) \sim (0, 1, 1)$ (разметки записаны как вектора длины 3).

Бисимуляции разметок в сетях Петри по определению обладают всеми свойствами общего определения бисимуляции для *LTS* (так как сети Петри есть частный случай *LTS*).

Известно, что для некоторых важных подклассов сетей Петри бисимуляция разрешима:

1. Ограниченные сети Петри — так как они совпадают с классом конечных автоматов.
2. Сети Петри с одной неограниченной позицией — так как они совпадают с классом автоматов с одним счетчиком.
3. Сети Петри с пометками один-к-одному (сети, в которых для каждой метки из *Act* существует не более одного помеченного ею перехода).
4. Бисимулярно-детерминированные сети Петри (в бисимулярно-детерминированных сетях разные переходы, имеющие одну и ту же метку, могут быть возбуждены одновременно,

только если их срабатывания приводят к бисимулярным разметкам).

П.Жанкар доказал [48], что бисимуляция разметок неразрешима для всего класса сетей Петри. Более того, бисимуляция неразрешима даже для сетей с двумя неограниченными позициями. Это весьма существенное ограничение, не позволяющее использовать бисимуляцию разметок при изучении свойств достаточно широких классов систем с бесконечным числом состояний. Необходимо искать более сильные эквивалентности, лучше поддающиеся алгоритмическому анализу.

Глава 2

Подобие ресурсов

2.1 Конечное представление отношений

Одним из ключевых препятствий при определении “практичных” отношений на основе бисимуляции разметок является тот факт, что само по себе множество разметок сети Петри бесконечно. Таким образом, мы имеем дело с отношениями на бесконечном множестве. Очевидно, что в практическом плане интересны только те из них, которые можно описать при помощи конечного описания.

В этом разделе исследуются возможности представления бесконечных бинарных отношений на мультимножествах при помощи конечных базисов.

2.1.1 Базисы отношений

Пусть X — конечное непустое множество.

Определим операцию сложения на множестве пар мультимножеств над X .

Пусть $(M_1, M_2), (M'_1, M'_2) \in \mathcal{M}(X) \times \mathcal{M}(X)$ — пары мультимножеств над множеством X . Определим их сумму как

$$(M_1, M_2) + (M'_1, M'_2) =_{def} (M_1 + M'_1, M_2 + M'_2).$$

Пусть $B \subseteq \mathcal{M}(X) \times \mathcal{M}(X)$ — бинарное отношение на множестве мультимножеств над X .

Определение 2.1. Для данного отношения B его *аддитивным замыканием* B^A назовем наименьшее (по вложению) подмножество множества $\mathcal{M}(X) \times \mathcal{M}(X)$, такое, что

1. $B \subseteq B^A$;
2. $\forall (M_1, M_2), (M'_1, M'_2) \in B^A \quad (M_1 + M'_1, M_2 + M'_2) \in B^A$.

Определение 2.2. Для данного отношения B его *транзитивным замыканием* B^T назовем наименьшее (по вложению) подмножество множества $\mathcal{M}(X) \times \mathcal{M}(X)$, такое, что

1. $B \subseteq B^T$;
2. $\forall (M_1, M_2), (M_2, M_3) \in B^T \quad (M_1, M_3) \in B^T$.

Определение 2.3. Для данного отношения B его *аддитивным транзитивным замыканием* B^{AT} назовем наименьшее (по вложению) подмножество множества $\mathcal{M}(X) \times \mathcal{M}(X)$, такое, что

1. $B \subseteq B^{AT}$;
2. $\forall (M_1, M_2), (M'_1, M'_2) \in B^{AT} \quad (M_1 + M'_1, M_2 + M'_2) \in B^{AT}$;
3. $\forall (M_1, M_2), (M_2, M_3) \in B^{AT} \quad (M_1, M_3) \in B^{AT}$.

В дальнейшем будем использовать сокращения A -замыкание, T -замыкание и AT -замыкание соответственно.

Заметим, что в случае отношений на мультимножествах AT -замыкание отношения не всегда совпадает с аддитивным замыканием транзитивного замыкания отношения и с транзитивным замыканием аддитивного замыкания отношения. Строго говоря, имеет место следующая вложенность друг в друга различных видов замыканий отношений на мультимножествах (см. также диаграмму на рисунке 2.1):

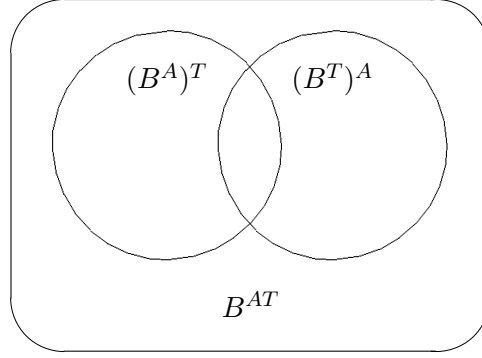


Рис. 2.1: Вложенность различных видов замыканий

Лемма 2.1.

1. $\forall X, B \in \mathcal{M}(X) \times \mathcal{M}(X)$ выполняется $(B^A)^T \subseteq B^{AT}$;
2. $\forall X, B \in \mathcal{M}(X) \times \mathcal{M}(X)$ выполняется $(B^T)^A \subseteq B^{AT}$;
3. $\exists X, B \in \mathcal{M}(X) \times \mathcal{M}(X) : (B^A)^T \subset B^{AT}$;
4. $\exists X, B \in \mathcal{M}(X) \times \mathcal{M}(X) : (B^T)^A \subset B^{AT}$;
5. $\exists X, B \in \mathcal{M}(X) \times \mathcal{M}(X) : (B^A)^T \not\subseteq (B^T)^A$;
6. $\exists X, B \in \mathcal{M}(X) \times \mathcal{M}(X) : (B^T)^A \not\subseteq (B^A)^T$.

Доказательство. 1)–2) — непосредственно из определений.

В качестве примера для случаев 3)–5) рассмотрим множество X , состоящее из одного элемента, и симметричное отношение $B = \{(1, 2), (2, 1)\}$ (один элемент эквивалентен двум).

3) Имеем $B^{AT} = \{(i, j) \mid i, j \geq 1\}$. Следовательно, пара $(1, 3)$ принадлежит B^{AT} . Покажем, что она не принадлежит $(B^A)^T$.

Предположим противное: в B^A существует конечная последовательность пар

$$(1, a), (a, b), (b, c), \dots, (d, e), (e, 3) \in B^A,$$

транзитивное замыкание которой дает пару $(1, 3)$.

Заметим, что для любой пары $(x, y) \in B^A$ выполняется $x + y = 3i$, где $i \in \text{Nat}$ (так как при сложении пар из B кратность суммы их компонентов трем сохраняется). Следовательно, для последней пары последовательности выполняется $e + 3 = 3i$, то есть e также кратно трем. Последовательно применив это рассуждение для всех остальных пар (справа налево), получим, что a кратно трем. Тогда $1 + a$ не может быть кратно трем, то есть пара $(1, a)$ не принадлежит B^A — противоречие.

4) Имеем $B^T = \{(1, 1), (2, 2), (1, 2), (2, 1)\}$. Следовательно, и в данном случае выполняется $(1, 3) \notin (B^T)^A$.

5) Рассмотрим пару $(2, 3)$. Имеем:

- $(2, 3) \in (B^T)^A$, так как $(2, 3) = (1, 1) + (1, 2)$.
- $(2, 3) \notin (B^A)^T$, так как в противном случае пара $(1, 3)$ принадлежала бы отношению $(B^A)^T$ (как транзитивное замыкание пар $(1, 2)$ и $(2, 3)$), что мы уже опровергли в доказательстве пункта 3.

Следовательно, $(B^A)^T \not\subseteq (B^T)^A$.

6) В качестве примера рассмотрим множество X из одного элемента и отношение $B = \{(1, 2)\}$. Рассмотрим пару $(1, 4)$. Имеем:

- $(1, 4) \in (B^A)^T$ (как транзитивное замыкание пар $(1, 2)$ и $(2, 4)$).
- $(1, 4) \notin (B^T)^A$ (очевидно).

Следовательно, $(B^T)^A \not\subseteq (B^A)^T$. □

Таким образом, AT -замыкание — наиболее эффективный (из представленных) способ построения бесконечных отношений (получаемые с его помощью бесконечные отношения самые слабые).

На практике в теории формальных моделей мы почти всегда имеем дело с транзитивно замкнутыми отношениями на множестве состояний исследуемой системы (или даже отношениями

эквивалентности). Накладывая дополнительное требование аддитивной замкнутости, мы получаем возможность выделять некоторые “структурированные” подмножества базового отношения (возможно, представимые конечным базисом). Как показано в лемме 2.1, аддитивную замкнутость можно вводить *три* различными способами.

Определим, в каких случаях конечный базис существует всегда (у любого отношения B , удовлетворяющего заданным ограничениям).

Пусть $\alpha \in \{A, T, AT\}$.

Определение 2.4. Отношение $B' \subseteq \mathcal{M}(X) \times \mathcal{M}(X)$ называется α -базисом отношения B , если $(B')^\alpha = B^\alpha$.

Базис B' называется *минимальным α -базисом отношения B* , если не существует $B'' \subset B'$, такого что $(B'')^\alpha = B^\alpha$.

Лемма 2.2. 1. Все минимальные α -базисы отношения B либо конечны, либо бесконечны.

2. Если у B существует конечный α -базис, то все минимальные α -базисы отношения B конечны;

3. Если у B существует бесконечный минимальный α -базис, то все α -базисы отношения B бесконечны.

Доказательство. 1) Предположим противное. Пусть B_f и B_i — различные минимальные базисы отношения B , причем B_f — конечный, а B_i — бесконечный.

Каждый элемент базиса B_f может быть выражен через конечное число элементов B_i . Поскольку число элементов B_f конечно, весь базис B_f может быть выражен через конечное подмножество базиса B_i . Следовательно, если B_i не конечен, то он не минимален — противоречие.

2) В качестве одного из минимальных базисов можно взять часть данного конечного базиса. Тогда из первой части леммы следует конечность всех минимальных базисов.

3) Предположим противное — существует конечный базис. В качестве одного из минимальных базисов можно взять часть данного конечного базиса. Тогда из первой части леммы следует конечность всех минимальных базисов — противоречие. $\square$

Мы хотим определить, при каких условиях α -замкнутое и в общем случае бесконечное отношение B обладает конечным α -базисом, то есть может быть представлено конечным числом элементов. Мы не рассматриваем в качестве α симметричные замыкания, так как с их помощью невозможно получить из конечного отношения бесконечное. Также мы не рассматриваем рефлексивные замыкания, так как получаемые с их помощью бесконечные отношения имеют простую структуру.

В качестве возможных ограничений, накладываемых на отношение B , мы рассматриваем требования его симметричности, рефлексивности и транзитивности.

Рассмотрим в качестве α аддитивные и транзитивные замыкания. Оказывается, даже если B — отношение эквивалентности, у него могут существовать бесконечные минимальные базисы.

Утверждение 2.1. 1) *Существуют отношения эквивалентности с бесконечными минимальными A -базисами.*

2) *Существуют отношения эквивалентности с бесконечными минимальными T -базисами.*

Пример 2.1. В качестве примера для обоих случаев рассмотрим множество X из одного элемента и отношение эквивалентности

$$B = \{(i, i), (1, i + 2), (i + 2, 1) \mid i \in \text{Nat}\}$$

(пары мультимножеств представлены как пары чисел).

Рассмотрим случай A -замыкания.

Пусть $B' = \{(0, 0), (1, 1)\} \cup \{(1, i + 2), (i + 2, 1) \mid i \in \text{Nat}\}$. Отношение B' является A -базисом отношения B^A . Докажем минимальность B' . Предположим противное: пусть некоторая пара

$(x, y) \in B'$ может быть получена аддитивным замыканием других пар из B' . Однако при любом сложении мультимножеств видов $(1, i+2)$, $(i+2, 1)$ или $(1, 1)$ (пустую пару можно не рассматривать) в итоговой паре оба коэффициента становятся больше единицы — противоречие.

Рассмотрим случай T -замыкания.

Пусть $B' = \{(0, 0)\} \cup \{(1, i+2), (i+2, 1) \mid i \in \text{Nat}\}$. Очевидно, что B' является T -базисом отношения B^T (любая пара вида (i, i) может быть получена транзитивным замыканием двух симметричных пар). Докажем минимальность B' . Предположим противное: некая пара $(x, y) \in B'$, где $x \neq y$, может быть получена транзитивным замыканием других пар из B' . Однако при любом транзитивном замыкании мультимножеств видов $(1, i+2)$ и $(i+2, 1)$ может получиться только рефлексивная пара — противоречие. $\square$

Итак, ни аддитивное, ни транзитивное замыкание не могут быть использованы для генерации бесконечных отношений на основе конечных базисов, даже если B — отношение эквивалентности. Кроме того, легко показать, что существуют аддитивные и транзитивные замыкания отношений эквивалентности, которые сами не являются отношениями эквивалентности.

Рассмотрим в качестве α аддитивное транзитивное замыкание.

Нам потребуется одно техническое определение:

Определение 2.5. Отношение $R \in \mathcal{M}(X) \times \mathcal{M}(X)$ называется *1-рефлексивным*, если R содержит все пары вида (x, x) , где $|x| = 1$.

В случае аддитивных транзитивных замыканий вместо полной рефлексивности отношения B достаточно требовать только 1-рефлексивности, так как аддитивное замыкание 1-рефлексивного отношения рефлексивно (с точностью до пустой пары $(\emptyset, \emptyset)$). Использовать же полную рефлексивность неудобно чисто технически, так как рефлексивные отношения на мультимножествах бесконечны а priori.

Утверждение 2.2. 1) Существуют симметричные отношения с бесконечными минимальными AT -базисами.

2) Существуют 1-рефлексивные отношения с бесконечными минимальными AT -базисами.

В качестве доказательства рассмотрим примеры:

Пример 2.2. Рассмотрим множество X из двух элементов и симметричное отношение

$$B = \{((i, i+1), (i, i+1)) \mid i \in \text{Nat}, i > 0\}$$

(мультимножества представлены как вектора длины 2).

Покажем, что базис B — минимальный. Пусть

$$B_k = \{((i, i+1), (i, i+1)) \mid i, k \in \text{Nat}, k \geq i > 0\}$$

— первые k элементов базиса B . Нам достаточно показать, что его $(k+1)$ -й элемент не может быть выражен через предыдущие, то есть

$$((k+1, k+2), (k+1, k+2)) \notin B_k^{AT}$$

Заметим, что и в левой, и в правой части всех пар векторов из B второй коэффициент на единицу больше первого. При сложении любых двух пар в левой и правой части получатся вектора, где второй коэффициент больше первого на 2. Только при транзитивном замыкании таких пар разность коэффициентов не меняется. Однако, очевидно, что одним транзитивным замыканием пары $((k+1, k+2), (k+1, k+2))$ из B_k не получить. $\square$

Пример 2.3. Рассмотрим множество X из двух элементов и 1-рефлексивное отношение

$$B = \{((1, 0), (1, 0)), ((0, 1), (0, 1))\} \cup \{((1, 0), (i, 1)) \mid i \in \text{Nat}, i \geq 2\}$$

(мультимножества представлены как вектора длины 2).

Покажем, что базис B — минимальный. Пусть

$$B_k = \{((1, 0), (1, 0)), ((0, 1), (0, 1))\} \cup \\ \{((1, 0), (i, 1)) \mid i, k \in \mathbb{N}, k \geq i \geq 2\}$$

— первые $k+2$ элементов базиса B . Нам достаточно показать, что его $(k+3)$ -й элемент не может быть выражен через предыдущие, то есть

$$((1, 0), (k+1, 1)) \notin B_k^{AT}$$

Рассмотрим все нерелексивные элементы отношения B_k . Они имеют вид $((1, 0), (i, 1))$, где $i \geq 2$. Первый коэффициент в правой части пары на $i-1$ больше, чем первый коэффициент в левой. Второй коэффициент в правой части пары на 1 больше, чем второй коэффициент в левой.

Рассмотрим релексивные пары $((1, 0), (1, 0))$ и $((0, 1), (0, 1))$. В них левые и правые части совпадают.

Если сложить любые две нерелексивные пары, то в полученной паре разность между вторым коэффициентом в правой и левой части увеличится и станет равной двум. При любом транзитивном замыкании эта разность уменьшиться не сможет, поскольку в B_k во всех парах “правая” координата не меньше “левой”. В то же время для построения пары $((1, 0), (k+1, 1))$ нам необходимо по крайней мере одно суммирование нерелексивных пар — чтобы увеличить на единицу разность по первому коэффициенту. $\square$

Итак, по отдельности симметричность и 1-релексивность не гарантируют существования конечного AT -базиса. Однако, как будет показано далее, использование обоих ограничений приводит к полностью конечному случаю.

При рассмотрении конкретных классов отношений требование симметричности и 1-релексивности, как правило, возникает естественным образом и не является существенным ограничением. Например, максимальная бисимуляция разметок сети Петри симметрична и 1-релексивна по построению.

2.1.2 Конечность базиса АТ-замыкания

Далее будет определен базис специального вида.

Определение 2.6. Определим частичный порядок $\sqsubseteq$ на множестве $B \subseteq \mathcal{M}(X) \times \mathcal{M}(X)$ пар мультимножеств: для рефлексивных пар (кроме $(\emptyset, \emptyset)$)

$$(r_1, r_1) \sqsubseteq (r_2, r_2) \stackrel{def}{\iff} r_1 \subseteq r_2 \wedge r_1 \neq \emptyset;$$

для нерефлексивных пар рефлексивная часть и нерефлексивный остаток сравниваются отдельно

$$(r_1 + o_1, r_1 + o'_1) \sqsubseteq (r_2 + o_2, r_2 + o'_2) \stackrel{def}{\iff} \\ \stackrel{def}{\iff} o_1 \cap o'_1 = \emptyset \wedge o_2 \cap o'_2 = \emptyset \wedge r_1 \subseteq r_2 \wedge o_1 \subseteq o_2 \wedge o'_1 \subseteq o'_2.$$

Заметим, что рефлексивные и нерефлексивные пары отношения B не сравнимы по $\sqsubseteq$.

Через B_s обозначим множество минимальных (относительно $\sqsubseteq$) элементов B^{AT} . В случае, когда $(\emptyset, \emptyset) \in B$, к B_s также добавляется пара $(\emptyset, \emptyset)$.

Теорема 2.1. Пусть отношение B симметрично и 1-рефлексивно. Тогда отношение B_s является его базисом и B_s конечно.

Доказательство. Минимальными элементами содержащегося в B отношения 1-рефлексивности являются пары вида $([x], [x])$, где $[x]$ обозначает мультимножество, содержащее единственный элемент $x \in X$. Поскольку X — конечное множество, число минимальных элементов отношения 1-рефлексивности также конечно.

Рассмотрим нерефлексивные пары из B^{AT} . Покажем, что всякая такая пара принадлежит АТ-замыканию отношения B_s .

Любая нерефлексивная пара (r, r') может быть разложена на наибольшую рефлексивную часть и нерефлексивный остаток:

$$(r, r') = (r + o, r + o'), \text{ где } o \cap o' = \emptyset.$$

(*) Пусть

$$(r + o, r + o') \in B^{AT} \quad (2.1)$$

— нерефлексивная пара. Если $(r + o, r + o')$ — минимальная пара, то $(r + o, r + o') \in B_s$.

Если пара $(r + o, r + o')$ не минимальна, то существует минимальная пара $(r_1 + o_1, r_1 + o'_1) \in B_s$, такая что $r_1 \subseteq r$, $o_1 \subseteq o$ и $o'_1 \subseteq o'$.

Если $o_1 = o$ и $o'_1 = o'$, то исходная пара $(r + o, r + o')$ является суммой рефлексивной пары и минимальной нерефлексивной пары, и, следовательно, может быть представлена как сумма элементов B_s .

В противном случае, пусть $o = o_1 + o_2$, $o' = o'_1 + o'_2$, $r = r_1 + r_2$. Тогда (2.1) может быть записано как

$$(r_1 + r_2 + o_1 + o_2, r_1 + r_2 + o'_1 + o'_2) \in B^{AT} \quad (2.2)$$

Поскольку B_s содержит все минимальные рефлексивные пары, мы имеем

$$\begin{aligned} & (r_1 + o_1, r_1 + o'_1) + (r_2 + o'_2, r_2 + o'_2) = \\ & = (r_1 + r_2 + o_1 + o'_2, r_1 + r_2 + o'_1 + o'_2) \in (B_s)^{AT} \end{aligned} \quad (2.3)$$

Замкнув по транзитивности пару (2.2) и пару, симметричную паре (2.3), получим

$$(r_3 + o_2, r_3 + o'_2) = (r_1 + r_2 + o_1 + o_2, r_1 + r_2 + o_1 + o'_2) \in B^{AT}, \quad (2.4)$$

где $o_2 \subset o$, $o'_2 \subset o'$, и, следовательно, $(o_2, o'_2) \sqsubset (o, o')$. Таким образом, мы нашли пару (2.4), нерефлексивная часть которой строго меньше, чем нерефлексивная часть (2.1). При этом (2.1) принадлежит АТ-замыканию отношения $B_s \cup \{(r_3 + o_2, r_3 + o'_2)\}$.

Далее повторяем все построения для (2.4), начиная со (*). Нереклексивная часть снова строго уменьшится, или же мы получим пару, разложимую по B_s .

Известно, что любое множество попарно несравнимых целочисленных векторов с неотрицательными коэффициентами конечно. Следовательно, не может существовать бесконечной последовательности строго уменьшающихся нереклексивных пар ресурсов, и описанный выше процесс завершится через конечное число шагов. Таким образом, любая пара вида (2.1) разложима по B_s (причем за конечное число шагов).

Докажем конечность B_s . Рассматривая пару мультимножеств как вектор тройной длины (сначала реклексивная часть, затем нереклексивные остатки), мы сведем частичный порядок $\sqsubseteq$ к отношению сравнения векторов $\leq$. Следовательно, множество минимальных относительно $\sqsubseteq$ элементов B конечно. $\square$

Мы будем называть базис B_s *основным базисом* отношения B .

Из теоремы 2.1 и второго утверждения леммы 2.2 вытекает:

Теорема 2.2. *Если отношение B симметрично и 1-рефлексивно, то все его минимальные AT -базисы конечны.*

Это утверждение (хотя и в несколько другой формулировке и в терминах конгруэнтностей в коммутативных полугруппах) было доказано Л.Редди [61], а позднее (независимо) Й.Хиршфельдом [44]. Приводимое здесь доказательство представляется более удачным, поскольку конструктивно описывает структуру элементов конечного базиса.

Приведем пример основного базиса.

Пример 2.4. Рассмотрим множество X из одного элемента и симметричное 1-рефлексивное отношение

$$B = \{(1, 1), (1, 2), (2, 1), (1, 8), (8, 1), (3, 3), (4, 5), (5, 4)\}$$

(пары мультимножеств представлены как пары чисел).

Основной базис — $B_s = \{(1, 1), (1, 2), (2, 1)\}$.

Основной базис отношения не всегда является минимальным. Например, в базисе из примера 2.4 избыточной является пара $(1, 1)$, которая может быть получена транзитивным замыканием двух других пар. Однако важным достоинством основного базиса является хорошая структурированность. К тому же он по определению единственен для любого B , тогда как минимальных базисов может существовать бесконечно много (несмотря на то, что все они конечны):

Утверждение 2.3. *Существуют отношения эквивалентности с бесконечным числом минимальных АТ-базисов.*

Доказательство. Рассмотрим множество X из одного элемента и отношение эквивалентности

$$B = \{(n, m) \mid n, m \in \text{Nat}, n, m > 0\}$$

(пары мультимножеств представлены как пары чисел).

Рассмотрим отношения

$$B^0 = \{(1, 2), (2, 1)\},$$

$$B^i = \{(1, 2i + 1), (2i + 1, 1), (1, 4), (4, 1)\}, \text{ где } i \in \text{Nat}, i > 0.$$

Отношение B^0 — АТ-базис отношения B . Кроме того, пары $(1, 2)$ и $(2, 1)$ могут быть выражены через любое отношение B^i при помощи сложения и транзитивного замыкания элементов. Например, при $i = 1$

$$B^1 = \{(1, 3), (3, 1), (1, 4), (4, 1)\},$$

пара $(1, 2)$ получается транзитивным замыканием $(1, 4)$ и $(4, 2)$, где $(4, 2) = (3, 1) + (1, 1)$, а пара $(1, 1)$ получена транзитивным замыканием пар $(1, 3)$ и $(3, 1)$.

Докажем, что базис B^i — минимален.

Во-первых, пару $(1, 2)$ нельзя получить замыканием пар вида $(1, 2i + 1)$ и $(2i + 1, 1)$, так как в них разность первой и второй

“координат” — четная, а в паре $(1, 2)$ — нечетная. При сложении и транзитивном замыкании четность этой разности сохраняется.

Во-вторых, пару $(1, 2)$ нельзя получить при помощи АТ-замыкания пар $(1, 4)$ и $(4, 1)$, так как разность “координат” любой пары, полученной АТ-замыканием пар $(1, 4)$ и $(4, 1)$, всегда остается кратной трем. $\square$

Основным базисом рассмотренного в доказательстве утверждения 2.3 отношения B является отношение

$$B_s = \{(1, 1), (1, 2), (2, 1)\}.$$

Далее для доказательства ряда теорем нам потребуется следующее важное свойство 1-рефлексивности базиса АТ-замыкания:

Лемма 2.3. *Пусть B — конечное 1-рефлексивное симметричное отношение. Любая пара $(r_1, r_2) \in B^{AT}$ может быть получена транзитивным замыканием конечного числа пар из B^A .*

Доказательство. Пусть $(a, b), (b, d), (c, e) \in B$.

Замкнем пары (a, b) и (b, d) по транзитивности — получится пара (a, d) . Теперь прибавим к ней пару (c, e) — получится пара $(a + c, d + e)$.

В процессе построения пары $(a + c, d + e)$ мы сначала использовали транзитивное замыкание, а затем сумму. Покажем, что эта пара также может быть получена транзитивным замыканием сумм элементов B .

Отношение B является 1-рефлексивным, следовательно,

$$(c, c), (d, d) \in B^A.$$

Прибавим пару (c, c) к парам (a, b) и (b, d) , а пару (d, d) — к паре (c, e) . Получим пары:

$$(a + c, b + c), (b + c, d + c), (d + c, d + e)$$

Замкнув эти три пары по транзитивности, получим искомую пару $(a + c, d + e)$.

Очевидно, что заменив таким образом в конечной процедуре построения произвольной пары $(x, y) \in B^{AT}$ последовательности вида транзитивность-аддитивность (возможно, вложенные) на последовательности вида аддитивность-транзитивность, мы получим искомую процедуру построения (x, y) как транзитивного замыкания элементов B^A . $\square$

Другими словами, для 1-рефлексивного симметричного отношения B выполняется $B^{AT} = (B^A)^T$.

2.1.3 Свойства основного базиса

Важным достоинством основного базиса (кроме его конечности) является то, что при помощи процедуры разложения, описанной в доказательстве теоремы 2.1, мы можем за конечное число шагов проверить принадлежность произвольной пары ресурсов замыканию B^{AT} .

Несколько упростить проверку позволяет следующее свойство аддитивного транзитивного замыкания:

Лемма 2.4. *Пусть B — симметричное 1-рефлексивное отношение. Тогда из $(r + o, r + o') \in B$ следует, что для любого целого $k > 0$ выполняется $(r + ko, r + ko') \in B^{AT}$.*

Доказательство. Из 1-рефлексивности B следует, что B^{AT} — рефлексивно. В частности, выполняется $((k - 1)o, (k - 1)o) \in B^{AT}$. Сложив данную рефлексивную пару и исходную пару $(r + o, r + o')$, получим

$$(r + ko, r + o' + (k - 1)o) \in B^{AT}. \quad (2.5)$$

Аналогично, сложив исходную пару $(r + o, r + o')$ и рефлексивную пару $(o' + (k - 2)o, o' + (k - 2)o)$, получим

$$(r + o' + (k - 1)o, r + 2o' + (k - 2)o) \in B^{AT}. \quad (2.6)$$

Транзитивным замыканием пар 2.5 и 2.6 является пара

$$(r + ko, r + 2o' + (k - 2)o) \in B^{AT}.$$

Повторив подобное рассуждение еще $k - 2$ раз (заменяя в правой части $r + o$ на $r + o'$), получим $(r + ko, r + ko') \in B^{AT}$. $\square$

Алгоритм проверки, использующий лемму 2.4:

Алгоритм 2.1. (проверка принадлежности AT -замыканию)

ввод: Основной базис B_s некоторого отношения B ,
пара $(r + o, r + o') \in \mathcal{M}(X) \times \mathcal{M}(X)$ (полагаем $o \cap o' = \emptyset$,
 $(o, o') \neq (\emptyset, \emptyset)$).

вывод: *True*, если $(r + o, r + o') \in (B_s)^{AT}$,
False в противном случае.

шаг 1:

Построим

$$B' = \{(s + t, s + t') \in B_s \mid t \cap t' = \emptyset \wedge (t, t') \neq (\emptyset, \emptyset) \wedge t \leq o \wedge t' \leq o'\}.$$

B' — это множество элементов основного базиса, чья
нерефлексивная часть не больше пары (o, o') .

Введем переменные x, y и y' типа “мультимножество”.

Присвоим $x := r$, $y := o$, $y' := o'$

(тогда $(x + y, x + y') = (r + o, r + o')$).

шаг 2:

Переберем множество B' , найдем первую пару

$(s + t, s + t') \in B'$, такую, что $s \leq x$, $t \leq y$, $t' \leq y'$.

Если такой пары не существует, то возвращаем *False*.

Если такая пара существует, причем $t = ny$, $t' = ny'$,

где $n \in \text{Nat}$, то возвращаем *True*.

шаг 3:

Определим $k = \max\{m \in \text{Nat} \mid mt \leq y \wedge mt' \leq y'\}$.

Присвоим $x := x + kt$, $y := y - kt$, $y' := y' - kt'$,

$B' := B' \setminus \{(s + t, s + t')\}$.

Возвратимся на шаг 2.

Рассмотрим более подробно шаг 3 алгоритма.

Согласно лемме 2.4, выполняется $(s + kt, s + kt') \in (B_s)^{AT}$. Прибавив к этой паре рефлексивную, получим

$$\begin{aligned} & (s + kt, s + kt') + (x - s + y' - kt', x - s + y' - kt') = \\ & = (x + kt + y' - kt', x + y') \in (B_s)^{AT}, \end{aligned} \quad (2.7)$$

Замкнув по транзитивности пару $(x + y, x + y')$ и пару, симметричную паре 2.7, получим

$$(x + y, x + kt + y' - kt') \in (B_s \cup \{(x + y, x + y')\})^{AT}.$$

Аналогично, мы можем выразить пару $(x + y, x + y')$ через базис B_s и пару $(x + y, x + kt + y' - kt')$:

$$(x + y, x + y') \in (B_s \cup \{(x + y, x + kt + y' - kt')\})^{AT}.$$

Итак, выполняется

$$(x + y, x + y') \in (B_s)^{AT} \Leftrightarrow (x + y, x + kt + y' - kt') \in (B_s)^{AT},$$

следовательно, вместо исходной пары $(x + y, x + y')$ мы можем рассматривать пару $(x + y, x + kt + y' - kt')$, нерелексивная часть которой (равная $(y - kt, y' - kt')$) строго меньше нерелексивной части исходной.

Заметим, что $y - kt < t$, $y' - kt' < t'$. Поскольку все пары, которые будут получаться на следующих итерациях алгоритма, будут иметь ещё меньшую нерелексивную часть, мы можем выбросить пару $(s + t, s + t')$ из рассматриваемого множества B' . Все выброшенные на предыдущих итерациях алгоритма пары также не могут нам еще раз пригодиться, поскольку их нерелексивная часть строго больше $(y - kt, y' - kt')$.

Оценим трудоемкость алгоритма. Обозначим B'_i — множество B' на i -й итерации алгоритма. На i -й итерации из множества

B'_i выбрасывается один элемент, то есть максимальное возможное число итераций — $|B'_1|$, а мощность i -го множества перебора $|B'_i| = |B'_1| - i$. На каждом шаге перебора происходит покоординатное сравнение двух целочисленных векторов длины $2|X|$ (нерефлексивных частей (y, y') и (t, t')). Следовательно, трудоемкость алгоритма 2.1 может быть оценена как

$$O\left(2|X|\frac{|B'_1|(|B'_1| - 1)}{2}\right) = O(|X||B'_1|(|B'_1| - 1)). \quad (2.8)$$

Обозначим $B_s^{nr} = \{(r+o, r+o') \in B_s \mid o \cap o' \neq \emptyset \wedge (o, o') \neq (\emptyset, \emptyset)\}$ — множество нерефлексивных пар базиса B_s . Из симметричности основного базиса следует, что отношение B_s^{nr} — также симметрично. Заметим, что нерефлексивные части любых двух симметричных пар не пересекаются, следовательно, на первом шаге алгоритма в множество B'_1 может попасть только одна из них. Тогда $|B'_1| \leq |B_s^{nr}|/2$, и оценку 2.8 можно записать как

$$O\left(\frac{|X||B_s^{nr}|^2}{8}\right).$$

Введем специальное обозначение $C\text{Test}(B_s)$ для оценки трудоемкости алгоритма проверки принадлежности пары мультимножеств AT -замыканию данного основного базиса B_s . Убрав константу, получим

$$C\text{Test}(B_s) = O(|X||B_s|^2).$$

Рассмотрим пример работы алгоритма 2.1.

Пример 2.5. Пусть X состоит из одного элемента, тогда $\mathcal{M}(X) = \text{Nat}$, пары мультимножеств представляют собой пары целых положительных чисел.

Пусть $B_s = \{(1, 1), (2, 4), (4, 2), (0, 5), (5, 0)\}$.

а) Рассмотрим пару $(2, 9)$.

Протокол работы (в квадратных скобках — номера шагов из определения алгоритма):

- 1[1] $x := 2, y := 0, y' := 7, B' := \{(2, 4), (0, 5)\}$.
 2[2] Найдем перебором пару $(s + t, s + t') = (0, 5)$
 $(s := 0, t := 0, t' := 5)$.
 3[3] Получим $k = 1$. Тогда $x := 2, y := 0, y' := 2$,
 $B' := \{(2, 4), (4, 2), (5, 0)\}$.
 4[2] Найдем перебором пару $(s + t, s + t') = (2, 4)$.
 $(t, t') = (y, y')$, следовательно, возвращаем True.
 б) Рассмотрим пару $(1, 7)$.
 Протокол работы алгоритма:
 1[1] $x := 1, y := 0, y' := 6, B' := \{(2, 4), (0, 5)\}$.
 2[2] Найдем перебором пару $(s + t, s + t') = (0, 5)$
 $(s := 0, t := 0, t' := 5)$.
 3[3] Получим $k = 1$. Тогда $x := 1, y := 0, y' := 1$,
 $B' := \{(2, 4), (4, 2), (5, 0)\}$.
 4[2] Так как подходящей пары $(s + t, s + t') = (2, 4)$ не существует,
 возвращаем False.

На практике рассматриваемое отношение может задаваться произвольным конечным базисом (не обязательно минимальным и не обязательно основным). Однако любой конечный симметричный 1-рефлексивный базис можно преобразовать в основной базис при помощи эффективной процедуры:

Алгоритм 2.2. (построение основного базиса)

ввод: Конечное симметричное 1-рефлексивное отношение B .

вывод: Основной базис B_s отношения B .

шаг 1: Построим $B' = B \setminus C$, где

$$C = \{(r + o, r + o') \in B \mid \exists (s + o, s + o') \in B, s < r\}.$$

шаг 2: Проверим, выполняется ли для всех элементов B' свойство минимальности относительно частичного порядка $\sqsubseteq$.

- Если да, то возвратим B' — искомый основной базис.

- В противном случае существуют пары $(a, b), (c, d) \in B'$, такие, что $(a, b) \sqsubseteq (c, d)$, $(a, b) \neq (c, d)$. Очевидно, что обе эти пары — нерефлексивные и выполняется

$$(a, b) = (r + o, r + o'), (c, d) = (r + r_1 + ko + o_1, r + r_1 + ko' + o'_1),$$

где $o \cap o' = \emptyset$, $(o + o_1) \cap (o' + o'_1) = \emptyset$, $(o_1, o'_1) \neq (\emptyset, \emptyset)$, $k \in \text{Nat}$.

В силу леммы 2.4 выполняется $(r + ko, r + ko') \in (B')^{AT}$. Поскольку отношение B' 1-рефлексивно, к этой паре мы можем прибавить рефлексивную пару $(r_1 + o'_1, r_1 + o'_1)$:

$$(r + r_1 + ko + o'_1, r + r_1 + ko' + o'_1) \in (B')^{AT}. \quad (2.9)$$

Замкнув по транзитивности пару (c, d) и пару, симметричную паре 2.9, получим

$$(e, f) = (r + r_1 + ko + o_1, r + r_1 + ko + o'_1) \in (B')^{AT}$$

Выбросим пары (c, d) и (d, c) из B' .

Если не существует пары $(e', f') \in B'$, такой, что

$$(e', f') = (r' + o_1, r' + o'_1), \quad r' \leq r + r_1 + ko,$$

то добавим к B' пары (e, f) и (f, e) .

Возвратимся на шаг 2.

Оценим трудоемкость алгоритма (обозначим ее $CBuild(B)$).

На каждой итерации мы либо уменьшаем размер множества B' , либо заменяем один из его элементов на элемент со строго меньшей нерефлексивной частью ((c, d) на (e, f)), причем количество “уменьшений” для каждого элемента не может быть больше, чем мощность B' (так как мы по сути дела последовательно “вычитаем” нерефлексивные составляющие других элементов). Следовательно, максимальное количество итераций может быть оценено как $O(|B|^2)$. На каждой итерации самая трудоемкая процедура — поиск несравнимых по $\sqsubseteq$ элементов B' , то есть последовательный перебор всевозможных пар элементов множества B' и

сравнение этих элементов как векторов тройной длины (сначала рефлексивная часть, затем нерефлексивные остатки). Трудоемкость такого поиска — $O(|X||B|^2)$. Следовательно, общую трудоемкость алгоритма 2.2 можно оценить как

$$CBuild(B) = O(|X||B|^4).$$

Итак, трудоемкость приведенного алгоритма полиномиально зависит от мощности рассматриваемого отношения B . Однако необходимо заметить, что на практике мощность B очень часто находится в экспоненциальной зависимости от мощности основного множества X .

Рассмотрим пример работы алгоритма 2.2.

Пример 2.6. Пусть X состоит из одного элемента, тогда $\mathcal{M}(X) = Nat$, пары мультимножеств представляют собой пары целых положительных чисел.

Пусть $B = \{(1, 1), (1, 2), (2, 1), (1, 3), (3, 1), (2, 6), (6, 2)\}$.

Протокол работы алгоритма:

1. $B' = \{(1, 1), (1, 2), (2, 1), (1, 3), (3, 1), (2, 6), (6, 2)\}$

$$(a, b) = (1, 2), (c, d) = (1, 3), k = 1$$

$$(e, f) = (c, d - 1 + 0) = (1, 2)$$

Удалим из B' пары $(1, 3)$ и $(3, 1)$.

Пара (e, f) уже содержится в B' .

2. $B' = \{(1, 1), (1, 2), (2, 1), (2, 6), (6, 2)\}$

$$(a, b) = (1, 2), (c, d) = (2, 6), k = 4$$

$$(e, f) = (c, d - 4 + 0) = (2, 2)$$

Удалим из B' пары $(2, 6)$ и $(6, 2)$.

Пару (e, f) добавлять не будем, так как в B' содержится пара $(1, 1)$.

3. Возвращаем $B_s = \{(1, 1), (1, 2), (2, 1)\}$.

2.2 Подобие ресурсов

В этом разделе вводятся и исследуются ключевые понятия ресурса и подобия ресурсов обыкновенных сетей Петри.

2.2.1 Определение

Определим понятие ресурса сети Петри.

Определение 2.7. Пусть $N = (P, T, F)$ — обыкновенная сеть Петри. *Ресурсом* сети N называется мультимножество над множеством позиций P .

Другими словами, ресурсами сети N являются все элементы множества $\mathcal{M}(P)$.

Формально определение ресурса не отличается от определения разметки. Каждая разметка является ресурсом и каждый ресурс является разметкой. Мы различаем эти понятия из-за их различной интерпретации. Ресурсы являются частями разметок (в некотором смысле неделимыми), обеспечивающими то или иное поведение сети при любом её состоянии. Например, в сети на рисунке 1.5 две молекулы водорода и одна молекула кислорода составляют ресурс, достаточный для срабатывания перехода (т.е. для производства двух молекул воды).

Мы могли бы использовать термин “подмножество разметки”, но предпочитаем “ресурс”, поскольку рассматриваем ресурс не как “подмножество данной разметки”, а как “общую часть всех разметок, содержащих данное (мульти)множество фишек”.

Суть различий наиболее ярко проявляется в определении подобных ресурсов:

Определение 2.8. Пусть $N = (P, T, F, l)$ — помеченная сеть Петри. Ресурсы r и s называются *подобными* (обозначается $r \approx s$), если для любой разметки R , такой, что $r \subseteq R$, выполняется

$$R \sim R - r + s.$$

Отношением *подобия ресурсов* сети Петри N называется максимальное по вложению отношение на множестве ресурсов $\mathcal{M}(P)$, такое, что любые два связанных им ресурса подобны (обозначается $\approx$).

Если ресурсы подобны, то в любой разметке мы можем заменить один из них на другой, и при этом дальнейшее наблюдаемое поведение системы не изменится (в смысле бисимулярности). Таким образом, подобие ресурсов отслеживает все ресурсы в сети Петри, обладающие идентичными свойствами при любом состоянии системы в целом.

Заметим, что для сети Петри, представляющей реакцию синтеза воды (рис. 1.5), одна молекула кислорода эквивалентна одной молекуле водорода с точки зрения бисимулярности разметок, так как и в том и в другом случае переход сети не сработает. Однако подобными ресурсами указанные две молекулы не являются.

Примером подобных ресурсов могут служить два набора монет — одна монета по десять копеек и две монеты по пять копеек, при условии кратности цен на интересующие нас товары десяти копейкам. В данном случае различный номинал монет соответствует различным позициям сети Петри, количество монет — разметке соответствующей позиции, покупка товара — срабатыванию перехода, а цена на товар — кратности исходящих из позиций дуг.

Рассмотрим модель подобной ситуации, изображенную на рисунке 2.2. Покупается товар стоимостью 20 копеек. Позиция “магазин” моделирует количество товара в магазине, “куплено” — количество уже купленного товара, “10к” и “5к” — количество неизрасходованных монет. Покупка возможна тремя способами — переходы t_1 , t_2 и t_3 . Во всех трех случаях тратится различное число различных монет.

В такой ситуации использование пятикопеечных монет избыточно, что и отражается в подобии ресурсов $“10к” \approx “5к” + “5к”$.

Примеры подобных ресурсов приведены на рисунке 2.3.

На рисунке а) изображена сеть Петри, содержащая два перехо-

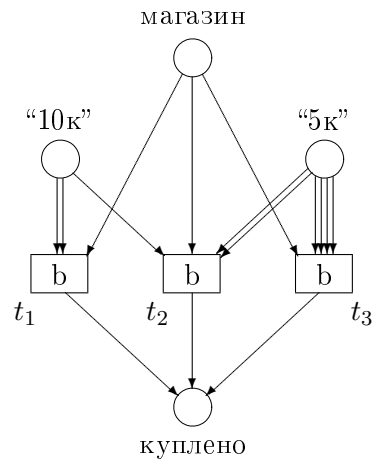
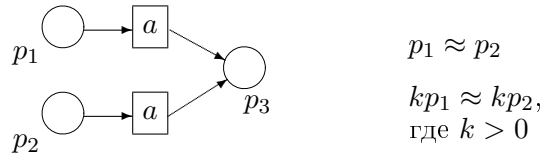


Рис. 2.2: Модель покупки товара с использованием двух видов монет

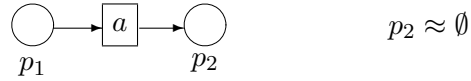
да, помеченных одинаковой меткой a и приводящих к одинаковой разметке p_3 . Здесь подобны ресурсы p_1 и p_2 , так как они приводят к полностью идентичному поведению сети — срабатыванию перехода с меткой a с помещением фишки в позицию p_3 . Более того, подобны все ресурсы, содержащие одинаковое число фишек в позициях p_1 и p_2 .

На рисунке б) изображена простейшая сеть, состоящая из одного перехода. В данном случае ресурс p_2 подобен пустому ресурсу, так как он никак не влияет на поведение сети (сколько бы фишек мы ни поместили в позицию p_2 , переход сработать все равно не сможет).

На рисунке в) изображен цикл, состоящий из одного перехода и одной позиции. Заметим, что множество разметок данной сети можно разделить на два непересекающихся подмножества — пустую разметку и все прочие. При пустой разметке переход сработать не может, при всех прочих — может сработать любое количество раз (более того, все прочие разметки бисимулярны между



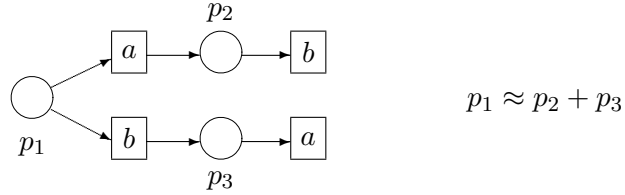
а) параллельные срабатывания



б) тупик



в) циклические срабатывания



г) ресурсы различной мощности

Рис. 2.3: Подобные ресурсы

$2p_1 \not\sim 2p_2$ (при разметке $2p_1$ может сработать переход t_3 , помеченный b).

Подобие ресурсов гораздо сильнее, чем бисимуляция разметок, так как оно не учитывает конкретной (начальной) разметки сети Петри, а выявляет закономерности, общие для ВСЕХ возможных разметок. Это снижает его выразительность по сравнению с бисимуляцией, однако добавляет полезное свойство аддитивной замкнутости, которое может быть использовано для построения конечного базиса.

Подобие ресурсов рефлексивно, симметрично, транзитивно и замкнуто относительно сложения:

Лемма 2.5. Пусть $N = (P, T, F, l)$ — помеченная сеть Петри, r, s, u, v — ресурсы сети N . Тогда

1. $r \approx r$;
2. $r \approx s \Rightarrow s \approx r$;
3. $r \approx s \wedge s \approx u \Rightarrow r \approx u$;
4. $r \approx s \wedge u \approx v \Rightarrow r + u \approx s + v$.

Доказательство. Утверждения 1–3) следуют непосредственно из определения подобных ресурсов и из того, что отношение максимальной бисимуляции разметок (отношение $\sim$) является отношением эквивалентности (утверждение 1.3).

4) Предположим противное: существует разметка $M \in \mathcal{M}(P)$, такая, что $r + u \subseteq M$ и $M \not\sim M - r - u + s + v$.

Из $r + u \subseteq M$ следует $r \subseteq M$. Тогда по определению подобных ресурсов для пары $r \approx s$ имеем

$$M \sim M - r + s. \quad (2.10)$$

Выполняется $r + u \subseteq M \Rightarrow u \subseteq M - r \Rightarrow u \subseteq M - r + s$, следовательно, по определению подобия для пары $u \approx v$ имеем

$$M - r + s \sim M - r + s - u + v = M - r - u + s + v. \quad (2.11)$$

Используя свойство транзитивности максимальной бисимуляции разметок, из 2.10 и 2.11 получим $M \sim M - r - u + s + v$ — противоречие. $\square$

Таким образом, подобие ресурсов обладает всеми свойствами AT -замыкания некоторого отношения (другими словами, отношения $\approx$ и $(\approx)^{AT}$ совпадают).

Например, из подобия ресурсов $r \approx s$ немедленно следует подобие ресурсов $2r \approx 2s$, $2r + s \approx r + 2s$, $5r \approx 3r + 2s$ и так далее, что может быть описано формулой

$$i_r r + i_s s \approx j_r r + j_s s, \text{ где } i_r, i_s, j_r, j_s \in Nat, i_r + i_s = j_r + j_s.$$

Из подобия ресурсов $r \approx 2r$ немедленно следует подобие ресурсов:

$$ir \approx jr, \text{ где } i, j \in Nat, i > 0, j > 0.$$

То есть любая пара подобных ресурсов может рассматриваться в качестве базиса для некоторого бесконечного отношения. С другой стороны, всё отношение подобия ресурсов также обладает конечным базисом:

Теорема 2.3. *Подобие ресурсов в помеченных сетях Петри обладает конечным AT -базисом.*

Доказательство. Непосредственно из теоремы 2.1 и леммы 2.5, используя тот факт, что рефлексивность отношения гарантирует его 1-рефлексивность. $\square$

Наличие конечного базиса — ключевое отличие подобия ресурсов от бисимуляции разметок. Поскольку подобие ресурсов кроме транзитивности еще и замкнуто относительно сложения, в любой сети оно имеет конечный базис.

В то же время в некоторых сетях бисимуляция разметок также может обладать конечным базисом:

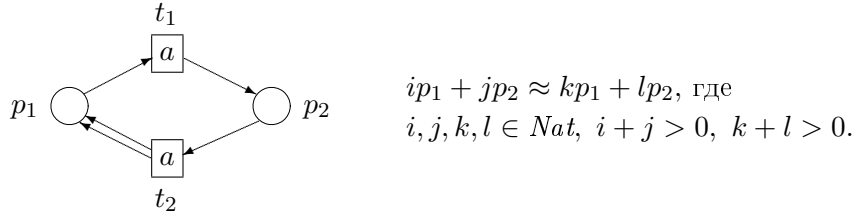


Рис. 2.5: Пример сети Петри, в которой подобие ресурсов и бисимуляция разметок совпадают

Пример 2.7. На рисунке 2.5 изображена сеть Петри, представляющая собой цикл, состоящий из двух позиций и двух переходов, помеченных одним и тем же символом a .

Поскольку один из переходов (p_1) при срабатывании сохраняет общее количество фишек в разметке сети, а другой (p_2) — увеличивает это количество на единицу, при любой непустой начальной разметке обе позиции — неограниченные, и переходы могут сработать сколь угодно много раз.

Таким образом, любые две непустые разметки — бисимулярны, то есть для данной сети мы можем построить следующую бисимуляцию разметок:

$$(\sim) = \{\emptyset, \emptyset\} \cup \{(ip_1 + jp_2, kp_1 + lp_2) \mid i, j, k, l \in Nat, i + j > 0, k + l > 0\}.$$

Это же отношение является подобием ресурсов данной сети:

$$(\approx) = (\sim).$$

Подобие ресурсов задается следующим основным базисом:

$$(\approx)_s = \{(\emptyset, \emptyset), (p_1, p_1), (p_2, p_2), (p_1, p_2), (p_2, p_1),$$

$$(p_1, 2p_1), (2p_1, p_1), (p_2, 2p_2), (2p_2, p_2)\}.$$

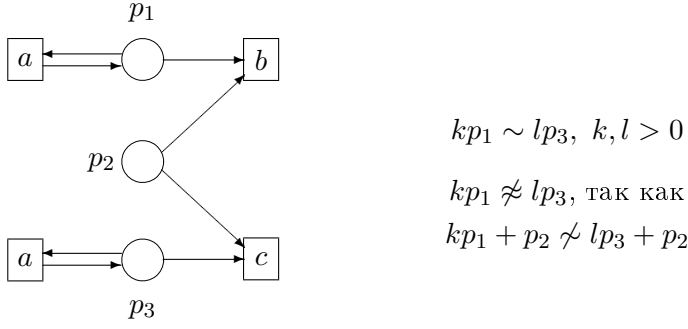


Рис. 2.6: Пример сети Петри, в которой существует АТ-замкнутое сужение бисимуляции разметок, не входящее в подобие ресурсов

Имеющее место в рассмотренном случае совпадение подобия ресурсов и бисимуляции разметок является достаточно нехарактерной ситуацией, поскольку, как правило, бисимуляция содержит гораздо больше пар, чем подобие ресурсов, и не замкнута относительно сложения.

Заметим также, что подобие ресурсов не является наибольшим по вложению АТ-замкнутым сужением бисимуляции разметок:

Пример 2.8. На рисунке 2.6 изображена сеть Петри, состоящая из трех позиций и четырех переходов. Бисимуляция разметок этой сети содержит пары вида $kp_1 \sim lp_3$, где $k, l > 0$. Действительно, при любой из разметок вида kp_1 или lp_3 могут сработать только переходы, помеченные "a" (сколько угодно много раз). Множество всех таких пар замкнуто относительно сложения и относительно транзитивности. С другой стороны, очевидно, что ни одна из них не принадлежит подобию ресурсов, так как $kp_1 + p_2 \not\sim lp_3 + p_2$.

Бисимуляция разметок в каком-то смысле существенно более выразительна, чем подобие ресурсов — в ней могут существовать и другие обладающие конечными базисами подмножества пар.

Итак, определив понятие подобия ресурсов, мы получили бес-

конечное корректное с точки зрения бисимулярности отношение на множестве $\mathcal{M}(P)$, у которого всегда существует конечный базис. Проблема состоит в построении этого базиса. Как будет показано далее, эта проблема все еще неразрешима.

2.2.3 Неразрешимость

В работах Н.С. Сидоровой и Ф. Шнобелена ([21],[65]) для бисимуляционной редукции помеченных сетей Петри было предложено понятие слияния позиций:

Определение 2.9. [21] Пусть $N = (P, T, F, l)$ — помеченная сеть Петри, $B \subseteq P \times P$ — отношение эквивалентности на множестве позиций сети N . Отношение B называется *корректным слиянием позиций* для сети N , если для любой разметки $M \in \mathcal{M}(P)$ и для любой пары позиций $(p, q) \in B$, такой, что $M(p) > 0$, выполняется $M \sim M'$, где

$$M'(p) = M(p) - 1,$$

$$M'(q) = M(q) + 1,$$

$$M'(r) = M(r) \text{ для любой позиции } r \in P, \text{ такой что } r \neq p, r \neq q.$$

Известно, что для любой помеченной сети N существует максимальное по вложению корректное слияние позиций (обозначается $R(N)$), причем оно является отношением эквивалентности.

Отношение названо “слиянием”, так как входящие в него пары позиций можно слить, получив сеть меньшего размера с тем же поведением (в смысле бисимулярности). При этом фишки начальной разметки сливаемых позиций также “суммируются” и помещаются в новую позицию.

Пример корректного слияния позиций представлен на рисунке 2.7. Слиты позиции p_1 и p_2 (получена позиция q), и это максимальное возможное слияние позиций для данной сети.

Позиции сети Петри можно рассматривать как ресурсы. Строго говоря, множество позиций — это множество ресурсов мощно-

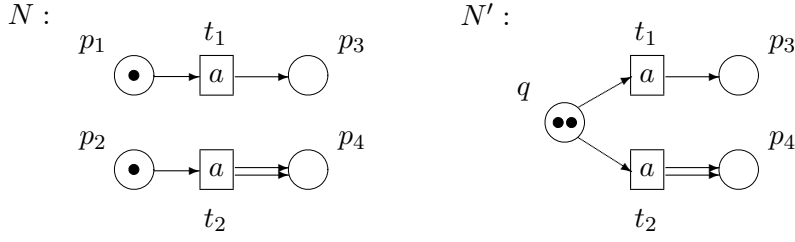


Рис. 2.7: Пример корректного слияния позиций. Для сети N выполняется $R(N) = Id(P) \cup \{(p_1, p_2), (p_2, p_1)\}$, позиция q в сети N' получена слиянием позиций p_1 и p_2 .

сти 1. Следовательно, слияние позиций является отношением на множестве ресурсов.

Непосредственно из определений подобия ресурсов и корректного слияния позиций следует их взаимосвязь:

Утверждение 2.5. Пусть $N = (P, T, F, l)$ — помеченная сеть Петри, $p, q \in P$ — позиции сети N . Тогда выполняется

$$(p, q) \in R(N) \Leftrightarrow p \approx q.$$

Таким образом, корректное слияние позиций является сужением подобия ресурсов. Однако, в силу конечности множества P , отношение $R(N)$ для любой сети конечно. Более того, оно охватывает достаточно небольшую часть конечного базиса подобия ресурсов, поскольку включает в себя только пары ресурсов единичной мощности. Например, среди всех подобных ресурсов, представленных на рисунке 2.3, только одна пара принадлежит слиянию позиций — пара $p_1 \approx p_2$ из случая (а). Подобие ресурсов предоставляет существенно более богатую информацию о бисимуляции сети Петри.

Рассмотрим, например, подобие ресурсов сети N , изображенной на рисунке 2.7. Оно включает в себя не только пару $p_1 \approx p_2$, но и набор подобных ресурсов $p_3 \approx 2p_4 \approx \emptyset$. Эта информация позволяет еще более упростить исходную сеть (рисунок 2.8), убрав

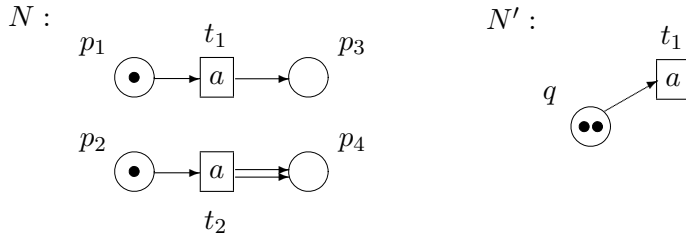


Рис. 2.8: Пример редукции на основе подобия ресурсов. Для сети N выполняется $p_1 \approx p_2$, $p_3 \approx 2p_4 \approx \emptyset$. Позиция q в сети N' получена слиянием позиций p_1 и p_2 , позиции p_3 , p_4 и переход t_2 удалены.

избыточные позиции p_3 и p_4 и переход t_2 . Подробнее редукция сетей Петри на основе подобия ресурсов описана в главе 6.

В 1998 году Н.С. Сидоровой было доказано, что корректное слияние позиций неразрешимо для обыкновенных сетей Петри:

Теорема 2.4. [21] *Проблема распознавания максимального (по вложению) корректного слияния позиций для данной помеченной сети N неразрешима.*

При доказательстве этой теоремы в работе [21] использован метод П. Жанкара построения сети Петри, имитирующей в слабом смысле машину Минского, и доказательства эквивалентности сложности исследуемой проблемы сложности проблемы останова машины Минского. Показано, что неразрешима следующая алгоритмическая проблема:

Даны сеть $N = (P, T, F, l)$, позиции $p, q \in P$. Является ли отношение $Id(P) \cup \{(p, q), (q, p)\}$ корректным слиянием позиций для N ?

Из неразрешимости этой проблемы следует неразрешимость

проблемы распознавания максимального корректного слияния позиций.

Подобие ресурсов содержит в себе максимальное корректное слияние позиций, то есть принадлежность пары позиций слиянию позиций является достаточным условием для их подобия как ресурсов. Из невозможности распознавания корректного слияния позиций (теорема 2.4) и утверждения 2.5 получаем

Следствие 2.1. *Проблема распознавания подобия ресурсов для данной помеченной сети N неразрешима.*

Таким образом, даже несмотря на существование конечного базиса, подобие ресурсов не может быть эффективно построено.

Глава 3

Сужения подобия ресурсов

3.1 Ограниченное подобие ресурсов

Неразрешимость фундаментального отношения подобия ресурсов вынуждает искать более грубые способы выявления бисимулярных структур на множестве ресурсов. Одним из вариантов могло бы стать ограничение сверху размера рассматриваемых множеств. Однако, как будет далее показано, это также приводит к алгоритмической неразрешимости.

В данном разделе вводится понятие ограниченного подобия ресурсов в обыкновенных сетях Петри. Такое отношение не учитывает пары ресурсов, мощность хотя бы одного из которых превышает заданный параметр.

3.1.1 Определение и основные свойства

Определение 3.1. Пусть $n \in \text{Nat}$ — целое неотрицательное число, $N = (P, T, F, l)$ — помеченная сеть Петри, $(\approx)$ — множество всех пар подобных ресурсов сети N . Через $(\approx_n)$ обозначим подмножество множества $(\approx)$, состоящее только из пар, получаемых аддитивно-транзитивным замыканием элементов $(\approx)$, мощности

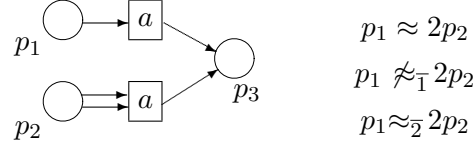


Рис. 3.1: Подобные ресурсы n -ограниченно подобны не для всех n

левой и правой частей которых не превышают n :

$$(\approx_n) =_{def} \{(r, s) \in (\approx) \mid (|r| \leq n \ \& \ |s| \leq n) \vee (r, s) \in (\approx_n)^{AT}\}.$$

Соответствующее отношение назовем *n -ограниченным подобием ресурсов*.¹

На рисунке 3.1 приведен пример 2-ограниченных подобных ресурсов. Две фишки в позиции p_2 эквивалентны одной фишке в позиции p_1 . Очевидно, что эта эквивалентность не будет обнаружена, если рассматривать только ресурсы мощности не более единицы.

Рассмотрим основные свойства ограниченного подобия.

Утверждение 3.1. Пусть $N = (P, T, F, l)$ — помеченная сеть Петри, r, s, u, v — ресурсы сети N , $n \in \text{Nat}$. Тогда

1. $r \approx_n r$ для $n > 0$;
2. $r \approx_n s \Rightarrow s \approx_n r$;
3. $r \approx_n s \wedge s \approx_n u \Rightarrow r \approx_n u$;
4. $r \approx_n s \wedge u \approx_n v \Rightarrow r + u \approx_n s + v$.

Доказательство. Из определений. □

¹Возможно, правильное было бы назвать отношение подобием n -ограниченных ресурсов, однако такой термин очень неудобен в использовании.

Следствие 3.1. *n -ограниченное подобие ресурсов обладает конечным АТ-базисом.*

Доказательство. Следует из аддитивной и транзитивной замкнутости отношения $\approx_{\bar{n}}$. $\square$

Таким образом, ограниченное подобие ресурсов обладает теми же структурными свойствами, что и полное подобие ресурсов.

3.1.2 Ограниченное подобие как способ приближения полного подобия

Рассмотрим, как связаны между собой полное и ограниченное подобия.

Утверждение 3.2. *Для любого $n \in \text{Nat}$ выполняется*

$$r \approx_{\bar{n}} s \implies r \approx_{\overline{n+1}} s.$$

Другими словами, $(\approx_{\bar{n}}) \subseteq (\approx_{\overline{n+1}})$.

Доказательство. Из определений. $\square$

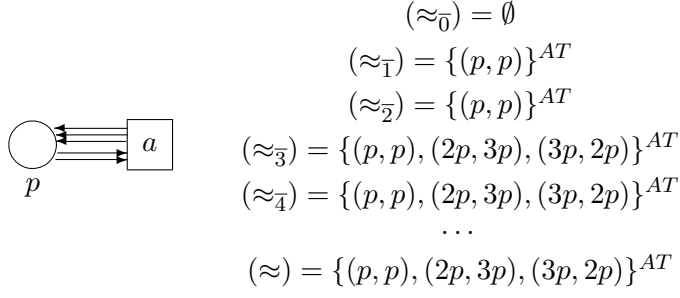
Утверждение 3.3. *Для любой помеченной сети Петри N найдется число $n \in \text{Nat}$, такое что $(\approx) = (\approx_{\bar{n}})$.*

То есть подобие ресурсов — предел n -ограниченного подобия ресурсов.

Доказательство. В качестве n можно взять максимальную мощность компонентов конечного базиса отношения $(\approx)$ (по следствию 2.3 такой базис всегда существует). $\square$

Итак, последовательность ограниченных подобий ресурсов стабилизируется начиная с некоторого n , причем пределом этой последовательности является полное подобие ресурсов:

$$(\approx_0) \subseteq (\approx_1) \subseteq \dots \subseteq (\approx_{\overline{n-1}}) \subseteq (\approx_{\bar{n}}) = (\approx_{\overline{n+1}}) = \dots = (\approx).$$

Рис. 3.2: “Стабилизация” ограниченного подобия при $n = 3$

На рисунке 3.2 приведен пример такой ситуации. При увеличении параметра n ограниченное подобие “обрастает” новыми парами ресурсов: при $n = 1$ добавляется пара (p, p) , при $n = 3$ — пары $(2p, 3p)$ и $(3p, 2p)$. Однако начиная с $n = 3$ множество эквивалентных пар стабилизируется, становясь равным полному подобию ресурсов $(\approx)$.

Использовать на практике ограниченное подобие в качестве приближения полного подобия проблематично, так как

Теорема 3.1. *n -ограниченное подобие ресурсов неразрешимо для любого $n > 0$.*

Доказательство. Слияние позиций, исследованное в [21, 65], совпадает с 1-ограниченным подобием ресурсов. В [21] было доказано, что слияние позиций неразрешимо.

Для любого $n > 1$ из разрешимости n -ограниченного подобия ресурсов следовала бы и разрешимость 1-ограниченного подобия — достаточно было бы взять только пары ресурсов, компоненты которых содержат не более одной фишки. $\square$

Следствие 3.2. *n -ограниченное подобие ресурсов невычислимо для любого $n > 0$.*

3.2 Бисимуляция ресурсов

В этом разделе рассматривается ещё один способ сужения полного подобия ресурсов. Ограничение накладывается не на размер ресурсов, а на их зависимость между собой через срабатывания переходов в сети. Рассматриваются подмножества множества пар подобных ресурсов, замкнутые относительно срабатываний.

Вводится понятие бисимуляции ресурсов в обыкновенных сетях Петри. Рассматриваются алгоритмы проверки бисимулярности отношения и построения аппроксимации максимальной бисимуляции ресурсов.

3.2.1 Определение

В работах Ф. Шнобелена и др. ([26, 27]) при определении вычислимых сохраняющих бисимулярность отношений на множестве позиций сети Петри был использован способ замыкания отношения относительно срабатываний переходов. Отношение на множестве разметок замкнуто относительно срабатываний переходов, если оно является бисимуляцией разметок.

Применим этот подход для случая ресурсов.

Определение 3.2. Пусть $N = (P, T, F, l)$ — помеченная сеть Петри. Симметричное 1-рефлексивное отношение $B \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$ называется *бисимуляцией ресурсов* сети N , если B^{AT} есть бисимуляция разметок сети N .

Определение 3.2 достаточно абстрактно, прежде чем его прокомментировать, нам придется привести важное свойство бисимуляций ресурсов, показывающее их связь с подобием ресурсов.

Теорема 3.2. Пусть $N = (P, T, F, l)$ — помеченная сеть Петри, $B \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$ — бисимуляция ресурсов сети N и $(r, s) \in B$. Тогда $r \approx s$.

Другими словами, любые бисимулярные ресурсы подобны.

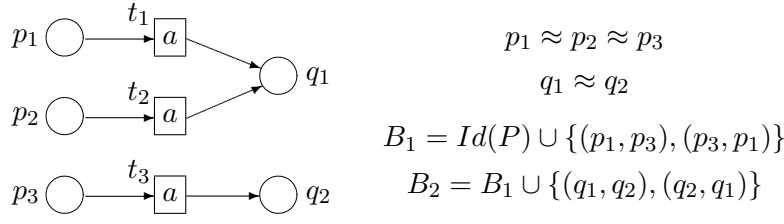


Рис. 3.3: Отношение B_1 — не бисимуляция ресурсов, отношение B_2 — бисимуляция ресурсов

Доказательство. Предположим противное: $r \not\approx s$. Тогда существует разметка $M \in \mathcal{M}(P)$, где $r \subseteq M$, замена в которой ресурса r на s дает небисимулярную разметку: $M \not\sim M - r + s$.

Однако из 1-рефлексивности отношения B и аддитивной замкнутости отношения B^{AT} для любой разметки $M' \in \mathcal{M}(P)$ имеем $(M' + r, M' + s) \in B^{AT}$. В качестве M' мы можем взять $M - r$, тогда $(M, M - r + s) \in B^{AT}$.

Мы знаем (утверждение 1.3), что максимальная бисимуляция разметок $\sim$ содержит в себе все прочие. Следовательно, она содержит и B^{AT} , то есть $B^{AT} \subseteq (\sim)$. Но тогда $M \sim M - r + s$ — противоречие. $\square$

Таким образом, всякая бисимуляция ресурсов есть сужение подобия ресурсов. Однако не всякое сужение подобия ресурсов есть бисимуляция ресурсов. Необходимо также, чтобы это отношение было замкнуто относительно срабатываний переходов, то есть было еще и бисимуляцией разметок.

Пример 3.1. Рассмотрим сеть, изображенную на рисунке 3.3. Отношение B_1 является подмножеством подобия ресурсов, однако оно не замкнуто относительно срабатывания переходов.

Рассмотрим пару разметок $(p_1, p_3) \in (B_1)^{AT}$. Срабатывание $p_1 \xrightarrow{t_1} q_1$ может быть имитировано при разметке p_3 только срабатыванием перехода t_3 : $p_3 \xrightarrow{t_3} q_2$. Однако пара (q_1, q_2) не при-

надлежит отношению $(B_1)^{AT}$. Следовательно, $(B_1)^{AT}$ не является бисимуляцией разметок, и B_1 не является бисимуляцией ресурсов.

Дополним отношение B_1 парами (q_1, q_2) и (q_2, q_1) . Полученное отношение B_2 является отношением бисимуляции ресурсов, так как $(B_2)^{AT}$ является бииммуляцией разметок.

Заметим, что отношение B_2 всё ещё не содержит всех пар подобных ресурсов (например, оно не содержит пару $p_1 \approx p_2$).

Итак, *бисимуляция ресурсов — это множество пар подобных ресурсов, являющееся к тому же бисимуляцией разметок*. Это определение является на самом деле всего лишь переформулировкой определения 3.2. Далее нам будет удобнее использовать именно определение 3.2, поскольку в нём бисимуляция ресурсов определена напрямую через бисимуляцию разметок, без упоминания термина “подобие”.

Следующая теорема описывает основные свойства бисимуляции ресурсов.

Теорема 3.3. Пусть $N = (P, T, F, l)$ — помеченная сеть Петри. Тогда

1. если $B_1, B_2 \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$ — бисимуляции ресурсов, то $B_1 \cup B_2$ — бисимуляция ресурсов;
2. для сети N существует максимальная по вложению бисимуляция ресурсов (обозначается как $\simeq$);
3. $(\simeq)$ является отношением эквивалентности.

Доказательство. 1) Пусть $B = B_1 \cup B_2$. Покажем, что B^{AT} — бисимуляция разметок.

Предположим противное: свойство переноса для B^{AT} не выполняется, то есть существуют $(M_1, M_2) \in B^{AT}$, $t \in T$, где $M_1 \xrightarrow{t} M'_1$, такие, что не найдется моделирующего перехода $u \in T$ с той же меткой $l(t) = l(u)$, для которого бы выполнялось $M_2 \xrightarrow{u} M'_2$ и $(M'_1, M'_2) \in B^{AT}$.

Рассмотрим пару разметок $(M_1, M_2) \in B^{AT}$. Согласно лемме 2.3 она получена транзитивным замыканием нескольких пар из B^A — аддитивного замыкания B :

$$(H_1, H_2), (H_2, H_3), \dots, (H_{k-1}, H_k) \in B^A, \text{ где } H_1 = M_1, H_k = M_2.$$

Рассмотрим пару (H_1, H_2) .

$$(H_1, H_2) = (L_1 + L_2, R_1 + R_2), \text{ где } (L_1, R_1) \in (B_1)^{AT}, (L_2, R_2) \in (B_2)^{AT}$$

В силу 1-рефлексивности B_1 в $(B_1)^{AT}$ входит пара $(L_1, R_1) + (L_2, L_2) = (H_1, R_1 + L_2)$. Заметим, что $H_1 = M_1$, тогда в силу свойства переноса для $(B_1)^{AT}$ существует переход v , имитирующий срабатывание перехода t при разметке $R_1 + L_2$:

$$R_1 + L_2 \xrightarrow{v} G_1, \text{ где } l(v) = l(t) \text{ и } (M'_1, G_1) \in (B_1)^{AT}.$$

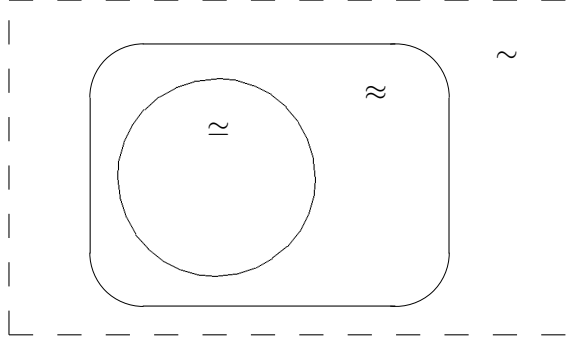
Аналогично, в силу 1-рефлексивности отношения B_2 в $(B_2)^{AT}$ входит пара $(L_2, R_2) + (R_1, R_1) = (R_1 + L_2, H_2)$. В силу свойства переноса для $(B_2)^{AT}$ существует переход w , имитирующий срабатывание перехода v при разметке H_2 :

$$H_2 \xrightarrow{w} G_2, \text{ где } l(w) = l(v) = l(t) \text{ и } (G_1, G_2) \in (B_2)^{AT}.$$

Очевидно, что $(B_1)^{AT} \in B^{AT}$ и $(B_2)^{AT} \in B^{AT}$. Кроме того, отношение B^{AT} транзитивно замкнуто, так что мы можем замкнуть пары (M'_1, G_1) и (G_1, G_2) , получив пару $(M'_1, G_2) \in B^{AT}$. Итак, срабатывание перехода w имитирует срабатывание перехода t в смысле B^{AT} .

Повторив это построение последовательно для всех пар вида (H_i, H_{i+1}) , мы получим в итоге искомый переход u , имитирующий t при разметке M_2 .

2) Второе утверждение является непосредственным следствием первого, поскольку максимальная бисимуляция ресурсов для N есть объединение всех бисимуляций ресурсов.

Рис. 3.4: Взаимная вложенность отношений на $\mathcal{M}(P)$

3) Известно, что $Id(\mathcal{M}(P))$ — бисимуляция разметок (утверждение 1.3, часть 1). Очевидно, что $(Id(\mathcal{M}(P)))^{AT} = Id(\mathcal{M}(P))$, следовательно, отношение $Id(\mathcal{M}(P))$ является также бисимуляцией ресурсов и содержится в $(\simeq)$. Отсюда $(\simeq)$ — рефлексивно.

Пусть B — некоторая бисимуляция ресурсов, B^{AT} — соответствующая бисимуляция разметок. Из утверждения 1.3(2) следует, что отношение $(B^{AT})^{-1}$ также является бисимуляцией разметок. Очевидно, что $(B^{-1})^{AT} = (B^{AT})^{-1}$, следовательно, B^{-1} — бисимуляция ресурсов. Объединение бисимуляций ресурсов является бисимуляцией ресурсов (второе утверждение данной теоремы), следовательно, симметричное отношение $B \cup B^{-1}$ является бисимуляцией ресурсов. Таким же образом мы могли бы из $(\simeq)$ получить симметричное отношение $(\simeq) \cup (\simeq)^{-1}$, следовательно, $(\simeq)$ симметрично в силу своей максимальности.

Заметим, что $(\simeq) = (\simeq)^{AT}$. Следовательно, максимальная бисимуляция ресурсов $(\simeq)$ транзитивна. $\square$

Очевидно, что максимальная бисимуляция ресурсов $(\simeq)$ бесконечна и содержится в максимальной бисиммуляции разметок. Однако в силу теоремы 2.1 даже максимальная бисимуляция ресур-

сов может быть представлена конечным числом пар, то есть конечной бисимуляцией ресурсов.

Взаимная вложенность максимальной бисимуляции разметок, подобия ресурсов и максимальной бисимуляции ресурсов изображена на рисунке 3.4. Сплошной линией обозначены отношения, обладающие конечным AT -базисом (в силу своей аддитивной и транзитивной замкнутости).

3.2.2 Слабое свойство переноса

Как и любая бисимуляция разметок, AT -замыкание бисимуляции ресурсов обладает свойством переноса. Однако хорошая структурированность бисимуляции ресурсов (по сравнению с произвольной бисимуляцией разметок) позволяет использовать и более слабый (локальный) вариант свойства переноса — когда для перехода t рассматриваются только те разметки, при которых он может сработать.

Определение 3.3. Отношение $B \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$ обладает *слабым свойством переноса*, если для всех $(r, s) \in B$, $t \in T$, таких что $\bullet t \cap r \neq \emptyset$, найдется имитирующий переход $u \in T$, такой что $l(t) = l(u)$ и, обозначив $M_1 = \bullet t \cup r$ и $M_2 = \bullet t - r + s$, мы имеем $M_1 \xrightarrow{t} M_1'$ и $M_2 \xrightarrow{u} M_2'$, где $(M_1', M_2') \in B^{AT}$.

Слабое свойство переноса может быть представлено в виде следующей диаграммы:

$$\begin{array}{ccc}
 & r & B & s \\
 \bullet t \cup r & & & \bullet t - r + s \\
 \\
 \downarrow t & & & \downarrow (\exists)u, l(u) = l(t) \\
 \\
 M_1' & \sim_{B^{AT}} & & M_2'
 \end{array}$$

Поясним данную диаграмму. Пусть имеются ресурсы r и s , связанные отношением B . Ресурс r при этом пересекается с пред-условием перехода t . Обозначим результат срабатывания перехода t как M'_1 :

$$\bullet t \cup r \xrightarrow{t} M'_1.$$

Заменяв в разметке $\bullet t \cup r$ ресурс r на ресурс s , получим новую разметку $\bullet t - r + s$. Слабое свойство переноса состоит в том, что обязательно найдется переход u с той же меткой $l(t)$, который может сработать от разметки $\bullet t - r + s$:

$$\bullet t - r + s \xrightarrow{u} M'_2,$$

и при этом результат его срабатывания — разметка M'_2 — будет связана с разметкой M'_1 отношением B^{AT} .

Теорема 3.4. *Симметричное и 1-рефлексивное отношение $B \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$ обладает слабым свойством переноса тогда и только тогда, когда B — бисимуляция ресурсов.*

Доказательство. ($\Rightarrow$) Докажем по схеме, использованной в доказательстве первого утверждения теоремы 3.3.

Предположим противное: свойство переноса для B^{AT} не выполняется, то есть существуют $(M_1, M_2) \in B^{AT}$, $t \in T$, где $M_1 \xrightarrow{t} M'_1$, такие, что не найдется моделирующего перехода $u \in T$ с той же меткой $l(t) = l(u)$, для которого бы выполнялось $M_2 \xrightarrow{u} M'_2$ и $(M'_1, M'_2) \in B^{AT}$.

Рассмотрим пару разметок $(M_1, M_2) \in B^{AT}$. Согласно лемме 2.3 она получена транзитивным замыканием нескольких пар из B^A — аддитивного замыкания B :

$$(H_1, H_2), (H_2, H_3), \dots, (H_{k-1}, H_k) \in B^A, \text{ где } H_1 = M_1, H_k = M_2.$$

Рассмотрим пару (H_1, H_2) .

$$(H_1, H_2) = (r_1 + r_2 + \dots + r_l, s_1 + s_2 + \dots + s_l), \text{ где } (r_i, s_i) \in B.$$

$H_1 = \bullet t \cup r_1 + F_1$. В силу слабого свойства переноса для пары (r_1, s_1) существует переход v , такой, что $l(t) = l(v)$, $\bullet t \cup r_1 \xrightarrow{t} G_1$ и $\bullet t - r_1 + s_1 \xrightarrow{v} G_2$, где $(G_1, G_2) \in B^{AT}$.

Так как $\bullet t \cup r_1 \subseteq H_1$, к пред- и постусловиям срабатываний мы можем добавить ресурс $F = H_1 - \bullet t \cup r_1$:

$$\begin{array}{l} \bullet t \cup r_1 + F \xrightarrow{t} G_1 + F \\ \bullet t - r_1 + s_1 + F \xrightarrow{v} G_2 + F \end{array}$$

В силу 1-рефлексивности B и аддитивной замкнутости B^{AT} полученная пара разметок также разложима по $B : (G_1 + F, G_2 + F) \in B^{AT}$.

Получаем новая разметка $H'_1 = \bullet t - r_1 + s_1 + F = H_1 - r_1 + s_1$. Заметим, что она содержит $r_2 + \dots + r_l$. То есть можно еще раз применить использованный выше прием для замены ресурса r_2 на бисимулярный ресурс s_2 .

Применив его еще в общей сложности $l - 1$ раз и используя свойство транзитивной замкнутости B^{AT} , в итоге получим переход w , имитирующий t при разметке H_2 .

Перейдем к следующей паре (H_2, H_3) , повторим все для перехода w . И т.д. вплоть до последней пары (H_{k-1}, H_k) . В итоге мы получим переход u , имитирующий t при разметке $H_k = M_2$.

($\Leftarrow$) Очевидно, так как слабое свойство переноса — это свойство переноса, ограниченное конечным числом пар. $\square$

3.2.3 Проверка бисимулярности отношения

Итак, для определения того, является ли данное конечное симметричное 1-рефлексивное отношение B бисимуляцией ресурсов, достаточно проверить выполнение слабого свойства переноса для конечного числа пар ресурсов.

Рассмотрим алгоритм, определяющий, является или нет данное конечное отношение на множестве ресурсов данной сети Петри бисимуляцией ресурсов.

Алгоритм 3.1. (проверки бисимулярности)

ввод: Помеченная сеть Петри $N = (P, T, F, l)$, конечное симметричное 1-рефлексивное отношение $B \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$.

вывод: *True*, если B является для N бисимуляцией ресурсов, *False* в противном случае.

шаг 1: Построим B_s — простой базис отношения B (при помощи алгоритма 2.2). Множество B_s распадается на два непересекающихся подмножества — множество рефлексивных пар B_s^r и множество нерефлексивных пар B_s^{nr} .

шаг 2: Проверим, обладает ли B_s слабым свойством переноса. Достаточно перебрать только элементы из B_s^{nr} (рефлексивные пары слабым свойством переноса обладают всегда).

- Если все пары обладают слабым свойством переноса, возвращаем *True*.
- В противном случае существуют $(r, s) \in B_s^{nr}$, $t \in T$, где $\bullet t \cap r \neq \emptyset$ и срабатывание $M_1 \xrightarrow{t} M_1'$, где $M_1 = \bullet t \cup r$, не может быть имитировано срабатыванием какого-либо перехода с той же меткой и предусловием $M_2 = \bullet t - r + s$, то есть либо

1) Не существует перехода с меткой $l(t)$, готового к срабатыванию при разметке M_2 ;

либо

1) Для любого перехода $u \in T$, такого, что $M_2 \xrightarrow{u} M_2'$ и $l(u) = l(t)$, выполняется $(M_1', M_2') \notin (B_s)^{AT}$ (для проверки принадлежности пары AT -замыканию используем алгоритм 2.1).

Возвращаем *False*.

Приведенный алгоритм возвращает *True* в единственном случае — когда рассматриваемое отношение обладает слабым свойством переноса, то есть является бисимуляцией ресурсов. В силу

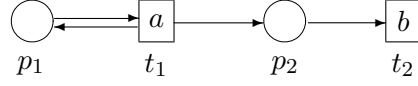


Рис. 3.5: Пример сети Петри

конечности перебираемых множеств B и T алгоритм завершит свою работу за конечное число шагов.

Оценим трудоемкость алгоритма. Как было показано ранее, трудоемкость алгоритма 2.2 построения основного базиса

$$CBuild(B) = O(|X||B|^4),$$

где X — множество, над которым определены мультимножества (в рассматриваемом случае это множество позиций P). Следовательно, трудоемкость первого шага алгоритма 3.1 —

$$O(|P||B|^4).$$

На втором шаге происходит перебор всех элементов множества B_s^{nr} , при этом на каждой итерации происходит перебор всевозможных пар переходов сети и для каждой пары производится проверка принадлежности постусловий их срабатываний отношению B^{AT} (трудоемкость $CTest(B_s) = O(|X||B_s|^2)$). Следовательно, трудоемкость второго шага алгоритма 3.1 можно оценить как

$$O(|T|^2|P||B_s|^3).$$

Итого, общую трудоемкость алгоритма 3.1 можно оценить как

$$O(\max\{|P||B|^4, |T|^2|P||B_s|^3\}).$$

Рассмотрим примеры работы алгоритма.

Пример 3.2. Пусть N — сеть Петри, изображенная на рисунке 3.5,

$$B = \{ ((1, 0), (1, 0)), ((0, 1), (0, 1)), \\ ((1, 0), (2, 0)), ((2, 0), (1, 0)), \\ ((7, 1), (4, 1)), ((4, 1), (7, 1)) \}.$$

На первом шаге алгоритма получим

$$B_s = \{ ((1, 0), (1, 0)), ((0, 1), (0, 1)), ((1, 0), (2, 0)), ((2, 0), (1, 0)) \}.$$

На втором шаге алгоритма проверим выполнение слабого свойства переноса для всех нерелексивных пар ресурсов из B_s — то есть для пары $((1, 0), (2, 0))$. Необходимо убедиться в наличии перехода, имитирующего срабатывание $(1, 0) \xrightarrow{t_1} (1, 1)$ при разметке $(2, 0)$. Очевидно, что таким переходом является сам переход t_1 :

$$(2, 0) \xrightarrow{t_1} (2, 1) \wedge ((1, 1), (2, 1)) \in (B_s)^{AT}.$$

Следовательно, возвращаем True — отношение B является отношением бисиммуляции ресурсов.

Пример 3.3. Пусть N — сеть Петри, изображенная на рисунке 3.5,

$$B = \{ ((1, 0), (1, 0)), ((0, 1), (0, 1)), ((1, 0), (0, 1)), ((0, 1), (1, 0)) \}.$$

На первом шаге алгоритма получим $B_s = B$. На втором шаге алгоритма проверим выполнение слабого свойства переноса для всех нерелексивных пар ресурсов из B_s — то есть для пары $((1, 0), (0, 1))$. Рассмотрим срабатывание $(1, 0) \xrightarrow{t_1} (1, 1)$. Оно не может быть имитировано срабатыванием перехода с меткой a при разметке $(0, 1)$ — так как возбужден только переход t_2 с меткой b . Следовательно, слабое свойство переноса не выполняется — возвращаем False — отношение B не является отношением бисиммуляции ресурсов.

3.2.4 Построение аппроксимации максимальной бисиммуляции ресурсов

Необходимо уметь не только проверять заданное отношение на бисиммулярность. Хотелось бы уметь строить по данной сети Петри

максимальную бисимуляцию ресурсов (отношение $(\simeq)$). В частности, проверка бисимулярности двух ресурсов сводится к проверке принадлежности данной пары отношению $(\simeq)$ (которое не известно заранее). Однако существование алгоритма построения $(\simeq)$ до сих пор остается открытой проблемой.

Одним из вариантов приближенного решения является построение аппроксимации максимальной бисимуляции ресурсов данной сети. Если рассматривать не все множество ресурсов сети (по определению бесконечное), а только его конечное подмножество (например, только ресурсы, чья мощность не превышает заданный параметр), то конечным становится и число пар в рассматриваемых отношениях, и мы можем использовать проверку слабого свойства переноса.

Пусть $N = (P, T, F, l)$ — помеченная сеть Петри, $q \in \text{Nat}$ — некоторый параметр. Через $\mathcal{M}_q(P)$ обозначим множество всех ресурсов, содержащих не более чем q фишек во всех позициях сети:

$$\mathcal{M}_q(P) = \{r \in \mathcal{M}(P) : |r| < q\}.$$

Легко убедиться, что существует наибольшая бисимуляция ресурсов над $\mathcal{M}_q(P)$, определяемая как объединение всех бисимуляций ресурсов над $\mathcal{M}_q(P)$. Обозначим ее как $B(N, q)$.

Поскольку множество $\mathcal{M}_q(P)$ конечно, мы можем использовать слабое свойство переноса для вычисления $B(N, q)$.

Приведем алгоритм, вычисляющий для данной сети Петри N максимальную бисимуляцию над множеством ресурсов, мощность которых не превышает данного параметра q . Получаемое отношение является корректной аппроксимацией максимальной бисимуляции ресурсов сети N .

Алгоритм 3.2. (*построения аппроксимации максимальной бисимуляции ресурсов*)

ввод: Помеченная сеть Петри $N = (P, T, F, l)$, параметр $q \in \text{Nat}$.

вывод: Отношение $B(N, q)$.

шаг 1: Пусть $C = \{(\emptyset, \emptyset)\}$ — пустое множество пар (рассматриваемое как отношение над $\mathcal{M}_q(P)$).

шаг 2: Построим $B = (\mathcal{M}_q(P) \times \mathcal{M}_q(P)) \setminus C$. В силу конечности $\mathcal{M}_q(P)$ множество пар B также конечно.

шаг 3: Построим B_s — основной базис B (при помощи алгоритма 2.2).

шаг 4: Проверим, обладает ли B_s слабым свойством переноса. Для этого переберем элементы из B_s^{nr} — подмножества нереклексивных элементов.

- Если все пары обладают слабым свойством переноса, то возвращаем B — искомую бисимуляцию.
- В противном случае существуют $(r, s) \in B_s^{nr}$, $t \in T$, где
 - $t \cap r \neq \emptyset$, такие, что срабатывание $M_1 \xrightarrow{t} M_1'$, где $M_1 =$
 - $t \cup r$, не может быть имитировано срабатыванием какого-либо перехода с той же меткой и предусловием $M_2 = \bullet t - r + s$. В этом случае добавляем (r, s) и (s, r) в C и возвращаемся на шаг 2.

Оценим трудоемкость алгоритма. Обозначим $\mathcal{R} = |\mathcal{M}_q(P)|$ — мощность рассматриваемого множества ресурсов.

По сути дела алгоритм 3.2 — это тот же алгоритм проверки бисимулярности 3.1, последовательно применяемый к уменьшающемуся на каждой итерации отношению B . Общее число таких итераций не может превысить $\mathcal{R}$. Тогда из оценки трудоемкости алгоритма 3.1 и из того факта, что $|B| \leq \mathcal{R}^2$, следует, что алгоритм 3.2 имеет трудоемкость

$$O(\max\{|P| \mathcal{R}^9, |T|^2 |P| \mathcal{R}^7\}).$$

Рассмотрим пример работы алгоритма 3.2.

Пример 3.4. Пусть N — сеть, изображенная на рисунке 3.6, $q = 2$.

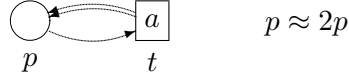


Рис. 3.6: Пример сети Петри

Имеем $\mathcal{M}_q(P) = \{0, 1, 2\}$ (ресурсы задаются как числа и соответствуют количеству фишек в единственной позиции p).

Протокол работы алгоритма:

1. $C = \emptyset$,
 $B = \{ (0, 0), (0, 1), (0, 2),$
 $(1, 0), (1, 1), (1, 2),$
 $(2, 0), (2, 1), (2, 2) \}.$

Применив алгоритм 2.2, получим

$$B_s = \{ (0, 0), (1, 1), (0, 1), (1, 0) \}.$$

Проверим выполнение слабого свойства переноса. Оно не выполняется для пары $(1, 0)$, так как срабатывание $(1) \xrightarrow{a} (2)$ не может быть имитировано при разметке (0) (нет ни одного возбужденного перехода). Добавим пары $(0, 1)$ и $(1, 0)$ в C .

2. $C = \{ (0, 1), (1, 0) \},$
 $B = \{ (0, 0), (0, 2),$
 $(1, 1), (1, 2),$
 $(2, 0), (2, 1), (2, 2) \}.$

Применив алгоритм 2.2, получим

$$B_s = \{ (0, 0), (1, 1), (0, 2), (2, 0), (1, 2), (2, 1) \}.$$

Проверим выполнение слабого свойства переноса. Оно не выполняется для пары $(2, 0)$, так как срабатывание $(2) \xrightarrow{a} (3)$

не может быть имитировано при разметке (0) (нет ни одного возбужденного перехода). Добавим пары (0, 2) и (2, 0) в C .

3. $C = \{ (0, 1), (1, 0), (0, 2), (2, 0) \},$
 $B = \{ (0, 0),$
 $(1, 1), (1, 2),$
 $(2, 1), (2, 2) \}.$

Применив алгоритм 2.2, получим

$$B_s = \{ (0, 0), (1, 1), (1, 2), (2, 1) \}.$$

Проверим выполнение слабого свойства переноса. Оно выполняется для любой пары ресурсов из B_s . Возвращаем B_s .

Итак, применив алгоритм 3.2, мы получили

$$B(N, 2) = \{ (0, 0), (1, 1), (1, 2), (2, 1) \}.$$

Увеличивая параметр q , мы можем получать более точные приближения максимальной бисимуляции ($\simeq$). В силу конечности основного базиса любой бисимуляции ресурсов (в том числе и максимальной) для всякой помеченной сети N найдется такой параметр $q_0(N)$, при котором $B(N, q_0(N)) = (\simeq)$. Например, для сети из предыдущего примера выполняется $q_0(N) = 2$, то есть мы получили именно максимальную бисимуляцию.

3.2.5 Гипотеза о неразрешимости

Существование оценки для $q_0(N)$ напрямую связано с разрешимостью максимальной бисимуляции ресурсов:

Утверждение 3.4. *Универсальная оценка для $q_0(N)$ существует в том и только том случае, если бисимуляция ресурсов в сетях Петри разрешима, то есть существует алгоритм, определяющий для данной помеченной сети $N = (P, T, F, l)$ и пары ресурсов $r, s \in \mathcal{M}(P)$, выполняется или нет $(r, s) \in (\simeq)$.*

Доказательство. ($\Rightarrow$) Если известно $q_0(N)$, то для построения $(\simeq)$ можно применить алгоритм 3.2 при $q = q_0(N)$.

($\Leftarrow$) Если известна бисимуляция $(\simeq)$, то она однозначно определяет $q_0(N)$ как максимальную мощность ресурса, входящего в основной базис отношения $(\simeq)$. $\square$

До настоящего времени вопрос о разрешимости бисимуляции ресурсов остается открытым. Ниже приводится гипотеза, которая кажется наиболее вероятной (хотя и достаточно неожиданной).

Гипотеза 3.1. 1. Для любой помеченной сети Петри N выполняется $(\simeq) = (\approx)$, то есть максимальная бисимуляция ресурсов совпадает с подобием ресурсов.

2. Бисимуляция ресурсов в сетях Петри неразрешима.

Таким образом, в соответствии с этой гипотезой на диаграмме, изображенной на рисунке 3.4, множества $\approx$ и $(\simeq)$ должны совпадать. Доказать первый пункт пока не удалось. Но и контрпример также пока не найден. Контрпримером являлась бы пара подобных ресурсов, не принадлежащих максимальной бисимуляции ресурсов. Второй пункт непосредственно следует из первого, так как подобие ресурсов неразрешимо (следствие 2.1).

Если приведенная гипотеза верна, то описанный в предыдущей главе способ приближения максимальной бисимуляции ресурсов (при помощи параметризованной аппроксимации) — единственно возможный, так как универсальной оценки для $q_0(N)$ не может существовать. Складывается очень интересная картина — имея некоторое отношение $B \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$, мы можем сказать, является ли это отношение бисимуляцией ресурсов (алгоритм 3.1), но не можем сказать, является ли оно *максимальной* бисимуляцией ресурсов.

Глава 4

Расширения подобия ресурсов

4.1 Условное подобие ресурсов

Рассмотрим еще один способ определения корректного с точки зрения бисимулярности отношения на $\mathcal{M}(P)$. При этом способе мы выделяем некоторую разметку (начальное условие) и рассматриваем все возможные пары ресурсов, которые одинаково влияют на поведение сети *при условии наличия данного начального ресурса*.

В этом разделе определяется и исследуется понятие условного подобия ресурсов, с его помощью доказывается полулинейность множества пар подобных ресурсов.

4.1.1 Определение условного подобия

Определение 4.1. Пусть $r, s, b \in \mathcal{M}(P)$. Ресурсы r и s называются *подобными при условии b* (обозначается $r \approx|_b s$), если для любого ресурса $t \in \mathcal{M}(P)$, такого, что $b \subseteq t$, выполняется $t + r \sim t + s$.

Ресурсы r и s называются *условно подобными* (обозначается

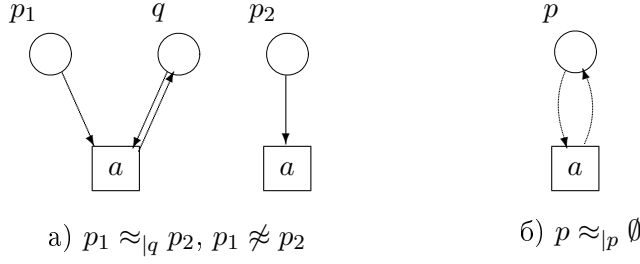


Рис. 4.1: Примеры условно подобных ресурсов

$r \approx| s$), если существует ресурс $b' \in \mathcal{M}(P)$, такой, что $r \approx|_{b'} s$.

Условное подобие ресурсов обладает естественной интерпретацией. В некоторых ситуациях использовать именно условное подобие удобнее, чем обычное подобие ресурсов.

Рассмотрим сеть, изображенную на рисунке 4.1(а). Ресурсы p_1 и p_2 не подобны, так как при разметке p_1 ни один из переходов не может сработать, тогда как при разметке p_2 может сработать переход a . С другой стороны, они подобны *при условии* q , то есть при наличии в сети ресурса q ресурсы p_1 и p_2 полностью взаимозаменяемы.

Еще один пример приведен на рисунке 4.1(б). Очевидно, что в данной сети Петри любое количество фишек в позиции p может быть заменено любым другим ненулевым количеством фишек, но только *при условии* наличия в позиции p еще одной фишки.

Условное подобие ресурсов улавливает те же структуры во множестве состояний сети Петри, что и обычное подобие ресурсов. Однако, благодаря отдельному рассмотрению общей части (условия), с его помощью проще исследовать некоторые аспекты подобия ресурсов. В частности, с использованием условного подобия мы докажем полулинейность множества пар подобных ресурсов.

4.1.2 Свойства условного подобия ресурсов

Следующее утверждение устанавливает некоторые важные свойства условного подобия ресурсов:

Утверждение 4.1. Пусть $r, s, b, b', m, m' \in \mathcal{M}(P)$. Тогда

1. $m + r \approx m + s \Leftrightarrow r \approx_{|m} s$.
2. $m \approx_{|m'} m' \wedge r \approx_{|s} s \Rightarrow m + r \approx_{|m'} m' + s$.
3. $r \approx_{|b} s \wedge b \subseteq b' \Rightarrow r \approx_{|b'} s$.
4. $m + r \approx_{|b} m + s \Leftrightarrow r \approx_{|b+m} s$.
5. $m + r \approx_{|m} m + s \Leftrightarrow r \approx_{|s} s$.
6. $m \approx m' \wedge m + r \approx m' + s \Rightarrow r \approx_{|s} s$.
7. $m \approx_{|b} m' \wedge m + r \approx_{|b'} m' + s \Rightarrow r \approx_{|s} s$.
8. $m \approx_{|b} m' \wedge r \approx_{|b'} r' \Rightarrow m + r \approx_{|b \cup b'} m' + r'$.

Доказательство. 1), 3), 4) Непосредственно из определений.

2) Пусть $m \approx_{|b} m'$ и $r \approx_{|b'} s$. Тогда из первого утверждения получим

$$m + b \approx m' + b \text{ и } r + b' \approx s + b'.$$

По свойству подобия ресурсов мы можем сложить левые и правые части:

$$m + r + b + b' \approx m' + s + b + b'.$$

Применив утверждение 4.1.1 еще раз, получим

$$m + r \approx_{|b+b'} m' + s.$$

5) Непосредственное следствие утверждения 4.1.4.

6) По свойству подобия ресурсов из $m \approx m'$ и $m + r \approx m' + s$ получим

$$m + r \approx m + s, \text{ то есть } r \approx_{|m} s.$$

7) Из утверждения 4.1.1 получим

$$m + b \approx m' + b \text{ и } m + r + b' \approx m' + s + b'.$$

Поскольку подобие замкнуто относительно сложения, получим

$$m + b + b' \approx m' + b + b' \text{ и } m + r + b' + b \approx m' + s + b' + b.$$

Из утверждения 4.1.6 получим $r \approx_{|s}$.

8) Из $m + b \approx m' + b$ и $r + b' \approx r' + b'$ по свойству подобия получим

$$m + r + b \cup b' \approx m' + r' + b \cup b'. \quad \square$$

Итак, условное подобие ресурсов замкнуто относительно сложения. Кроме того, оно инвариантно относительно расширения условия. Утверждения 4.1.4 и 4.1.5 показывают, что общая часть может быть удалена из подобных ресурсов. Утверждения 4.1.6 и 4.1.7 показывают, что разность как пар подобных, так и пар условно подобных ресурсов является парой условно подобных ресурсов. Кроме того, условное подобие ресурсов замкнуто относительно сложения ресурсов и объединения условий (утверждение 4.1.8).

Отметим, что, в отличие от подобия ресурсов, условное подобие замкнуто относительно вычитания. Это свойство будет далее использовано при конструировании базиса отношения условного подобия ресурсов.

Следующее утверждение устанавливает связь между подобием ресурсов и условным подобием ресурсов:

Утверждение 4.2. Пусть $r, s, m, m' \in \mathcal{M}(P)$, $m \approx m'$. Тогда

$$m + r \approx m' + s \Leftrightarrow r \approx_{|m} s.$$

Доказательство. ($\Rightarrow$) Пусть $m + r \approx m' + s$. Поскольку $m \approx m'$, по свойству подобия получим $m + r \approx m + s$. Тогда из утверждения 4.1.1 получим $r \approx_{|m} s$.

($\Leftarrow$) Пусть $r \approx_{|m} s$. Из утверждения 4.1.1 получим $m + r \approx m + s$. Тогда из $m \approx m'$ по свойству подобия получим $m + r \approx m' + s$. $\square$

4.1.3 Полулинейность множества пар подобных ресурсов

Отношение $(\approx|)$ более хорошо структурировано, чем отношение $(\approx)$. В частности, оно замкнуто не только относительно сложения пар, но и относительно вычитания. Поэтому оно должно обладать более простым базисом. Построим такой базис.

Определение 4.2. Пусть $r, s, r', s', r'', s'' \in \mathcal{M}(P)$. Пара условно подобных ресурсов $r \approx| s$ называется *минимальной*, если ее нельзя представить как сумму двух других пар условно подобных ресурсов, то есть для любой непустой пары $r' \approx| s'$ из $r = r' + r''$ и $s = s' + s''$ следует $r = r'$ и $s = s'$.

Из утверждения 4.1.7 получаем

Следствие 4.1. *Любая пара условно подобных ресурсов может быть представлена в виде суммы минимальных пар условно подобных ресурсов.*

Поскольку пара условно подобных ресурсов может быть представлена как вектор длины $2|P|$ с целочисленными неотрицательными координатами, а минимальная пара условно подобных ресурсов минимальна также и относительно обычного покоординатного сравнения таких векторов, для любой помеченной сети Петри множество минимальных пар условно подобных ресурсов конечно. Следовательно,

Утверждение 4.3. *Множество всех пар условно подобных ресурсов является аддитивным замыканием конечного множества минимальных пар условно подобных ресурсов.*

Определение 4.3. Пара подобных ресурсов $r \approx s$ называется *минимальной*, если ее нельзя представить как сумму непустой пары подобных ресурсов и пары условно подобных ресурсов, то есть для любой непустой пары подобных ресурсов $r' \approx s'$ из $r = r' + r''$ и $s = s' + s''$ следует $r = r'$ и $s = s'$.

Из утверждений 4.1.6 и 4.3 имеем

Следствие 4.2. *Любая пара подобных ресурсов может быть представлена в виде суммы одной минимальной пары подобных ресурсов и нескольких минимальных пар условно подобных ресурсов.*

Заметим, что по свойству покоординатного сравнения векторов с целочисленными неотрицательными компонентами, для любой пары условно подобных ресурсов $r \approx_1 s$ множество ее минимальных условий (относительно покоординатного сравнения) конечно.

Обозначение 4.1. Пусть $R \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$ — некоторое подмножество множества пар условно подобных ресурсов ($r \approx_1 s$ для всех $(r, s) \in R$). Пусть

$$B = \{ (u, v) \in \mathcal{M}(P) \times \mathcal{M}(P) \mid u \approx v \wedge \forall (r, s) \in R \quad u + r \approx v + s \}$$

— множество всех общих условий для R . Через $Cond(R)$ обозначим множество минимальных элементов B (относительно покоординатного сравнения, рассматривая элементы B как вектора длины $2|P|$).

Заметим, что из утверждения 4.2 следует, что для любой пары $(u, v) \in Cond(R)$ оба ресурса u и v являются условиями для любой пары $(r, s) \in R$. Кроме того, в силу свойства покоординатного сравнения векторов, для любого R множество $Cond(R)$ конечно.

Обозначение 4.2. Пусть $u, v \in \mathcal{M}(P)$ и $u \approx v$. Через $S(u, v)$ обозначим множество всех потенциальных дополнений к паре (u, v) (корректных относительно подобия ресурсов):

$$S(u, v) = \{ (r, r') \in \mathcal{M}(P) \times \mathcal{M}(P) \mid u + r \approx v + r' \}.$$

Через $S_{\min}(u, v)$ обозначим множество всех минимальных относительно покоординатного сравнения элементов множества $S(u, v)$.

Утверждение 4.4. Пусть $u, v, u', v' \in \mathcal{M}(P)$ и $u \approx v$. Тогда

- 1) $S(u, v)$ является отношением эквивалентности;
- 2) $S(u, v)$ замкнуто относительно сложения;
- 3) $u \approx v, u' \approx v', (u, v) \leq (u', v') \Rightarrow S(u, v) \subseteq S(u', v')$;
- 4) $S_{\min}(u, v)$ конечно.

Доказательство. 1) Рефлексивность. Очевидно, что $u \approx v \Rightarrow u + r \approx v + r$ для любого r .

Симметричность.

Пусть $(r, s) \in S(u, v)$. Покажем, что $(s, r) \in S(u, v)$.

По определению $S(u, v)$ выполняется $u + r \approx v + s$. Из $u \approx v$, $s \approx s$ и свойства аддитивной замкнутости подобия имеем $u + s \approx v + s$. Замкнув пары $u + r \approx v + s$ и $u + s \approx v + s$ по транзитивности, получим $u + s \approx u + r$. По свойству подобных ресурсов мы можем заменить в правой части u на v : $u + s \approx v + r$.

Транзитивность.

Пусть $(r, s), (s, t) \in S(u, v)$. Покажем, что $(r, t) \in S(u, v)$.

Из симметричности $S(u, v)$ имеем $(t, s) \in S(u, v)$, то есть $u + t \approx v + s$. Замкнув по транзитивности эту пару и пару $u + r \approx v + s$, получим $u + r \approx u + t$. По свойству подобных ресурсов мы можем заменить в правой части u на v : $u + r \approx v + t$.

2) Пусть $(r, s), (r', s') \in S(u, v)$. Покажем, что $(r + r', s + s') \in S(u, v)$.

Из $u + r \approx v + s$ получим $u + r + r' \approx v + s + r'$. Из симметричности $S(u, v)$ следует $(s', r') \in S(u, v)$, то есть $u + s' \approx v + r'$. Тогда $u + s' + s \approx v + r' + s$. Замкнув пары по транзитивности, получим $u + r + r' \approx u + s + s'$. По свойству подобных ресурсов мы можем заменить в правой части u на v : $u + r + r' \approx v + s + s'$.

3) Обозначим $(u', v') = (u, v) + (w, w')$. Пусть $u + r \approx v + r'$ для некоторой пары (r, r') . Тогда $u + w + r \approx v + w + r' \approx u + w + r' \approx v + w' + r'$, то есть $u' + r \approx v' + r'$.

4) Из свойств покоординатного сравнения целочисленных векторов с неотрицательными коэффициентами. $\square$

Из первой и второй части утверждения следует, что множество $S(u, v)$ обладает конечным АТ-базисом. В частности, согласно теореме 2.1, оно обладает основным базисом, составленным из минимальных относительно $\sqsubseteq$ элементов.

Обозначение 4.3. Пусть N — помеченная сеть Петри. Через $A(N)$ обозначим множество всех множеств потенциальных дополнений в N :

$$A(N) = \{H \mid \exists(u, v) : u \approx v \wedge H = S(u, v)\}.$$

Утверждение 4.5. *Множество $A(N)$ конечно для любой помеченной сети Петри N .*

Доказательство. Предположим противное: существует бесконечно много различных множеств потенциальных дополнений. Рассмотрим соответствующие пары подобных ресурсов.

Существует бесконечно много таких пар, следовательно, существует бесконечная возрастающая последовательность (u_i, v_i) пар подобных ресурсов, таких, что $S(u_i, v_i) \neq S(u_j, v_j)$ для любых $i \neq j$. Поскольку $(u_i, v_i) < (u_{i+1}, v_{i+1})$ для любого i , из утверждения 4.4.3 получим $S(u_i, v_i) \subset S(u_{i+1}, v_{i+1})$.

Напомним, что любое $S(u_i, v_i)$ обладает конечным основным базисом, состоящим из минимальных относительно $\sqsubseteq$ пар. Увеличение множества $S(u_i, v_i)$ может проявляться в его основном базисе двумя способами:

- 1) появлением новых элементов базиса (новых минимальных относительно $\sqsubseteq$ элементов $S(u_i, v_i)$) при сохранении минимальности старых;
- 2) исчезновением старых элементов базиса и появлением новых, строго меньших относительно $\sqsubseteq$.

Очевидно, что подобное изменение базиса не может продолжаться бесконечно. $\square$

Обозначение 4.4. Пусть $R \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$. Через $lc(R)$ обозначим множество всех *линейных комбинаций* над R :

$$lc(R) = \{(r, s) \mid (r, s) = (r_1, s_1) + \dots + (r_k, s_k) : \\ (r_i, s_i) \in R \forall i = 1, \dots, k\}.$$

Пусть также $S \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$. Через $R + S$ обозначим множество всех сумм пар из R и S :

$$R + S = \{(u, v) \mid (u, v) = (r + r', s + s') : (r, s) \in R, (r', s') \in S\}.$$

Определение 4.4. Множество R называется *линейным*, если выполняется $R = lc(R)$.

Множество R называется *полулинейным*, если оно является объединением конечного числа линейных множеств.

Теорема 4.1. Пусть N — помеченная сеть Петри, $(\approx)$ — множество всех пар подобных ресурсов в N , $(\approx|)$ — множество всех пар условно подобных ресурсов в N . Множество $(\approx)$ полулинейно: существует конечное множество $R \subseteq (\approx|)$, такое, что

$$(\approx) = \bigcup_{\mathcal{R} \in 2^R} [Cond(\mathcal{R}) + lc(\mathcal{R})],$$

где 2^R — множество всех подмножеств множества R .

Доказательство. $(\supseteq)$ Очевидно, что для всех $\mathcal{R} \subseteq (\approx|)$ выполняется $[Cond(\mathcal{R}) + lc(\mathcal{R})] \subseteq (\approx)$.

$(\subseteq)$ Рассмотрим некоторую пару $u \approx v$. Пусть (u', v') — минимальная пара ресурсов, такая, что

- $(u', v') \leq (u, v)$;
- $u' \approx v'$;
- $S(u', v') = S(u, v)$.

Докажем, что $(u, v) \in (u', v') + lc(S_{\min}(u', v'))$.

Пусть $(u_1, v_1) =_{def} (u - u', v - v')$. Тогда $u_1 \approx_1 v_1$ и найдется пара $(w_1, w'_1) \in S_{\min}(u', v')$, такая что $(w_1, w'_1) \leq (u_1, v_1)$. Если $(w_1, w'_1) = (u_1, v_1)$, то мы получили искомое разложение.

Пусть $(w_1, w'_1) < (u_1, v_1)$. Тогда выполняется $(u', v') < (u' + w_1, v' + w'_1) < (u' + u_1, v' + u'_1) = (u, v)$. Из $S(u', v') = S(u, v)$ получим $S(u' + w_1, v' + w'_1) = S(u, v)$.

Рассмотрим $(u_2, v_2) =_{def} (u_1 - w_1, v_1 - w'_1)$. Использованным ранее способом мы можем показать, что $u_2 \approx_1 v_2$ и, следовательно, существует пара $(w_2, w'_2) \in S_{\min}(u', v')$, такая что $(w_2, w'_2) \leq (u_2, v_2)$. Если $(w_2, w'_2) = (u_2, v_2)$, то мы получили искомое разложение. Если $(w_2, w'_2) < (u_2, v_2)$, то, повторив рассуждение еще раз, получим пары (u_3, v_3) и (w_3, w'_3) и т.д.

Поскольку $(u_1, v_1) > (u_2, v_2) > (u_3, v_3) > \dots$, на некотором шаге мы получим $(w_j, w'_j) = (u_j, v_j)$ и, следовательно,

$$(u, v) = (u', v') + (w_1, w'_1) + \dots + (w_j, w'_j) \in (u', v') + lc(S_{\min}(u', v')).$$

Докажем, что множество R конечно. Достаточно показать, что существует только конечное число пар-кандидатов на роль пары (u', v') в предыдущем рассуждении для всех возможных пар подобных ресурсов.

Согласно утверждению 4.5, существует конечное число различных множеств $S(u, v)$. По свойству покоординатного сравнения существует также только конечное число минимальных пар $(u', v') \in (\approx)$, для которых $S(u', v') = S(u, v)$. $\square$

Теорема демонстрирует взаимосвязь между обычным подобием ресурсов и условным подобием ресурсов. Подобие ресурсов представимо при помощи некоторого конечного набора пар условно подобных ресурсов (причем в виде полулинейного множества). Может возникнуть вопрос: нельзя ли использовать только *минимальные* условно подобные ресурсы в этом разложении? Действительно, это было бы гораздо удобнее. Однако, к сожалению, это



Рис. 4.2: Цикл со сдвоенными дугами

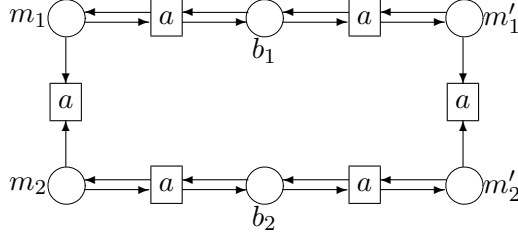


Рис. 4.3: Пример сети

невозможно. Только лишь минимальные условно подобные ресурсы не отражают всей структуры подобия ресурсов.

Рассмотрим в качестве примера сеть Петри, изображенную на рисунке 4.2. Минимальной парой условно подобных ресурсов для данной сети является пара $0 \approx_{|2} 1$. Одна фишка подобна любому числу фишек при условии наличия как минимум двух других фишек в единственной позиции сети. Однако существует и другая (не минимальная) условно подобная пара $1 \approx_{|1} 2$ с меньшим минимальным условием — только одной фишкой.

На рисунке 4.3 представлен другой пример, показывающий, что сумма минимальных условно подобных пар может иметь меньшее минимальное условие, чем слагаемые. Пары $m_1 \approx_{|b_1} m'_1$ и $m_2 \approx_{|b_2} m'_2$ являются минимальными парами условно подобных ресурсов, но пара $m_1 + m_2 \approx m'_1 + m'_2$ обладает пустым условием!

При аддитивном разложении подобных ресурсов неизбежно приходится учитывать не только минимальные условно подобные пары, но и некоторое количество других условно подобных пар (не обязательно минимальных). Выбор соответствующих пар зависит от раскладываемых ресурсов.

Итак, условное подобие сильно взаимосвязано с обычным подобием ресурсов. Кроме того, из следствия 2.1 и утверждения 4.1.1 получим очевидное

Следствие 4.3. *Проблема распознавания условного подобия ресурсов неразрешима для сетей Петри, то есть невозможно построить алгоритм, отвечающий на вопрос, являются ли данные ресурсы подобными при данном условии в данной сети Петри.*

Как и подобие ресурсов, условное подобие ресурсов в общем случае не может быть эффективно построено.

4.2 Расслоенное подобие ресурсов

Ограниченные подобия и бисимуляции ресурсов являются сужениями полного подобия ресурсов, то есть их можно рассматривать как его приближения “снизу”. В этом разделе мы рассмотрим способ приближения подобия ресурсов “сверху”, то есть при помощи расширения.

4.2.1 Расслоенная бисимуляция разметок

В литературе существует понятие расслоенной (stratified) бисимуляции разметок [51] (обозначается $\sim_n$). Оно определяется индуктивно:

Во-первых, полагаем $M_1 \sim_0 M_2$ для любых $M_1, M_2 \in \mathcal{M}(P)$. Далее, для любого $n \in \text{Nat}$ полагаем $M_1 \sim_{n+1} M_2$, если для любого $a \in \text{Act}$:

- если $M_1 \xrightarrow{a} M'_1$, то $M_2 \xrightarrow{a} M'_2$, причем $M'_1 \sim_n M'_2$; и
- если $M_2 \xrightarrow{a} M'_2$, то $M_1 \xrightarrow{a} M'_1$, причем $M'_1 \sim_n M'_2$.

Другими словами, n -расслоенная бисимуляция разметок — это бисимуляция разметок в том случае, когда нам не важно, что будет происходить с системой после n -го шага. Все последующее поведение сети просто игнорируется.

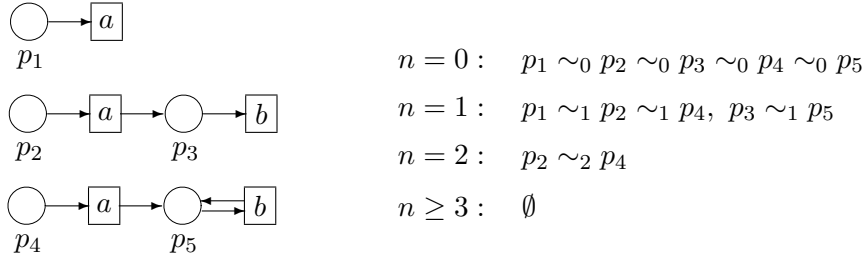


Рис. 4.4: Расслоенные бисимуляции разметок (перечислены только пары разметок, содержащих ровно одну фишку)

Известно, что для любого n отношение $\sim_n$ является отношением эквивалентности, кроме того, $(\sim_{n+1}) \subseteq (\sim_n)$. Также выполняется $M_1 \sim M_2 \Leftrightarrow M_1 \sim_n M_2$ для любого $n \in \text{Nat}$. Наибольшая бисимуляция разметок является пределом последовательности расслоенных бисимуляций: $(\sim) = (\sim_\infty)$.

На рисунке 4.4 приведен пример расслоенных бисимуляций разметок. При увеличении n количество пар в отношении уменьшается. Заметим, что на рисунке приведены не все пары разметок, а только те, которые содержат ровно одну фишку. На самом деле пар бисимулярных разметок существенно больше (например, $2p_3 \sim_2 p_5$). Пределом последовательности n -расслоенных бисимуляций разметок является достаточно простое отношение:

$$(\sim_\infty) = (\sim) = \{(M, M) \mid M \in \mathcal{M}(P)\} \cup \{(M + kp_3 + lp_5, M + mp_3 + np_5) \mid M \in \mathcal{M}(P), k, m \geq 0, l, n > 0\}.$$

Известно [51], что проблема n -расслоенной бисимулярности разметок разрешима для любого n .

4.2.2 Определение расслоенного подобия ресурсов

Определение 4.5. Пусть $n \in \text{Nat}$ — целое неотрицательное число, $N = (P, T, F, l)$ — помеченная сеть Петри, $(\sim_n)$ — n -расслоен-

ная бисимуляция разметок сети N . Через $(\approx_n)$ обозначим множество пар ресурсов, задающее на $(\sim_n)$ отношение подобия:

$$(\approx_n) =_{\text{def}} \{ (r, s) \in \mathcal{M}(P) \times \mathcal{M}(P) \mid$$

$$\forall M \in \mathcal{M}(P) (M + r, M + s) \in (\sim_n) \}.$$

Соответствующее отношение назовем *n -расслоенным подобием ресурсов*.

Для n -расслоенного подобия ресурсов верны аналоги утверждения 3.1 и следствия 3.1 (доказываются аналогично).

Ключевым отличием от уже рассмотренных ранее отношений на множестве ресурсов является тот факт, что при увеличении параметра n множество пар не увеличивается, а уменьшается:

Утверждение 4.6. *Для любого $n \in \text{Nat}$, $n > 0$, выполняется*

$$r \approx_{n+1} s \implies r \approx_n s.$$

Другими словами, $(\approx_{n+1}) \subseteq (\approx_n)$.

Доказательство. Из определений. □

Кроме того, n -расслоенное подобие ресурсов не содержится в полном подобии, а содержит его в себе:

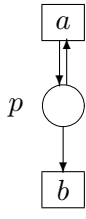
Утверждение 4.7. *Для любого $n \in \text{Nat}$ выполняется $(\approx) \subseteq (\approx_n)$.*

Доказательство. Следует из определений и свойства $(\sim) = (\sim_\infty) \subseteq (\sim_n)$. □

Предел у последовательности существует:

Утверждение 4.8. $(\approx) = (\approx_\infty)$.

Доказательство. Так как $(\sim) = (\sim_\infty)$. □



Обозначим $I = \{(\emptyset, \emptyset), (p, p)\}$, тогда:

$$(\approx) = I^{AT}$$

$$(\approx_0) = (I \cup \{(p, \emptyset), (\emptyset, p)\})^{AT}$$

$$(\approx_1) = (I \cup \{(2p, p), (p, 2p)\})^{AT}$$

...

$$(\approx_n) = (I \cup \{((n+1)p, np), (np, (n+1)p)\})^{AT}$$

Рис. 4.5: Пример сети, в которой для любого $n \in \text{Nat}$ выполняется $(\approx) \subset (\approx_n)$

То есть подобие ресурсов — предел расслоенного подобия ресурсов (как и в случае ограниченного подобия). Однако, в отличие от ограниченного подобия, последовательность не всегда стабилизируется:

Замечание 4.1. Существует сеть Петри $N = (P, T, F, l)$, такая что для любого $n \in \text{Nat}$ выполняется $(\approx) \subset (\approx_n)$. В качестве примера можно рассмотреть сеть на рисунке 4.5.

4.2.3 Свойства расслоенного подобия ресурсов

Рассмотрим некоторые свойства расслоенного подобия ресурсов.

Утверждение 4.9. $r \approx_0 s$ для любых r и s .

Доказательство. Из определений. □

Пусть $M \in \mathcal{M}(P)$. Через $\text{Act}(M)$ обозначим множество меток переходов, готовых к срабатыванию:

$$\text{Act}(M) =_{\text{def}} \{a \in \text{Act} \mid \exists t \in T : (M \xrightarrow{t} M' \wedge l(t) = a)\}.$$

Утверждение 4.10.

$$r \approx_1 s \iff \forall m \in \mathcal{M}(P) : \text{Act}(r + m) = \text{Act}(s + m).$$

Доказательство. Из определений. $\square$

Другими словами, для 1-расслоенного подобия двух ресурсов необходимо и достаточно того, чтобы множества активизируемых ими переходов обладали одинаковыми наборами меток. Следующее утверждение дает несколько более конструктивное толкование этого же свойства:

Утверждение 4.11. $r \approx_1 s \iff \forall t \in T$

1. $\exists t_1 \in T : l(t_1) = l(t) \ \& \ \bullet t_1 \subseteq (\bullet t \setminus r + s);$
2. $\exists t_2 \in T : l(t_2) = l(t) \ \& \ \bullet t_2 \subseteq (\bullet t \setminus s + r).$

Условия вида $(\bullet t \setminus r + s) \subseteq \bullet t_1$ фактически означают возможность срабатывания перехода t_1 .

Доказательство. ($\implies$) Докажем первое утверждение правой части (второе аналогично).

Предположим противное: найдется t , такой, что для любого t_1 с той же меткой $\bullet t_1 \not\subseteq (\bullet t \setminus r + s)$.

Рассмотрим разметку $r \cup \bullet t$. Из $r \approx_1 s$ следует

$$r \cup \bullet t \sim_1 r \cup \bullet t - r + s = (\bullet t \setminus r + r) - r + s = \bullet t \setminus r + s.$$

Из 1-бисимулярности $r \cup \bullet t$ и $\bullet t \setminus r + s$ следует существование имитирующего перехода u , такого, что $\bullet u \subseteq (\bullet t \setminus r + s)$ — противоречие.

($\impliedby$)

Предположим противное: $r \not\approx_1 s$, то есть найдется ресурс x , такой, что $r + x \not\sim_1 s + x$. Небисимулярность означает невозможность симулирования разметки $r + x$ разметкой $s + x$ или разметки $s + x$ разметкой $r + x$. Рассмотрим первую ситуацию (для второй аналогично):

Существует переход $r + x \xrightarrow{t} M'$, для которого не найдется перехода u с той же меткой, такого, что $s + x \xrightarrow{u} M''$ при $M' \sim_0 M''$.

Поскольку выполняется $M' \sim_0 M''$ для любых разметок, небисимулярность означает отсутствие перехода с меткой $l(t)$, активного при разметке $s + x$.

Имеем $r + x \xrightarrow{t}$, т.е. $r + x = \bullet t + y$, следовательно, $x = \bullet t \setminus r + z$. Подставляя получившееся выражение для x в $s + x$, получим

$$s + x = s + \bullet t \setminus r + z.$$

Из первого утверждения правой части следует существование имитирующего перехода t_1 , такого, что $l(t_1) = l(t)$ и при этом $\bullet t_1 \subseteq (\bullet t \setminus r + s)$ — противоречие. $\square$

Определение 4.6. Пусть $\sigma \in T^*$ — некоторая последовательность переходов сети N . Обозначим через $\bullet\sigma$ минимальную относительно вложения разметку, при которой последовательность σ может сработать.

Обозначим через $Mark(n)$ минимальную относительно вложения разметку, при которой могут сработать все последовательности длины n :

$$Mark(n) =_{def} \bigcup_{\forall \sigma \in T^* : |\sigma|=n} \bullet\sigma.$$

Очевидно, что $Mark(n) \subseteq Mark(n+1)$.

Утверждение 4.12. Верны следующие свойства $Mark(n)$:

1. $r \sim_n s \implies (r \cap Mark(n)) \sim_n (s \cap Mark(n))$;
2. $r \approx_n s \implies (r \cap Mark(n)) \approx_n (s \cap Mark(n))$.

Другими словами, если разметки (ресурсы) n -расслоенно бисимулярны (подобны) и при этом в каких-то позициях содержат фишек больше, чем $Mark(n)$, то лишние фишки можно убрать, и при этом бисимулярность (подобие) не нарушится.

Доказательство. Так как учитываются только последовательности переходов длины не более чем n , избыточные фишки не влияют ни на бисимуляцию, ни на подобие. $\square$

Таким образом, для построения n -расслоенного подобия достаточно перебрать только ресурсы, не превышающие по размеру $Mark(n)$. Их число конечно, следовательно,

Теорема 4.2. *n -расслоенное подобие ресурсов вычислимо.*

В качестве доказательства приведем следующий алгоритм.

Алгоритм 4.1. *(построения n -расслоенного подобия ресурсов)*

ввод: Помеченная сеть Петри $N = (P, T, F, l)$, параметр $n \in Nat$.

вывод: Отношение $(\approx_n)$.

шаг 1: Построим мультимножество $Mark(n)$. Для этого просмотрим все последовательности переходов длины n .

шаг 2: Построим $Bis(n)$ — множество пар n -расслоенно бисимулярных разметок, не превышающих $Mark(n)$. Для этого просмотрим все пары разметок, не превышающих $Mark(n)$, и все готовые сработать последовательности переходов длины не более n .

шаг 3: Положим $X = \{(r, s) \mid r, s \subseteq Mark(n)\}$.

шаг 4: Удалим из X все пары ресурсов, которые не являются подобными в смысле n -расслоенного подобия. Для проверки одной пары нужно рассмотреть каждую разметку, не превышающую $Mark(n)$, прибавить к ней r и s и выяснить n -расслоенную бисимулярность получившихся двух разметок (то есть перебрать все элементы множества $Bis(n)$).

шаг 5: Возвратим $X \cup \{(r, s) \mid Mark(n) \subseteq r \ \& \ Mark(n) \subseteq s\}$ — искомое множество $(\approx_n)$.¹

¹Заметим, что бесконечное множество $\{(r, s) \mid Mark(n) \subseteq r \ \& \ Mark(n) \subseteq s\}$ однозначно задается конечным мультимножеством $Mark(n)$.

Представленный алгоритм вычисления ($\approx_n$) крайне неэффективен, однако он доказывает принципиальную возможность аппроксимации подобия ресурсов “сверху”.

4.3 Подобие обобщенных ресурсов

Во многих системах заменяемыми являются не только статические ресурсы (моделируемые фишками в сетях Петри), но и динамические составляющие процесса — действия и события (моделируемые переходами). Например, в сетях потоков работ (workflow) под переходами зачастую понимаются сотрудники или устройства, выполняющие ту или иную индивидуальную работу [2].

В связи с этим возникают вопросы по поводу их взаимозаменяемости, сравнительной эффективности, избыточности и т.п. Проблема поиска эквивалентных переходов и поведений также важна для поддержки методологии адаптивного управления системой, согласно которой структура сети может изменяться непосредственно в ходе ее функционирования, например, в ответ на изменения внешних условий или же при возникновении внутрисистемных событий (болезнь сотрудника, отказ оборудования, внедрение новых элементов системы и т.п.).

В этом разделе определяется и исследуется отношение подобия на множестве так называемых обобщенных ресурсов. Обобщенный ресурс может содержать не только мультимножество фишек (материальная часть), но и мультимножество переходов (инструментальная часть). Два обобщенных ресурса подобны, если при любой разметке сети мы можем заменить фишки и срабатывания одного на фишки и срабатывания другого, и при этом поведение сети не изменится. Отношение подобия обобщенных ресурсов включает обычное подобие (“подобие материальных ресурсов”), а также позволяет отслеживать многие интересные свойства динамической составляющей системы, например, “сравнительную эффективность” и “эквивалентность при условии”.

4.3.1 Обобщенные ресурсы сети Петри

Определение 4.7. Пусть $N = (P, T, F, l)$ — помеченная сеть Петри. Пара (r, α) , где $r \in \mathcal{M}(P)$, $\alpha \in \mathcal{M}(T)$ и $\bullet\alpha \subseteq r$, называется *обобщенным ресурсом* сети Петри N .

Множество всех обобщенных ресурсов помеченной сети Петри N обозначим как $\Phi(N)$.

Другими словами, обобщенный ресурс можно рассматривать как мультимножество над множеством $P \cup T$ вершин графа сети Петри. Мы используем запись (r, α) , поскольку из соображений синтаксиса более удобно явно разделять позиции и переходы.

Итак, обобщенный ресурс (r, α) включает в себя две составляющие — материальный ресурс r и инструментальный ресурс α . Первый показывает наличие в системе “вещественных”, статичных элементов (“ресурсов” в обычном понимании этого слова), второй — динамическую составляющую процесса, то есть выполнение в данный момент тех или иных действий. Поскольку мы рассматриваем мультимножества, и материалы и инструменты могут обладать количеством.

Условие “правильности” $\bullet\alpha \subseteq r$ — естественное требование, которое гарантирует обеспеченность выполняемых действий необходимыми материальными ресурсами.

Обобщенные ресурсы также могут быть взаимозаменяемыми:

Определение 4.8. Обобщенные ресурсы (r, α) и (s, β) называются *подобными* (обозначается $(r, \alpha) \approx (s, \beta)$), если

1. $l(\alpha) = l(\beta)$;
2. для любой разметки $M \in \mathcal{M}(P)$ и параллельного срабатывания $M + r \xrightarrow{\alpha} M'$ возможно параллельное срабатывание $M + s \xrightarrow{\beta} M''$, где $M' \sim M''$.

Таким образом, если два обобщенных ресурса подобны, то мы можем без каких-либо последствий для наблюдаемого поведения

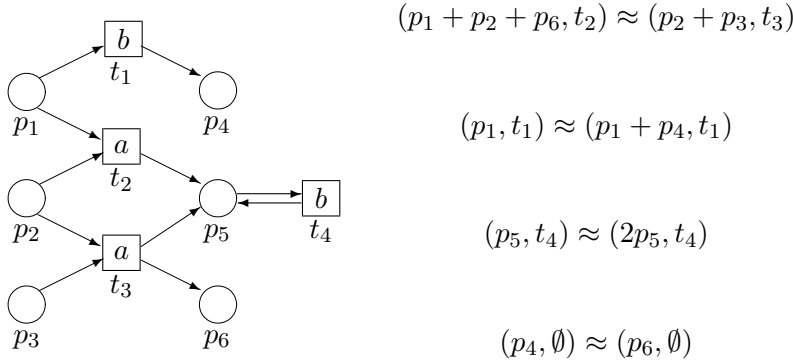


Рис. 4.6: Примеры подобных обобщенных ресурсов

процесса заменять один из них на другой. При этом замена материальной (статической) составляющей состоит в простом переносе соответствующих фишек, а замена инструментальной (динамической) составляющей означает “отмену” срабатывания первого набора переходов и запуск всех переходов из второго набора.

Несколько примеров подобных обобщенных ресурсов приведены на рисунке 4.6.

Подобие обобщенных ресурсов обладает естественной интерпретацией. Например, оно способно выразить возможность замены одного работника/подрядчика (выполняющего некоторую работу, описанную мультимножеством переходов в сети) на другого (другое мультимножество переходов), но только при условии одновременной замены мультимножества фишек в позициях сети (сырья на складе, денег на банковских счетах сотрудников) на другое мультимножество. При этом деньги и сырье — это материальный ресурс, а работник — инструментальный. Различные сотрудники потребляют и производят различные материальные ресурсы.

Заметим, что подобная “замена” не означает фактической за-

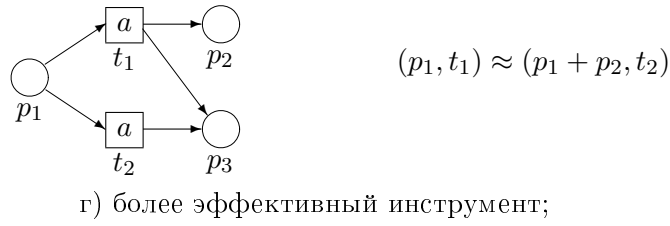
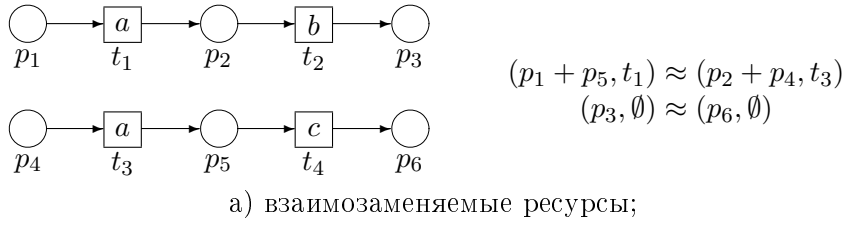


Рис. 4.7: Свойства системы, выявляемые при помощи подобия обобщенных ресурсов

мены переходов в графе сети. Структура графа остается той же. Мы заменяем *срабатывания* переходов в самом процессе (поведении сети). Образно говоря, первый работник не увольняется, а лишь заменяется вторым для выполнения данной конкретной работы. В дальнейшем он опять может быть привлечен для ее выполнения.

При помощи подобия обобщенных ресурсов можно выразить ряд дополнительных эквивалентностных свойств. Некоторые из них показаны на рисунке 4.7.

$(r, \alpha) \approx (s, \beta)$	Обобщенные ресурсы (r, α) и (s, β) взаимо-заменяемы в любом состоянии системы.
$(r, \alpha) \approx (r, \beta)$	Действия (инструменты) α и β эквивалентны при наличии материала r .
$(r, \alpha) \approx (s, \alpha)$	Материалы r и s эквивалентны при условии выполнения действия α .
$(r, \alpha) \approx (r + s, \beta)$	Действие (инструмент) α эффективнее , чем действие β .
$(r, \alpha) \approx (r + s, \alpha)$	Материал s избыточен при условии выполнения действия α .
$(r, \emptyset) \approx (r + s, \emptyset)$	Материал s избыточен .

4.3.2 Свойства подобия обобщенных ресурсов

Подобие обобщенных ресурсов в сетях Петри обладает рядом интересных свойств. Прежде всего, это отношение эквивалентности:

Утверждение 4.13. Пусть $(r, \alpha), (s, \beta), (u, \gamma) \in \Phi(N)$. Тогда

1. $(r, \alpha) \approx (r, \alpha)$;
2. $(r, \alpha) \approx (s, \beta) \Rightarrow (s, \beta) \approx (r, \alpha)$;
3. $(r, \alpha) \approx (s, \beta) \ \& \ (s, \beta) \approx (u, \gamma) \Rightarrow (r, \alpha) \approx (u, \gamma)$.

Доказательство. 1) Из определения.

2) Поскольку наибольшая бисимуляция разметок $\sim$ замкнута относительно симметричности.

3) Поскольку наибольшая бисимуляция разметок $\sim$ замкнута относительно транзитивности. $\square$

Простейшие нетривиальные пары подобных обобщенных ресурсов можно получить, используя следующий факт:

Утверждение 4.14. Пусть $\alpha, \beta \in \mathcal{M}(T)$. Тогда

$$l(\alpha) = l(\beta) \Rightarrow (\bullet\alpha + \beta^\bullet, \alpha) \approx (\bullet\beta + \alpha^\bullet, \beta).$$

Доказательство. Из определения. $\square$

Подобие обобщенных ресурсов замкнуто относительно “вычитания параллельного шага”:

Утверждение 4.15. Пусть $(r, \alpha), (s, \beta) \in \Phi(N)$, $\gamma, \delta \in \mathcal{M}(T)$. Тогда

$$\begin{aligned} (r, \alpha) \approx (s, \beta) \ \& \ l(\gamma) = l(\delta) \ \& \ \gamma \subseteq \alpha \ \& \ \delta \subseteq \beta \Rightarrow \\ \Rightarrow \ (r - \bullet\gamma + \gamma^\bullet, \alpha - \gamma) \approx (s - \bullet\delta + \delta^\bullet, \beta - \delta). \end{aligned}$$

Доказательство. Предположим противное:

$$(r - \bullet\gamma + \gamma^\bullet, \alpha - \gamma) \not\approx (s - \bullet\delta + \delta^\bullet, \beta - \delta).$$

Во-первых, заметим, что $(r - \bullet\gamma + \gamma^\bullet, \alpha - \gamma)$ и $(s - \bullet\delta + \delta^\bullet, \beta - \delta)$ являются “правильными” обобщенными ресурсами, поскольку $\bullet(\alpha - \gamma) \subseteq (r - \bullet\gamma + \gamma^\bullet)$ и $\bullet(\beta - \delta) \subseteq (s - \bullet\delta + \delta^\bullet)$. Из $(r, \alpha) \approx (s, \beta)$ и $l(\gamma) = l(\delta)$ получим $l(\alpha - \gamma) = l(\beta - \delta)$.

Следовательно, мультимножества переходов $\alpha - \gamma$ и $\beta - \delta$ могут сработать параллельно при наличии в разметке соответствующих материальных ресурсов.

Таким образом, из нашего предположения следует, что для некоторой разметки $M \in \mathcal{M}(P)$ и параллельных шагов

$$M + (r - \bullet\gamma + \gamma\bullet) \xrightarrow{\alpha-\gamma} M' \quad (4.1)$$

и

$$M + (s - \bullet\delta + \delta\bullet) \xrightarrow{\beta-\delta} M'' \quad (4.2)$$

выполняется $M' \not\sim M''$.

Рассмотрим α и r . Из условия $\gamma \subseteq \alpha$ имеем $\alpha = \gamma + \gamma'$. Из условий $\bullet\alpha \subseteq r$ и $\gamma \subseteq \alpha$ имеем

$$r = \bullet\gamma + \bullet\gamma' + r'. \quad (4.3)$$

Применив (4.3), мы можем переписать шаг (4.1) в виде

$$M + (r - \bullet\gamma + \gamma\bullet) = M + \gamma\bullet + \bullet\gamma' + r' \xrightarrow{\gamma'} M + \gamma\bullet + \gamma'\bullet + r'.$$

Следовательно, $M' = M + \gamma\bullet + \gamma'\bullet + r'$.

Аналогично получим $\beta = \delta + \delta'$, $s = \bullet\delta + \bullet\delta' + s'$ для некоторого s' и $M'' = M + \delta\bullet + \delta'\bullet + s'$.

Из подобия обобщенных ресурсов $(r, \alpha) \approx (s, \beta)$ следует

$$M + r \xrightarrow{\alpha} G', \quad M + s \xrightarrow{\beta} G'', \quad G' \sim G''. \quad (4.4)$$

Применив (4.3), мы можем переписать шаг $M + r \xrightarrow{\alpha} G'$ в виде

$$M + r = M + \bullet\gamma + \bullet\gamma' + r' \xrightarrow{\gamma+\gamma'} G' = M + \gamma\bullet + \gamma'\bullet + r'.$$

Следовательно, $G' = M'$. Аналогично $G'' = M''$. Таким образом, мы получили противоречие между (4.4) и предположением $M' \not\sim M''$. $\square$

Заметим, что представленная операция не является полноценным вычитанием ресурсов, поскольку мы вычитаем только инструментальную часть ресурса. Материальная часть не вычитается, а трансформируется в соответствии со свойствами шага.

Подобие обобщенных ресурсов замкнуто относительно “прибавления параллельного шага”:

Утверждение 4.16. Пусть $(r, \alpha), (s, \beta) \in \Phi(N)$, $\gamma, \delta \in \mathcal{M}(T)$. Тогда

$$\begin{aligned} (r, \alpha) \approx (s, \beta) \ \& \ l(\gamma) = l(\delta) \ \& \ \gamma^\bullet \subseteq (r - \bullet\alpha) \ \& \ \delta^\bullet \subseteq (s - \bullet\beta) \quad \Rightarrow \\ \Rightarrow \quad (r - \gamma^\bullet + \bullet\gamma, \alpha + \gamma) &\approx (s - \delta^\bullet + \bullet\delta, \beta + \delta). \end{aligned}$$

Доказательство. Аналогично доказательству утверждения 4.15 о замкнутости подобия обобщенных ресурсов относительно “вычитания параллельного шага”. $\square$

Это также не вполне сложение. Мы прибавляем только инструментальную компоненту, а материальная компонента трансформируется в зависимости от изменения инструментальной.

Однако мы можем складывать и обе компоненты одновременно. Подобие обобщенных ресурсов замкнуто относительно прибавления пар ресурсов:

Утверждение 4.17. Пусть $(r, \alpha), (s, \beta), (u, \gamma), (v, \delta) \in \Phi(N)$. Тогда

$$(r, \alpha) \approx (s, \beta) \ \& \ (u, \gamma) \approx (v, \delta) \quad \Rightarrow \quad (r + u, \alpha + \gamma) \approx (s + v, \beta + \delta).$$

Доказательство. Предположим противное:

$$(r + u, \alpha + \gamma) \not\approx (s + v, \beta + \delta).$$

Рассуждая так же, как в начале доказательства утверждения 4.15, получим, что для некоторой разметки $M \in \mathcal{M}(P)$ и параллельных шагов $M + r + u \xrightarrow{\alpha+\gamma} M'$ и $M + s + v \xrightarrow{\beta+\delta} M''$ выполняется $M' \not\approx M''$.

Рассмотрим (r, α) . Поскольку $\bullet\alpha \subseteq r$, мы имеем

$$r \xrightarrow{\alpha} (r - \bullet\alpha + \alpha\bullet).$$

Обозначим мультимножество позиций $r - \bullet\alpha + \alpha\bullet$ символом r' . Аналогично, обозначим “результатирующие материальные ресурсы” обобщенных ресурсов (s, β) , (u, γ) и (v, δ) символами s' , u' и v' :

$$r \xrightarrow{\alpha} r', \quad s \xrightarrow{\beta} s', \quad u \xrightarrow{\gamma} u', \quad v \xrightarrow{\delta} v'.$$

Очевидно, мы имеем $M' = M + r' + u'$ и $M'' = M + s' + v'$, и, следовательно, предположение может быть записано как:

$$M + r' + u' \not\sim M + s' + v'. \quad (4.5)$$

Рассмотрим $G = M + r'$. Из $(u, \gamma) \approx (v, \delta)$ имеем

$$G + u \xrightarrow{\gamma} G + u', \quad G + v \xrightarrow{\delta} G + v', \quad G + u' \sim G + v'. \quad (4.6)$$

Рассмотрим $H = M + v'$. Из $(r, \alpha) \approx (s, \beta)$ имеем

$$H + r \xrightarrow{\alpha} H + r', \quad H + s \xrightarrow{\beta} H + s', \quad H + r' \sim H + s'. \quad (4.7)$$

Бисимулярности (4.6) и (4.7) могут быть записаны как:

$$(M + r') + u' \sim (M + r') + v', \quad (M + v') + r' \sim (M + v') + s'.$$

Поскольку наибольшая бисимуляция разметок $\sim$ замкнута относительно транзитивности, а сложение мультимножеств коммутативно и ассоциативно, имеем

$$M + r' + u' \sim M + s' + v',$$

что противоречит предположению (4.5). $\square$

Замечание 4.2. Подобие обобщенных ресурсов не замкнуто относительно вычитания пар ресурсов. В частности, это обусловлено тем, что вычитание может нарушить “правильность” ресурса.

Следствие 4.4. *Подобие обобщенных ресурсов обладает конечным АТ-базисом.*

Доказательство. Следует из аддитивной и транзитивной замкнутости отношения. $\square$

Определим частичный порядок $\sqsubseteq$ на множестве $R \subseteq \Phi(N) \times \Phi(N)$ пар обобщенных ресурсов как “обобщение” случая $\mathcal{M}(P) \times \mathcal{M}(P)$:

$$\left((r, \alpha), (s, \beta) \right) \sqsubseteq \left((u, \gamma), (v, \delta) \right) \stackrel{\text{def}}{\iff} (r, s) \sqsubseteq (u, v) \ \& \ (\alpha, \beta) \sqsubseteq (\gamma, \delta).$$

Аналогично, через R_s обозначим множество всех минимальных относительно $\sqsubseteq$ элементов R , которое будем называть *основным базисом* отношения R .

Следствие 4.5. *Подобие обобщенных ресурсов обладает конечным основным базисом.*

4.3.3 Материальные и инструментальные ресурсы

Можно выделить два интересных специальных вида обобщенных ресурсов.

Определение 4.9. Обобщенный ресурс вида $(r, \emptyset)$ называется *материальным ресурсом*.

Обобщенный ресурс вида $(\bullet\alpha, \alpha)$ называется *инструментальным ресурсом*.

Рассматривая только пары подобных материальных ресурсов, мы получим отношение эквивалентности — *подобие материальных ресурсов*.

Легко видеть, что подобие материальных ресурсов совпадает с подобием ресурсов (с точностью до обозначений).

Утверждение 4.18. Пусть $r, s \in \mathcal{M}(P)$. Тогда

$$r \approx s \quad \Leftrightarrow \quad (r, \emptyset) \approx (s, \emptyset).$$

Доказательство. Из определений. □

Из неразрешимости подобия ресурсов (следствие 2.1) и утверждения 4.18 получаем

Следствие 4.6. Подобие материальных ресурсов неразрешимо.

Следствие 4.7. Подобие обобщенных ресурсов неразрешимо.

Рассматривая только пары подобных инструментальных ресурсов, мы получим *подобие инструментальных ресурсов*. Оказывается, оно менее выразительно, чем подобие материальных ресурсов, так как полностью определяется его специальным подмножеством:

Утверждение 4.19. Пусть $\alpha, \beta \in \mathcal{M}(T)$. Тогда

$$(\bullet\alpha, \alpha) \approx (\bullet\beta, \beta) \quad \Leftrightarrow \quad (\alpha^\bullet, \emptyset) \approx (\beta^\bullet, \emptyset).$$

Доказательство.

($\Rightarrow$) Из утверждения 4.15.

($\Leftarrow$) Из утверждения 4.16. □

Это довольно естественно — мы можем заменять “инструменты” без дополнительных условий тогда и только тогда, когда они “полностью” подобны, то есть производят эквивалентные материальные ресурсы.

Глава 5

Другие классы формальных моделей

5.1 Сети Петри с невидимыми переходами

Довольно часто при анализе системы возникает потребность абстрагироваться от излишней информации о ее поведении. Например, не учитывать срабатывания переходов, соответствующие неким внутренним действиям системы, “невидимым” внешнему наблюдателю. Получаемая при этом информация может быть полезной, в частности, для обнаружения дополнительных свойств системы в части ее взаимодействия с окружающим миром.

В этом разделе исследуются возможности эффективного построения сохраняющих бисимуляцию отношений на множестве ресурсов для расширения класса обыкновенных сетей Петри — класса сетей Петри с невидимыми переходами.

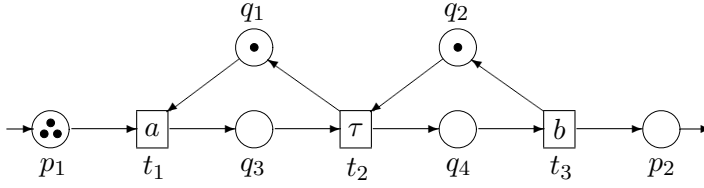


Рис. 5.1: Пример невидимого перехода в сети Петри

5.1.1 Определение сетей Петри с невидимыми переходами

Чтобы различить видимые и невидимые срабатывания переходов, в алфавит меток срабатываний добавляется особая метка τ .

Рассмотрим алфавит меток срабатываний $Act_\tau = Act \cup \{\tau\}$.

Определение 5.1. Помеченной сетью Петри с невидимыми переходами называется набор $N = (P, T, F, l)$, где (P, T, F) — сеть Петри, $l : T \rightarrow Act_\tau$ — помечающая функция, сопоставляющая каждому переходу либо метку из Act , либо символ τ .

Символом τ помечаются внутренние, невидимые для внешнего наблюдателя переходы системы. Невидимые переходы также называют τ -переходами.

На рисунке 5.1 изображена часть более крупной сети Петри, моделирующая элемент памяти “очередь” (FIFO) длины 2. Переход t_2 моделирует перенос данных из первой ячейки во вторую. Его срабатывание не важно с точки зрения поведения системы в целом (в некотором смысле это служебный переход) — нас интересует в первую очередь интерфейс элемента памяти, то есть позиции p_1 (вход) и p_2 (выход), а также срабатывания t_1 (поместить в очередь) и t_3 (достать из очереди). Поэтому переход t_2 помечен символом τ и его срабатывание не учитывается при рассмотрении поведения сети.

Обозначение 5.1. Пусть $\sigma, \sigma' \in (Act_\tau)^*$ — последовательности переходов. Полагаем $\sigma =_\tau \sigma' \Leftrightarrow_{def} \sigma|_{Act} = \sigma'|_{Act}$.

Запись “ $=_\tau$ ” означает равенство “по модулю τ ” между строками из символов алфавита Act_τ . Например, “ $\tau\tau a\tau$ ” $=_\tau$ “ a ”.

Определим бисимуляцию для сетей с невидимыми переходами.

Определение 5.2. Пусть $N = (P, T, F, l)$ — помеченная сеть Петри с невидимыми переходами. Отношение $R \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$ обладает *свойством τ -переноса*, если для любой пары разметок $(M_1, M_2) \in R$ и для любого перехода $t \in T$, такого что $M_1 \xrightarrow{t} M'_1$, найдется имитирующая последовательность переходов $\sigma \in T^*$, такая, что $l(t) =_\tau l(\sigma)$, $M_2 \xrightarrow{\sigma} M'_2$ и $(M'_1, M'_2) \in R$.

Свойство τ -переноса может быть проиллюстрировано следующей диаграммой:

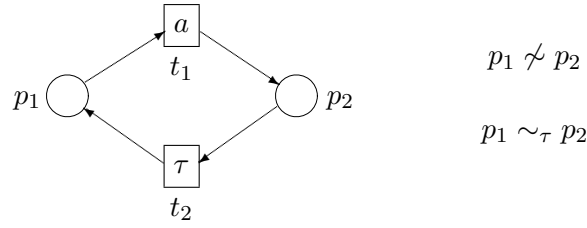
$$\begin{array}{ccc} M_1 & R & M_2 \\ \downarrow t & & \downarrow (\exists)\sigma, l(\sigma) =_\tau l(t) \\ M'_1 & R & M'_2 \end{array}$$

Определение 5.3. Если отношения R и R^{-1} обладают свойством τ -переноса, то отношение R называется *τ -бисимуляцией (разметок)*.

Максимальная τ -бисимуляция обозначается как $\sim_\tau$.

Обыкновенная бисимуляция разметок является частным случаем τ -бисимуляции (для сетей, в которых ни один из переходов не помечен τ). Поэтому очевидно, что она слабее. В качестве примера можно рассмотреть сеть на рисунке 5.2. Разметки p_1 и p_2 небисимулярны, поскольку при разметке p_2 не может сработать ни один переход с меткой a . В то же время они τ -бисимулярны, поскольку срабатывание невидимого перехода t_2 меняет разметку сети с p_2 на p_1 .

Из того, что τ -бисимуляция слабее обыкновенной бисимуляции, следует, в частности, неразрешимость τ -бисимуляции в сетях Петри с невидимыми переходами [48].

Рис. 5.2: τ -бисимуляция слабее бисимуляции

5.1.2 Подобие и бисимуляция ресурсов в сетях с невидимыми переходами

Определение подобия естественным образом обобщается на случай сетей с невидимыми переходами.

Определение 5.4. Пусть $N = (P, T, F, l)$ — помеченная сеть Петри с невидимыми переходами. Ресурсы r и s называются τ -подобными (обозначается $r \approx_\tau s$), если для любой разметки R , такой, что $r \subseteq R$, выполняется $R \sim_\tau R - r + s$.

Покажем, что τ -подобие обладает всеми основными свойствами подобия:

- Утверждение 5.1.** 1. τ -подобие ресурсов замкнуто относительно аддитивности и транзитивности и, следовательно, обладает конечным АТ-базисом.
2. τ -подобие ресурсов неразрешимо.

Доказательство. 1) Непосредственно из определений.

2) Из следствия 2.1 и того, что подобие ресурсов является частным случаем τ -подобия ресурсов (когда ни один из переходов сети не помечен символом τ). $\square$

Определение бисимуляции ресурсов обобщается на случай сетей с невидимыми переходами следующим образом:

Определение 5.5. Пусть $N = (P, T, F, l)$ — помеченная сеть Петри с невидимыми переходами. Симметричное 1-рефлексивное отношение $B \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$ называется τ -бисимуляцией ресурсов, если B^{AT} есть τ -бисимуляция разметок.

Для сетей с невидимыми переходами имеет место аналог теоремы 3.3:

Утверждение 5.2. Пусть $N = (P, T, F, l)$ — помеченная сеть Петри с невидимыми переходами. Тогда

1. если $B_1 \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$ — τ -бисимуляция ресурсов и выполняется $B_2 \subseteq B_1$, то отношение B_2 — τ -подобие ресурсов;
2. если $B_1, B_2 \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$ — τ -бисимуляции ресурсов, то отношение $B_1 \cup B_2$ — τ -бисимуляция ресурсов;
3. для N существует максимальная по вложению τ -бисимуляция ресурсов (обозначается как $\simeq_\tau$);
4. $(\simeq_\tau)$ является отношением эквивалентности.

Доказательство. Доказательство аналогично доказательству теоремы 3.3, с единственным отличием: при доказательстве утверждения 2 ищется не имитирующий переход, а имитирующая последовательность переходов. $\square$

Определим слабое свойство переноса для сетей Петри с невидимыми переходами.

Определение 5.6. Отношение $B \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$ обладает *слабым свойством τ -переноса*, если для всех $(r, s) \in B$, $t \in T$, таких что $\bullet t \cap r \neq \emptyset$, найдется имитирующая последовательность переходов $\sigma \in T^*$, такая, что $l(t) =_\tau l(\sigma)$ и, обозначив $M_1 = \bullet t \cup r$ и $M_2 = \bullet t - r + s$, имеем $M_1 \xrightarrow{t} M_1'$ и $M_2 \xrightarrow{\sigma} M_2'$, где $(M_1', M_2') \in B^{AT}$.

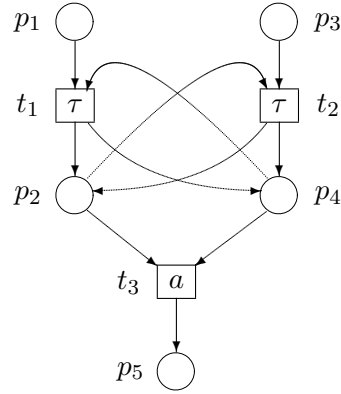


Рис. 5.3: Теорема 3.4 не выполняется в случае сетей с невидимыми переходами

Теорема 3.4 в случае сетей Петри с невидимыми переходами выполняется только “в одну сторону”:

Утверждение 5.3. *Из выполнения свойства τ -переноса следует выполнение слабого свойства τ -переноса; обратное неверно.*

Доказательство. ($\Rightarrow$) Так как слабое свойство τ -переноса — это свойство τ -переноса, ограниченное конечным числом пар.

($\nLeftarrow$) Рассмотрим сеть, представленную на рисунке 5.3, и отношение

$$B = Id(P) \cup \{(p_1, p_2), (p_2, p_1), (p_3, p_4), (p_4, p_3)\}.$$

Отношение B обладает слабым свойством τ -переноса. В то же время B не является τ -бисимуляцией ресурсов.

Действительно, рассмотрим разметки $M_1 = p_1 + p_3$ и $M_2 = p_2 + p_4$. Пара (M_1, M_2) принадлежит отношению B^{AT} , но не является бисимулярной, так как при разметке M_2 может сработать переход с меткой a (переход t_3), а при разметке M_1 — нет. $\square$

В силу этого слабое свойство τ -переноса не может быть использовано для построения бисимуляций. В случае сетей с невидимыми переходами требуется ещё большее усиление рассматриваемых отношений и ограничение рассматриваемого класса сетей.

5.1.3 Насыщенные сети Петри

Существует обширный подкласс сетей Петри с невидимыми переходами, для которых τ -бисимуляция ресурсов может быть построена при помощи слабого свойства переноса. Это предложенные Ф. Шнобеленом и др. [27] p -насыщенные сети, в которых срабатывание любой последовательности переходов с не более чем одной видимой меткой может быть имитировано одновременным (независимым) срабатыванием некоторого множества переходов с той же меткой (называемого “параллельным шагом”).

Обозначение 5.2. Введем обозначение для множества последовательностей переходов с не более чем одной видимой меткой:

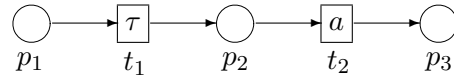
$$T^\times =_{def} \{\sigma \in T^* \mid l(\sigma) \in Act_\tau\}.$$

Определение 5.7. Пусть $U = \{t_1, \dots, t_k\} \in \mathcal{M}(T)$ — мультимножество переходов. Будем обозначать $\bullet U = \bullet t_1 + \dots + \bullet t_k$, $U^\bullet = t_1^\bullet + \dots + t_k^\bullet$ и говорить, что переходы из U могут *сработать одновременно при разметке* M , если $\bullet U \subseteq M$.

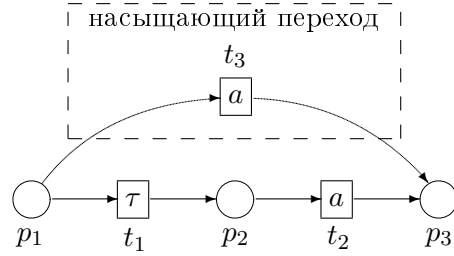
Тогда $M \xrightarrow{U} M'$, где $M' = M - \bullet U + U^\bullet$, а мультимножество U называется *параллельным шагом*.

Заметим, что параллельно могут сработать не только два различных перехода, но и две “копии” одного и того же перехода.

Определение 5.8. Помеченная сеть Петри с невидимыми переходами $N = (P, T, F, l)$ называется p -насыщенной (или просто *насыщенной*), если для любой последовательности переходов $\sigma \in T^\times$ найдется параллельный шаг $U \in \mathcal{M}(T)$, такой, что $l(U) = l(\sigma)$, $\bullet U = \bullet \sigma$ и $U^\bullet = \sigma^\bullet$.



а) ненасыщенная сеть



б) насыщенная сеть

Рис. 5.4: Пример насыщения сети Петри

Кроме насыщенных, существует еще более обширный класс насыщаемых сетей Петри, то есть таких, которые могут быть преобразованы в насыщенные добавлением конечного числа переходов при сохранении поведения сети (в смысле τ -бисимулярности).

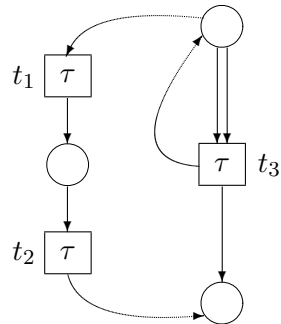
Известно [27], что сеть p -насыщена тогда и только тогда, когда она $2p$ -насыщена, то есть в сети насыщены все последовательности из $T^\times$ длины 2.

На рисунке 5.4 изображена насыщенная сеть, полученная добавлением насыщающего перехода t_3 к исходной ненасыщенной.

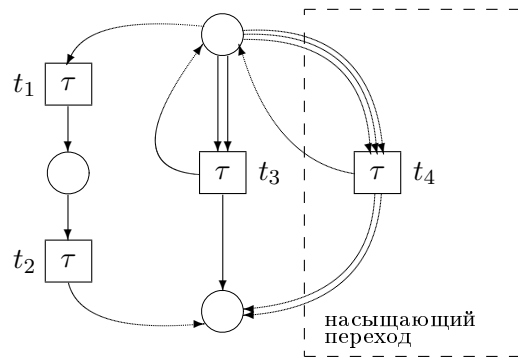
Известно [27], что не у всех конечных сетей существуют конечные насыщения.

Пример ненасыщаемой сети приведен на рисунке 5.5. В исходной сети ненасыщенной является последовательность срабатываний $t_1.t_2.t_3$. Для ее насыщения в сеть добавляется переход t_4 . Однако при этом появляется новая ненасыщенная последовательность — $t_1.t_2.t_4$. И так далее до бесконечности.

Приведем один критерий насыщаемости.



а) исходная ненасыщенная сеть



б) первый шаг "насыщения"

Рис. 5.5: Пример ненасыщаемой сети

Определение 5.9. Пусть $N = (P, T, F, l)$ — помеченная сеть Петри с невидимыми переходами. Назовем τ -подсетью сети N (обозначается N_τ) сеть, полученной из N удалением всех видимых переходов.

Теорема 5.1. Пусть $N = (P, T, F, l)$ — помеченная сеть Петри с невидимыми переходами. Сеть N p -насыщаема тогда и только тогда, когда p -насыщаема τ -подсеть N_τ .

Доказательство. ($\Rightarrow$) Очевидно, поскольку если N^* — p -насыщение N , то $(N^*)_\tau$ — p -насыщение N_τ .

($\Leftarrow$) Достаточно доказать, что сеть с насыщенной τ -подсетью насыщаема. Пусть N_τ — насыщенная. Докажем, что существует конечная насыщающая последовательность для N .

Все ненасыщенные последовательности переходов из множества $T^\times$ имеют вид $\sigma = \sigma_1 t \sigma_2$, где $l(\sigma_1) = l(\sigma_2) = \tau$, $l(t) \neq \tau$ (так как сеть N_τ — насыщена).

Любой новый переход, появляющийся в ходе применения любой стратегии насыщения, является насыщающим для одной из таких последовательностей. Сеть N_τ насыщена, т.е. у любой последовательности переходов $\sigma' \in T^*$, такой, что $l(\sigma') = \tau$, существует насыщающий параллельный шаг $U \in \mathcal{M}(T)$. Следовательно, насыщение последовательностей вида $\sigma_1 t \sigma_2$ эквивалентно насыщению последовательностей вида $U_1 t U_2$, где U_1 и U_2 — параллельные шаги. Разобьем эти шаги на подшаги:

$$U_1 = V_1 \cup W_1, \text{ где } V_1 = \{t' \in U_1: t'^\bullet \cap \bullet t \neq \emptyset\}, W_1 = U_1 \setminus V_1;$$

$$U_2 = V_2 \cup W_2, \text{ где } V_2 = \{t' \in U_2: \bullet t' \cap t^\bullet \neq \emptyset\}, W_2 = U_2 \setminus V_2.$$

Добавление в сеть нового перехода, насыщающего последовательность $U_1 t U_2$, можно заменить добавлением перехода, насыщающего $V_1 t V_2$ (это один и тот же переход).

Таким образом, для получения насыщенной сети достаточно добавить переходы, насыщающие все последовательности вида $V_1 t V_2$, где

$$t \in T, l(t) \neq \tau,$$

$$\begin{aligned}
V_1, V_2 &\in \mathcal{M}(T), \quad l(V_1) = l(V_2) = \tau, \\
\forall t' \in V_1 \quad t'^{\bullet} \cap t^{\bullet} &\neq \emptyset, \\
\forall t' \in V_2 \quad \bullet t' \cap \bullet t &\neq \emptyset.
\end{aligned}$$

Для любой конечной сети число таких последовательностей конечно (так как $|t^{\bullet}| \leq \infty$, $|\bullet t| \leq \infty$). $\square$

5.1.4 τ -бисимуляция ресурсов

В работе [27] Ф. Шнобеленом и др. было предложено более сильное по сравнению с τ -бисимуляцией разметок отношение, названное τ -бисимуляцией разметок. Переход в этом случае моделируется не последовательностью переходов, а параллельным шагом.

Определение 5.10. Пусть $N = (P, T, F, l)$ — помеченная насыщенная сеть Петри с невидимыми переходами. Скажем, что отношение $R \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$ обладает *свойством τ -переноса*, если для любой пары разметок $(M_1, M_2) \in R$ и для любого перехода $t \in T$, такого что $M_1 \xrightarrow{t} M'_1$, найдется имитирующий параллельный шаг $U \in \mathcal{M}(T)$, такой, что $l(t) =_{\tau} l(U)$, $M_2 \xrightarrow{U} M'_2$ и $(M'_1, M'_2) \in R$.

Определение 5.11. Если отношения R и R^{-1} обладают свойством τ -переноса, то отношение R называется *τ -бисимуляцией (разметок)*.

Известно [27], что для любой сети существует максимальная по вложению τ -бисимуляция разметок (обозначается как $\sim_{\tau p}$).

В насыщенных сетях Петри τ -бисимуляция разметок совпадает с τ -бисимуляцией разметок:

Утверждение 5.4. [27] Пусть $N = (P, T, F, l)$ — помеченная насыщенная сеть Петри с невидимыми переходами, $M_1, M_2 \in \mathcal{M}(P)$ — разметки сети N . Тогда

$$M_1 \sim_{\tau p} M_2 \quad \Leftrightarrow \quad M_1 \sim_{\tau} M_2.$$

Введем отношение τr -подобия ресурсов.

Определение 5.12. Пусть $N = (P, T, F, l)$ — помеченная насыщенная сеть Петри с невидимыми переходами. Ресурсы r и s называются τr -подобными (обозначается $r \approx_{\tau r} s$), если для любой разметки R , такой, что $r \subseteq R$, выполняется $R \sim_{\tau r} R - r + s$.

Из утверждения 5.4 вытекает

Следствие 5.1. Пусть $N = (P, T, F, l)$ — помеченная насыщенная сеть Петри с невидимыми переходами, $r, s \in \mathcal{M}(P)$ — ресурсы сети N . Тогда

$$r \approx_{\tau r} s \Leftrightarrow r \approx_{\tau} s.$$

Таким образом, для ресурсов в насыщенных сетях достаточно установить их τr -подобие.

Определение 5.13. Пусть $N = (P, T, F, l)$ — помеченная насыщенная сеть Петри с невидимыми переходами. Симметричное 1-рефлексивное отношение $B \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$ называется τr -бисимуляцией ресурсов, если B^{AT} является τr -бисимуляцией разметок.

Для τr -случая выполняются все основные свойства подобий ресурсов и бисимуляций ресурсов:

- Утверждение 5.5.**
1. τr -подобие ресурсов замкнуто относительно аддитивности и транзитивности и, следовательно, обладает конечным AT -базисом.
 2. τr -подобие ресурсов неразрешимо.
 3. Если $B_1 \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$ — τr -бисимуляция ресурсов и выполняется $B_2 \subseteq B_1$, то отношение B_2 — τr -подобие ресурсов.

4. Если $B_1, B_2 \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$ — τr -бисимуляции ресурсов, то отношение $B_1 \cup B_2$ — τr -бисимуляция ресурсов.
5. Для любой сети N существует максимальная по вложению τr -бисимуляция ресурсов (обозначается как $\simeq_{\tau r}$).
6. $\simeq_{\tau r}$ является отношением эквивалентности.

Доказательство.

- 1) Непосредственно из определения τr -подобия ресурсов.
- 2) Из следствий 2.1 и 5.1.
- 3) Непосредственно из определений.
- 4) Доказательство аналогично доказательству утверждения 1 теоремы 3.3, с единственным отличием: ищется не имитирующий переход, а имитирующий параллельный шаг.
- 5) В качестве максимальной можно взять объединение всех τr -бисимуляций ресурсов.
- 6) Доказательство аналогично доказательству утверждения 3 теоремы 3.3. $\square$

Введем понятие слабого свойства τr -переноса.

Определение 5.14. Пусть $N = (P, T, F, l)$ — помеченная насыщенная сеть Петри с невидимыми переходами. Отношение $B \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$ обладает *слабым свойством τr -переноса*, если для любой пары ресурсов $(r, s) \in B$ и для любого перехода $t \in T$, таких, что $\bullet t \cap r \neq \emptyset$, найдется имитирующий параллельный шаг $U \in \mathcal{M}(T)$, такой, что $l(t) = l(U)$ и, обозначив $M_1 = \bullet t \cup r$ и $M_2 = \bullet t - r + s$, мы имеем $M_1 \xrightarrow{t} M_1'$ и $M_2 \xrightarrow{U} M_2'$, где $(M_1', M_2') \in B^{AT}$.

В насыщенных сетях выполнение слабого свойства τr -переноса является необходимым и достаточным условием для выполнения его расширенной версии, гарантирующей имитацию уже не единичного перехода, а целого параллельного шага:

Определение 5.15. Пусть $N = (P, T, F, l)$ — помеченная насыщенная сеть Петри с невидимыми переходами. Отношение $B \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$ обладает *расширенным слабым свойством тр-переноса*, если для любой пары ресурсов $(r, s) \in B$ и для любого параллельного шага $V \in \mathcal{M}(T)$, таких, что $\bullet V \cap r \neq \emptyset$, найдется имитирующий параллельный шаг $U \in \mathcal{M}(T)$, такой, что $l(V) = l(U)$ и, обозначив $M_1 = \bullet V \cup r$ и $M_2 = \bullet V - r + s$, мы имеем $M_1 \xrightarrow{V} M_1'$ и $M_2 \xrightarrow{U} M_2'$, где $(M_1', M_2') \in B^{AT}$.

Лемма 5.1. Пусть $N = (P, T, F, l)$ — помеченная насыщенная сеть Петри с невидимыми переходами. Отношение $B \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$ обладает *слабым свойством тр-переноса* тогда и только тогда, когда оно обладает *расширенным слабым свойством тр-переноса*.

Доказательство. ($\Leftarrow$) Так как слабое свойство переноса является частным случаем расширенного слабого свойства переноса.

($\Rightarrow$) Предположим противное: расширенное свойство не выполняется, то есть существуют $(M_1, M_2) \in B^{AT}$, $V = \{t_1, \dots, t_k\} \in \mathcal{M}(T)$, где $M_1 \xrightarrow{V} M_1'$, такие, что не найдется моделирующего параллельного шага $U \in \mathcal{M}(T)$ с той же видимой меткой $l(V) = l(U)$, для которого бы выполнялось $M_2 \xrightarrow{U} M_2'$ и $(M_1', M_2') \in B^{AT}$.

Рассмотрим срабатывание $M_1 \xrightarrow{t_1} M_1^1$. Согласно слабому свойству *тр-переноса*, для него существует имитирующий параллельный шаг $M_2 \xrightarrow{W_1} M_2^1$, такой, что $(M_1^1, M_2^1) \in B^{AT}$.

Поскольку $V = \{t_1, \dots, t_k\}$ — параллельный шаг при разметке M_1 , после срабатывания одного из переходов все остальные остаются готовыми к срабатыванию. Следовательно, мы можем повторить предыдущее рассуждение для пары разметок $(M_1^1, M_2^1) \in B^{AT}$ и перехода t_2 . И так далее вплоть до перехода t_k :

$$\begin{array}{ccc}
M_1 & B^{AT} & M_2 \\
t_1 \downarrow & & \downarrow W_1 \\
M_1^1 & B^{AT} & M_2^1 \\
t_2 \downarrow & & \downarrow W_2 \\
M_1^2 & B^{AT} & M_2^2 \\
t_3 \downarrow & & \downarrow W_3 \\
\cdots & & \cdots \\
t_k \downarrow & & \downarrow W_k \\
M_1' = M_1^k & B^{AT} & M_2^k = M_2'
\end{array}$$

В итоге мы получили последовательность срабатываний параллельных шагов

$$M_2 \xrightarrow{W_1} M_2^1 \xrightarrow{W_2} M_2^2 \xrightarrow{W_3} \dots \xrightarrow{W_k} M_2^k = M_2',$$

имитирующую срабатывание параллельного шага $M_1 \xrightarrow{V} M_1'$. Поскольку сеть насыщенная, для любой последовательности переходов (а последовательность параллельных шагов может рассматриваться как последовательность переходов) существует параллельный шаг с той же меткой и с теми же пред- и постусловием, то есть искомым параллельный шаг $M_2 \xrightarrow{U} M_2'$ с меткой $l(V)$. $\square$

Заметим, что, в отличие от обычного слабого свойства переноса, расширенное слабое свойство переноса не может быть эффективно проверено перебором пар ресурсов, так как множество параллельных шагов в общем случае бесконечно.

Теорема 5.2. Пусть $N = (P, T, F, l)$ — помеченная насыщен-
ная сеть Петри с невидимыми переходами. Симметричное 1-
рефлексивное отношение $B \subseteq \mathcal{M}(P) \times \mathcal{M}(P)$ обладает слабым
свойством τp -переноса тогда и только тогда, когда B — τp -
бисимуляция ресурсов.

Доказательство. ($\Leftarrow$) Так как слабое свойство переноса — это
свойство переноса, ограниченное конечным числом пар.

($\Rightarrow$) Доказательство в целом аналогично доказательству тео-
ремы 3.4, с использованием леммы 5.1.

Предположим противное: пусть свойство τp -переноса для B^{AT}
не выполняется, то есть существуют $(M_1, M_2) \in B^{AT}$, $t \in T$, где
 $M_1 \xrightarrow{t} M'_1$, такие, что не найдется моделирующего параллельного
шага $U \in \mathcal{M}(T)$ с той же видимой меткой $l(t) = l(U)$, для которого
бы выполнялось $M_2 \xrightarrow{U} M'_2$ и $(M'_1, M'_2) \in B^{AT}$.

Рассмотрим пару разметок $(M_1, M_2) \in B^{AT}$. Согласно лем-
ме 2.3 она получена транзитивным замыканием нескольких пар
из B^A — аддитивного замыкания B :

$$(H_1, H_2), (H_2, H_3), \dots, (H_{k-1}, H_k) \in B^A, \text{ где } H_1 = M_1, H_k = M_2.$$

Рассмотрим пару (H_1, H_2) .

$$(H_1, H_2) = (r_1 + r_2 + \dots + r_l, s_1 + s_2 + \dots + s_l), \text{ где } (r_i, s_i) \in B$$

$H_1 = \bullet t \cup r_1 + F_1$. В силу выполнения слабого свойства переноса
для пары (r_1, s_1) существует имитирующий параллельный шаг
 $V \in \mathcal{M}(T)$, такой, что $l(t) = l(V)$, $\bullet t \cup r_1 \xrightarrow{t} G_1$ и $\bullet t - r_1 + s_1 \xrightarrow{V} G_2$,
где $(G_1, G_2) \in B^{AT}$.

Так как $\bullet t \cup r_1 \subseteq H_1$, к пред- и постусловиям срабатываний
мы можем добавить ресурс $F = H_1 - \bullet t \cup r_1$:

$$\begin{aligned} \bullet t \cup r_1 + F &\xrightarrow{t} G_1 + F \\ \bullet t - r_1 + s_1 + F &\xrightarrow{V} G_2 + F \end{aligned}$$

В силу 1-рефлексивности отношения B и аддитивной замкнутости B^{AT} полученная пара разметок также разложима по B : $(G_1 + F, G_2 + F) \in B^{AT}$.

У нас получилась новая разметка $H'_1 = \bullet t - r_1 + s_1 + F = H_1 - r_1 + s_1$. Заметим, что она содержит $r_2 + \dots + r_l$. То есть мы можем еще раз применить использованный прием для замены ресурса r_2 на бисимулярный ресурс s_2 , на это раз используя лемму 5.1 и находя имитирующий параллельный шаг не для перехода, а для параллельного шага V .

Применив его еще в общей сложности $l - 1$ раз и используя свойство транзитивной замкнутости B^{AT} , в итоге получим параллельный шаг W , имитирующий t при разметке H_2 .

Перейдем к следующей паре (H_2, H_3) , повторим все для параллельного шага W . И т.д., вплоть до последней пары (H_{k-1}, H_k) . В итоге мы получим параллельный шаг U , имитирующий срабатывание перехода t при разметке $H_k = M_2$. $\square$

Итак, слабое свойство τp -переноса может быть использовано при построении τp -бисимуляции ресурсов.

5.1.5 Алгоритм построения аппроксимации

Как и для обыкновенных сетей Петри, для насыщенных сетей с невидимыми переходами существует способ построения аппроксимации максимальной τp -бисимуляции ресурсов. Если рассматривать не бесконечное множество ресурсов сети, а только его конечное подмножество, то мы можем использовать проверку слабого свойства τp -переноса.

Пусть $N = (P, T, F, l)$ — помеченная насыщенная сеть Петри с невидимыми переходами, $q \in \text{Nat}$ — некоторый параметр. Через $\mathcal{M}_q(P)$ обозначим множество всех ресурсов, содержащих не более q фишек во всех позициях сети: $\mathcal{M}_q(P) = \{r \in \mathcal{M}(P) : |r| < q\}$.

Через $B_{\tau p}(N, q)$ обозначим наибольшую (по вложению) τp -бисимуляцию ресурсов над множеством ресурсов ограниченной

мощности $\mathcal{M}_q(P)$.

Приведем алгоритм, вычисляющий для данной насыщенной сети Петри N с невидимыми переходами максимальную τr -бисимуляцию над множеством ресурсов, мощность которых не превышает данного параметра q . Получаемое отношение является корректной аппроксимацией максимальной τr -бисимуляции ресурсов сети N .

Алгоритм 5.1. (построения аппроксимации максимальной τr -бисимуляции ресурсов)

ввод: Помеченная насыщенная сеть Петри с невидимыми переходами $N = (P, T, F, l)$, параметр $q \in \text{Nat}$.

вывод: Отношение $B_{\tau r}(N, q)$.

шаг 1: Пусть $C = \{(\emptyset, \emptyset)\}$ — пустое множество пар (рассматриваемое как отношение над $\mathcal{M}_q(P)$).

шаг 2: Построим $B = (\mathcal{M}_q(P) \times \mathcal{M}_q(P)) \setminus C$. В силу конечности $\mathcal{M}_q(P)$ множество пар B также конечно.

шаг 3: Построим B_s — основной базис B (при помощи алгоритма 2.2).

шаг 4: Проверим, обладает ли B_s слабым свойством τr -переноса. Для этого переберем элементы из B_s^{nr} — подмножества нереклексивных элементов.

- Если все пары обладают слабым свойством τr -переноса, то возвращаем B — искомую бисимуляцию.
- В противном случае существуют $(r, s) \in B_s^{nr}$ и $t \in T$, где
 - $t \cap r \neq \emptyset$, такие, что срабатывание $M_1 \xrightarrow{t} M_1'$, где $M_1 = \bullet t \cup r$, не может быть имитировано срабатыванием какого-либо параллельного шага с той же меткой и предусловием $M_2 = \bullet t - r + s$. В этом случае добавляем (r, s) и (s, r) в C и возвращаемся на шаг 2.

Поскольку для любой разметки множество готовых к срабатыванию параллельных шагов конечно, а также конечно и пе-

ребираемое множество пар ресурсов $\mathcal{M}_q(P) \times \mathcal{M}_q(P)$, алгоритм завершит свою работу за конечное число шагов.

Оценим трудоемкость алгоритма. Обозначим $\mathcal{R} = |\mathcal{M}_q(P)|$ — мощность рассматриваемого множества ресурсов.

Алгоритм 5.1 отличается от алгоритма 3.2 только тем, что на втором шаге проверяется выполнение не слабого свойства переноса, а слабого свойства tr -переноса. При такой проверке происходит перебор не множества переходов, а множества параллельных шагов с не более чем одной видимой меткой. При этом рассматриваются не все параллельные шаги (которых бесконечно много), а только те, которые могут сработать при разметке M_2 . Оценим трудоемкость такого перебора.

Каждый невидимый переход сети может входить в искомый параллельный шаг не более чем $|M_2|$ раз, так как он забирает из начальной разметки как минимум одну фишку.¹ Также в параллельный шаг может входить не более одного перехода, помеченного видимой меткой. Следовательно, нам необходимо перебрать не более $|T||M_2|^{|T|}$ мультимножеств переходов.

Мощность разметки $M_2 = \bullet t - r + s$ можно оценить как $O(|s|) = O(q)$. Используя оценку трудоемкости алгоритма 3.2 (для обыкновенных сетей Петри), получим следующую оценку трудоемкости для алгоритма 5.1:

$$O(\max\{|P| \mathcal{R}^9, |T|^2 q^{|T|} |P| \mathcal{R}^7\}).$$

Итак, при переходе к сетям с невидимыми переходами трудоемкость алгоритма существенно повысилась (линейная зависимость от $|T|$ сменилась экспоненциальной). Такой скачок объясняется переходом от множеств переходов к мультимножествам при определении бисимуляций.

¹Без ограничения общности мы можем полагать, что в сети нет невидимых переходов с пустым предусловием, так как такие переходы избыточны и всегда могут быть удалены из сети вместе с позициями, входящими в их постусловия.

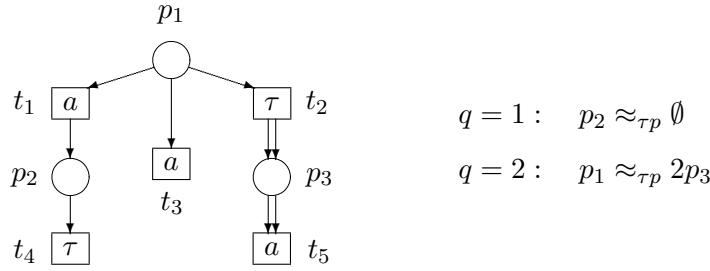


Рис. 5.6: Пример построения аппроксимации τp -бисимуляции ресурсов для насыщенной сети Петри с невидимыми переходами

Рассмотрим пример построения аппроксимации τp -бисимуляции ресурсов (рисунок 5.6). Положив параметр q равным единице, мы получили, что ресурс p_2 τp -подобен пустому ресурсу (то есть позиция p_2 избыточна). Увеличив параметр на единицу ($q = 2$), мы получили еще одну пару подобных ресурсов $p_1 \approx_{\tau p} 2p_3$.

5.2 Сети Петри высокого уровня

Класс обыкновенных сетей Петри не всегда удобен для моделирования систем, содержащих большие наборы одинаковых элементов или хорошо структурированные иерархические конструкции. Сети Петри слабо структурированы, поэтому существует достаточно много формализмов на их основе, в которых тем или иным способом вводятся конструкции для моделирования модульности и/или иерархичности.

Среди всех этих формализмов выделяется (как по степени изученности, так и по популярности в практическом применении) набор классов моделей, объединенных общим названием “сети Петри высокого уровня”. Это, например, базисные сети высокого уровня [67], предикатные сети [42], алгебраические сети [63], раскрашенные сети К. Йенсена [53].

В отличие от обыкновенных сетей Петри, где используется только один вид фишек (черная точка), в сетях высокого уровня разметка задается при помощи фишек различных типов (цветов). Количество типов конечно, каждый тип содержит конечное число индивидуальных фишек. Например, тип **Boolean** содержит два экземпляра фишек — **true** и **false**.

Чтобы управлять значениями переносимых фишек, в сетях высокого уровня дугам присваиваются специальные выражения некоторого типизированного языка, содержащие в том числе переменные. Таким образом, переходы могут срабатывать в различных режимах, в зависимости от переносимых по инцидентным дугам фишек. Кроме того, каждому переходу присваивается булевская охранная функция, разрешающая или запрещающая его срабатывание в зависимости от значений фишек на входе.

Сети высокого уровня позволяют строить гораздо более компактные и структурированные модели для многих классов систем. В них удобнее вносить элементы объектно-ориентированного подхода, конструкции модульности и иерархичности. При этом класс сетей высокого уровня (с конечным множеством индивидуальных фишек) по выразительной мощности совпадают с классом обыкновенных сетей и, следовательно, сохраняют все положительные свойства разрешимости обыкновенных сетей Петри.

Существуют простые алгоритмы, преобразующие сеть Петри высокого уровня в обыкновенную сеть (как правило, гораздо большего размера). Однако обратное преобразование весьма проблематично, поскольку может привести к совершенно неудачным с точки зрения наглядности и удобства анализа моделям. Если сравнивать сети Петри с языками программирования, то сети высокого уровня — это процедурные языки высокого уровня наподобие **C++** или **Pascal** (хотя и без рекурсии), а обыкновенные сети Петри — низкоуровневые машинные языки или **Assembler**. Известно, что восстановить по машинному коду исходный текст программы практически невозможно; в сетях Петри наблюдается

похожая ситуация.

Очевидно, что наибольший интерес представляют алгоритмы анализа именно для сетей высокого уровня, без перевода их в низкоуровневые сети.

В этом разделе исследуются возможности эффективного построения сохраняющих бисимуляцию отношений на множестве ресурсов для сетей Петри высокого уровня на примере одного из самых популярных классов сетей высокого уровня — класса раскрашенных сетей (Йенсена).

5.2.1 Раскрашенные сети Петри

Раскрашенные сети Петри (Coloured Petri Nets, CPN) были предложены в 1981 году К. Йенсеном [52, 53]. В настоящее время это один из наиболее популярных и широко применяемых на практике формализмов. CPN-модели используются во многих программных системах моделирования и анализа (среди которых нужно выделить в первую очередь системы Design/CPN и CPN Tools).

В раскрашенных сетях множество типов фишек называется множеством *цветов*. Позициям также приписан цвет, при этом позиция может содержать только фишки соответствующего цвета. На практике иногда используют модели на основе CPN, содержащие цвета с потенциально бесконечным набором различных фишек, например, тип **Integer**. Это не совсем корректно, так как раскрашенная сеть в таком случае равносильна машине Тьюринга и не поддается анализу. Но, с другой стороны, на практике тип **Integer** никогда не является бесконечным, он ограничивается возможностями ЭВМ и допусками конкретной системы программирования.

В данной работе рассматриваются раскрашенные сети Петри с конечными типами (цветами). Рассмотрим их формальное определение.

Пусть задан некоторый язык $\mathcal{L}$ типизированных выражений и некоторая конечная модель $\mathcal{U}$ этого языка. Выражения в языке

$\mathcal{L}$ построены из переменных и констант с использованием только операции сложения мультимножеств. Элементами $\mathcal{U}$ являются индивидуальные (различимые) фишки. Pin (цвет) определяется как конечное подмножество $\mathcal{U}$, типом суммы двух мультимножеств является объединение их типов.

Тип элемента e обозначается как $Type(e)$, тип выражения $\theta \in \mathcal{L}$ — как $Type(\theta)$. $Var(b)$ обозначает множество всех переменных в выражении b .

Определение 5.16. *Помеченной раскрашенной сетью Петри (Coloured Petri Net) называется набор $\mathcal{N} = (\Omega, N, Type, W, G, l)$, где*

- Ω — конечное непустое множество типов (цветов);
- $N = (P, T, F)$ — сеть Петри;
- $Type : P \rightarrow \Omega$ — функция раскраски позиций;
- $W : F \rightarrow \mathcal{L}$ — функция, сопоставляющая дугам выражения языка $\mathcal{L}$, такие, что для всех $p \in P, t, u \in T$, где $(t, p), (p, u) \in F$, выполняется $Type(W(t, p)) = Type(W(p, u)) = Type(p)$;
- $G : T \rightarrow \mathbb{B}$ — булевская охранный функция;
- $l : T \rightarrow Act$ — помечающая функция.

В раскрашенных сетях все дуги имеют кратность 1. Однако это не означает, что по дуге всегда транспортируется только одна фишка. Кратность переносится из графовой структуры на уровень выражений языка $\mathcal{L}$, помечающих дуги. Например, выражение $x + x$ (записываемое также как $2x$), где x — переменная, показывает, что по дуге транспортируются две *одинаковые* фишки. Выражение $x + y$, где x и y — переменные, показывает, что транспортируются две *произвольные* (не обязательно одинаковые) фишки.

Определим поведение раскрашенной сети Петри.

Определение 5.17. Пусть $\mathcal{N} = (\Omega, N, Type, W, G, l)$ — помеченная раскрашенная сеть Петри.

- Разметкой сети $\mathcal{N}$ называется функция $M : P \rightarrow \mathcal{M}(\mathcal{U})$, такая, что

$$\forall p \in P \quad Type(M(p)) = Type(p).$$

Множество всех разметок раскрашенной сети $\mathcal{N}$ обозначим как $\mathbf{M}(\mathcal{N})$.

- Для перехода $t \in T$ через $Var(t)$ обозначим множество всех переменных в выражении $G(t)$ и во всех выражениях на входных и выходных дугах данного перехода.
- Означиванием перехода t (переменных из множества $Var(t)$) называется функция $b : Var(t) \rightarrow \mathcal{M}(\mathcal{U})$, такая, что

$$\forall v \in Var(t) \quad b(v) \in Type(v).$$

Переход t при означивании b обозначается как $t[b]$. Множество всех возможных означиваний перехода t обозначим как $Y(t)$. Через $\mathcal{T}(\mathcal{N})$ обозначим множество означенных переходов сети $\mathcal{N}$:

$$\mathcal{T}(\mathcal{N}) = \{t[b] \mid t \in T, b \in Y(t)\}.$$

- Определим *пред-* и *постусловие* означенного перехода $t[b]$:

$$\bullet t[b] = \sum_{p \in P} W(p, t)[b], \quad t \bullet [b] = \sum_{p \in P} W(t, p)[b].$$

- Означенный переход $t[b]$ *готов к срабатыванию* при разметке M , если $G(t)[b] = true$ и $\bullet t[b] \subseteq M$.
- Готовый к срабатыванию при разметке M означенный переход $t[b]$ может *сработать*, порождая новую разметку $M' = M - \bullet t[b] + t \bullet [b]$ (обозначается $M \xrightarrow{t[b]} M'$).

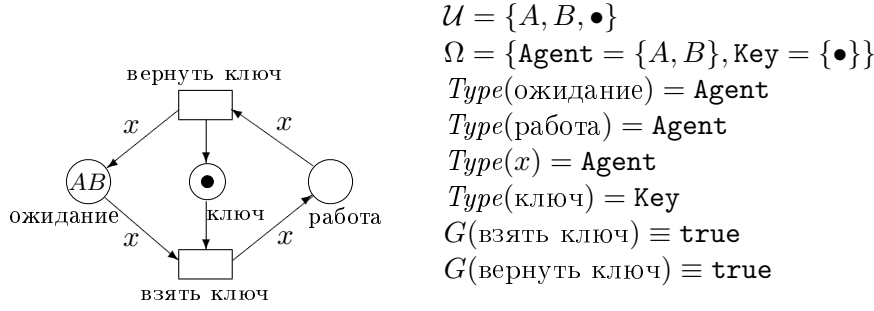


Рис. 5.7: Разделенный доступ к общему ресурсу для двух агентов (процессов)

Таким образом, при одной и той же разметке переход может сработать в разных режимах (в зависимости от означивания). Число возможных означиваний конечно.

Пример раскрашенной сети Петри приведен на рисунке 5.7. Эта сеть моделирует ту же самую систему разделения доступа, что и обыкновенная сеть Петри, изображенная на рисунке 1.7. Однако она гораздо компактней (три позиции вместо пяти, два перехода вместо четырех) и структурированной. Дублирование двух идентичных по свойствам и поведению агентов (процессов) перенесено из графовой структуры сети на уровень различных фишек. Дальнейшее увеличение количества агентов в исходной системе при моделировании сведется к простому увеличению количества фишек в типе **Agent**.

Широко известно следующее свойство раскрашенных сетей:

Факт 5.1. *Класс раскрашенных сетей Петри совпадает по выразительной мощности с классом обыкновенных сетей Петри.*

Доказательство. ($\supseteq$) Очевидно, так как обыкновенные сети Петри — это частный случай раскрашенных сетей Петри, для которых выполняется $\mathcal{U} = \{\bullet\}$ и $G(t) \equiv \text{true}$ для любого $t \in T$.

($\subseteq$) Существует простой способ “разворачивания” раскрашенной сети в обыкновенную: каждая раскрашенная позиция p заменяется на набор позиций (по одной на каждый элемент $C(p)$), каждый переход t заменяется на набор переходов (по одному на каждое означивание, допустимое в соответствии с $G(t)$), соответствующим образом перенаправляются дуги. Примером такого “разворачивания” является сеть на рисунке 1.7 (по отношению к сети на рисунке 5.7). $\square$

Бисимуляция разметок для раскрашенных сетей определяется так же, как и для обыкновенных сетей Петри (но с “поправкой” на означивания переходов):

Определение 5.18. Пусть $\mathcal{N} = (\Omega, N, Type, W, G, l)$ — помеченная раскрашенная сеть Петри. Отношение $R \subseteq \mathbf{M}(\mathcal{N}) \times \mathbf{M}(\mathcal{N})$ на множестве разметок обладает *свойством переноса*, если для любой пары $(M_1, M_2) \in R$ и для любого означенного перехода $t[b] \in \mathcal{T}(\mathcal{N})$, такого что $M_1 \xrightarrow{t[b]} M'_1$, найдется имитирующий означенный переход $u[v] \in \mathcal{T}(\mathcal{N})$, такой что $l(t) = l(u)$, $M_2 \xrightarrow{u[v]} M'_2$ и $(M'_1, M'_2) \in R$.

Определение 5.19. Если отношения R и R^{-1} обладают свойством переноса, то R называется *бисимуляцией разметок*.

Из неразрешимости бисимуляции разметок для обыкновенных сетей Петри [48] и леммы 5.1 следует неразрешимость бисимуляции разметок для раскрашенных сетей Петри.

5.2.2 Элементарные ресурсы

В обыкновенных сетях Петри позиция представляет собой достаточно простой объект, состояние которого характеризуется целым неотрицательным числом (количеством фишек в данный момент времени). В раскрашенных сетях позиция обладает некоторой внутренней структурой, поэтому ее состояние задается век-

тором (мультимножеством над множеством индивидуальных фишек данного цвета). Чтобы выделить каждую составляющую этого вектора, мы вводим понятие *элементарного ресурса*.

Определение 5.20. *Элементарным ресурсом* в раскрашенной сети Петри $\mathcal{N} = (\Omega, N, Type, W, G, l)$ называется пара

$$(p, d) \in P \times \mathcal{U}, \text{ где } d \in Type(p).$$

Множество всех элементарных ресурсов раскрашенной сети $\mathcal{N}$ обозначается как $\mathbf{E}(\mathcal{N})$.

Элементарный ресурс можно рассматривать как “часть” раскрашенной позиции, соответствующую некоторой индивидуальной фишке. Например, в сети на рисунке 5.7 позиции *ожидание* соответствует два элементарных ресурса — $(\text{ожидание}, A)$ и $(\text{ожидание}, B)$.

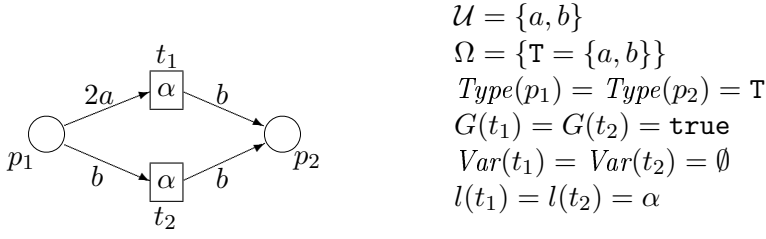
Элементарный ресурс мы будем в дальнейшем использовать как аналог позиции обыкновенной сети Петри.

Определение 5.21. *Ресурсом* в раскрашенной сети Петри $\mathcal{N}$ называется мультимножество над множеством её элементарных ресурсов $\mathbf{E}(\mathcal{N})$.

Заметим, что разметку раскрашенной сети Петри также можно рассматривать как мультимножество над множеством элементарных ресурсов $\mathbf{E}(\mathcal{N})$. Как и в случае обыкновенных сетей Петри, каждый ресурс является разметкой, и каждая разметка является ресурсом. Поэтому для обозначения множества ресурсов мы будем использовать выражение $\mathbf{M}(\mathcal{N})$ (то же самое, что и для множества разметок).

5.2.3 Подобие и бисимуляция ресурсов в раскрашенных сетях

Определим понятие подобия ресурсов раскрашенной сети Петри.

Рис. 5.8: Пример подобных ресурсов $((p_1, b) \approx 2 \cdot (p_1, a))$

Определение 5.22. Пусть $\mathcal{N} = (\Omega, N, Type, W, G, l)$ — раскрашенная сеть Петри. Ресурсы r и s называются *подобными* (обозначается $r \approx s$), если для любой разметки R , такой, что $r \subseteq R$, выполняется $R \sim R - r + s$.

Сохраняются все свойства подобия ресурсов.

Утверждение 5.6. 1. *Подобие ресурсов раскрашенных сетей Петри замкнуто относительно аддитивности и транзитивности и обладает конечным АТ-базисом.*

2. *Подобие ресурсов раскрашенных сетей Петри неразрешимо.*

Доказательство. 1) Следует из определения подобия ресурсов и теоремы 2.1.

2) Легко доказать, используя неразрешимость подобия ресурсов в обыкновенных сетях Петри (следствие 2.1) и равносильность классов раскрашенных и обыкновенных сетей (лемма 5.1). $\square$

Простой пример подобных ресурсов в раскрашенной сети Петри приведен на рисунке 5.8. В данном случае одна фишка b в позиции p_1 подобна двум фишкам a в той же самой позиции p_1 .

Определение 5.23. Пусть $\mathcal{N} = (\Omega, N, Type, W, G, l)$ — раскрашенная сеть Петри. Симметричное 1-рефлексивное отношение $B \subseteq \mathbf{M}(\mathcal{N}) \times \mathbf{M}(\mathcal{N})$ называется *бисимуляцией ресурсов*, если B^{AT} есть бисимуляция разметок.

Для раскрашенных сетей верно утверждение, аналогичное теореме 3.3 для обыкновенных сетей:

Утверждение 5.7. Пусть $\mathcal{N} = (\Omega, N, \text{Туре}, W, G, l)$ — раскрашенная сеть Петри. Тогда

1. если $B_1 \subseteq \mathbf{M}(\mathcal{N}) \times \mathbf{M}(\mathcal{N})$ — бисимуляция ресурсов и выполнено $B_2 \subseteq B_1$, то отношение B_2 — подобие ресурсов;
2. если $B_1, B_2 \subseteq \mathbf{M}(\mathcal{N}) \times \mathbf{M}(\mathcal{N})$ — бисимуляции ресурсов, то отношение $B_1 \cup B_2$ — бисимуляция ресурсов;
3. для $\mathcal{N}$ существует максимальная по вложению бисимуляция ресурсов (обозначается как $\simeq$);
4. $\simeq$ является отношением эквивалентности.

Доказательство. 1) Непосредственно из определений.

2) Доказательство аналогично доказательству теоремы 3.3, но с учетом специфики раскрашенных сетей (моделируются срабатывания *означенных* переходов):

Пусть $B = B_1 \cup B_2$. Покажем, отношение что B^{AT} — бисимуляция разметок.

Предположим противное: свойство переноса для B^{AT} не выполняется, то есть существуют $(M_1, M_2) \in B^{AT}$, $t[b] \in \mathcal{T}(\mathcal{N})$, где $M_1 \xrightarrow{t[b]} M'_1$, такие, что не найдется моделирующего означенного перехода $u[v] \in \mathcal{T}(\mathcal{N})$ с той же меткой $l(t) = l(u)$, для которого бы выполнялось $M_2 \xrightarrow{u[v]} M'_2$ и $(M'_1, M'_2) \in B^{AT}$.

Рассмотрим пару разметок $(M_1, M_2) \in B^{AT}$. Согласно лемме 2.3 она получена транзитивным замыканием нескольких пар из B^A — аддитивного замыкания B :

$$(H_1, H_2), (H_2, H_3), \dots, (H_{k-1}, H_k) \in B^A, \text{ где } H_1 = M_1, H_k = M_2.$$

Рассмотрим пару (H_1, H_2) .

$$(H_1, H_2) = (L_1 + L_2, R_1 + R_2), \text{ где } (L_1, R_1) \in (B_1)^{AT}, (L_2, R_2) \in (B_2)^{AT}$$

В силу 1-рефлексивности B_1 в $(B_1)^{AT}$ входит пара $(L_1, R_1) + (L_2, L_2) = (H_1, R_1 + L_2)$. Заметим, что $H_1 = M_1$, тогда в силу свойства переноса для $(B_1)^{AT}$ существует означенный переход $w[x]$, имитирующий срабатывание перехода $t[b]$ при разметке $R_1 + L_2$:

$$R_1 + L_2 \xrightarrow{w[x]} G_1, \text{ где } l(w) = l(t) \text{ и } (M'_1, G_1) \in (B_1)^{AT}.$$

Аналогично, в силу 1-рефлексивности отношения B_2 в $(B_2)^{AT}$ входит пара $(L_2, R_2) + (R_1, R_1) = (R_1 + L_2, H_2)$. В силу свойства переноса для $(B_2)^{AT}$ существует означенный переход $y[z]$, имитирующий срабатывание перехода $w[x]$ при разметке H_2 :

$$H_2 \xrightarrow{y[z]} G_2, \text{ где } l(y) = l(w) = l(t) \text{ и } (G_1, G_2) \in (B_2)^{AT}.$$

Очевидно, что $(B_1)^{AT} \in B^{AT}$ и $(B_2)^{AT} \in B^{AT}$. Кроме того, отношение B^{AT} транзитивно замкнуто, так что мы можем замкнуть пары (M'_1, G_1) и (G_1, G_2) , получив пару $(M'_1, G_2) \in B^{AT}$. Итак, срабатывание перехода $y[z]$ имитирует срабатывание перехода $t[b]$ в смысле B^{AT} .

Повторив это построение последовательно для всех пар вида (H_i, H_{i+1}) , мы получим в итоге искомый означенный переход $u[v]$, имитирующий $t[b]$ при разметке M_2 .

3)–4) Аналогично схеме доказательства утверждений 2) и 3) теоремы 3.3. $\square$

Определение слабого свойства переноса для CPN:

Определение 5.24. Пусть $\mathcal{N} = (\Omega, N, Type, W, G, l)$ — раскрашенная сеть Петри. Отношение $B \subseteq \mathbf{M}(\mathcal{N}) \times \mathbf{M}(\mathcal{N})$ обладает *слабым свойством переноса*, если для всех $(r, s) \in B$, $t[b] \in \mathcal{T}(\mathcal{N})$, таких что $\bullet t[b] \cap r \neq \emptyset$, найдется имитирующий означенный переход $u[v] \in \mathcal{T}(\mathcal{N})$, такой что $l(t) = l(u)$ и, обозначив $M_1 = \bullet t[b] \cup r$

и $M_2 = \bullet t[b] - r + s$, мы имеем $M_1 \xrightarrow{t[b]} M_1'$ и $M_2 \xrightarrow{u[v]} M_2'$, где $(M_1', M_2') \in B^{AT}$.

Теорема 5.3. Пусть $\mathcal{N} = (\Omega, N, \text{Туре}, W, G, l)$ — раскрашенная сеть Петри. Симметричное 1-рефлексивное отношение $B \subseteq \mathbf{M}(\mathcal{N}) \times \mathbf{M}(\mathcal{N})$ обладает слабым свойством переноса тогда и только тогда, когда B — бисимуляция ресурсов.

Доказательство. $(\Rightarrow)$ Предположим противное: свойство переноса для B^{AT} не выполняется, то есть существуют $(M_1, M_2) \in B^{AT}$ и $t[b] \in \mathcal{T}(\mathcal{N})$, где $M_1 \xrightarrow{t[b]} M_1'$, такие, что не найдется моделирующего означенного перехода $u[v] \in \mathcal{T}(\mathcal{N})$ с той же меткой $l(t) = l(u)$, для которого бы выполнялось $M_2 \xrightarrow{u[v]} M_2'$ и $(M_1', M_2') \in B^{AT}$.

Рассмотрим пару разметок $(M_1, M_2) \in B^{AT}$. Согласно лемме 2.3 она получена транзитивным замыканием нескольких пар из B^A — аддитивного замыкания B :

$$(H_1, H_2), (H_2, H_3), \dots, (H_{k-1}, H_k) \in B^A, \text{ где } H_1 = M_1, H_k = M_2.$$

Рассмотрим пару (H_1, H_2) .

$$(H_1, H_2) = (r_1 + r_2 + \dots + r_l, s_1 + s_2 + \dots + s_l), \text{ где } (r_i, s_i) \in B$$

$H_1 = \bullet t[b] \cup r_1 + F_1$. В силу слабого свойства переноса для пары (r_1, s_1) всегда найдется означенный переход $w[x]$ с той же меткой $l(t) = l(w)$, такой, что $\bullet t[b] \cup r_1 \xrightarrow{t[b]} G_1$ и $\bullet t[b] - r_1 + s_1 \xrightarrow{w[x]} G_2$, где $(G_1, G_2) \in B^{AT}$.

Так как $\bullet t[b] \cup r_1 \subseteq H_1$, к пред- и постусловиям срабатываний мы можем добавить ресурс $F = H_1 - \bullet t[b] \cup r_1$:

$$\begin{aligned} & \bullet t[b] \cup r_1 + F \xrightarrow{t[b]} G_1 + F \\ & \bullet t[b] - r_1 + s_1 + F \xrightarrow{w[x]} G_2 + F \end{aligned}$$

В силу 1-рефлексивности B и аддитивной замкнутости B^{AT} полученная пара разметок также разложима по $B : (G_1 + F, G_2 + F) \in B^{AT}$.

Получаем новая разметка $H'_1 = \bullet t[b] - r_1 + s_1 + F = H_1 - r_1 + s_1$. Заметим, что она содержит $r_2 + \dots + r_l$. То есть можно еще раз применить использованный выше прием для замены ресурса r_2 на бисимулярный ресурс s_2 .

Применив его еще в общей сложности $l - 1$ раз и используя свойство транзитивной замкнутости B^{AT} , в итоге получим означенный переход $y[z]$, имитирующий $t[b]$ при разметке H_2 .

Перейдем к следующей паре (H_2, H_3) , повторим все для перехода w . И т.д. вплоть до последней пары (H_{k-1}, H_k) . В итоге мы получим означенный переход $u[v]$, имитирующий $t[b]$ при разметке $H_k = M_2$.

($\Leftarrow$) Очевидно, так как слабое свойство переноса — это свойство переноса, ограниченное конечным числом пар. $\square$

5.2.4 Алгоритм построения аппроксимации

Для раскрашенных сетей возможно построение аппроксимации максимальной бисимуляции ресурсов. Как и в случае обыкновенных сетей Петри, при рассмотрении конечного подмножества множества ресурсов сети мы можем использовать проверку слабого свойства переноса.

Пусть $\mathcal{N} = (\Omega, N, Type, W, G, l)$ — раскрашенная сеть Петри, $q \in Nat$ — некоторый параметр.

Через $B(\mathcal{N}, q)$ обозначим наибольшую (по вложению) бисимуляцию ресурсов над множеством ресурсов ограниченной мощности $\mathbf{M}_q(\mathcal{N})$ (где $\mathbf{M}_q(\mathcal{N}) = \{r \in \mathbf{M}(\mathcal{N}) \mid |r| \leq q\}$).

Приведем алгоритм, вычисляющий для данной раскрашенной сети Петри $\mathcal{N}$ максимальную бисимуляцию над множеством ресурсов, мощность которых не превышает данного параметра q . Получаемое отношение является корректным приближением максимальной бисимуляции ресурсов сети $\mathcal{N}$.

Алгоритм 5.2. (построения аппроксимации максимальной бисимуляции ресурсов раскрашенной сети Петри)

ввод: Раскрашенная сеть Петри $\mathcal{N} = (\Omega, N, \text{Type}, W, G, l)$, параметр $q \in \text{Nat}$.

вывод: Отношение $B(\mathcal{N}, q)$.

шаг 1: Пусть $C = \{(\emptyset, \emptyset)\}$ — пустое множество пар (рассматриваемое как отношение над $\mathbf{M}_q(\mathcal{N})$).

шаг 2: Построим $B = (\mathbf{M}_q(\mathcal{N}) \times \mathbf{M}_q(\mathcal{N})) \setminus C$. В силу конечности $\mathbf{M}_q(\mathcal{N})$ множество пар B также конечно.

шаг 3: Построим B_s — основной базис отношения B (по алгоритму 2.2).

шаг 4: Проверим, обладает ли B_s слабым свойством переноса. Для этого переберем элементы из B_s^{nr} — подмножества нереклексивных элементов.

- Если все пары обладают слабым свойством переноса, то возвращаем B — искомую бисимуляцию.
- В противном случае существуют $(r, s) \in B_s^{nr}$, $t[b] \in \mathcal{T}(\mathcal{N})$, где $\bullet t[b] \cap r \neq \emptyset$, такие, что срабатывание $M_1 \xrightarrow{t[b]} M_1'$, где $M_1 = \bullet t[b] \cup r$, не может быть имитировано срабатыванием какого-либо перехода с той же меткой и предусловием $M_2 = \bullet t[b] - r + s$. В этом случае добавляем (r, s) и (s, r) в C и возвращаемся на шаг 2.

Поскольку для любой разметки множество готовых к срабатыванию означенных переходов конечно, а также конечно и перебираемое множество пар ресурсов $\mathbf{M}_q(\mathcal{N}) \times \mathbf{M}_q(\mathcal{N})$, алгоритм завершит свою работу за конечное число шагов.

Оценим трудоемкость алгоритма. Обозначим $\mathcal{R} = |\mathbf{M}_q(\mathcal{N})|$ — мощность рассматриваемого множества ресурсов.

Алгоритм 5.2 отличается от алгоритма 3.2 множествами перебора. Во-первых, мы рассматриваем ресурс не как мультимножество позиций, а как мультимножество элементарных ресурсов

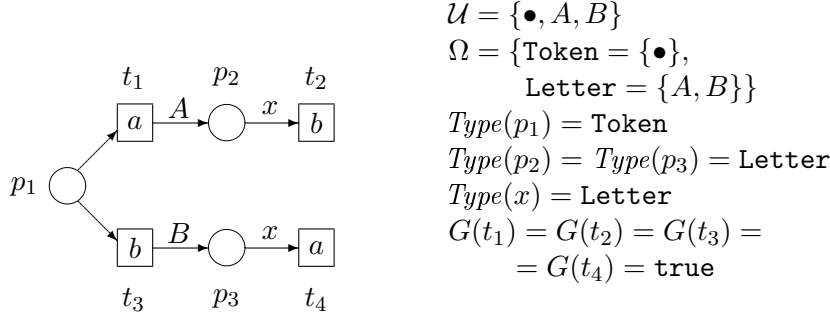


Рис. 5.9: Пример раскрашенной сети Петри

(вместо множества P перебираем множество $\mathbf{E}(\mathcal{N})$). Еще одно отличие состоит в том, что на втором шаге проверяется выполнение слабого свойства переноса для раскрашенных сетей. При такой проверке происходит перебор не множества переходов (как в случае обыкновенных сетей), а множества *означенных* переходов. Мощность множества означенных переходов сети $\mathcal{N}$ равна $|\mathcal{T}(\mathcal{N})|$.

Заменяя в оценке трудоемкости алгоритма 3.2 (для обыкновенных сетей Петри) $|P|$ на $|\mathbf{E}(\mathcal{N})|$ и $|T|$ на $|\mathcal{T}(\mathcal{N})|$, получим следующую оценку трудоемкости для алгоритма 5.2:

$$O(\max\{|\mathbf{E}(\mathcal{N})| \mathcal{R}^9, |\mathcal{T}(\mathcal{N})|^2 |\mathbf{E}(\mathcal{N})| \mathcal{R}^7\}).$$

Рассмотрим пример построения аппроксимации бисимуляции ресурсов в раскрашенной сети Петри (Рис. 5.9). Положив параметр q равным двум, получим следующий набор подобных ресурсов:

$$(p_1, \bullet) \approx (p_2, A) + (p_3, A) \approx (p_2, A) + (p_3, B) \approx$$

$$\approx (p_2, B) + (p_3, A) \approx (p_2, B) + (p_3, B).$$

5.3 Вложенные сети Петри

Формализм вложенных сетей Петри (Nested Petri Nets, NP-сети) был предложен И.А. Ломазовой [14, 15, 17, 55]. Во вложенных сетях Петри в качестве фишек выступают обычные сети Петри, которые могут функционировать как автономно, так и во взаимодействии между собой (горизонтальная синхронизация) и с системной сетью (вертикальная синхронизация). Этот механизм позволяет интуитивно ясно моделировать сложные иерархические и мультиагентные системы, имеющие динамическую структуру.

Вложенные сети Петри занимают строго промежуточное по выразительности положение между обыкновенными сетями Петри и универсальными вычислительными моделями. В частности, для них неразрешимы проблемы достижимости, живости и ограниченности, разрешимые для обыкновенных сетей. В то же время вложенные сети Петри являются вполне структурированными системами переходов, для них разрешимы проблемы останова и покрытия.

В этом разделе исследуются возможности эффективного построения сохраняющих бисимуляцию отношений на множестве ресурсов для расширения класса обыкновенных сетей Петри — так называемых вложенных сетей Петри.

5.3.1 Определение вложенных сетей Петри

Далее приводятся формальные определения класса двухуровневых вложенных сетей Петри.

Пусть $Lab_v = \{l_1, l_2, \dots\}$ и $Lab_h = \{\lambda_1, \lambda_2, \dots\}$ — два непересекающихся множества меток. Для каждой метки из $l \in Lab_v$ и для каждой метки $\lambda \in Lab_h$ имеется *двойственная* метка $\bar{l} \in Lab_v$, соответственно, $\bar{\lambda} \in Lab_h$, такие что

1. $\bar{l}_1 \neq \bar{l}_2$ для $l_1, l_2 \in Lab_v$, $l_1 \neq l_2$; $\bar{\lambda}_1 \neq \bar{\lambda}_2$ для $\lambda_1, \lambda_2 \in Lab_h$, $\lambda_1 \neq \lambda_2$;

$$2. \bar{l} =_{def} l; \bar{\lambda} =_{def} \lambda.$$

Метки множества Lab_v будут использоваться для вертикальной синхронизации переходов, метки множества Lab_h — для горизонтальной синхронизации. Полагаем $Lab =_{def} Lab_v \cup Lab_h$.

Как и в случае обыкновенных сетей Петри, для наблюдения за поведением вложенных сетей их переходы помечаются символами из алфавита меток срабатываний Act . Необходимо, чтобы эти метки не конфликтовали с метками из Lab . Без ограничения общности здесь и далее мы предполагаем, что множество Act содержит все синхронизирующие метки из Lab и пары двойственных меток вида $(l, \bar{l})$ или $(\lambda, \bar{\lambda})$.

Пусть задан некоторый язык выражений $\mathcal{L}$. Выражения в языке $\mathcal{L}$ построены из переменных и констант с использованием только операции сложения мультимножеств. Множество констант языка $\mathcal{L}$ распадается на два непересекающихся подмножества: $\mathfrak{A}$ — конечное множество атомарных констант (используемых для обозначения обычных цветных фишек) и $\mathfrak{D}$ — множество объектных констант (используемых для обозначения сетевых фишек).

$Var(b)$ обозначает множество всех переменных в выражении b . $Const(b)$ обозначает множество всех констант в выражении b .

Определение 5.25. Помеченной вложенной сетью Петри (NP-сетью) называется набор $NPN = (\mathfrak{A}, \{EN_1, \dots, EN_k\}, SN, Lab, \Lambda)$, где

1. $\mathfrak{A}$ — конечное множество атомарных фишек.
2. $\{EN_1, \dots, EN_k\}$ — конечное множество элементарных сетей, где $EN_i = (P_i, T_i, F_i, l_i)$ — обыкновенная помеченная сеть Петри.

$\mathfrak{D} = \bigcup_{i=1 \dots k} \{(EN_i, M) \mid M \in \mathcal{M}(P_i)\}$ — множество сетевых фишек, то есть множество маркированных элементарных сетей. В общем случае это множество бесконечно, так как бесконечно множество разметок обыкновенной сети Петри.

3. $SN = (N, W)$ — помеченная сеть Петри высокого уровня, где

- $N = (P, T, F, l)$ — помеченная сеть Петри, такая, что все дуги в F имеют кратность 1.
- W — функция, сопоставляющая каждой дуге $(x, y) \in F$ некоторое выражение $W(x, y)$ языка $\mathcal{L}$. При этом на выражения, приписанные входным дугам переходов, накладываются следующие ограничения:
 - эти выражения не содержат констант из $\mathfrak{D}$ (то есть только константы из $\mathfrak{A}$ и переменные);
 - любая переменная встречается в выражении $W(p, t)$ не более одного раза;
 - для выражений $W(p_1, t)$ и $W(p_2, t)$, приписанных различным входным дугам одного и того же перехода t , выполняется $Var(W(p_1, t)) \cap Var(W(p_2, t)) = \emptyset$.

На выражения, приписанные выходным дугам, накладывается следующее ограничение: выходное выражение для перехода t может содержать только переменные, которые входят в одно из входных выражений для этого же перехода.

4. $Lab = Lab_v \cup Lab_h$ — множество меток, удовлетворяющих приведенным выше условиям (согласованности с Act).

5. Λ — частичная функция пометки переходов, помечающая некоторые переходы системной сети метками из Lab_v и некоторые переходы элементарных сетей метками из $Lab_v \cup Lab_h$.

Ограничения на выражения, приписанные дугам системной сети, необходимы для того, чтобы получаемый формализм не был слишком выразительным. Без любого из этих ограничений вложенные сети Петри стали бы равномогущны универсальным вычислительным системам.

Определение 5.26. Разметкой помеченной вложенной сети Петри $NPN = (\mathfrak{A}, \{EN_1, \dots, EN_k\}, SN, Lab, \Lambda)$ называется конечное мультимножество M пар вида (p, C) , где $p \in P$, $C \in \mathfrak{A} \cup \mathfrak{D}$.

Заметим, что разметку можно рассматривать также как отображение, сопоставляющее каждой позиции системной сети SN некоторое конечное мультимножество атомарных и сетевых фишек:

$$M : P \rightarrow \mathcal{M}(\mathfrak{A} \cup \mathfrak{D}).$$

Множество всех разметок вложенной сети NPN обозначим как $\mathfrak{M}(NPN)$.

Определение 5.27. Маркированной помеченной вложенной сетью Петри называется пара (NPN, M_0) , где

1. $NPN = (\mathfrak{A}, \{EN_1, \dots, EN_k\}, SN, Lab, \Lambda)$ — помеченная вложенная сеть Петри;
2. $M_0 \in \mathfrak{M}(NPN)$ — некоторая выделенная разметка сети NPN , называемая начальной.

Определим поведение вложенной сети Петри.

Определение 5.28. Пусть $NPN = (\mathfrak{A}, \{EN_1, \dots, EN_k\}, SN, Lab, \Lambda)$ — помеченная вложенная сеть Петри.

- Для перехода системной сети $t \in T$ через $Var(t)$ обозначим множество всех переменных во всех выражениях на его входных и выходных дугах:

$$Var(t) = \bigcup_{p \in P} \{Var(W(p, t)) \cup Var(W(t, p))\}.$$

Через $Const(t)$ обозначим множество всех констант во всех выражениях на его входных и выходных дугах:

$$Const(t) = \bigcup_{p \in P} \{Const(W(p, t)) \cup Const(W(t, p))\}.$$

- *Означиванием* перехода $t \in T$ (переменных из множества $Var(t)$) называется функция $b : Var(t) \rightarrow \mathcal{M}(\mathfrak{A} \cup \mathfrak{D})$.

Переход t при означивании b обозначается как $t[b]$.

Сетевые фишки, являющиеся значениями переменных в выражениях на входных дугах перехода $t \in T$ при означивании b , называются *задействованными* при срабатывании $t[b]$.

- Определим *пред-* и *постусловие* означенного перехода системной сети $t[b]$:

$$\bullet t[b] = \sum_{p \in P} W(p, t)[b], \quad t \bullet [b] = \sum_{p \in P} W(t, p)[b].$$

- Означенный переход системной сети $t[b]$ *готов к срабатыванию* при разметке M , если $\bullet t[b] \subseteq M$.

Определение 5.29. Для вложенной сети Петри определяются следующие четыре вида *шагов срабатывания*:

- **системно-автономный шаг:** Пусть t — переход системной сети SN , такой, что $\Lambda(t) = \emptyset$. Если переход t готов к срабатыванию при разметке M и означивании b , то он может сработать, порождая новую разметку системной сети $M' = M - \bullet t[b] + t \bullet [b]$ (обозначается $M \xrightarrow{t[b]} M'$).

Такое срабатывание перехода t в сети SN называется *системно-автономным шагом* вложенной сети NPN . Разметки сетевых фишек не меняются, но некоторые из них перемещаются из одной позиции в другую (копируются, исчезают), а также возможно появление новых сетевых фишек.

- **элементно-автономный шаг:** Пусть M — разметка сети NPN , $p \in P$ — некоторая позиция системной сети SN и $\alpha_1 \in M(p)$ — сетевая фишка, находящийся в позиции p при разметке M . Пусть $\alpha = (EN_i = (P_i, T_i, F_i, l_i), m)$. Пусть

в обыкновенной сети Петри EN_i имеется переход $t \in T_i$, такой, что $\Lambda(t) = \emptyset$ и $\bullet t \subseteq m$, причем $m \xrightarrow{t} m'$. Тогда переход t может сработать, порождая новую разметку M' системной сети SN , получаемую из разметки M заменой в позиции p сетевой фишки $\alpha = (EN, m)$ на сетевую фишку $\alpha' = (EN, m')$.

Такое срабатывание перехода t в сети EN называется элементарно-автономным шагом вложенной сети NPN .

- **шаг горизонтальной синхронизации:** Пусть M — разметка сети NPN , $p \in P$ — позиция системной сети SN , $\alpha_1 = (EN_1, m_1)$, $\alpha_2 = (EN_2, m_2)$ — сетевые фишки, содержащиеся при разметке M в позиции p . Пусть в обыкновенной сети Петри EN_1 имеется переход t_1 , такой что $\Lambda(t_1) = \lambda \in Lab_h$ и $m_1 \xrightarrow{t_1} m'_1$. Пусть при этом в сети EN_2 имеется переход t_2 , такой что $\Lambda(t_2) = \bar{\lambda} \in Lab_h$ и $m_2 \xrightarrow{t_2} m'_2$. Тогда эти два перехода в элементарных сетях могут сработать одновременно, порождая в системной сети N разметку M' , полученную заменой в позиции p фишки $\alpha_1 = (EN_1, m_1)$ на фишку $\alpha'_1 = (EN_1, m'_1)$ и фишки $\alpha_2 = (EN_2, m_2)$ на фишку $\alpha'_2 = (EN_2, m'_2)$.

Такое синхронное срабатывание переходов t_1 и t_2 в сетях EN_1 и EN_2 называется шагом горизонтальной синхронизации для вложенной сети NPN . При этом внешний наблюдатель видит единую метку $(\lambda, \bar{\lambda})$ из Act .

- **шаг вертикальной синхронизации:** Пусть M — разметка сети NPN , t — переход системной сети SN , $\Lambda(t) = l \in Lab_v$ и t готов к срабатыванию при разметке M и означивании b . Пусть далее $\{\alpha_1, \dots, \alpha_k\} \subset \mathfrak{D}$ есть множество задействованных в срабатывании $t[b]$ сетевых фишек, где $\alpha_i = (EN_i, m_i)$, и пусть для каждого $1 \leq i \leq k$ в сети EN_i найдется переход t_i , такой что t_i возбужден в EN_i при разметке m_i , $m_i \xrightarrow{t_i} m'_i$ и $\Lambda(t_i) = \bar{l}$.

Тогда переход t в системной сети может сработать одновременно со всеми переходами t_i в задействованных сетевых фишках, порождая в системной сети SN разметку M' , полученную по правилу:

$$\forall p \in P \quad M'(p) = M(p) - W(p, t)[b] + W'(t, p)[b],$$

где $W'(t, p)[b]$ есть мультимножество атомарных и сетевых фишек, полученное из мультимножества $W(t, p)[b]$ заменой сетевой фишки $\alpha_i = (EN_i, m_i)$ на фишку $\alpha'_i = (EN_i, m'_i)$ для всех $1 \leq i \leq k$.

Такое синхронное срабатывание перехода t при означивании b в системной сети и переходов $t_1, \dots, t_k$ в сетях $\alpha_1, \dots, \alpha_k$ называется шагом вертикальной синхронизации для вложенной сети NPN . Внешний наблюдатель видит единую метку $(l, \bar{l})$ из Act .

Множество переходов вложенной сети Петри NPN (всех перечисленных выше типов) обозначим как $T(NPN)$.

Рассмотрим пример вложенной сети.

На рисунке 5.10 изображена вложенная сеть Петри, моделирующая разгрузку вагона. Имеется n мешков груза, m штук носилок (неиспользуемые носилки лежат около вагона) и конечное множество G грузчиков. Грузчики могут работать в одиночку, перенося из вагона на склад один мешок за один раз, либо парами (при наличии свободных носилок), перенося вдвоем за один раз 4 мешка. При этом поведение грузчика достаточно свободное — он может носить мешки в одиночку, даже если есть свободные носилки и ищущий напарника другой грузчик. Единственное ограничение состоит в том, что уж если грузчик пришел к складу с носилками (в паре с другим грузчиком), то он обязан следующим “ходом” вернуть носилки “на исходную позицию” к вагону (хотя и не обязательно те же самые носилки с тем же самым напарником).

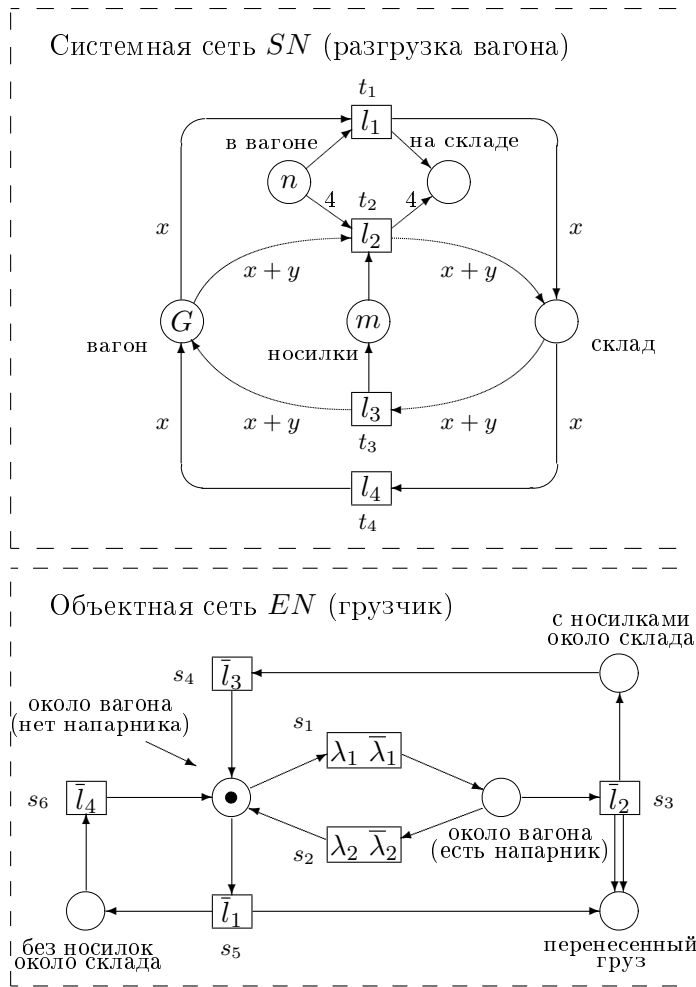


Рис. 5.10: Вложенная сеть Петри, моделирующая разгрузку вагона, содержащего n мешков груза. Имеется конечное множество G грузчиков и m носилок. Грузчики могут работать в одиночку, перенося за один раз один мешок, либо парами (при наличии свободных носилок), перенося вдвоем за один раз 4 мешка.

Сам процесс разгрузки моделируется системной сетью SN , поведение грузчика моделируется сетевой фишкой EN .

Позиции в системной сети означают следующее:

- ☐ “в вагоне” — количество мешков в вагоне (не разгруженных);
- ☐ “на складе” — количество мешков на складе (разгруженных);
- ☐ “вагон” — грузчики у вагона (содержит сетевые фишки);
- ☐ “склад” — грузчики на складе (содержит сетевые фишки);
- ☐ “носилки” — носилки у вагона (неиспользуемые).

Переходы в системной сети означают следующее:

- ☐ t_1 — какой-то грузчик переносит в одиночку 1 мешок;
- ☐ t_2 — какие-то двое грузчиков переносят на носилках 4 мешка;
- ☐ t_3 — какие-то двое грузчиков возвращаются к вагону с пустыми носилками;
- ☐ t_4 — какой-то грузчик возвращаются к вагону без носилок.

Позиции в объектной сети означают следующее:

- ☐ “около вагона (нет напарника)” — грузчик находится около вагона, но он еще не принял решение брать носилки;
- ☐ “около вагона (есть напарник)” — грузчик находится около вагона, и он принял решение брать носилки;
- ☐ “с носилками около склада” — грузчик находится около склада, причем он пришел сюда с носилками;
- ☐ “без носилок около склада” — грузчик находится около склада, причем он пришел сюда без носилок;
- ☐ “перенесенный груз” — количество мешков, уже перенесенных грузчиком.

Переходы в объектной сети означают следующее:

- ☐ s_1 — грузчик вместе с каким-то другим грузчиком решают взять носилки;
- ☐ s_2 — грузчик вместе с каким-то другим грузчиком решают все-таки не брать носилок;
- ☐ s_3 — грузчик в паре с каким-то другим грузчиком переносят на носилках 4 мешка из вагона на склад;
- ☐ s_4 — грузчик в паре с каким-то другим грузчиком возвраща-

ются к вагону с носилками;

□ s_5 — грузчик в одиночку переносит 1 мешок из вагона на склад;

□ s_6 — грузчик возвращаются к вагону без носилок.

В данной сети представлены оба вида синхронизации — вертикальная и горизонтальная. Вертикальная синхронизация моделирует использование объектами (грузчиками) системного ресурса (носилки). Горизонтальная синхронизация моделирует взаимодействие между объектами (грузчиками) при их совместной деятельности (принятие решения брать/не брать носилки).

Например, переход t_2 в системной сети может сработать только при наличии вертикальной синхронизации с обеими переносимыми им сетевыми фишками (x и y), то есть при наличии как минимум двух грузчиков, готовых нести носилки. Переход s_1 в объектной сети может сработать только совместно с таким же переходом в “соседней” фишке, то есть грузчик принимает решение брать носилки только одновременно с другим грузчиком. Этим гарантируется наличие четного числа грузчиков, готовых нести носилки.

Таким образом, системная сеть описывает правила поведения агентов (грузчиков) с точки зрения системы и системных ресурсов (носилки и мешки). Объектная сеть моделирует поведение отдельного агента. Такое разделение очень удобно при моделировании мультиагентных систем. Например, если мы захотим ввести в модель еще один вид грузчиков, которые не умеют пользоваться носилками, то нам не придется менять системную сеть — достаточно добавить еще один вид сетевых фишек. С другой стороны, если мы захотим добавить в модель явление недетерминированной пропажи груза со склада/из вагона (без участия грузчиков), то нам не придется менять описание поведения грузчика, достаточно добавить соответствующие переходы в системную сеть. Менять оба уровня модели (системный и объектный) придется только при появлении нового вида взаимодействия между агентом и системой или нового вида системных ресурсов (например, тачек).

Определим бисимуляцию разметок для вложенных сетей.

Определение 5.30. Пусть $NPN = (\mathfrak{A}, \{EN_1, \dots, EN_k\}, SN, Lab, \Lambda)$ — помеченная вложенная сеть Петри. Скажем, что отношение $R \subseteq \mathfrak{M}(NPN) \times \mathfrak{M}(NPN)$ обладает *свойством переноса*, если для любой пары разметок $(M_1, M_2) \in R$ и для любого срабатывания перехода $t \in \mathcal{T}(NPN)$, такого, что $M_1 \xrightarrow{t} M'_1$, найдется имитирующее срабатывание $u \in \mathcal{T}(NPN)$, такое, что $l(t) = l(u)$, $M_2 \xrightarrow{u} M'_2$ и $(M'_1, M'_2) \in R$.

Заметим, что из определений Lab и Act следует, что автономный шаг должен быть имитирован автономным шагом (системно-или элементарно-автономным), шаг горизонтальной синхронизации — обязательно шагом горизонтальной синхронизации, шаг вертикальной синхронизации — шагом вертикальной синхронизации.

Определение 5.31. Если отношения R и R^{-1} обладают свойством переноса, то R называется *бисимуляцией разметок*.

Обыкновенные сети Петри являются частным случаем вложенных сетей Петри (при $\mathfrak{A} = \{\bullet\}$ и $\mathfrak{S} = \emptyset$). Следовательно, бисимуляция разметок неразрешима и для вложенных сетей.

5.3.2 Объектные ресурсы

Одним из замечательных свойств сетей Петри является то, что состояние моделируемой системы задается при помощи набора локальных состояний (позиций). Таким образом, существует два уровня иерархии: глобальное состояние — разметка сети в целом, и локальное состояние — количество фишек именно в данной позиции. Срабатывание отдельного перехода зависит не от глобального состояния системы, а от конечного набора локальных состояний (определяемого входными позициями данного перехода).

В отличие от обыкновенных сетей Петри и созданных на их основе сетей высокого уровня, двухуровневые вложенные сети пред-

ставляют собой формализм, позволяющий выделить не один, а три существенно различных вида локальных состояний:

1. Локальные состояния позиций элементных сетей.
2. Локальные состояния позиций системной сети, то есть мультимножества сетевых фишек, расположенных в данный момент в данной позиции, с учетом их разметок.
3. Локальные состояния позиций системной сети без учета состояний сетевых фишек.

Ресурс в обыкновенной сети Петри представляет собой обобщение понятия позиции. Во вложенных сетях Петри появляется возможность при определении ресурса использовать любой из перечисленных выше видов “позиции”. Как будет показано далее, некоторые из подобных, индуцированных этими тремя видами ресурсов, эффективно вычислимы.

Определим понятие ресурса, использующее первый (внутри-элементный) вид локальности.

Определение 5.32. *Объектным ресурсом* помеченной вложенной сети Петри $NPN = (\mathfrak{A}, \{EN_1, \dots, EN_k\}, SN, Lab, \Lambda)$ называется пара $\langle EN_i, r \rangle$, где $r \in \mathcal{M}(P_i)$.

Объектными ресурсами вложенной сети являются ресурсы её элементных сетей, которые представляют собой обыкновенные сети Петри.

Пусть $N = (P, T, F, l)$ — обыкновенная помеченная сеть Петри. Для удобства обозначим максимальную бисимуляцию разметок сети N как $\sim_N$.

Определение 5.33. Пусть $NPN = (\mathfrak{A}, \{EN_1, \dots, EN_k\}, SN, Lab, \Lambda)$ — помеченная вложенная сеть Петри, EN_i — одна из её элементных сетей.

Объектные ресурсы $\langle EN_i, r \rangle$ и $\langle EN_i, s \rangle$ называются *подобными* (обозначается $\langle EN_i, r \rangle \approx_{EN_i} \langle EN_i, s \rangle$), если для любой разметки $M \subseteq \mathcal{M}(P_i)$, такой что $r \subseteq M$, выполняется $M \sim_{EN_i} M - r + s$.

Поскольку элементные сети являются обыкновенными помеченными сетями Петри, бисимуляция ресурсов (в данном случае называемая бисимуляцией объектных ресурсов) определяется и вычисляется для них в точности так же, как и для обыкновенных сетей. Выполняются все свойства подобия и бисимуляции ресурсов обыкновенных помеченных сетей Петри.

Подобные объектные ресурсы эквивалентны с точки зрения бисимуляции разметок вложенной сети в целом, то есть их можно заменять друг на друга, и при этом поведение всей сети не изменится (в смысле бисимilarityности):

Теорема 5.4. Пусть $NPN = (\mathcal{A}, \{EN_1, \dots, EN_k\}, SN, Lab, \Lambda)$ — помеченная вложенная сеть Петри, $\langle EN_i, r \rangle$ и $\langle EN_i, s \rangle$ — подобные объектные ресурсы сети NPN ($\langle EN_i, r \rangle \approx_{EN_i} \langle EN_i, s \rangle$).

Тогда для любой разметки $M \in \mathfrak{M}(NPN)$, содержащей сетевую фишку (EN_i, t) , такую, что $r \subseteq t$, выполняется

$$M \sim M[(EN_i, t)/(EN_i, t - r + s)],$$

где $M[(EN_i, t)/(EN_i, t - r + s)]$ — разметка, полученная из разметки M заменой фишки (EN_i, t) на фишку $(EN_i, t - r + s)$.

Доказательство. Поскольку функции пометки переходов метками из Act и Lab согласованы (в частности $Lab \subseteq Act$), при замене подобных объектных ресурсов не происходит никаких изменений в наблюдаемом поведении сетевой фишки с точки зрения синхронизации (как вертикальной, так и горизонтальной). Следовательно, системная сеть и соседние сетевые фишки (находящиеся в той же позиции системной сети) “не замечают” никаких изменений в данной сетевой фишке. $\square$

Таким образом, элементные сети могут рассматриваться отдельно от всей вложенной сети (единственным отличием является дополнительное рассмотрение меток из Lab). При этом могут быть найдены возможные избыточности во внутренней структуре элементной сети, удаление которых (и упрощение структуры сетевой фишки) никак не повлияет на функционирование системы в целом.

Пример 5.1. Рассмотрим элементную сеть EN , моделирующую грузчика (рисунок 5.10). Выполняется

$$\text{“перенесенный груз”} \approx_{EN} \emptyset,$$

то есть с точки зрения наблюдаемого поведения системы (в смысле бисимулярности) несущественно, сколько мешков уже перенес данный грузчик. Позицию “перенесенный груз” можно просто удалить, при этом все поведенческие свойства вложенной сети сохранятся, а ее анализ упростится.

5.3.3 Системные ресурсы

Во вложенных сетях Петри можно выделить ресурсы другого типа — так называемые *системные* ресурсы, использующие второй вид локальностей вложенной сети Петри. Заметим, что разметка системной сети может рассматриваться как ресурс:

Определение 5.34. *Системным ресурсом* помеченной вложенной сети Петри $NPN = (\mathfrak{A}, \{EN_1, \dots, EN_k\}, SN, Lab, \Lambda)$ называется конечное мультимножество M пар вида (p, C) , где $p \in P$, $C \in \mathfrak{A} \cup \mathfrak{Q}$.

Другими словами, системными ресурсами являются элементы множества $\mathfrak{M}(NPN)$. Разметка вложенной сети Петри является системным ресурсом и наоборот.

Определение 5.35. Системные ресурсы r и s из $\mathfrak{M}(NPN)$ называются *подобными* (обозначается $\approx$), если для любой разметки $M \subseteq \mathfrak{M}(NPN)$, такой что $r \subseteq M$, выполняется $M \sim M - r + s$.

В случае обыкновенных сетей Петри ресурс есть мультимножество над конечным множеством позиций сети. В случае вложенных сетей системный ресурс является мультимножеством над *бесконечным* множеством пар вида (p, C) , где $p \in P$, $C \in \mathfrak{A} \cup \mathfrak{D}$. Это множество бесконечно, так как в общем случае бесконечно множество размеченных сетевых фишек $\mathfrak{D}$. Поэтому мы не можем говорить о конечной базирруемости подобия системных ресурсов.

Сохраняются следующие свойства подобия ресурсов:

Утверждение 5.8. *1. Подобие системных ресурсов во вложенных сетях Петри замкнуто относительно аддитивности и транзитивности.*

2. Подобие системных ресурсов во вложенных сетях Петри неразрешимо.

Доказательство. 1) Следует из определения подобия системных ресурсов.

2) Легко доказать, используя неразрешимость подобия ресурсов в обыкновенных сетях Петри (следствие 2.1) и равномощность класса обыкновенных сетей Петри подклассу вложенных сетей при ограничениях на множество фишек $\mathfrak{A} = \{\bullet\}$ и $\mathfrak{D} = \emptyset$. $\square$

Пример 5.2. Рассмотрим вложенную сеть Петри, моделирующую разгрузку вагона (рисунок 5.10). Выполняется

$$(\text{“на складе”}, 1) \approx \emptyset,$$

то есть с точки зрения наблюдаемого поведения системы (в смысле бисимулярности) несущественно, сколько мешков уже разгружено. Позицию системной сети “на складе” можно удалить как избыточную.

По аналогии со случаем обыкновенных сетей Петри определим бисимуляцию системных ресурсов вложенной сети.

Определение 5.36. Пусть $NPN = (\mathfrak{A}, \{EN_1, \dots, EN_k\}, SN, Lab, \Lambda)$ — помеченная вложенная сеть Петри. Симметричное 1-рефлексивное отношение $B \subseteq \mathfrak{M}(NPN) \times \mathfrak{M}(NPN)$ называется *бисимуляцией системных ресурсов*, если B^{AT} есть бисимуляция разметок вложенной сети NPN .

Аналогом теоремы 3.3 является следующее

Утверждение 5.9. Пусть $NPN = (\mathfrak{A}, \{EN_1, \dots, EN_k\}, SN, Lab, \Lambda)$ — помеченная вложенная сеть Петри. Тогда

1. если $B_1 \subseteq \mathfrak{M}(NPN) \times \mathfrak{M}(NPN)$ — бисимуляция системных ресурсов и $B_2 \subseteq B_1$, то отношение B_2 — подобие системных ресурсов;
2. если $B_1, B_2 \subseteq \mathfrak{M}(NPN) \times \mathfrak{M}(NPN)$ — бисимуляции системных ресурсов, то отношение $B_1 \cup B_2$ — бисимуляция системных ресурсов;
3. для NPN существует максимальная по вложению бисимуляция ресурсов (обозначается как $\simeq$);
4. $\simeq$ является отношением эквивалентности.

Доказательство. 1) Непосредственно из определений.

2) Докажем по схеме доказательства первого утверждения теоремы 3.3, используя тот факт, что системная разметка (системный ресурс) есть мультимножество пар вида (p, C) , где $p \in P$ — позиция системной сети, $C \in \mathfrak{A} \cup \mathfrak{D}$ — фишка (атомарная или сетевая):

Пусть $B = B_1 \cup B_2$. Покажем, что B^{AT} — бисимуляция разметок. Предположим противное: свойство переноса для B^{AT} не выполняется, то есть существуют $(M_1, M_2) \in B^{AT}$, $t \in T(NPN)$, где $M_1 \xrightarrow{t} M'_1$, такие, что не найдется моделирующего перехода $u \in T(NPN)$ с той же меткой $l(t) = l(u)$, для которого бы выполнялось $M_2 \xrightarrow{u} M'_2$ и $(M'_1, M'_2) \in B^{AT}$.

Рассмотрим пару разметок $(M_1, M_2) \in B^{AT}$. Согласно лемме 2.3 она получена транзитивным замыканием нескольких пар из B^A — аддитивного замыкания B :

$$(H_1, H_2), (H_2, H_3), \dots, (H_{k-1}, H_k) \in B^A, \text{ где } H_1 = M_1, H_k = M_2.$$

Рассмотрим пару (H_1, H_2) .

$$(H_1, H_2) = (L_1 + L_2, R_1 + R_2), \text{ где } (L_1, R_1) \in (B_1)^{AT}, (L_2, R_2) \in (B_2)^{AT}$$

В силу 1-рефлексивности B_1 в $(B_1)^{AT}$ входит пара $(L_1, R_1) + (L_2, L_2) = (H_1, R_1 + L_2)$. Заметим, что $H_1 = M_1$, тогда в силу свойства переноса для $(B_1)^{AT}$ существует переход $v \in \mathcal{T}(NPN)$, имитирующий срабатывание перехода t при разметке $R_1 + L_2$:

$$R_1 + L_2 \xrightarrow{v} G_1, \text{ где } l(v) = l(t) \text{ и } (M'_1, G_1) \in (B_1)^{AT}.$$

Аналогично, в силу 1-рефлексивности отношения B_2 в $(B_2)^{AT}$ входит пара $(L_2, R_2) + (R_1, R_1) = (R_1 + L_2, H_2)$. В силу свойства переноса для $(B_2)^{AT}$ существует переход $w \in \mathcal{T}(NPN)$, имитирующий срабатывание перехода v при разметке H_2 :

$$H_2 \xrightarrow{w} G_2, \text{ где } l(w) = l(v) = l(t) \text{ и } (G_1, G_2) \in (B_2)^{AT}.$$

Очевидно, что $(B_1)^{AT} \in B^{AT}$ и $(B_2)^{AT} \in B^{AT}$. Кроме того, отношение B^{AT} транзитивно замкнуто, так что мы можем замкнуть пары (M'_1, G_1) и (G_1, G_2) , получив пару $(M'_1, G_2) \in B^{AT}$. Итак, срабатывание перехода w имитирует срабатывание перехода t в смысле B^{AT} .

Повторив это построение последовательно для всех пар вида (H_i, H_{i+1}) , мы получим в итоге искомый переход u , имитирующий t при разметке M_2 .

3)–4) Аналогично схеме доказательства утверждений 2) и 3) теоремы 3.3. $\square$

Определим слабое свойство переноса для бисимуляции системных ресурсов во вложенных сетях Петри.

В слабом свойстве переноса для обыкновенных сетей рассматриваемые разметки ограничены конечным мультимножеством $\bullet t$ входных позиций данного перехода. При определении аналога слабого свойства переноса нам потребуется аналогичное ограничение на рассматриваемые разметки вложенной сети, позволяющее для каждого перехода определить некий конечный ресурс, достаточный для его срабатывания.

Рассмотрим частичный порядок $\preceq$ на множестве разметок вложенной сети Петри, определенный в [17].

Определение 5.37. Пусть $NPN = (\mathfrak{A}, \{EN_1, \dots, EN_k\}, SN, Lab, \Lambda)$ — помеченная вложенная сеть Петри, $M_1, M_2 \in \mathfrak{M}(NPN)$. Скажем, что $M_1 \preceq M_2$, если для любой позиции $p \in P$ системной сети SN существует вложение $j_p : M_1(p) \rightarrow M_2(p)$, такое, что для любой фишки $\alpha \in M_1(p)$ выполняется одно из двух условий:

- $\alpha = j_p(\alpha)$

или

- $\alpha = (EN_i, m)$ и $j_p(\alpha) = (EN_i, m')$, где для любой позиции $q \in P_i$ элементной сети EN_i выполняется $m(q) \subseteq m'(q)$.

Другими словами, $M_1 \preceq M_2$ если для любой позиции p системной сети мультимножество фишек $M_2(p)$ может быть преобразовано в мультимножество $M_1(p)$ удалением некоторых фишек и/или заменой части сетевых фишек на фишки с меньшей разметкой (но той же сетевой структуры).

Обозначение 5.3. Пусть $NPN = (\mathfrak{A}, \{EN_1, \dots, EN_k\}, SN, Lab, \Lambda)$ — помеченная вложенная сеть Петри, $t \in \mathcal{T}(NPN)$ — переход сети NPN . Через ${}^\circ t$ обозначим множество всех минимальных относительно $\preceq$ разметок, при которых t может сработать.

В [17] доказано, что для любого перехода t множество ${}^{\circ}t$ конечно и может быть эффективно построено.

Обозначение 5.4. Пусть $NPN = (\mathfrak{A}, \{EN_1, \dots, EN_k\}, SN, Lab, \Lambda)$ — помеченная вложенная сеть Петри, $t \in \mathcal{T}(NPN)$ — переход в сети NPN , $B \subseteq \mathfrak{M}(NPN) \times \mathfrak{M}(NPN)$ — некоторое симметричное рефлексивное отношение на множестве системных ресурсов. Через ${}^{\circ}t(B)$ обозначим множество минимальных относительно $\preceq$ разметок из множества

$$\mathfrak{M}_B(NPN) = \{M \in \mathfrak{M}(NPN) \mid (M, M') \in B^{AT}\}.$$

Другими словами, ${}^{\circ}t(B)$ — это множество минимальных относительно $\preceq$ разметок среди задействованных в B^{AT} .

Множество ${}^{\circ}t(B)$ конечно для любого B , так как конечно число сетевых структур элементарных фишек вложенной сети и мы можем выбрать среди фишек одной структуры конечное число минимальных с возбужденными синхронизирующими переходами. В том случае, когда B задается конечным базисом и этот базис известен, множество ${}^{\circ}t(B)$ может быть эффективно построено.

Определение 5.38. Пусть $NPN = (\mathfrak{A}, \{EN_1, \dots, EN_k\}, SN, Lab, \Lambda)$ — помеченная вложенная сеть Петри. Скажем, что отношение $B \subseteq \mathfrak{M}(NPN) \times \mathfrak{M}(NPN)$ обладает *слабым свойством переноса*, если для всех $(r, s) \in B$, $t \in \mathcal{T}(NPN)$, $m \in {}^{\circ}t(B)$, где $m \cap r \neq \emptyset$, найдется имитирующий переход $u \in \mathcal{T}(NPN)$, такой, что $l(t) = l(u)$ и, обозначив $M_1 = m \cup r$ и $M_2 = m - r + s$, мы имеем $M_1 \xrightarrow{t} M_1'$ и $M_2 \xrightarrow{u} M_2'$, где $(M_1', M_2') \in B^{AT}$.

Так же, как и в случае обыкновенных сетей Петри, слабое свойство переноса гарантирует выполнение полного свойства переноса.

Теорема 5.5. Пусть $NPN = (\mathfrak{A}, \{EN_1, \dots, EN_k\}, SN, Lab, \Lambda)$ — помеченная вложенная сеть Петри. Симметричное рефлексив-

ное отношение $B \subseteq \mathfrak{M}(NPN) \times \mathfrak{M}(NPN)$ обладает слабым свойством переноса тогда и только тогда, когда B — бисимуляция системных ресурсов.

Доказательство. Докажем по схеме доказательства теоремы 3.4, используя тот факт, что системная разметка (системный ресурс) есть мультимножество пар вида (p, C) , где $p \in P$ — позиция системной сети, $C \in \mathfrak{A} \cup \mathfrak{D}$ — фишка (атомарная или сетевая):

($\Rightarrow$) Предположим противное: свойство переноса для B^{AT} не выполняется, то есть существуют $(M_1, M_2) \in B^{AT}$, $t \in \mathcal{T}(NPN)$, где $M_1 \xrightarrow{t} M'_1$, такие, что не найдется моделирующего перехода $u \in \mathcal{T}(NPN)$ с той же меткой $l(t) = l(u)$, для которого бы выполнялось $M_2 \xrightarrow{u} M'_2$ и $(M'_1, M'_2) \in B^{AT}$.

Рассмотрим пару разметок $(M_1, M_2) \in B^{AT}$. Согласно лемме 2.3 она получена транзитивным замыканием нескольких пар из B^A — аддитивного замыкания B :

$$(H_1, H_2), (H_2, H_3), \dots, (H_{k-1}, H_k) \in B^A, \text{ где } H_1 = M_1, H_k = M_2.$$

Рассмотрим пару (H_1, H_2) .

$$(H_1, H_2) = (r_1 + r_2 + \dots + r_l, s_1 + s_2 + \dots + s_l), \text{ где } (r_i, s_i) \in B$$

$H_1 = \bullet t \cup r_1 + F_1$. В силу слабого свойства переноса для пары (r_1, s_1) существует переход $v \in \mathcal{T}(NPN)$, такой, что $l(t) = l(v)$, $\bullet t \cup r_1 \xrightarrow{t} G_1$ и $\bullet t - r_1 + s_1 \xrightarrow{v} G_2$, где $(G_1, G_2) \in B^{AT}$.

Так как $\bullet t \cup r_1 \subseteq H_1$, к пред- и постусловиям срабатываний мы можем добавить ресурс $F = H_1 - \bullet t \cup r_1$:

$$\begin{aligned} \bullet t \cup r_1 + F &\xrightarrow{t} G_1 + F \\ \bullet t - r_1 + s_1 + F &\xrightarrow{v} G_2 + F \end{aligned}$$

В силу 1-рефлексивности B и аддитивной замкнутости B^{AT} полученная пара разметок также разложима по B : $(G_1 + F, G_2 + F) \in B^{AT}$.

Получаем новая разметка $H'_1 = \bullet t - r_1 + s_1 + F = H_1 - r_1 + s_1$. Заметим, что она содержит $r_2 + \dots + r_l$. То есть можно еще раз применить использованный выше прием для замены ресурса r_2 на бисимулярный ресурс s_2 .

Применив его еще в общей сложности $l - 1$ раз и используя свойство транзитивной замкнутости B^{AT} , в итоге получим переход $w \in \mathcal{T}(NPN)$, имитирующий t при разметке H_2 .

Перейдем к следующей паре (H_2, H_3) , повторим все для перехода w . И т.д. вплоть до последней пары (H_{k-1}, H_k) . В итоге мы получим переход $u \in \mathcal{T}(NPN)$, имитирующий t при разметке $H_k = M_2$.

($\Leftarrow$) Очевидно, так как слабое свойство переноса — это свойство переноса, ограниченное конечным числом пар. $\square$

Приведем аналог алгоритма аппроксимации максимальной бисимуляции для случая системных ресурсов вложенной сети Петри. Заметим, что системный ресурс может быть ограничен по двум параметрам — количеству фишек в разметке системной сети и количеству фишек в разметках элементарных сетей.

Пусть $NPN = (\mathfrak{A}, \{EN_1, \dots, EN_k\}, SN, Lab, \Lambda)$ — помеченная вложенная сеть Петри, $q, d \in Nat$ — некоторые целочисленные параметры. Обозначим как $\mathbf{M}_{q,d}(NPN)$ множество системных ресурсов, определяемое следующим образом:

$$\mathbf{M}_{q,d}(NPN) = \{r = \{(p, o), \dots\} \in \mathfrak{M}(NPN) \mid$$

$$p \in P \wedge o \in \mathfrak{A} \cup \mathfrak{S} \wedge |r| \leq q \wedge (o = (EN_i, M) \Rightarrow |M| \leq d)\}$$

Другими словами, $\mathbf{M}_{q,d}(NPN)$ — множество системных ресурсов, содержащих не более q фишек в системной сети, причем у всех сетевых фишек разметка ограничена сверху параметром d .

Через $B(NPN, q, d)$ обозначим наибольшую (по вложению) бисимуляцию системных ресурсов над множеством $\mathbf{M}_{q,d}(NPN)$ (легко показать, что она существует и является объединением всех бисимуляций системных ресурсов над $\mathbf{M}_{q,d}(NPN)$).

Алгоритм 5.3. (построения аппроксимации максимальной бисимуляции системных ресурсов вложенной сети Петри)

ввод: Помеченная вложенная сеть Петри NPN , параметры $q, d \in \mathbb{N}$.

вывод: Отношение $B(NPN, q, d)$.

шаг 1: Пусть $C = \{(\emptyset, \emptyset)\}$ — пустое множество пар (рассматриваемое как отношение над $\mathbf{M}_{q,d}(NPN)$).

шаг 2: Построим $B = (\mathbf{M}_{q,d}(NPN) \times \mathbf{M}_{q,d}(NPN)) \setminus C$. В силу конечности $\mathbf{M}_{q,d}(NPN)$ множество пар B также конечно.

шаг 3: Построим B_s — основной базис отношения B (по алгоритму 2.2).

шаг 4: Проверим, обладает ли B_s слабым свойством переноса. Для этого переберем элементы из B_s^{nr} — подмножества нереклексивных элементов.

- Если все пары обладают слабым свойством переноса, то возвращаем B — искомую бисимуляцию.
- В противном случае существуют $(r, s) \in B_s^{nr}$, $t \in T(NPN)$, $m \in {}^\circ t$, где $r \cap m \neq \emptyset$, такие, что срабатывание $M_1 \xrightarrow{t} M_1'$, где $M_1 = m \cup r$, не может быть имитировано срабатыванием какого-либо перехода с той же меткой и предусловием $M_2 = m - r + s$. В этом случае добавляем (r, s) и (s, r) в C и возвращаемся на шаг 2.

Поскольку для любой разметки множество готовых к срабатыванию переходов вложенной сети конечно, а также конечно и перебираемое множество пар системных ресурсов $\mathbf{M}_{q,d}(NPN) \times \mathbf{M}_{q,d}(NPN)$, алгоритм завершит работу за конечное число шагов.

Оценим трудоемкость алгоритма.

Алгоритм 5.3 отличается от алгоритма 3.2 множествами перебора. Во-первых, мы рассматриваем системный ресурс не как мультимножество позиций, а как мультимножество над множеством пар вида (p, o) , где $p \in P$, $o \in \mathfrak{A} \cup \mathfrak{D}$, причем если $o =$

(EN_i, M) , то выполняется $|M| \leq p$. Мощность этого множества

$$\mathbf{R} = |P|(|\mathfrak{D}| + \sum_{i=1}^k p^{|P_i|}).$$

Другое отличие от алгоритма 3.2 состоит в том, что на втором шаге происходит перебор множества переходов вложенной сети. Его мощность равна $|\mathcal{T}(NPN)|$.

Заменяя в оценке трудоемкости алгоритма 3.2 значение $|P|$ на $\mathbf{R}$ и $|T|$ на $|\mathcal{T}(NPN)|$, получим следующую оценку трудоемкости для алгоритма построения аппроксимации:

$$O(\max\{\mathbf{R} |\mathbf{M}_{q,d}(NPN)|^9, |\mathcal{T}(NPN)|^2 \mathbf{R} |\mathbf{M}_{q,d}(NPN)|^7\}).$$

Таким образом, в случае вложенных сетей алгоритм построения аппроксимации имеет еще большую трудоемкость, чем даже в случае сетей с невидимыми переходами. Экспоненциальность возникает в трех местах: оценке мощности рассматриваемого множества “позиций” (множитель $\mathbf{R}$), оценке мощности множества ресурсов $(|\mathbf{M}_{q,d}(NPN)|)$ и оценке мощности множества переходов вложенной сети $(|\mathcal{T}(NPN)|)$.

5.3.4 Системно-автономные ресурсы

Рассмотрим последний вид ресурса во вложенных сетях Петри — локальное состояние системной позиции сети без учета разметок сетевых фишек.

Заметим, что в общем случае системную сеть нельзя анализировать как обыкновенную “плоскую” сеть Петри, рассматривая сетевые фишки как простые “черные” или даже “цветные” фишки. Пример на рисунке 5.11 показывает, какие проблемы при этом возникают.

На рисунке 5.11 изображена системная сеть некоторой вложенной сети. Вертикальной синхронизации в данной сети нет. Внутренняя структура сетевых фишек для нас не важна, мы знаем

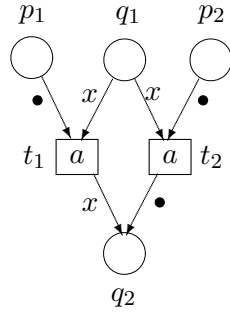


Рис. 5.11: Пример системной сети SN . Синхронизации нет, x — переменная. Выполняется $p_1 \approx_{SN} p_2$ (если рассматривать SN как обыкновенную сеть Петри), однако $(p_1, \bullet) \not\approx (p_2, \bullet)$, поскольку $(p_1, \bullet) + (q_1, C) \not\approx (p_2, \bullet) + (q_1, C)$, где C — сетевая фишка с внутренним поведением.

только, что они могут автономно функционировать. Дугам приписаны два выражения — переменная x и константа $\bullet$.

Если рассматривать эту сеть как обыкновенную сеть Петри, то ресурсы p_1 и p_2 (позиции) подобны. Однако эти ресурсы для поведения вложенной сети существенно различны, так как при срабатывании перехода t_1 в позицию q_2 переносится та же фишка (возможно, сетевая), которая забирается из позиции q_1 ; в то время как при срабатывании перехода t_2 эта фишка исчезает и возникает фишка $\bullet$, которая уже не может автономно функционировать (и породить новое поведение системы в целом).

Следовательно, при определении подобия ресурсов необходимо в какой-то мере учитывать различия между фишками либо различия между пометками на дугах.

Первый подход сводится к анализу состояния системы в целом и рассмотрен в предыдущей главе (подобие системных ресурсов).

Рассмотрим второй подход.

Определение 5.39. *Системно-автономным ресурсом* вложен-

ной сети $NPN = (\mathfrak{A}, \{EN_1, \dots, EN_k\}, SN, Lab, \Lambda)$ называется мультимножеством над множеством позиций системной сети SN .

Так как внутренняя структура сетевых фишек не учитывается, при замене фишек в системной разметке мы можем заменять только одинаковые фишки:

Определение 5.40. Системно-автономные ресурсы $r = \{p_1, p_2, \dots, p_k\}$ и $s = \{q_1, q_2, \dots, q_l\}$ называются *подобными* (обозначается $r \approx_o s$), если для любой разметки $M \subseteq \mathfrak{M}(NPN)$, такой, что в ней существует подмножество

$$M_r^a = \{(p_1, a), (p_2, a), \dots, (p_k, a)\} \subseteq M, \text{ где } a \in \mathfrak{A} \cup \mathfrak{D},$$

$$\text{выполняется } M \sim M - M_r^a + M_s^a, \text{ где} \\ M_s^a = \{(q_1, a), (q_2, a), \dots, (q_l, a)\}.$$

Подобные системно-автономные ресурсы — это мультимножества позиций системной сети, которые, во-первых, подобны как ресурсы при рассмотрении системной сети как обыкновенной сети Петри, и, во-вторых, неразличимы с точки зрения находящихся в них сетевых фишек. При этом сетевые фишки различают позиции системной сети только по тем ограничениям, которые накладывает на их поведение вертикальная синхронизация.

Пример 5.3. Рассмотрим системную сеть, изображенную на рисунке 5.12. Поскольку в данной сети нет никакой синхронизации, с точки зрения переходов t_2 и t_3 внутренняя структура фишек в позициях p_1 и p_2 несущественна. Для этих переходов важно только наличие/отсутствие *какой-нибудь* фишки во входной позиции. И обратно: сетевая фишка не может отличить позицию p_1 от позиции p_2 , так как вертикальная синхронизация отсутствует.

Следовательно, мы можем переносить сетевую фишку из позиции p_1 в позицию p_2 и наоборот без последствий для наблюдаемого поведения сети (в смысле бисимуляции). Таким образом, выполняется $p_1 \approx_o p_2$.

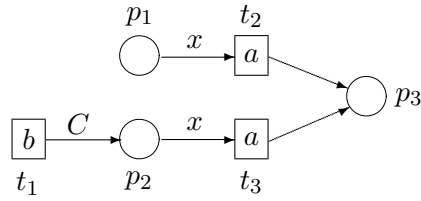


Рис. 5.12: Пример подобия системно-автономных ресурсов. Изображена системная сеть SN некоторой вложенной сети Петри. Синхронизация отсутствует, $C \in \mathfrak{D}$ — константа (сетевая фишка), x — переменная. Выполняется $p_1 \approx_o p_2$.

Поскольку мы не рассматриваем внутреннее поведение сетевых фишек, горизонтальная синхронизация не может быть отслежена при помощи подобия системно-автономных ресурсов:

Утверждение 5.10. *Для вложенных сетей с действующей горизонтальной синхронизацией (при $|Lab_h| > 0$) максимальным (по вложению) подобием системно-автономных ресурсов является отношение $Id(P)$.*

Доказательство. Предположим противное: подобны две различные позиции системной сети — p и q . Рассмотрим разметку M , содержащую две горизонтально синхронизированные сетевые фишки в позиции p и ни одной фишки в позиции q . Если мы в соответствие со свойством подобия p и q переместим одну из этих фишек в q , то синхронизирующий переход не сможет сработать. $\square$

5.3.5 Рекурсивные вложенные сети Петри

Существует расширение класса вложенных сетей Петри — многоуровневые вложенные сети Петри, в которых в качестве фишек в свою очередь могут выступать вложенные сети. Необходимо разделять два случая — собственно многоуровневые сети (в которых число слоев ограничено структурой сети) и рекурсивные вложенные сети с потенциально бесконечным числом слоев.

Рекурсивные вложенные сети [16, 56] удобны для моделирования систем, имеющих рекурсивную природу. Этот формализм реализует одновременно и принцип параллелизма и принцип рекурсии, что делает его весьма привлекательным для моделирования вычислительных систем с динамическим распараллеливанием, сложных иерархических структур и т.п.

В этой главе исследуются свойства языков рекурсивных вложенных сетей Петри. Доказывается, что класс контекстно-свободных языков строго вкладывается в класс тупиковых языков рекурсивных сетей.

Определение 5.41. Тупиковым языком помеченной рекурсивной сети назовем множество слов, порождаемых всеми возможными ее исполнениями, приводящими к тупиковым разметкам (то есть таким разметкам, при которых никакое срабатывание невозможно).

Напомним определение контекстно-свободных языков.

Определение 5.42. *Порождающей формальной грамматикой* называется четверка $G = (V, W, S, R)$, где

V — алфавит терминальных (основных) символов;

W — алфавит нетерминальных (вспомогательных) символов, причем $V \cap W = \emptyset$;

R — конечное множество правил вывода вида $\xi \rightarrow \eta$, где ξ и η — цепочки (слова) в алфавите $V \cup W$;

S — начальный символ (аксиома грамматики).

Языком $L(G)$, порождаемым грамматикой G , называется множество всех цепочек в терминальном алфавите V , выводимых из S в соответствие с правилами R .

Грамматика называется *контекстно-свободной* (КС-грамматикой), если все ее правила вывода имеют вид $A \rightarrow \alpha$, где $A \in W$, $\alpha \in (V \cup W)^*$.

Определим несколько преобразований над простейшими рекурсивными сетями.

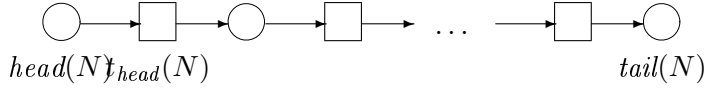


Рис. 5.13: Сеть-цепочка

Цепочкой назовем сеть $N = (P, T, F)$, имеющую вид, представленный на рисунке 5.13. В цепочке выделяются головная позиция $head(N)$, хвостовая позиция $tail(N)$ и головной переход $t_{head}(N)$.

Определим следующие две операции над цепочками:

- *Конкатенация.*

Пусть $N_1 = (P_1, T_1, F_1)$ и $N_2 = (P_2, T_2, F_2)$ — цепочки, тогда

$$N_1 \cdot N_2 =_{def} (P_1 \cup P_2 \setminus \{head(N_2)\}, T_1 \cup T_2,$$

$$F_1 \cup (F_2 \setminus \{(head(N_2), t_{head}(N_2))\}) \cup \{(tail(N_1), t_{head}(N_2))\}).$$

- *Склеивание головных позиций.*

Пусть $N_1 = (P_1, T_1, F_1), \dots, N_k = (P_k, T_k, F_k)$ — цепочки, тогда

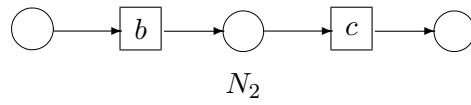
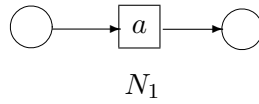
$$Link_{i=1}^k(N_i) =_{def} \left(\bigcup_{i=1}^k (P_i \setminus \{head(N_i)\}) \cup \{p\}, \bigcup_{i=1}^k T_i,$$

$$\bigcup_{i=1}^k (F_i \setminus \{(head(N_i), t_{head}(N_i))\}) \cup \{(p, t_{head}(N_i))\} \right).$$

Результат применения данных операций к цепочкам показан на рисунке 5.14. Очевидно, что в результате конкатенации опять получается цепочка, тогда как склеивание n цепочек производит сеть более сложной структуры, у которой одна головная и n хвостовых позиций.

Введем также три шаблона элементарных рекурсивных сетей, которые будут играть роль атомарных структур, из которых будет строиться итоговая сеть (рисунок 5.15). На рисунке метки из

Исходные сети:



Результат применения операций над сетями:

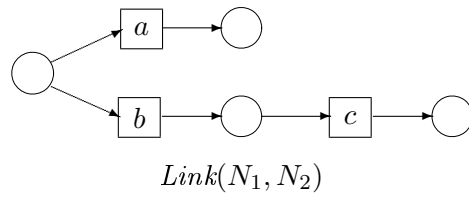
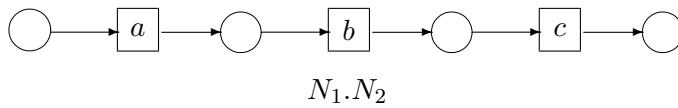


Рис. 5.14: Операции над сетями

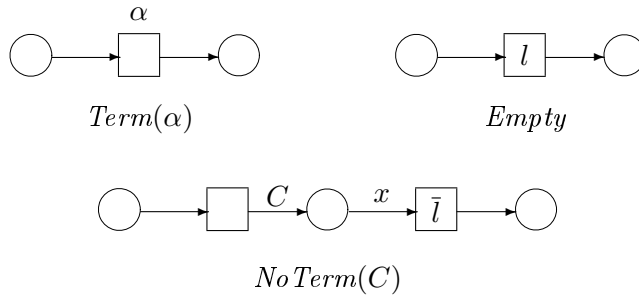


Рис. 5.15: Шаблоны

Lab поставлены внутри переходов, метки из *Act* — над переходами. Эти шаблоны имеют структуру цепочки, поэтому к ним применимы операции конкатенации и склеивания.

Теорема 5.6. *Для любой КС-грамматики G существует помеченная рекурсивная вложенная сеть, порождающая язык $L(G)$ в качестве своего тупикового языка.*

Доказательство. В качестве доказательства приведем алгоритм построения помеченной рекурсивной вложенной сети по произвольной КС-грамматике $G = (V, W, S, R)$:

1. Для каждого правила вывода R_i построим цепочку $N(R_i)$:
 - шаг 1'. Пусть $R_i : T \rightarrow \xi$. Введем переменную $\eta := \xi$. $N(R_i) := (\{p\}, \emptyset, \emptyset)$.
 - шаг 2'. Если длина строки η не равна 0, то возможны два случая (два вида символов в начале строки):
 - нетерминальный: $\eta = T'\eta'$, где $T' \in W$, $\eta' \in (V \cup W)^*$. Тогда $N(R_i) := \text{Term}(N_{T'}) \cdot N(R_i)$, $\eta := \eta'$, и возвращаемся на шаг 2'.
 - терминальный: $\eta = a\eta'$, где $a \in V$, $\eta' \in (V \cup W)^*$. Тогда $N(R_i) := \text{NoTerm}(a) \cdot N(R_i)$, $\eta := \eta'$, возвращаемся на шаг 2'.

шаг 3'. $N(R_i) := N(R_i).Empty$.

2. Обозначим $I_T = \{i \mid R_i \in R \& R_i = T \rightarrow \xi\}$. Построим для каждого $T \in W$ сеть $N_T = Link_{i \in I_T}(N(R_i))$.
3. Искомая рекурсивная сеть состоит из системной сети N_S и множества элементарных сетей $\{N_T\}_{T \in W}$. Каждой константе на дугах сопоставлена сеть с соответствующим именем и одной черной фишкой в головной позиции в качестве разметки. Начальная разметка состоит из одной черной фишки в головной позиции системной сети N_S . Имеется одна метка l для вертикальной синхронизации. Множество Act помеченных сетей совпадает с множеством V терминальных символов грамматики.

Для конечных грамматик данный алгоритм конечен, так как, во-первых, R конечно, поэтому конечно число шагов 1 и 2, и, во-вторых, на каждом шаге вида 2' происходит уменьшение рассматриваемой строки, а длина строк в R также конечна.

Полученная сеть в любой момент своего функционирования содержит конечное число слоев (уровней вложенности) и ровно по одной сети в каждом слое. Недетерминированность возможна только в самом внутреннем слое, если соответствующий объект получен склеиванием, и в нем не сработал еще ни один переход. Это естественным образом соответствует случаю недетерминированного выбора из нескольких правил вывода в случае грамматик.

Вообще, сработать всегда может только переход во внутреннем слое (возможно, синхронно с переходом в соседнем слое, причем при этом самый внутренний слой просто исчезнет), поскольку, в соответствии со структурой сети $NoTerm$, содержащая объект сеть всегда дожидается его исчезновения, прежде чем сможет сработать сама. Правильная вложенность объектов друг в друга (в соответствии с правилами вывода) позволяет в этом случае гарантировать правильность порождения языка $L(G)$. $\square$

Известно [47], что класс контекстно-свободных языков не сравним с классом языков, порождаемых обыкновенными сетями Петри. Поэтому из приведенной теоремы следует, что рекурсивные вложенные сети обладают гораздо большей выразительной мощностью. В частности, это является косвенным признаком того, что корректные с точки зрения бисимулярности отношения эквивалентности на множестве ресурсов рекурсивных сетей не могут быть построены так же эффективно, как и в случае обыкновенных сетей Петри.

Глава 6

Применение эквивалентностей ресурсов

6.1 Редукция (оптимизация) модели

Подобие ресурсов может использоваться для редукции сети Петри, то есть сокращения ее размера при сохранении наблюдаемого поведения (в смысле бисимулярности разметок). В частности, такая редукция представляет собой достаточно мощное средство оптимизации сложных (в том числе распределенных) процессов и систем [27]. Спектр сходных задач весьма широк: от реинжиниринга бизнес-процессов до оптимизирующей компиляции параллельных программ.

Для определения эквивалентности поведения сетей различной структуры нам потребуется понятие бисимуляции разметок *между* двумя сетями Петри:

Определение 6.1. Пусть $N_1 = (P_1, T_1, F_1, l_1)$ и $N_2 = (P_2, T_2, F_2, l_2)$ — помеченные сети Петри. Отношение $R \subseteq \mathcal{M}(P_1) \times \mathcal{M}(P_2)$ обладает *свойством переноса*, если для любой пары разметок $(M_1, M_2) \in R$ и для любого перехода $t \in T_1$, такого, что в сети

N_1 существует срабатывание $M_1 \xrightarrow{t} M'_1$, найдется имитирующий переход $u \in T_2$, такой, что $l_1(t) = l_2(u)$ и в сети N_2 существует срабатывание $M_2 \xrightarrow{u} M'_2$, где $(M'_1, M'_2) \in R$.

Если отношения R и R^{-1} обладают свойством переноса, то R называется *бисимуляцией разметок между* N_1 и N_2 .

Маркированные сети Петри (N_1, M_1) и (N_2, M_2) , такие, что существует отношение бисимуляции R , для которого $(M_1, M_2) \in R$, называются *бисимулярными* (обозначается $(N_1, M_1) \sim (N_2, M_2)$).

Известно, что для любых сетей Петри N_1 и N_2 существует максимальная бисимуляция разметок между ними, обозначаемая как $\sim_{[N_1, N_2]}$.

Рассмотрим некоторые способы редукции сетей Петри на основе подобию ресурсов.

По определению подобные ресурсы при всех возможных разметках сети полностью взаимозаменяемы, то есть фишки одного можно заменить на фишки другого. Уже это позволяет в некотором смысле “редуцировать” сеть, заменяя в её начальной разметке больший (по количеству фишек) ресурс на меньший.

Однако наибольший интерес представляет изменение не начальной разметки сети, а её графовой структуры, то есть сокращение количества позиций, переходов и дуг. Информацию для такой редукции также можно получить из подобию ресурсов.

Заметим, что всякому подобию ресурсов (мультимножеств позиций) можно сопоставить “подобие” выходных дуг переходов. Другими словами, если срабатывание перехода t помещает в его выходные позиции некоторый ресурс r , у которого есть подобный ресурс s , то мы можем заменить у перехода t все выходные дуги, соответствующие r , на выходные дуги, соответствующие s , и наблюдаемое поведение полученной сети ничем не будет отличаться от поведения исходной (в смысле бисимулярности).

Утверждение 6.1. (замена “подобных” выходных дуг)

Пусть $N = (P, T, F, l)$ — помеченная сеть Петри; $r, s \in \mathcal{M}(P)$

— ресурсы сети N , такие, что $r \approx s$; $t \in T$ — переход, такой, что $r \subseteq \bullet t$. Пусть $N' = (P, T, F', l)$ — помеченная сеть Петри, такая, что

$$\forall p \in P \quad F'(t, p) = F(t, p) - r(p) + s(p).$$

Тогда для любой $M \in \mathcal{M}(P)$ выполняется $(N, M) \sim (N', M)$.

Другими словами, для сети N' , полученной из исходной сети N заменой “подобных” дуг, выполняется $Id(\mathcal{M}(P)) \subseteq (\sim_{[N, N']})$.

Доказательство. Пусть R — отношение максимальной бисимуляции разметок сети N (отношение $\sim$).

Поскольку множества позиций (и, следовательно, разметок) сетей N и N' совпадают, мы можем рассматривать R и в качестве отношения между разметками сетей N и N' . Поскольку по свойству бисимуляции (утверждение 1.3) выполняется $Id(\mathcal{M}(P)) \in R$, для доказательства теоремы достаточно доказать, что отношение R — бисимуляция между сетями N и N' .

1) Покажем, что R обладает свойством переноса.

Предположим противное: существуют $(M_1, M_2) \in R$, переход $u \in T$, такие, что в сети N возможно срабатывание $M_1 \xrightarrow{u} M'_1$, причем в сети N' для него не существует имитирующего (в смысле R) срабатывания при разметке M_2 .

С другой стороны, поскольку R — бисимуляция разметок в сети N , существует переход $v \in T$, такой, что в сети N возможно срабатывание $M_2 \xrightarrow{v} M'_2$, где $(M'_1, M'_2) \in R$. Рассмотрим в сети N' в качестве имитирующего перехода тот же переход v . Возможны два случая:

- $v \neq t$. Тогда в результате срабатывания v в сети N' получается в точности разметка M'_2 : $M_2 \xrightarrow{v} M'_2$.
- $v = t$. Тогда в результате срабатывания v в сети N' получается разметка $M''_2 = M'_2 - r + s$. По определению подобных

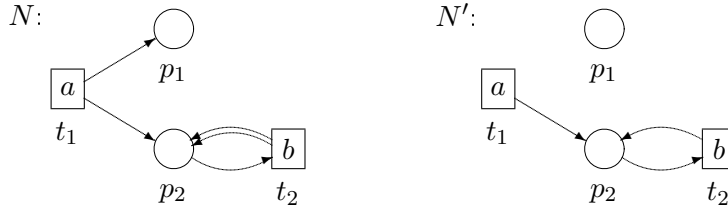


Рис. 6.1: Редукция заменой “подобных” выходных дуг. Для сети N выполняется $p_1 \approx \emptyset$, $2p_2 \approx p_2$. Соответствующие этим ресурсам дуги заменены у переходов t_1 и t_2 .

ресурсов имеем $(M'_2 - r + s, M'_2) \in R$, то есть $(M''_2, M'_2) \in R$. Тогда из $(M'_1, M'_2) \in R$ и транзитивности бисимуляции разметок имеем $(M'_1, M''_2) \in R$.

2) Покажем, что R^{-1} обладает свойством переноса.

Заметим, что постусловие измененного перехода t (в сети N') подобно постусловию исходного перехода (в сети N). Следовательно, R — бисимуляция разметок также и для сети N' .

Далее доказательство выполнения свойства переноса аналогично доказательству для R , используя тот факт, что R — бисимуляция разметок в сети N' .

Итак, отношение R является бисимуляцией разметок между N и N' , что и требовалось доказать. $\square$

Таким образом, при одинаковых начальных разметках поведение сетей N и N' неразличимо (с точки зрения бисимулярности), и мы можем использовать для моделирования исходной системы сеть с меньшим числом дуг.

Замена подобных дуг во многих случаях позволяет добиваться существенного упрощения структуры сети.

Рассмотрим редукцию сети Петри N , изображенной на рисунке 6.1. Сеть N' получена из исходной сети N при помощи следующих замен “подобных” дуг:

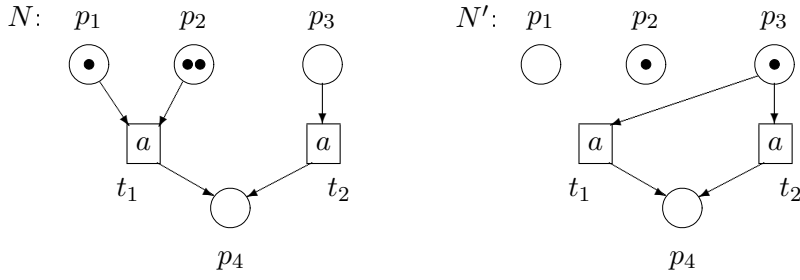


Рис. 6.2: Пример редукции заменой входных дуг и изменением начальной разметки. Для сети N выполняется $p_1 + p_2 \approx p_3$.

1. $p_1 \approx \emptyset$. Дуга (t_1, p_1) заменена на $(t_1, \emptyset)$ (то есть удалена).
2. $2p_2 \approx p_3$. Две дуги (t_2, p_2) заменены на одну такую же.

Замена “подобных” выходных дуг может производиться даже у немаркированных сетей, поскольку одинаковые разметки исходной и редуцированной сетей бисимулярны.

Применить аналогичный прием для входных дуг переходов не удастся без существенных модификаций.

Во-первых, заметим, что в случае входных дуг необходимо учитывать начальную разметку. Рассмотрим сеть, изображенную на рисунке 6.2. В данном случае дуги (p_1, t_1) и (p_2, t_1) заменены на дугу (p_3, t_1) , в начальной разметке ресурс $p_1 + p_2$ заменен на p_3 . Если бы мы не изменили начальную разметку, то в сети N' ни один переход не смог бы сработать.

Однако и замены начальной разметки не всегда достаточно для корректной редукции. Рассмотрим сети, изображенные на рисунке 6.3. Сеть N' получена из сети N тем же преобразованием, что и в предыдущем случае. Однако маркированные сети $(N, p_1 + 2p_2)$ и $(N', p_2 + p_3)$ не бисимулярны. Действительно, в сети $(N, p_1 + 2p_2)$ возможна последовательность срабатываний $t_3.t_1.t_1$, помеченная словом “baa”, тогда как в сети $(N', p_2 + p_3)$ переход, помеченный символом “a”, может сработать только один раз.

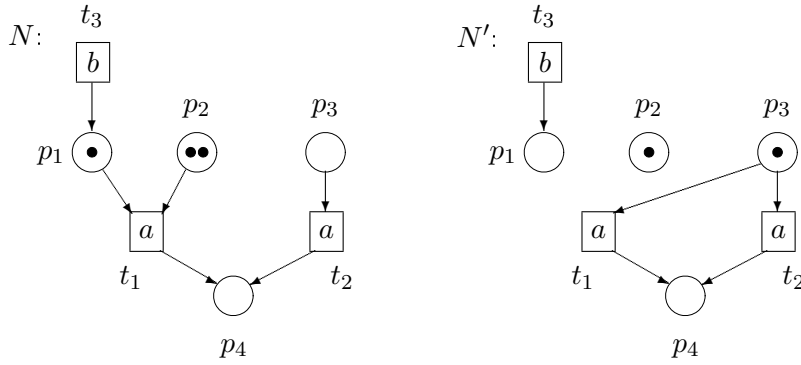


Рис. 6.3: Пример некорректной редукции. Для сети N выполняется $p_1 + p_2 \approx p_3$, однако $(N, p_1 + 2p_2) \not\approx (N', p_2 + p_3)$

Следовательно, требуются дополнительные ограничения.

Рассмотрим следующее условие. Пусть заменяемый ресурс “неделим” с точки зрения начальной разметки сети и с точки зрения предусловий и постусловий ВСЕХ переходов. Другими словами, этот ресурс входит в начальную разметку “целое” число раз и всякий переход при срабатывании забирает из входных позиций и помещает в выходные позиции “целое” число таких ресурсов.

Это условие достаточно сильное, однако оно позволяет производить полную замену одного ресурса на другой (подобный):

Утверждение 6.2. (замена подобных ресурсов)

Пусть (N, M_0) — помеченная маркированная сеть Петри, где $N = (P, T, F, l)$ — помеченная сеть Петри. Пусть $r, s \in \mathcal{M}(P)$ — ресурсы сети N , такие, что

- $r \approx s$;
- $M_0 = nr + \overline{M}_0$, где $n \in \text{Nat}$, $\overline{M}_0 \cap r = \emptyset$;
- для любого перехода $t \in T$ выполняется
 $\bullet t = i_t r + g_t$, где $i_t \in \text{Nat}$, $g_t \cap r = \emptyset$,

$$t^\bullet = j_t r + h_t, \text{ где } j_t \in \text{Nat}, h_t \cap r = \emptyset.$$

Пусть $M'_0 = ns + \overline{M}_0$, $N' = (P, T, F', l)$ — помеченная сеть Петри, такая, что $\forall t \in T, \forall p \in P$

$$F'(p, t) = F(p, t) - i_t r(p) + i_t s(p),$$

$$F'(t, p) = F(t, p) - j_t r(p) + j_t s(p).$$

Тогда $(N, M_0) \sim (N', M'_0)$.

Доказательство. Построим множество $\text{Reach}(M_0)$ разметок, достижимых в сети N от начальной разметки M_0 :

$$\text{Reach}(M_0) = \{M \in \mathcal{M}(P) \mid M_0 \rightarrow M\}$$

Поскольку все переходы сети N забирают из входной разметки и помещают в выходную разметку “целые” ресурсы r , любой элемент множества $\text{Reach}(M_0)$ обладает тем же свойством, что и начальная разметка M_0 — он содержит “целое” число ресурсов r :

$$\forall M \in \text{Reach}(M_0) \quad M = i_M r + m_M, \text{ где } i_M \in \text{Nat}, m_M \cap r = \emptyset.$$

Пусть R — отношение максимальной бисимуляции разметок сети N (отношение $\sim$). Построим отношение E по следующему правилу:

$$(M, M') \in R \wedge M \in \text{Reach}(M_0) \quad \Rightarrow \quad (M, M' - i_M r + i_M s) \in E.$$

Так как ресурсы r и s подобны, имеем $M \sim M' - i_M r + i_M s$, то есть $E \subseteq R$.

По определению подобных ресурсов имеем $M_0 \sim M'_0$, следовательно, $(M_0, M'_0) \in E$.

Поскольку множества позиций (и, следовательно, разметок) сетей N и N' совпадают, мы можем рассматривать E в качестве отношения между разметками сетей N и N' . Так как $(M_0, M'_0) \in E$, для доказательства теоремы достаточно доказать, что E — бисимуляция между N и N' .

1) Покажем, что E обладает свойством переноса.

Предположим противное: существуют $(M_1, M_2) \in E$, переход $u \in T$, такие, что в сети N возможно срабатывание $M_1 \xrightarrow{u} M'_1$, причем в сети N' для него не существует имитирующего (в смысле E) срабатывания при разметке M_2 .

Рассмотрим в сети N' в качестве имитирующего перехода переход v , полученный из u заменой дуг. Он может сработать при разметке M_2 , так как она получена из разметки M_1 заменой всех ресурсов r на s , а входные дуги перехода v получены из входных дуг перехода u заменой всех (т.е. i_u) ресурсов r на ресурсы s .

Пусть в сети N' срабатывание перехода v приводит к разметке $L : M_2 \xrightarrow{v} L$. Так как постусловие перехода v отличается от постусловия u также на целое число ресурсов (на j_u), имеем $L = M'_2 - j_u r + j_u s$, то есть $(M'_2, L) \in E$.

2) Покажем, что E^{-1} обладает свойством переноса. Аналогично доказательству для E .

Итак, отношение E является бисимуляцией разметок между N и N' , что и требовалось доказать. $\square$

Очевидно, что после такой замены многие позиции могут остаться несвязанными ни с одним переходом, и их можно будет просто удалить из сети. Строго говоря,

Утверждение 6.3. Пусть выполнены все условия утверждения 6.2, $p \in P$. Тогда

$$r(p) > 0 \wedge s(p) = 0 \Rightarrow \forall t \in T \quad F'(p, t) = 0 \wedge F'(t, p) = 0.$$

Другими словами, позиция p избыточна, если она принадлежит ресурсу r и не принадлежит ресурсу s .

Доказательство. 1) Докажем отсутствие входных дуг. Предположим противное: существует переход $t \in T$, такой, что $F'(p, t) > 0$.

Из условия утверждения 6.2 имеем

$$\bullet t = i_t r + g_t, \text{ где } i_t \in Nat, \quad g_t \cap r = \emptyset.$$

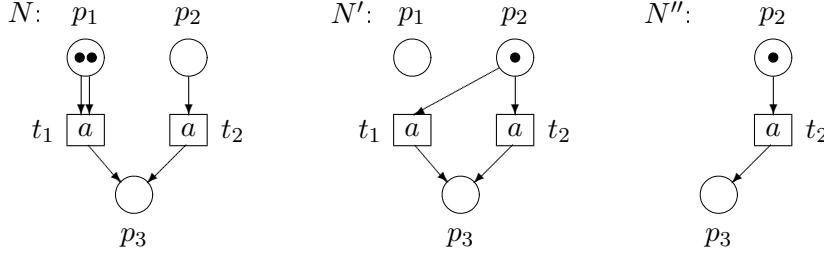


Рис. 6.4: Пример редукции заменой подобных ресурсов. Для сети N выполняется $2p_1 \approx p_2$. Произведена замена при $r = 2p_1$, $s = p_2$.

Так как $r(p) > 0$, выполняется $g_t(p) = 0$, то есть $F(p, t) = i_t r(p)$.

Из условия утверждения 6.2 имеем $F'(p, t) = F(p, t) - i_t r(p) + i_t s(p)$, то есть выполняется

$$F'(p, t) = i_t r(p) - i_t r(p) + i_t s(p) = i_t s(p) = 0.$$

2) Случай выходных дуг — аналогично. $\square$

Это не единственный “побочный” эффект от замены подобных ресурсов. При перемещении дуг по правилу, описанному в утверждении 6.2, в сети может возникнуть много одинаковых переходов. Очевидно, что при наличии в сети двух одинаковых переходов один из них можно удалить без последствий для наблюдаемого поведения.

На рисунке 6.4 приведен пример редукции заменой подобных ресурсов. Сеть N' получена из сети N заменой ресурса $2p_1$ на ресурс p_2 . Сеть N'' получена из сети N' удалением избыточной позиции p_1 и перехода t_1 , дублирующего переход t_2 .

Итак, мы описали два способа редукции обыкновенной сети Петри с использованием подобия ресурсов. Во-первых, можно редуцировать немаркированную сеть, заменяя “подобные” выходные дуги переходов. Во-вторых, можно редуцировать маркированную сеть с заданной начальной разметкой, полностью заменяя

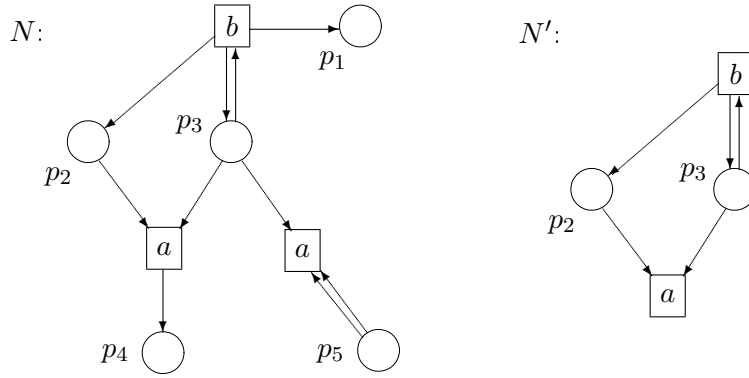


Рис. 6.5: Пример редукции сети Петри на основе подобия ресурсов. Для сети N выполняется $p_1 \approx p_4 \approx \emptyset$, $p_2 \approx 2p_5$.

один ресурс на другой (при выполнении дополнительных условий, накладываемых на заменяемый ресурс). Очевидно, что оба этих способа можно использовать и одновременно, получая еще более сильную редукцию. Получаемая при этом сеть меньшего размера будет иметь то же поведение, что и исходная (с точки зрения бисимулярности). Это гарантируется свойствами подобия ресурсов.

Пример “комплексной” редукции сети Петри на основе подобия ресурсов приведен на рисунке 6.5.

В данном случае объединены ресурсы p_2 и $2p_5$, удалены ресурсы (позиции) p_1 и p_4 (как тупиковые, то есть подобные пустому ресурсу $\emptyset$). Переходы с меткой a слиты в один переход.

В приведенных утверждениях использовано подобие ресурсов, однако очевидно, что на практике для редукции можно использовать и любые сужения подобия. Фактически, если известна только одна пара эквивалентных ресурсов, её уже можно использовать для редукции. Приведенные в этом разделе подходы могут с равным успехом применяться для редукции на основе полного подобия, ограниченного подобия и бисимуляции ресурсов.

6.2 Адаптивное управление процессом

Другая важная область применения отношений эквивалентности ресурсов — адаптивное управление.

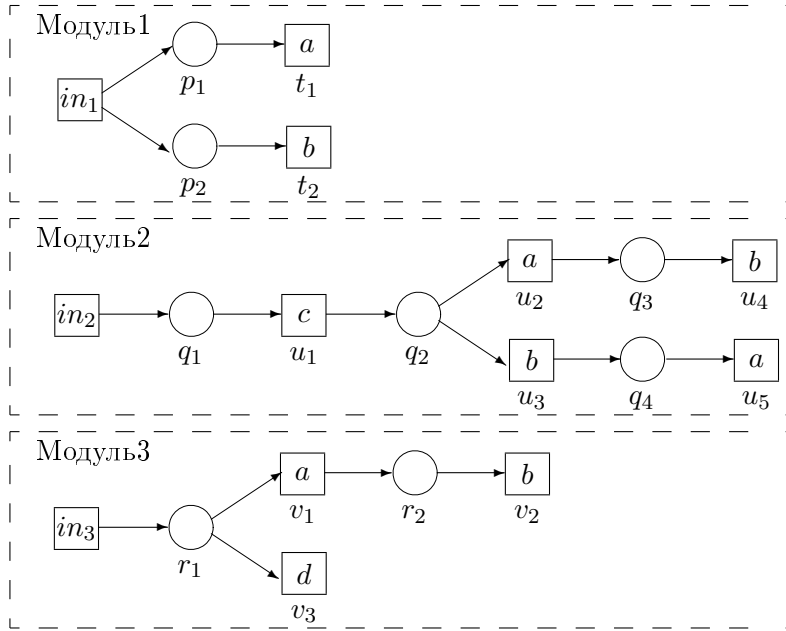
На практике в ходе эксплуатации автоматизированных систем управления (в частности, систем управления потоками работ — Workflow Management Systems [2]) в процесс зачастую приходится вносить изменения непосредственно в ходе его выполнения. Это может происходить по причине доступности/недоступности тех или иных ресурсов, сбоев в отдельных модулях, в силу необходимости обрабатывать нестандартные ситуации/запросы и по многим другим причинам. Перенастройка системы вручную в таких ситуациях может быть сделана только специалистом, хорошо знающим все детали процесса и архитектуру системы, и является, таким образом, очень трудоемкой и затратной процедурой. К тому же внесение изменений в систему может приводить к появлению ошибок в ее функционировании.

В связи с этим необходимо уметь автоматически находить эквивалентные замены для отказавших элементов системы. Можно рассматривать различные критерии эквивалентности ресурсов, в частности, бисимуляцию ресурсов [34] или подобие ресурсов и отношения на его основе.

6.2.1 Управление “без потерь” на основе подобия обобщенных ресурсов

Рассмотрим небольшой пример, показывающий один из возможных способов практического использования подобия обобщенных ресурсов для адаптивного управления процессами.

На рисунке 6.6 изображена модель некоей довольно абстрактной системы обработки запросов. Это может быть организация, система web-сервисов или вычислительное устройство. Система состоит из трех отдельных модулей (подразделений, серверов или процессоров). Каждый из них умеет обрабатывать запросы толь-



$$\begin{aligned}
 (p_1, \emptyset) &\approx (q_4, \emptyset), \quad (p_2, \emptyset) \approx (q_3, \emptyset) \approx (r_2, \emptyset) \\
 (p_1 + p_2, \emptyset) &\approx (q_2, \emptyset) \approx (q_3 + q_4, \emptyset) \\
 (p_1, t_1) &\approx (q_4, u_5), \quad (p_2, t_2) \approx (q_3, u_4) \approx (r_2, v_2) \\
 (p_1 + p_2, t_1) &\approx (q_2, u_2) \approx (q_2 + q_3, u_5) \approx (r_1, v_1) \\
 (p_1 + p_2, t_2) &\approx (q_2, u_3) \approx (q_2 + q_3, u_4)
 \end{aligned}$$

Рис. 6.6: Адаптивная обработка запросов

ко одного определенного типа. Алгоритмы обработки различных типов запросов в целом различны, однако некоторые из операций совпадают. А именно, в каждом модуле выполняются операции “а” и “б”. Поэтому мы хотим знать, каким образом эта похожесть модулей может быть использовано для *адаптивной обработки запросов* — обмена запросами (ресурсами) между модулями в целях уменьшения общей загрузки системы и поддержания ее работоспособности в случае отказа отдельных подсистем. При этом мы хотим сохранять наблюдаемое поведение системы.

Для решения таких задач интересные возможности предоставляет подобие обобщенных ресурсов. Например, в представленной модели системы обработки запросов можно выделить целый ряд нетривиальных подобных ресурсов.

Подобие материальных ресурсов (“обычное” подобие ресурсов) дает удобный набор правил для управления загрузкой подсистем. Например, используя правило $(q_2, \emptyset) \approx (p_1 + p_2, \emptyset)$, мы можем уменьшить загруженность подсистемы 2 (ценой увеличения загруженности подсистемы 1), удаляя фишку из q_2 и помещая фишки в p_1 и p_2 . (Для простоты мы не вдаемся здесь в технические подробности процесса переноса запроса между подсистемами.)

Управление загрузкой обычно производится под воздействием каких-то внешних факторов и причин. В свою очередь, обработка исключительных ситуаций чаще всего имеет дело с внутренними проблемами системы. Рассмотрим ситуацию, при которой в подсистеме 3 происходит сбой в момент выполнения задачи v_1 (или же эта задача просто выполняется слишком долго, в то время как другие — возможно, более мощные — подсистемы простаивают). В такой ситуации правило подобия $(r_1, v_1) \approx (p_1 + p_2, t_1) \approx (q_2, u_2)$ позволяет нам перенести всю входную информацию из подсистемы 3 в другую подсистему (1 или 2), а затем (незаметно для пользователя) рестартовать задачу “а”. Такая манипуляция не является традиционным откатом транзакции (rollback), поскольку мы запоминаем не только состояние системы (разметку), но и мно-

жество выбранных действий (переходов).

Заметим, что изложенный выше подход позволяет изменять структуру системы без каких-либо потерь в её поведении (относительно бисимуляции).

6.2.2 Управление в условиях ограниченного времени на основе расслоенного подобию

Ещё один вид адаптивного управления — управление “с потерями”. Дается гарантия на сохранение поведения только до определенного времени (шага). В дальнейшем системе позволено изменить свое поведение по сравнению с эталонным. Такое управление может потребоваться при обработке каких-либо кризисных ситуаций, когда нужно быстро и правильно выполнить срочные (тактические) действия, и при этом не так важно, будет ли в будущем соблюдена долговременная стратегия (или есть возможность через несколько шагов опять изменить структуру процесса).

Рассмотрим пример на рисунке 6.7. Здесь представлены два циклических бизнес-процесса (потока работ) некоей абстрактной фирмы. За каждый процесс отвечает конкретное подразделение, которое имеет жесткую структуру и поэтому может функционировать только по раз и навсегда определенному алгоритму.

Очевидно, что подразделения фирмы не являются взаимозаменяемыми. То есть в случае проблем в одном из процессов (на любой его стадии) мы не можем “увести” его в другое подразделение “навсегда”. В этом ключевое отличие от примера из предыдущего параграфа — там некоторые состояния различных модулей были полностью эквивалентными. Однако *на время* подразделения все-таки могут обмениваться друг с другом работой. Действительно, из рисунка видно, что на один “такт” мы можем перенести задание из состояния p_2 первого подразделения в состояние q_1 второго (и наоборот).

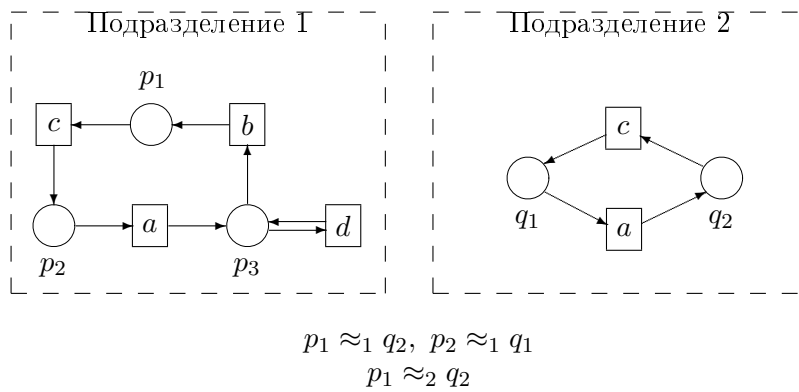


Рис. 6.7: Тактическое адаптивное управление

Для решения задач тактического адаптивного управления бизнес-процессами удобнее всего использовать расслоенное подобие ресурсов. Это отношение позволяет адекватно учесть границы временного интервала, кроме того, оно разрешимо (то есть, в отличие от обычного подобия, здесь не требуется аппроксимация).

Вычислив расслоенные подобия для сети Петри на рисунке 6.7, мы можем сделать вывод о допустимости следующих переносов заданий:

- На один такт: между p_1 и q_2 , между p_2 и q_1 .
- На два такта: между p_1 и q_2 .

Необходимо заметить, что в данном случае под квантами “времени” понимаются срабатывания переходов сети, а не реальные временные интервалы.

Литература

1. Алгоритмы, математическое обеспечение и архитектура много-процессорных вычислительных систем. Под ред. Ершова А.П. Новосибирск: Наука. 1982
2. ван дер Аалст В., ван Хей К. Управление потоками работ: модели, методы и системы. М.: Научный мир. 2007.
3. Ачасова С.М., Бандман О.Л. Корректность параллельных вычислительных процессов. Новосибирск: Наука. 1990.
4. Башкин В.А. Методы насыщения сетей Петри с невидимыми переходами // Моделирование и анализ информационных систем. Ярославль: ЯрГУ. 1999. Вып.6. С.16–20.
5. Башкин В.А. Бисимуляция ресурсов в сетях Петри с невидимыми переходами // Современные проблемы математики и информатики: Сборник научных трудов молодых ученых, аспирантов и студентов. Ярославль: ЯрГУ. 2002. Вып.5. С.79–85.
6. Башкин В.А. Конечное представление отношений на множествах // Моделирование и анализ информационных систем. Ярославль: ЯрГУ. 2003. Вып.10. С.56–59.

7. Башкин В.А. Бисимуляция ресурсов в сетях Петри // Тез. доклада на IV ежегодной научно-практической конференции студентов, аспирантов и молодых ученых Ярославской области. Ярославль, 2003. С.15.
8. Башкин В.А., Ломазова И.А. Бисимуляция ресурсов в сетях Петри // Известия РАН: Теория и системы управления. 2003. №4. С.115–123.
9. Башкин В.А., Ломазова И.А. О языках вложенных рекурсивных сетей Петри // Интеллектуальное управление: новые интеллектуальные технологии в задачах управления (ISIT'99). Труды международной конференции, Переславль-Залесский, 6–9 декабря 1999. М.:Наука. Физматлит, 1999. С.9–13.
10. Башкин В.А., Ломазова И.А. Подобие обобщенных ресурсов в сетях Петри // Труды МСО-2005. М.: МГУ. 2005. С.330–336.
11. Башкин В.А., Ломазова И.А. Эквивалентность ресурсов в моделях потоков работ // Труды международной конференции “Программные системы: теория и приложения”. ИПС РАН, г.Переславль-Залесский. 2006. Т.1. С.323–338.
12. Башкин В.А., Ломазова И.А. О параметризованном построении подобия ресурсов в сетях Петри // Труды IX-ой Международной конференции “Интеллектуальные системы и компьютерные науки”. Изд-во мехмата МГУ, М.2006. Т.1, №1. С.56–58.
13. Котов В.Е. Сети Петри. М.: Наука. 1984.
14. Ломазова И.А. Моделирование мультиагентных динамических систем вложенными сетями Петри // Программные системы: Теоретические основы и приложения. М.: Наука. 1999. С.143–156.

15. Ломазова И.А. Некоторые алгоритмы анализа для многоуровневых вложенных сетей Петри // Известия РАН: Теория и системы управления. 2000. №6. С.965–974.
16. Ломазова И.А. Рекурсивные вложенные сети Петри: анализ семантических свойств и выразительность // Программирование. 2001. №4. С.21–35.
17. Ломазова И.А. Объектно-ориентированные сети Петри: формальная семантика и анализ // Системная информатика. Новосибирск. 2002. №8. Вып.8. С.143–205.
18. Ломазова И.А. Вложенные сети Петри: моделирование и анализ распределенных систем с объектной структурой. М.: Научный мир. 2005.
19. Питерсон Дж. Сети Петри и моделирование систем. М.: Мир. 1984.
20. Подловченко Р.И. Эквивалентные преобразования схем программ с коммутирующими и монотонными операторами // Программирование. 2002. №6. С.301–313.
21. Сидорова Н.С. Преобразования сетей Петри: Дис. канд. физ.-мат. наук. Ярославль: ЯрГУ. 1998.
22. Сидорова Н.С., Соколов В.А. О редукции сетей Петри // Тез. докладов на Третьей международной конференции по алгебре памяти М.И.Каргаполова. Красноярск. 1993. С.305–306.
23. Соколов В.А., Кушнаренок О.Б. Рекурсивно-параллельное программирование и сети Петри: моделирование, анализ и верификация программ // Моделирование и анализ информационных систем. Ярославль: ЯрГУ. 1994. Вып.2. С.98–102.

24. Тарасюк И.В. Эквивалентностные понятия для моделей параллельных и распределенных систем: Дис. канд. физ.-мат. наук. Новосибирск: ИСИ СО РАН. 1997.
25. W. van der Aalst and K. van Hee. Workflow Management: Models, Methods and Systems. MIT Press, 2002.
26. Autant C., Schnoebelen Ph. Place bisimulations in Petri nets // Proc. of ICATPN'92. Lecture Notes in Computer Science. 1992. V.616. P.45–61.
27. Autant C., Pfister W., Schnoebelen Ph. Place bisimulations for the reduction of labeled Petri nets with silent moves // Proc. of ICCI'94. 1994.
28. Baeten J.C.M., Bergstra J.A., Klop J.W. Decidability of bisimulation equivalence for processes generating context-free languages // Journal of the ACM. 1993. V.40. P.653–682.
29. Bashkin V.A., Lomazova I.A. Reduction of Coloured Petri nets based on resource bisimulation // Joint Bulletin of NCC & IIS, Comp. Science. Novosibirsk. 2000. V.13. P.12–17.
30. Bashkin V.A., Lomazova I.A. Resource bisimulations in Nested Petri Nets // Proc. of CS&P'2002. Vol.1. Informatik Berichte №161. Humboldt-Universitat zu Berlin. Berlin. 2002. P.39–52.
31. Bashkin V.A., Lomazova I.A. Petri Nets and resource bisimulation // Fundamenta Informaticae. 2003. V.55. №2. P.101–114.
32. Bashkin V.A., Lomazova I.A. Resource similarities in Petri net models of distributed systems // Proc. of PACT'2003. Lecture Notes in Computer Science. 2003. V.2763. P.35–48.

- 33. Bashkin V.A., Lomazova I.A. Similarity of Generalized Resources in Petri Nets // Proc. of PaCT'2005. Lecture Notes in Computer Science. 2005. V.3606. P.27–41.
- 34. Bashkin V.A. Applications of Marking Bisimulation for Adaptive Workflow Management // Proc. of CS&P'2005. Warsaw University. 2005. P.41–49.
- 35. Bashkin V.A., Lomazova I.A. Resource equivalence in workflow nets // Proc. of CS&P'2006. Vol.1. Humboldt-Universitat zu Berlin. Berlin. 2006. P.80–91.
- 36. Christensen S., Hüttel H., Stirling C. Bisimulation relation is decidable for all context-free processes // Proc. of CONCUR'92. Lecture Notes in Computer Science. 1993. V.630. P.138–147.
- 37. Christensen S., Hirshfeld Y., Moller F. Decomposability, decidability and axiomatisability for bisimulation equivalence on basic parallel processes // Proc. of LICS'93. 1993. P.386–396.
- 38. Christensen S., Hirshfeld Y., Moller F. Bisimulation relation is decidable for basic parallel processes // Proc. of CONCUR'93. Lecture Notes in Computer Science. 1993. V.715. P.143–157.
- 39. Esparza J., Nielsen M. Decidability Issues for Petri Nets — a Survey // Bulletin of the EATCS. 1994. V.52. P.245–262.
- 40. Esparza J. Decidability and complexity of Petri net problems — an introduction // Lecture Notes in Computer Science. 1998. V.1491. P.374–428.
- 41. Esser R. An Object Oriented Petri Net Approach to Embedded System Design. PhD theses. Zurich: Swiss Federal Institute of Technology. 1996.

42. Genrich H., Lautenbach K. System modeling with high-level Petri nets // Theoretical Computer Science. 1981. V.13. P.109–136.
43. Haddad S., Poitrenaud D. Theoretical aspects of recursive Petri nets // Proc. of ATPN'99. Lecture Notes in Computer Science. 1999. V.1639. P.228–247.
44. Hirshfeld Y. Congruences in commutative semigroups. Research report ECS-LFCS-94-291. Edinburgh: University of Edinburgh, Department of Computer Science. 1994.
45. Hirshfeld Y., Jerrum M., Moller F. A polynomial algorithm for deciding bisimulation equivalence of normed basic parallel processes // Mathematical Structures in Computer Science. 1996. V.6. P.251–259.
46. Hirshfeld Y., Jerrum M. Bisimulation equivalence is decidable for normed Process Algebra // Proc. of ICALP'99. Lecture Notes in Computer Science. 1999. V.1644.
47. Jantzen M. Language Theory of Petri Nets // Lecture Notes in Computer Science. 1987. V.254.
48. Jančar P. Decidability questions for bisimilarity of Petri nets and some related problems // Proc. of STACS'94. Lecture Notes in Computer Science. 1993. V.775. P.581–592.
49. Jančar P. Bisimulation equivalence is decidable for one-counter processes // Proc. of ICALP'97. Lecture Notes in Computer Science. 1997. V.1256. P.549–559.
50. Jančar P., Kučera P., Moller F. Simulation and Bisimulation over One-Counter Processes // Proc. of STACS'2000. Theoretical Computer Science. 2000.

51. Jančar P., Moller F. Techniques for decidability and undecidability of bisimilarity // Proc. of CONCUR'99. Lecture Notes in Computer Science. 1999. V.1664. P.30–45.
52. Jensen K. Coloured Petri Nets and the Invariant Method // Theoretical Computer Science. 1981. V.14.
53. Jensen K. Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Volume 1: Basic Concepts, 1992. Volume 2: Analysis Methods, 1994. EATCS Monographs in Computer Science. Springer.
54. Jensen K., Christensen S., Huber P., Holla M. Design/CPN Reference Manual. Computer Science Department, University of Aarhus, Denmark. Online:<http://www.daimi.au.dk/designCPN>.
55. Lomazova I.A. Nested Petri nets — a Formalism for Specification and Verification of Multi-Agent Distributed Systems // Fundamenta Informaticae. 2000. V.43. P.195–214.
56. Lomazova I.A. Nested Petri nets: multi level and recursive systems // Fundamenta Informaticae. 2001. V.47. P.283–293.
57. Milner R. A Calculus of Communicating Systems // Lecture Notes in Computer Science. 1980. V.92.
58. Moller F. Infinite results // Proc. of CONCUR'96. Lecture Notes in Computer Science. 1996. V.1119. P.195–216.
59. Park D.M.R. Concurrency and automata on infinite sequences // Lecture Notes in Computer Science. 1981. V.104.
60. Petri C.A. Kommunikation mit Automaten. PhD theses. Bonn: Institute für Instrumentelle Mathematik. 1962.

61. Redei L. The theory of finitely generated commutative semigroups. Oxford University Press. 1965.
62. Reisig W. Petri nets. Springer. 1985.
63. Reisig W. Petri nets and algebraic specifications // Theoretical Computer Science. 1991. V.80. P.1–34.
64. Reisig W. Petri net models of distributed algorithms // Lecture Notes in Computer Science. 1995. V.1000.
65. Schnoebelen Ph., Sidorova N. Bisimulation and the reduction of Petri nets // Proc. of ICATPN'2000. 2000.
66. Smith E. A Survey on High-Level Petri-Net Theory // Bulletin EATCS. 1996. V.59. P.267–293.
67. Smith E. Principles of high-level net theory. Lectures on Petri nets // Lecture Notes in Computer Science. 1998. V.1491. P.174–210.
68. Stirling C. Decidability of bisimulation equivalence for normed pushdown processes // Theoretical Computer Science. 1998. V.195. P.113–131.
69. Tarasyuk I.V. τ -Equivalences and Refinement for Petri Net Based Design. Technische Berichte FI00-11. Dresden: Technische Universität zu Dresden. 2000.
70. Valk R. Petri Nets as Token Objects: An Introduction to Elementary Object Nets // Proc. of ICATPN'98. Lecture Notes in Computer Science. 1998. V.1420. P.1–25.
71. <http://www.daimi.au.dk/PetriNets> — Petri Net WWW page of Aarhus University (неофициальный вебсайт Petri Net Community), Aarhus, Denmark.

72. <http://www.lsv.ens-cachan.fr> — Laboratoire Spécification et Vérification, Ecole Normale Supérieure de Cachan, Cachan, France.
73. <http://www.dcs.ed.ac.uk> — Department of Computer Science, University of Edinburgh, Edinburgh, UK.
74. <http://citeseer.nj.nec.com> — NEC Digital Library.

Научное издание

В. А. Башкин, И. А. Ломазова

ЭКВИВАЛЕНТНОСТЬ РЕСУРСОВ
В СЕТЯХ ПЕТРИ