

РОССИЙСКАЯ АКАДЕМИЯ НАУК  
СИБИРСКОЕ ОТДЕЛЕНИЕ  
ИНСТИТУТ СИСТЕМ ИНФОРМАТИКИ  
ИМ. А.П. ЕРШОВА

На правах рукописи

ТАРАСЮК ИГОРЬ ВАЛЕРЬЕВИЧ

**Эквивалентные понятия для моделей  
параллельных и распределенных систем**

*Специальность 05.13.11 — математическое и программное обеспечение  
вычислительных машин, комплексов, систем и сетей*

*ДИССЕРТАЦИЯ  
на соискание ученой степени  
кандидата физико-математических наук*

Научные руководители:  
к.ф.-м.н., с.н.с. ИСИ СО РАН, доцент ВИРВИЦКАЙТЕ И.Б.,  
д.ф.-м.н., директор ИСИ СО РАН, профессор ПОТТОСИН И.В.

Новосибирск  
1997

# Оглавление

<b>Введение</b>	<b>3</b>
<b>1 Сети Петри</b>	<b>9</b>
1.1 Основные определения	10
1.1.1 Мультимножества	10
1.1.2 Помеченные сети	10
1.1.3 Маркированные сети	12
1.1.4 Частично упорядоченные множества	13
1.1.5 Структуры событий	14
1.1.6 С-процессы	15
1.1.7 О-процессы	17
1.2 Сети Петри с видимыми переходами	18
1.2.1 Базисные эквивалентности	18
1.2.2 Обратные-прямые эквивалентности	30
1.2.3 Эквивалентности мест	39
1.2.4 Сохранение эквивалентностей при детализациях	46
1.2.5 Эквивалентности на подклассах сетей Петри	53
1.3 Сети Петри с невидимыми переходами	62
1.3.1 Базисные $\tau$ -эквивалентности	64
1.3.2 Обратные-прямые $\tau$ -эквивалентности	73
1.3.3 Сравнение эквивалентностей с $\tau$ -эквивалентностями	78
1.3.4 Сохранение $\tau$ -эквивалентностей при детализациях	79
1.3.5 $\tau$ -эквивалентности на подклассах сетей с невидимыми переходами	84
<b>2 Временные сети Петри</b>	<b>89</b>
2.1 Основные определения	90
2.2 Временные сети Петри с видимыми переходами	91
2.2.1 Временные эквивалентности	91
2.2.2 Не-временные эквивалентности	92
2.2.3 Региональные эквивалентности	92
2.2.4 Сравнение эквивалентностей	94
2.2.5 Сохранение эквивалентностей при временных детализациях	96
2.2.6 Эквивалентности на не-временных сетях Петри	103
2.3 Временные сети Петри с невидимыми переходами	104
2.3.1 Временные $\tau$ -эквивалентности	105
2.3.2 Не-временные $\tau$ -эквивалентности	105
2.3.3 Региональные $\tau$ -эквивалентности	106
2.3.4 Сравнение $\tau$ -эквивалентностей	107
2.3.5 Сравнение эквивалентностей и $\tau$ -эквивалентностей	108
2.3.6 Сохранение $\tau$ -эквивалентностей при временных детализациях	108
2.3.7 $\tau$ -эквивалентности на подклассах временных сетей Петри с невидимыми переходами	109

<b>3</b>	<b>Алгебры процессов</b>	<b>111</b>
3.1	Алгебра конечных процессов $AFP_2$	112
3.1.1	Синтаксис	112
3.1.2	Семантика	113
3.1.3	Эквивалентности на $A$ -сетях	117
3.1.4	Сравнение сетевых и алгебраических эквивалентностей	119
3.1.5	Автоматизация проверки алгебраической эквивалентности	121
3.2	Алгебра конечных помеченных процессов $AFLP_2$	126
3.2.1	Синтаксис	127
3.2.2	Семантика	128
3.2.3	Эквивалентности на слабо помеченных $A$ -сетях	134
3.2.4	Сравнение сетевых и алгебраических эквивалентностей	138
3.2.5	Введение алгебраических аналогов сетевых эквивалентностей	140
	<b>Заключение</b>	<b>145</b>
	<b>A Доказательства</b>	<b>146</b>
	<b>B Описание программы CANON</b>	<b>155</b>
	<b>C Примеры работы программы CANON</b>	<b>160</b>

# Введение

**Актуальность проблемы.** В последние годы широкое распространение получили вычислительные машины и комплексы с параллельной и распределенной архитектурой, позволяющие успешно справляться с постоянно растущим объемом вычислительных задач. Однако проблема поведенческого анализа для таких параллельных систем является значительно более сложной, чем для обычных последовательных систем, в силу того, что некоторые их компоненты могут работать частично или полностью независимо друг от друга. В связи с этим все более важное значение приобретает такая область компьютерных наук, как теория параллельных систем и процессов, в которой с помощью различных математических формализмов исследуется поведение указанных систем.

В теории параллелизма с самого начала уделялось большое внимание разработке формальных моделей, предназначенных для спецификации и анализа систем с независимым выполнением действий. В дополнение к таким стандартным моделям, как языки, автоматы и системы переходов [185, 238], введены сети Петри [232], процессные алгебры [198, 204, 168, 169], следы Хоара [175], следы Мазуркевича [194, 195], деревья синхронизации [289] и структуры событий [215, 287, 290, 291].

В работах [254, 255] проведена классификация и сравнение основных моделей в соответствии со следующими дихотомиями:

1. структура / поведение;
2. интерливинг / истинный параллелизм;
3. линейное / ветвистое время.

В структурных моделях явно описаны состояния систем, и поведенческие аспекты сокрыты, а в поведенческих моделях представлены действия, выполняемые системами, и состояния извлекаются из поведения.

В интерливинговых моделях параллелизм моделируется недетерминированным чередованием действий (интерливингом), в то время как в моделях истинного параллелизма он трактуется как причинная независимость действий.

В моделях линейного времени не учитывается конфликт, то есть тот момент в поведении, когда происходит выбор между несколькими альтернативными путями дальнейших вычислений (несколькими “будущими” или “ветвями”), тогда как в моделях ветвистого времени данная информация представлена полно.

Переход от одной модели к другой соответствует смене уровня абстракции, на котором рассматривается поведение моделируемых систем, то есть смене семантики.

Альтернативный (и более удобный) подход заключается в выборе достаточно выразительной (структурной или поведенческой) формальной модели и рассмотрению различных эквивалентностей, задающих семантику или критерий равенства на необходимом уровне абстракции в соответствии с двумя оставшимися дихотомиями.

Среди таких моделей наибольшую известность приобрели структурная модель сетей Петри и поведенческая модель алгебр процессов. Как отмечено в [273], основные достоинства сетей Петри состоят в ясном описании параллелизма, а также в накопленном опыте спецификации и анализа параллельных систем и удобном графическом представлении. Главные преимущества процессных алгебр — модульность, хорошо разработанные понятия эквивалентностей, алгебраические правила и полные системы доказательств. Совместное рассмотрение этих моделей (в частности, определение сетевой семантики алгебраических формул) позволяет сочетать лучшие свойства данных формализмов.

Понятие эквивалентности является важнейшим понятием любой теории систем. Поведенческие эквивалентности позволяют сравнивать параллельные 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

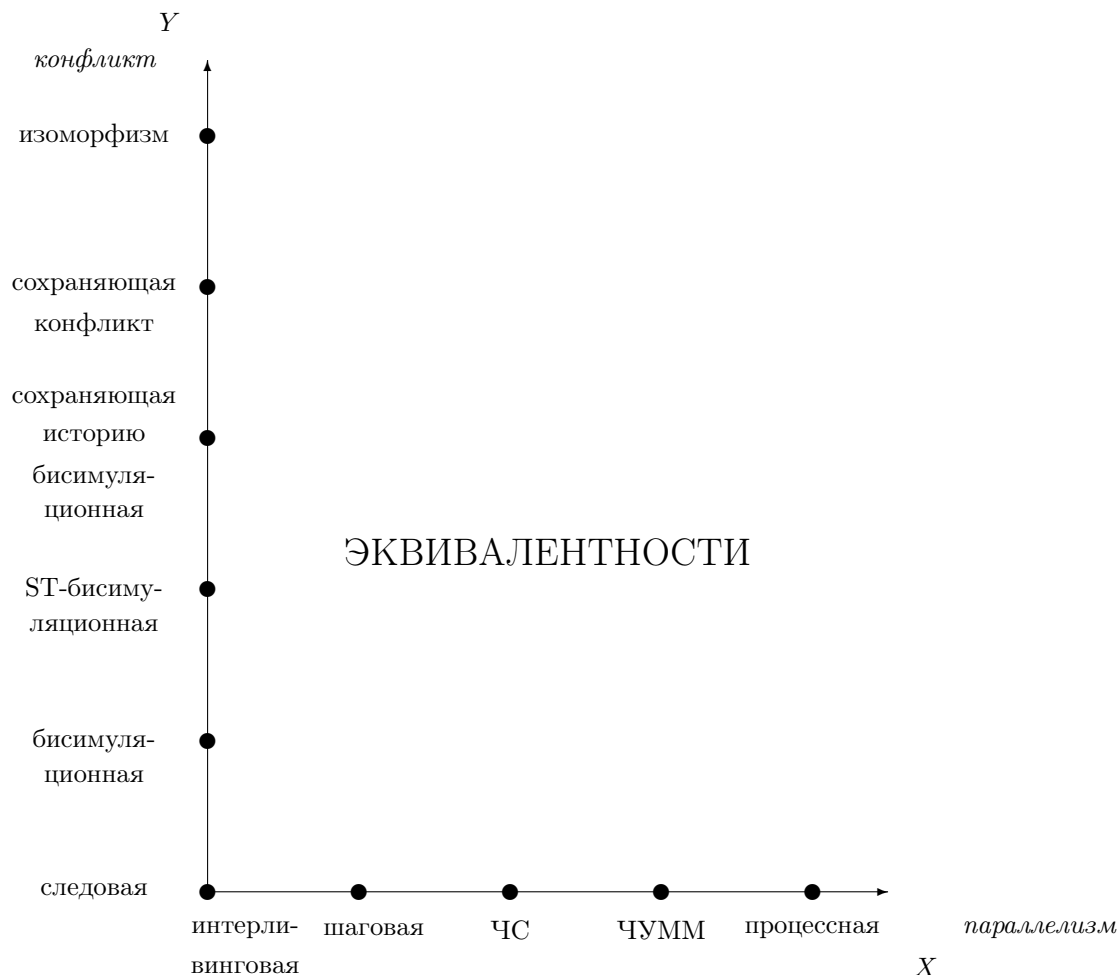


Рис. 1: Семантическая координатная плоскость

ожидаемое или высокоуровневое (спецификация) и реальное или низкоуровневое (применение) поведения систем.

В настоящее время определен ряд эквивалентностей, из которых наиболее известной является бисимуляционная. Ее огромное значение для сравнения и редукции параллельных систем, доказательства их корректности неоднократно отмечалось в литературе.

Основные известные семантики, определяемые эквивалентностями, можно изобразить точками на координатной плоскости, представленной на рисунке 0.1.

Движению вправо по оси  $X$  будет соответствовать возрастание степени моделирования параллелизма: от интерливинговой семантики до семантик истинного параллелизма.

При движении вверх по оси  $Y$  будет возрастать степень моделирования конфликта, ветвистой структуры: от семантики линейного времени (следовой) до семантик ветвистого времени.

Известны следующие точки на оси  $X$ .

**Интерливинговая семантика** Процессу (то есть возможному функционированию системы) соответствует последовательность действий, выписанных в порядке их срабатывания.

**Шаговая семантика** Процессу соответствует последовательность мультимножеств параллельных действий.

**Семантика частичных слов (ЧС)** Процессу соответствует частично упорядоченное мультимножество действий (ЧУММ) с порядком по отношению причинной зависимости. В этой семантике одна система

может моделировать другую, если отношение причинной зависимости между действиями в ее ЧУММ менее строгое или такое же, как в ЧУММ другой системы.

**ЧУММ семантика** Процессу соответствует ЧУММ, и при моделировании должно сохраняться отношение причинной зависимости действий.

**Процессно-сетевая семантика** Процессу соответствует ациклическая безконфликтная сеть (так называемая, С-сеть). Для краткости будем называть эквивалентности, связанные с этой семантикой, “процессными”.

Известны следующие точки на оси  $Y$ .

**Следовая семантика** Поведение системы полностью определяется множеством ее процессов. Две системы считаются равными в данной семантике, если совокупности их процессов совпадают.

**Бисимуляционная семантика** Принимается во внимание ветвистая структура поведения системы, то есть учитываются точки выбора между несколькими альтернативными расширениями текущего процесса. При этом требуется взаимное моделирование расширяющих процессов.

**ST-бисимуляционная семантика** Предполагается, что действия имеют некоторую внутреннюю структуру (или выполняются не мгновенно, а за определенный промежуток времени). При моделировании учитывается причинная зависимость действий расширяющего процесса и работающих действий текущего процесса.

**Сохраняющие историю бисимуляционная семантика** Эта семантика учитывает “прошлое” процессов, то есть то как расширяющий процесс причинно связан с текущим процессом.

**Сохраняющие конфликт семантика** Полностью учитывается информация о конфликте.

**Семантика изоморфизма** Это самая сильная семантика. Две системы изоморфны, если они отличаются лишь конкретными именами своих компонент.

В указанной области следующие проблемы требуют решения и являются предметом углубленных исследований.

- До сих пор нет достаточно полного набора эквивалентностей во всех рассмотренных семантиках. Некоторые из претендентов на заполнение пустых мест координатной плоскости были определены в рамках разных моделей, поэтому возникают трудности в их сравнении. Как отмечено в [166], для систематического исследования семантики параллельных систем полезно рассмотреть все возможности для них быть эквивалентными. Это ведет к лучшему пониманию важнейших свойств систем. С другой стороны, для практических целей спецификации и верификации этих свойств, необходимо иметь ряд подходящих понятий эквивалентности, чтобы всегда иметь возможность выбора наиболее простой точки зрения на системы.
- Сравнение эквивалентностей позволяет понять их взаимосвязь и природу, избавляет от дублирования известных семантик, ведет к пониманию, какие из них надо доопределить. К настоящему времени не установлена взаимосвязь ряда важных эквивалентностных отношений.
- Актуальна задача логической характеристики поведенческих эквивалентностей, решение которой позволяет рассуждать о поведении систем в терминах формул темпоральных логик.
- Очевидно значение эффективных методов редукции систем с сохранением поведения по модулю эквивалентностей.
- При нисходящей разработке систем и программ происходит переход с высокого на низкий уровень абстракции от их структуры. Очень важно, чтобы системы, имевшие сходное поведение на одном из уровней, сохраняли его на всех более низких. Таким образом, необходимо проверить, какие из эквивалентностей сохраняются при операции детализации, заменяющей элементарные компоненты систем на более сложные структуры. В настоящее время этот вопрос остается открытым для ряда эквивалентностей.

- Взаимосвязь параллелизма и недетерминизма (конфликта), а также другие особенности параллельных систем зачастую сильно усложняют анализ их поведения. Поэтому в литературе был введен ряд более простых подклассов формальных моделей, пригодных для практических целей. Важно исследовать эквивалентности на упомянутых подклассах для упрощения проверки этих отношений, лучшего понимания их природы и восстановления сложной информации о поведении по более простой. Данная проблема также не вполне разработана.
- Важная задача — разработка эквивалентностных понятий для расширений известных формальных моделей (например, понятием абстракции от невидимых действий, введением понятия времени, пометки элементарных событий).
- Актуален вопрос установления взаимосвязи эквивалентностей, определенных в рамках различных формализмов, что дало бы возможность переходить к спецификациям в разных моделях без изменения поведенческих свойств.

**Цель** диссертационной работы состоит в разработке ряда методов и средств, которые способствовали бы решению указанных выше проблем.

**Методы исследования** заключаются в использовании трех важнейших формальных моделей параллелизма: сетей Петри, временных сетей Петри [196] и алгебр процессов, а также их подклассов (последовательных сетей, строго помеченных сетей, T-сетей), расширений (сетей с невидимыми переходами, алгебр помеченных процессов) и аппарата темпоральных логик.

**Научная новизна** данной работы состоит в следующем.

1. В рамках сетей Петри с видимыми и невидимыми переходами введен и исследован широкий набор поведенческих эквивалентностей в семантиках от интерливинговой до истинного параллелизма и от линейного до ветвистого времени, позволяющих абстрагироваться от структурных и поведенческих свойств моделируемых систем.
  - Получена диаграмма взаимосвязей указанных выше эквивалентностей. Дана логическая характеристика ряда эквивалентностей, позволяющая рассуждать о поведении параллельных систем в терминах формул темпоральных логик. Описан метод эффективной редукции сетей с сохранением их поведения по модулю эквивалентностей.
  - Исследованы композиционные аспекты сохранения поведенческих свойств моделируемых параллельных систем.
  - Установлена взаимосвязь эквивалентностных отношений на подклассах сетей с целью упрощения сравнения их поведения и лучшего понимания природы эквивалентностей.
2. На временных сетях Петри с видимыми и невидимыми переходами исследован ряд временных, невременных и региональных эквивалентностей, способных в разной степени учитывать временные аспекты поведения моделируемых систем.
  - Выяснено соотношение указанных эквивалентностей. Дана региональная характеристика временных эквивалентностей, позволяющая упростить проверку последних.
  - Разработан композиционный подход к проверке эквивалентности временных систем.
  - Установлена взаимосвязь эквивалентностей на подклассах временных сетей.
3. Исследованы семантические эквивалентности алгебраических исчислений и их расширений, а также их связь с сетевыми эквивалентностными отношениями.
  - Разработано новое исчисление помеченных недетерминированных параллельных процессов  $AFLP_2$  — расширение известной алгебры  $AFP_2$  (введенной В.Е. Котовым и Л.А. Черкасовой) функцией пометки, что позволило специфицировать значительно более широкий класс процессов, чем в  $AFP_2$ .
  - Дана полная и корректная аксиоматизация эквивалентностей относительно денотационных семантик указанных алгебр, а также операционная характеристика этих эквивалентностей, позволяющая сравнивать их с поведенческими эквивалентностными отношениями.
  - Установлены взаимосвязи алгебраических и сетевых эквивалентностей, что дало возможность переходить от сетевых к алгебраическим спецификациям и обратно с сохранением поведения и объединило преимущества сетей и алгебр.

- Предложена система правил переписывания для автоматизации проверки семантических эквивалентностей алгебраических исчислений, на основе которой разработана программа для машинной проверки эквивалентности формул.

**Практическая ценность** работы заключается во введении ряда эквивалентностных отношений для сравнения и редукции параллельных систем и их систематическом исследовании, позволяющем разработчику выбрать наиболее подходящую модель и критерий равенства (семантику).

**Реализация результатов** работы заключается в их использовании в лаборатории теоретического программирования ИСИ СО РАН при создании модуля проверки сетевых эквивалентностей для системы PER (Programming Environment based on Petri nets) [59], разработанной в Институте информатики г. Хильдесхайма (Германия). Кроме того, автором написана программа CANON, реализующая проверку семантической эквивалентности алгебраических формул.

**Публикация результатов.** По теме диссертации было опубликовано 13 научных работ: [4, 256, 257, 5, 6, 258, 259, 260, 261, 262, 263, 264, 284].

Во время работы над диссертацией автор участвовал в следующих **научных проектах** (с указанием фондов поддержки):

1. *Исследование методов анализа и верификации параллельных вычислительных систем, программ и процессов.* Российский Фонд Фундаментальных Исследований (РФФИ), грант 93-01-00986, руководитель к.ф.-м.н. В.А. НЕПОМНЯЩИЙ, 1993–1995. (Публикации [256, 5, 6].)
2. *Models for formal semantics of reactive systems.* International Association for Promotion of Cooperation with Scientists from the Former Soviet Union (INTAS), грант 1010-СТ93-0048, руководитель к.ф.-м.н. В.А. НЕПОМНЯЩИЙ, 1993–1995. (Публикация [257].)
3. *Formal methods in design of concurrent / distributed systems.* Volkswagen Stiftung (VS), грант I/70 564, руководители Prof. Dr. EIKE BEST и к.ф.-м.н. И.Б. ВИРВИЦКАЙТЕ, 1995–1997. (Публикации [258, 259, 260, 261, 262, 263, 264, 284].)
4. *Разработка и исследование семантических методов и средств спецификации и верификации параллельных систем и процессов.* РФФИ, грант 96-01-01655, руководитель к.ф.-м.н. И.Б. ВИРВИЦКАЙТЕ, 1995–1997. (Публикации [260, 261, 262, 284].)
5. *Methods and Tools for Verification and Analysis of Distributed Systems.* INTAS-RFBR, грант 95-0378, руководитель к.ф.-м.н. В.А. НЕПОМНЯЩИЙ, 1997–1999. (Публикации [263, 264].)
6. International Soros Science Education Program (ISSEP), грант а97-683, 1997. Автору присвоено звание Соросовского Аспиранта. (Публикация [284].)

**Апробация** работы проведена на следующих международных научных конференциях и школах.

1. *4<sup>th</sup> International Conference on Applied Logics - 95 (AL'95)* и *2<sup>nd</sup> Workshop on Non-Standard Logics and Logical Foundations of Computer Sciences - 95 (NSL'95)*, Иркутск, Россия, 15–18 июня 1995.
2. *5<sup>th</sup> Workshop on Concurrency, Specification and Programming - 96 (CSP'96)*, Берлин, Германия, 25–27 сентября 1996.
3. *4<sup>th</sup> Symposium on Logical Foundations of Computer Science - 97 (LFCS'97)*, Ярославль, Россия, 6–12 июля 1997.
4. *International Summer School on Computational Logic - 97 (CL'97)*, Марктобердорф, Германия, 29 июля – 10 августа 1997.
5. *4<sup>th</sup> Workshop on Logic, Languages, Information and Computation - 97 (WoLLIC'97)*, Форталеза (Цеара), Бразилия, 19–22 августа 1997.

Кроме того, доклады по теме работы были сделаны на ряде семинаров Института информатики г. Хильдесхайма (Германия) во время работы автора с 01.12.95 по 29.02.96 и с 01.03.97 по 30.04.97 в рамках совместного научного проекта по гранту VS I/70 564. Полученные результаты обсуждались также на семинарах лаборатории теоретического программирования ИСИ СО РАН.

Сделаем краткий обзор **содержания** работы.

Глава 1 посвящена исследованию эквивалентностей на сетях Петри.



В разделе 1.1 даются основные определения. Раздел 1.2 посвящен исследованию эквивалентностей на сетях с видимыми переходами, а раздел 1.3 — с невидимыми переходами.

Глава 2 посвящена исследованию эквивалентностей на временных сетях Петри.

В разделе 2.1 даются основные определения. Раздел 2.2 посвящен исследованию эквивалентностей на временных сетях с видимыми переходами, а раздел 2.3 — с невидимыми переходами.

Глава 3 посвящена эквивалентностям на формулах процессов.

В разделе 3.1 рассматривается алгебра  $AFP_2$ , а в разделе 3.2 — исчисление  $AFLP_2$ .

В приложении А перенесен ряд больших по объему доказательств.

В приложении В описывается структура и принципы работы программы CANON, реализующей проверку семантической эквивалентности алгебраических формул.

В приложении С приводятся примеры преобразования формул  $AFP_2$  к каноническому виду с помощью этой программы.

Работы автора связаны с темами глав следующим образом. По тематике главы 1 — работы [256, 257, 5, 258, 260, 262, 263, 264], по тематике главы 2 — работа [284], и по тематике главы 3 — работы [4, 6, 259, 261].

# Глава 1

## Сети Петри

Сети Петри [232, 233, 234, 235, 247] — удобная и мощная популярная формальная модель для представления и анализа поведения параллельных и распределенных систем [226, 50, 51, 115]. Сети Петри получили широкое распространение благодаря ясному описанию таких базисных понятий, как причинная зависимость, недетерминизм и параллелизм, а также удобному графическому представлению и естественной семантике функционирования на основе “правила срабатывания”. Они активно используются как графическое средство моделирования параллельных систем и процессов на ранних стадиях разработки, а также в качестве мощного аналитического инструмента изучения их поведенческих свойств.

В этой главе эквивалентные понятия изучаются на двух классах сетей Петри: сети с видимыми переходами и сети с невидимыми переходами (переходы могут быть помечены символом невидимого действия  $\tau$ ).

Основные результаты данной главы следующие.

- Ряд известных эквивалентностей перенесен на сети из других формальных моделей (например, систем переходов, структур событий, поведенческих структур, автоматов). В результате возникла среда для их сравнения и выяснения, какие недостающие понятия следует определить.
- Введены новые эквивалентные отношения для сетей Петри с целью получения их полного набора и заполнения пустых мест на семантической плоскости. В результате появилась возможность выбора наиболее простой подходящей точки зрения на моделируемые системы и отпала необходимость введения новых семантик, дублирующих в новом контексте уже предложенные ранее.
- Даны параметризованные определения эквивалентностей, дающие возможность их трактовки в едином стиле и позволяющие лучше понять природу этих отношений.
- Проведено сравнение всех рассмотренных сетевых эквивалентностей и получена полная картина их взаимосвязей, демонстрирующая различающую силу этих понятий.
- Дана логическая характеристика ряда эквивалентностей, позволяющая рассуждать о поведении сетей с использованием представления их свойств формулами темпоральных логик.
- Приведены результаты по эффективному, семантически корректному упрощению сетей и показана роль взаимосвязи эквивалентностей в сохранении свойств редуцируемой сети.
- Все эквивалентности проверены на сохранение при операции детализации, соответствующей переходу на более низкий уровень структурной абстракции при моделировании систем. В результате выяснено, какие эквивалентные отношения могут быть использованы при разработке систем по методу “сверху-вниз”.
- Исследована взаимосвязь эквивалентностей на подклассах сетей с целью упрощения сравнения поведения сетей этих подклассов, лучшего понимания природы отношений эквивалентности и восстановления сложной информации о поведении по более простой. Проверено, какие эквивалентности хорошо различают сети из упомянутых подклассов.

## 1.1 Основные определения

В данном разделе приводятся основные определения, используемые в дальнейшем.

### 1.1.1 Мультимножества

Мультимножества — расширение понятия множеств допущением в них нескольких одинаковых элементов.

**Определение 1.1.1** Пусть  $X$  — некоторое множество. Мультимножество  $M$  над  $X$  — отображение  $M : X \rightarrow \mathbf{N}$ , где  $\mathbf{N}$  — множество натуральных чисел.

Обозначим через  $\mathcal{M}(X)$  множество всех конечных мультимножеств над  $X$ .

Для  $x \in X$   $M(x)$  — кратность  $x$  в  $M$ . Пишем  $x \in M$ , если  $M(x) > 0$ .

Когда  $\forall x \in X M(x) \leq 1$ ,  $M$  — обычное множество.  $M$  конечно, если  $M(x) = 0$  для всех  $x \in X$ , за исключением, быть может, конечного их числа. Мощность мультимножества  $M$  определяется так:  $|M| = \sum_{x \in X} M(x)$ . В дальнейшем будем рассматривать только конечные мультимножества.

Определим операцию вычитания до нуля следующим образом:  $\forall x, y \in X x \ominus y = \max\{0, x - y\}$ .

Понятия теории множеств стандартно расширяются на конечные мультимножества. Если  $M_1, M_2 \in \mathcal{M}(X)$  и  $x \in X$ , то:

$$\begin{aligned} (M_1 + M_2)(x) &= M_1(x) + M_2(x); \\ (M_1 - M_2)(x) &= M_1(x) \ominus M_2(x); \\ (M_1 \cup M_2)(x) &= \max\{M_1(x), M_2(x)\}; \\ (M_1 \cap M_2)(x) &= \min\{M_1(x), M_2(x)\}; \\ M_1 \subseteq M_2 &\Leftrightarrow \forall x \in X M_1(x) \leq M_2(x). \end{aligned}$$

Запись  $M + x - y$  часто используется вместо  $M + \{x\} - \{y\}$ . Символ  $\emptyset$  обозначает пустое мультимножество.

**Пример 1.1.1** Мультимножества представляют как множества с несколькими одинаковыми элементами. Например, запись  $M = \{x, x, x, y, z, z\}$  обозначает мультимножество  $M$  такое, что  $M(x) = 3$ ,  $M(y) = 1$ ,  $M(z) = 2$ , а на остальных элементах, отличных от  $x, y, z$ , функция  $M$  равна 0.

### 1.1.2 Помеченные сети

Помеченные сети — сети Петри [232], переходы которых помечены именами определенных действий.

Пусть  $Act = \{a, b, \dots\}$  — множество имен действий (меток).

**Определение 1.1.2** Помеченная сеть — четверка  $N = \langle P_N, T_N, F_N, l_N \rangle$ , где:

- $P_N = \{p, q, \dots\}$  — множество мест;
- $T_N = \{t, u, \dots\}$  — множество переходов;
- $F_N : (P_N \times T_N) \cup (T_N \times P_N) \rightarrow \mathbf{N}$  — функция инцидентности;
- $l_N : T_N \rightarrow Act$  — пометка переходов именами действий.

Предполагается, что  $P_N \cap T_N = \emptyset$ .

**Пример 1.1.2** На рисунке 1.1 дано графическое представление помеченной сети. Места изображаются окружностями, переходы — квадратами, в которые вписаны символы действий, помечающие эти переходы. Названия мест и переходов помещены рядом с ними (их часто не указывают, когда в этом нет необходимости). Функция инцидентности изображается направленными дугами, соединяющими места и переходы. Количество дуг соответствует ее значению на соответствующей паре место-переход (или переход-место). Если это количество больше единицы, то оно записывается соответствующей цифрой возле дуги.

Даны помеченные сети  $N = \langle P_N, T_N, F_N, l_N \rangle$  и  $N' = \langle P_{N'}, T_{N'}, F_{N'}, l_{N'} \rangle$ . Отображение  $\beta : P_N \cup T_N \rightarrow P_{N'} \cup T_{N'}$  — изоморфизм между  $N$  и  $N'$ , обозначение  $\beta : N \simeq N'$ , если:

1.  $\beta$  — биекция такая, что  $\beta(P_N) = P_{N'}$  и  $\beta(T_N) = T_{N'}$ ;

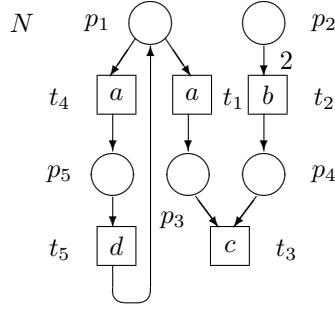


Рис. 1.1: Пример помеченной сети

2.  $\forall p \in P_N \forall t \in T_N F_N(p, t) = F_{N'}(\beta(p), \beta(t))$  and  $F_N(t, p) = F_{N'}(\beta(t), \beta(p))$ ;
3.  $\forall t \in T_N l_N(t) = l_{N'}(\beta(t))$ .

Помеченные сети  $N$  и  $N'$  *изоморфны*, обозначение  $N \simeq N'$ , если  $\exists \beta : N \simeq N'$ .

Пусть  $N = \langle P_N, T_N, F_N, l_N \rangle$  — помеченная сеть и  $X \subseteq P_N \cup T_N$ . *Ограничение*  $N$  на множество  $X$  — помеченная сеть  $N|_X = \langle P_N \cap X, T_N \cap X, F_N|_{(X \times X)}, l_N|_{(T_N \cap X)} \rangle$ .  $N'$  — *подсеть*  $N$ , обозначение  $N' \subseteq N$ , если  $\exists X \subseteq P_N \cup T_N N' = N|_X$ .

Пусть дана помеченная сеть  $N = \langle P_N, T_N, F_N, l_N \rangle$  и некоторый переход  $t \in T_N$ . *Предусловие* и *постусловие*  $t$ , обозначаемые соответственно  $\bullet t$  и  $t^\bullet$ , — это мультимножества, которые определяются следующим образом:  $(\bullet t)(p) = F_N(p, t)$  и  $(t^\bullet)(p) = F_N(t, p)$ . Аналогичные понятия введем для мест:  $(\bullet p)(t) = F_N(t, p)$  и  $(p^\bullet)(t) = F_N(p, t)$ . Введем также следующие обозначения:  ${}^\circ N = \{p \in P_N \mid p^\bullet = \emptyset\}$  — множество *начальных* (*входных*) мест  $N$  и  $N^\circ = \{p \in P_N \mid p^\bullet = \emptyset\}$  — множество *конечных* (*выходных*) мест  $N$ .

Переход  $t$  *нестабильный*, если  $\bullet t = \emptyset$ . Помеченная сеть  $N$  — *стабильная*, если в ней нет нестабильных переходов. В дальнейшем будем рассматривать только стабильные сети.

Помеченная сеть  $N$  — *без петель*, если  $\forall t \in T_N \bullet t \cap t^\bullet = \emptyset$ .

*Цикл* — множество переходов  $\{t_0, \dots, t_n\} \subseteq T_N$  такое, что  $t_{i-1}^\bullet \cap \bullet t_i \neq \emptyset$  ( $1 \leq i \leq n$ ) и  $t_0 = t_n$ . Помеченная сеть  $N$  — *ациклическая*, если в ней нет циклов.

Помеченная сеть  $N$  — *ординарная*, если  $\forall t \in T_N \bullet t$  и  $t^\bullet$  — обычные множества (а не мультимножества).

Помеченная сеть — *конечная*, если  $|P_N \cup T_N| < \infty$ .

Пусть  $N = \langle P_N, T_N, F_N, l_N \rangle$  — ациклическая ординарная помеченная сеть и  $x, y \in P_N \cup T_N$ . Введем следующие понятия.

- $x \prec_N y \Leftrightarrow x F_N^* y$ , где  $F_N^*$  — транзитивное замыкание отношения  $F_N$  (отношение *строгой причинной зависимости*);
- $x \preceq_N y \Leftrightarrow (x \prec_N y) \vee (x = y)$  (отношение *причинной зависимости*);
- $x \diamond_N y \Leftrightarrow (x \prec_N y) \vee (y \prec_N x)$  (отношение *линейной упорядоченности*);
- $x \#_N y \Leftrightarrow \exists t, u \in T_N (t \neq u, \bullet t \cap \bullet u \neq \emptyset, t \preceq_N x, u \preceq_N y)$  (отношение *конфликта*);
- $x \smile_N y \Leftrightarrow \neg(x \diamond_N y \vee x \#_N y \vee x = y)$  (отношение *причинной независимости* (*параллельности*));
- $\downarrow_N x = \{y \in P_N \cup T_N \mid y \prec_N x\}$  (множество *строгих предшественников*  $x$ );
- $\Downarrow_N x = \{y \in P_N \cup T_N \mid y \preceq_N x\}$  (множество *предшественников*  $x$ ).

Пусть  $T \subseteq T_N$ . Множество  $T$  *замкнуто влево* в  $N$ , если  $\forall t \in T (\downarrow_N t) \cap T_N \subseteq T$ . Множество  $T$  *безконфликтно* в  $N$ , если  $\forall t, u \in T \neg(t \#_N u)$ . Множество  $T$  — *конфигурация* в  $N$ , если оно конечно, замкнуто влево и безконфликтно в  $N$ .

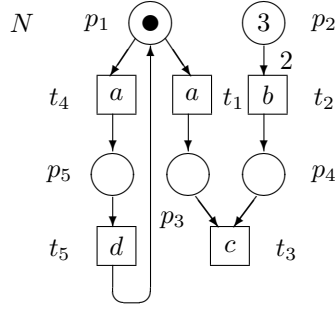


Рис. 1.2: Пример маркированной сети

### 1.1.3 Маркированные сети

Маркированные сети — помеченные сети, содержащие активные элементы, называемые “фишками” или “маркерами”, в своих местах, и эти места считаются “маркированными”. Функционированию маркированной сети соответствует перемещение маркеров по особым правилам “игры в фишки”.

Пусть  $N$  — помеченная сеть. *Маркировка*  $N$  есть мультимножество  $M \in \mathcal{M}(P_N)$ .

**Определение 1.1.3** Маркированная сеть — *пятерка*  $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$ , где:

- $\langle P_N, T_N, F_N, l_N \rangle$  — помеченная сеть;
- $M_N \in \mathcal{M}(P_N)$  — начальная маркировка.

**Пример 1.1.3** На рисунке 1.2 дано графическое представление маркированной сети. Маркировка изображается вписанными в места черными точками — “фишками”. Число фишек, большее единицы, записывается соответствующей цифрой.

Далее будем писать “сеть” вместо “маркированная сеть”.

Даны сети  $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$  и  $N' = \langle P_{N'}, T_{N'}, F_{N'}, l_{N'}, M_{N'} \rangle$ . Отображение  $\beta : P_N \cup T_N \rightarrow P_{N'} \cup T_{N'}$  — *изоморфизм* между  $N$  и  $N'$ , обозначение  $\beta : N \simeq N'$ , если:

1.  $\beta : \langle P_N, T_N, F_N, l_N \rangle \simeq \langle P_{N'}, T_{N'}, F_{N'}, l_{N'} \rangle$ ;
2.  $\forall p \in P_N M_N(p) = M_{N'}(\beta(p))$ .

Сети  $N$  и  $N'$  *изоморфны*, обозначение  $N \simeq N'$ , если  $\exists \beta : N \simeq N'$ .

Пусть  $M \in \mathcal{M}(P_N)$  — маркировка сети  $N$ . Переход  $t \in T_N$  *допустим* в  $M$ , запись  $M \xrightarrow{t}$ , если  $\bullet t \subseteq M$ . Если  $t$  допустим в  $M$ , то срабатывание этого перехода изменяет маркировку  $M$  на  $\widetilde{M} = M - \bullet t + t \bullet$ , запись  $M \xrightarrow{t} \widetilde{M}$  или  $M \xrightarrow{a} \widetilde{M}$ , если  $l_N(t) = a$ . Пишем  $M \rightarrow \widetilde{M}$ , если  $M \xrightarrow{t} \widetilde{M}$  для некоторого  $t$ . Маркировка  $\widetilde{M}$  сети  $N$  *достижима* из маркировки  $M$  этой сети, если:

1.  $\widetilde{M} = M$  или
2. существуют достижимая из  $M$  маркировка  $\widehat{M}$  сети  $N$  такая, что  $\widehat{M} \rightarrow \widetilde{M}$ .

Маркировка  $M$  сети  $N$  *достижимая*, если она достижима из  $M_N$ . Обозначим через  $Mark(N, M)$  множество всех достижимых из  $M$  маркировок сети  $N$  и через  $Mark(N)$  — множество всех достижимых маркировок сети  $N$ .

Пусть  $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$  — некоторая сеть.

Сеть  $N$  *безопасная*, если  $\forall M \in Mark(N) \forall p \in P_N M(p) \leq 1$ , то есть при любой достижимой маркировке ни в одном месте сети не содержится более одной фишки.

Действие  $a \in Act$  *автопараллельно* в  $N$ , если  $\exists M \in Mark(N) \exists t, u \in T_N l_N(t) = l_N(u) = a$  и  $\bullet t + \bullet u \subseteq M$ . Сеть  $N$  *без автопараллелизма*, если ни одно из действий не автопараллельно в  $N$ .

Действие  $a \in Act$  *самопараллельно* в  $N$ , если  $\exists M \in Mark(N) \exists t \in T_N l_N(t) = a$  и  $2 \bullet t \subseteq M$ . Сеть  $N$  *без самопараллелизма*, если ни одно из действий не самопараллельно в  $N$ .

### 1.1.4 Частично упорядоченные множества

Частично упорядоченные множества, рассмотренные в [242], — важный формализм, часто используемый в качестве семантической области для параллельных систем. Они позволяют представлять причинно-следственные связи событий моделируемой системы. Параллелизм трактуется как причинная независимость.

**Определение 1.1.4** Помеченное частично упорядоченное множество (ПЧУМ) — тройка  $\rho = \langle X, \prec, l \rangle$ , где:

- $X = \{x, y, \dots\}$  — множество событий;
- $\prec \subseteq X \times X$  — строгий частичный порядок, отношение причинной зависимости;
- $l : X \rightarrow Act$  — функция пометки.

**Пример 1.1.4** На рисунке 1.3(a) дано графическое представление ПЧУМ. События изображаются соответствующими символами, а помечающие их действия помещены в скобки. Отношение предшествования (только первичное, а не полученное в результате транзитивного замыкания) изображается направленными дугами.

ПЧУМ  $\rho = \langle X, \prec, l \rangle$  конечно, если  $|X| < \infty$ . Пустое ПЧУМ — тройка  $\emptyset = \langle \emptyset, \emptyset, \emptyset \rangle$ .

Пусть  $\rho = \langle X, \prec, l \rangle$  — ПЧУМ и  $Y \subseteq X$ . Ограничение ПЧУМ  $\rho$  на множество  $Y$  — ПЧУМ  $\rho|_Y = \langle Y, \prec \cap (Y \times Y), l|_Y \rangle$ .

Пусть  $x \in X$ . Тогда  $\downarrow x = \{y \in X \mid y \prec x\}$  — множество строгих предшественников  $x$ . Множество  $Y \subseteq X$  замкнуто влево, если  $\forall y \in Y \downarrow y \subseteq Y$ . ПЧУМ  $\rho_1$  — строгий префикс ЧУМ  $\rho_2$ , обозначение  $\rho_1 \triangleleft \rho_2$ , если  $\rho_1 = \rho_2|_Y$ , где  $Y \subset X$  — конечное замкнутое влево множество. ПЧУМ  $\rho_1$  — префикс ПЧУМ  $\rho_2$ , обозначение  $\rho_1 \trianglelefteq \rho_2$ , если  $\rho_1 \triangleleft \rho_2$  или  $\rho_1 = \rho_2$ .

Пусть  $\rho = \langle X, \prec, l \rangle$  и  $\rho' = \langle X', \prec', l' \rangle$  — ПЧУМ.

Отображение  $\beta : X \rightarrow X'$  — сохраняющая пометку биекция между  $\rho$  и  $\rho'$ , обозначение  $\beta : \rho \simeq \rho'$ , если:

1.  $\beta$  — биекция;
2.  $\forall x \in X \ l(x) = l'(\beta(x))$ .

Пишем  $\rho \simeq \rho'$ , если  $\exists \beta : \rho \simeq \rho'$ .

Отображение  $\beta : X \rightarrow X'$  — гомоморфизм между  $\rho$  и  $\rho'$ , обозначение  $\beta : \rho \sqsubseteq \rho'$ , если:

1.  $\beta : \rho \simeq \rho'$ ;
2.  $\forall x, y \in X \ x \prec y \Rightarrow \beta(x) \prec' \beta(y)$ .

Пишем  $\rho \sqsubseteq \rho'$ , если  $\exists \beta : \rho \sqsubseteq \rho'$ .

Отображение  $\beta : X \rightarrow X'$  — изоморфизм между  $\rho$  и  $\rho'$ , обозначение  $\beta : \rho \simeq \rho'$ , если  $\beta : \rho \sqsubseteq \rho'$  и  $\beta^{-1} : \rho' \sqsubseteq \rho$ . ПЧУМ  $\rho$  и  $\rho'$  изоморфны, запись  $\rho \simeq \rho'$ , если  $\exists \beta : \rho \simeq \rho'$ .

**Определение 1.1.5** Частично упорядоченное мультимножество (ЧУММ) — класс изоморфизма ПЧУМ.

**Пример 1.1.5** На рисунке 1.3(b) дано графическое представление ЧУММ. В отличие от ПЧУМ, не изображены события, так как мы абстрагируемся от них при изоморфизме.

**Определение 1.1.6** Частично упорядоченное множество (ЧУМ) — ПЧУМ  $\rho = \langle X, \prec, l \rangle$  такое, что его помечающая функция  $l$  инъективна, то есть  $\forall x, y \in X \ x \neq y \Rightarrow l(x) \neq l(y)$ .

Заметим, что в ЧУМ  $\rho = \langle X, \prec, l \rangle$  элементы множества  $X$  можно идентифицировать с действиями, помечающими их. Следовательно, из определения ЧУМ можно убрать функцию пометки и рассматривать ЧУМ как пару  $\rho = \langle X, \prec \rangle$ .

**Пример 1.1.6** На рисунке 1.3(c) дано графическое представление ЧУМ. В отличие от ПЧУМ, не изображены действия, так как можно считать, что они совпадают с соответствующими событиями.

Все понятия, введенные выше на ПЧУМ, стандартно переносятся на ЧУМ.

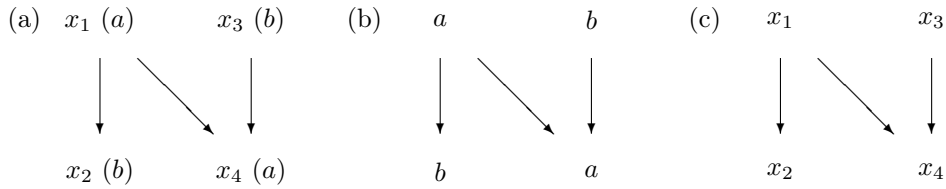


Рис. 1.3: Примеры ПЧУМ, ЧУММ и ЧУМ

### 1.1.5 Структуры событий

Структуры событий, введенные в [215, 287, 290, 291], позволяют представлять не только причинно-следственные связи событий, как ЧУМ, но и конфликт, то есть ситуацию, когда одно событие исключает другое.

**Определение 1.1.7** Помеченная структура событий (ПСС) — четверка  $\xi = \langle X, \prec, \#, l \rangle$ , где:

- $X = \{x, y, \dots\}$  — множество событий;
- $\prec \subseteq X \times X$  — строгий частичный порядок, отношение причинной зависимости, удовлетворяющее принципу конечности причин:  $\forall x \in X \mid \downarrow x \mid < \infty$ ;
- $\# \subseteq X \times X$  — иррефлексивное симметричное отношение конфликта, удовлетворяющее принципу наследования конфликта:  $\forall x, y, z \in X \ x \# y \prec z \Rightarrow x \# z$ ;
- $l : X \rightarrow Act$  — функция пометки.

**Пример 1.1.7** На рисунке 1.4(a) дано графическое представление ПСС. Отношение конфликта (только первичное, а не полученное в результате наследования) изображается пунктирными линиями.

ПСС  $\xi = \langle X, \prec, \#, l \rangle$  конечна, если  $|X| < \infty$ . Пустая ПСС — четверка  $\emptyset = \langle \emptyset, \emptyset, \emptyset, \emptyset \rangle$ .

Пусть  $\xi = \langle X, \prec, \#, l \rangle$  — ПСС и  $Y \subseteq X$ . Ограничение ПСС  $\xi$  на множество  $Y$  — ПСС  $\xi|_Y = \langle Y, \prec \cap (Y \times Y), \# \cap (Y \times Y), l|_Y \rangle$ .

Множество  $Y \subseteq X$  замкнуто влево в  $\xi$ , если  $\forall x \in Y \ \downarrow x \subseteq Y$ . Множество  $Y \subseteq X$  безконфликтно в  $\xi$ , если  $\forall x, y \in Y \ \neg(x \# y)$ . Конфигурация ПСС  $\xi$  — конечное замкнутое влево безконфликтное множество  $Y \subseteq X$ . Вычисление ПСС  $\xi$  — ПЧУМ  $\rho = \langle Y, \prec \cap (Y \times Y), l|_Y \rangle$ , где  $Y$  — конфигурация  $\xi$ .

Пусть  $\xi = \langle X, \prec, \#, l \rangle$  и  $\xi' = \langle X', \prec', \#', l' \rangle$  — ПСС.

Отображение  $\beta : X \rightarrow X'$  — изоморфизм между  $\xi$  и  $\xi'$ , обозначение  $\beta : \xi \simeq \xi'$ , если выполняются следующие условия.

1.  $\beta$  — биекция;
2.  $\forall x \in X \ l(x) = l'(\beta(x))$ ;
3.  $\forall x, y \in X \ x \prec y \Leftrightarrow \beta(x) \prec' \beta(y)$ ;
4.  $\forall x, y \in X \ x \# y \Leftrightarrow \beta(x) \# \beta(y)$ .

Пишем  $\xi \simeq \xi'$ , если  $\exists \beta : \xi \simeq \xi'$ .

**Определение 1.1.8** Мультиструктура событий (МСС) — класс изоморфизма ПСС.

**Пример 1.1.8** На рисунке 1.4(b) дано графическое представление МСС. В отличие от ПСС, не изображены события, так как мы абстрагируемся от них при изоморфизме.

**Определение 1.1.9** Структура событий (СС) — ПСС  $\xi = \langle X, \prec, \#, l \rangle$  такая, что ее помечающая функция  $l$  инъективна, то есть  $\forall x, y \in X \ x \neq y \Rightarrow l(x) \neq l(y)$ .

Заметим, что в СС  $\xi = \langle X, \prec, \#, l \rangle$  элементы множества  $X$  можно идентифицировать с действиями, помечающими их. Следовательно, из определения СС можно убрать функцию пометки и рассматривать СС как тройку  $\xi = \langle X, \prec, \# \rangle$ .

**Пример 1.1.9** На рисунке 1.4(c) дано графическое представление СС. В отличие от ПСС, не изображены действия, так как можно считать, что они совпадают с соответствующими событиями.

Все понятия, введенные выше на ПСС, стандартно переносятся на СС.

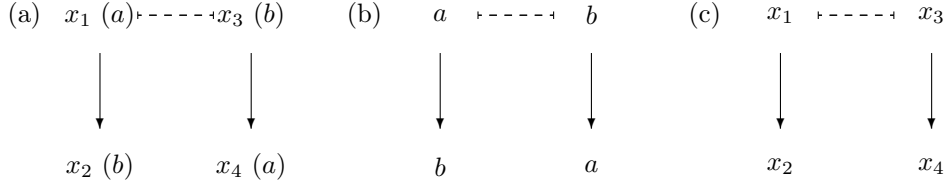


Рис. 1.4: Примеры ПСС, МСС и СС

### 1.1.6 С-процессы

С-процессы (процессы с С-сетями, causal nets) были введены в [41, 53, 157, 158]. С-процессы являются детерминированными процессами, так как в ходе выполнения С-процесса не происходит недетерминированного выбора между несколькими альтернативными действиями.

**Определение 1.1.10** С-сеть — ациклическая ординарная помеченная сеть  $C = \langle P_C, T_C, F_C, l_C \rangle$  со следующими свойствами.

1.  $\forall r \in P_C \ |\bullet r| \leq 1$  и  $|r\bullet| \leq 1$ , то есть нет ни прямых, ни обратных конфликтов;
2.  $\forall x \in P_C \cup T_C \ |\downarrow_C x| < \infty$ , то есть множество причин конечно.

Заметим, что любой С-сети  $C = \langle P_C, T_C, F_C, l_C \rangle$  можно сопоставить ПЧУМ  $\rho_C = \langle T_C, \prec_C \cap (T_C \times T_C), l_C \rangle$ .

Известно [23, 24] фундаментальное свойство С-сетей: для С-сети  $C$  существует последовательность срабатываний переходов  ${}^\circ C = L_0 \xrightarrow{v_1} \dots \xrightarrow{v_n} L_n = C^\circ$  такая, что  $L_i \subseteq P_C$  ( $0 \leq i \leq n$ ),  $P_C = \cup_{i=0}^n L_i$  и  $T_C = \{v_1, \dots, v_n\}$ . Такую последовательность назовем *полным выполнением*  $C$ .

**Определение 1.1.11** Даны сеть  $N$  и С-сеть  $C$ . Отображение  $\varphi : P_C \cup T_C \rightarrow P_N \cup T_N$  — встраивание  $C$  в  $N$ , обозначение  $\varphi : C \rightarrow N$ , если:

1.  $\varphi(P_C) \in \mathcal{M}(P_N)$  и  $\varphi(T_C) \in \mathcal{M}(T_N)$ , то есть сохраняются сорта;
2.  $\forall v \in T_C \ \bullet\varphi(v) = \varphi(\bullet v)$  и  $\varphi(v)\bullet = \varphi(v\bullet)$ , то есть учитывается отношение инцидентности;
3.  $\forall v \in T_C \ l_C(v) = l_N(\varphi(v))$ , то есть сохраняется пометка.

Так как встраивание учитывает отношение инцидентности, имеет место следующий факт. Если  ${}^\circ C \xrightarrow{v_1} \dots \xrightarrow{v_n} C^\circ$  — полное выполнение  $C$ , то  $M = \varphi({}^\circ C) \xrightarrow{\varphi(v_1)} \dots \xrightarrow{\varphi(v_n)} \varphi(C^\circ) = \widetilde{M}$  — последовательность срабатываний переходов в  $N$ , соответствующая этому полному выполнению, запись  $M \xrightarrow{C, \varphi} \widetilde{M}$ . Обратно, для любой последовательности срабатываний  $M \xrightarrow{t_1} \dots \xrightarrow{t_n} \widetilde{M}$  сети  $N$  существуют С-сеть  $C$  и встраивание  $\varphi : C \rightarrow N$  такие, что  $M = \varphi({}^\circ C)$ ,  $\widetilde{M} = \varphi(C^\circ)$ ,  $t_i = \varphi(v_i)$  ( $0 \leq i \leq n$ ) и  ${}^\circ C \xrightarrow{v_1} \dots \xrightarrow{v_n} C^\circ$  — полное выполнение  $C$ .

**Определение 1.1.12** С-процесс (процесс), допустимый в маркировке  $M$  сети  $N$  — пара  $\pi = (C, \varphi)$ , где  $C$  — С-сеть и  $\varphi : C \rightarrow N$  — встраивание такое, что  $M = \varphi({}^\circ C)$ . Допустимый в  $M_N$  процесс — процесс  $N$ .

Обозначим множество всех процессов, допустимых в маркировке  $M$  сети  $N$  через  $\Pi(N, M)$  и множество всех процессов сети  $N$  через  $\Pi(N)$ .

Если  $\pi \in \Pi(N, M)$ , то срабатывание этого процесса меняет маркировку  $M$  на  $\widetilde{M} = M - \varphi({}^\circ C) + \varphi(C^\circ) = \varphi(C^\circ)$ , обозначение  $M \xrightarrow{\pi} \widetilde{M}$ . Тогда  $\varphi(C^\circ)$  — маркировка, соответствующая процессу  $\pi$ . Процессы и достижимые маркировки сети  $N$  связаны следующим соотношением:  $Mark(N, M) = \{\varphi(C^\circ) \mid \pi = (C, \varphi) \in \Pi(N, M)\}$ .

*Начальный* процесс сети  $N$  — это процесс  $\pi_N = (C_N, \varphi_N) \in \Pi(N)$  такой, что  $T_{C_N} = \emptyset$ .

Пусть  $\pi = (C, \varphi) \in \Pi(N)$  и  $\pi' = (C', \varphi') \in \Pi(N')$ . Отображение  $\beta : P_C \cup T_C \rightarrow P_{C'} \cup T_{C'}$  — изоморфизм между  $\pi$  и  $\pi'$ , обозначение  $\beta : \pi \simeq \pi'$ , если:

1.  $\beta : C \simeq C'$ ;



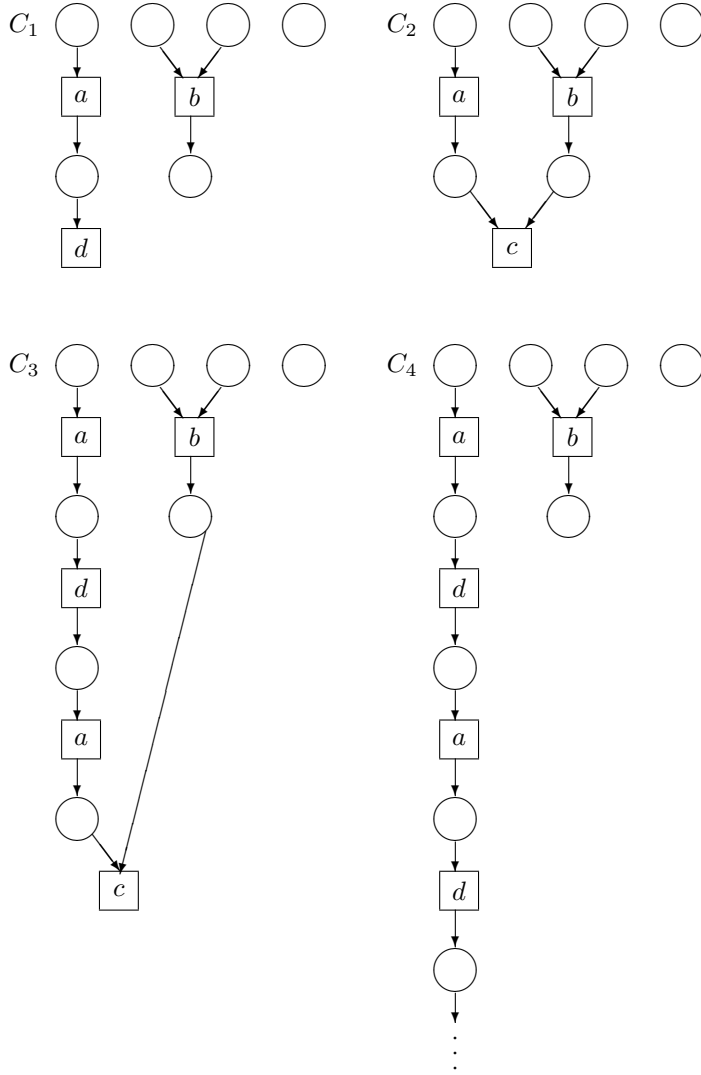


Рис. 1.5: C-сети процессов сети на рисунке 1.2

$$2. \forall x \in P_C \cup T_C \varphi(x) = \varphi'(\beta(x)).$$

Процессы  $\pi$  и  $\pi'$  *изоморфны*, обозначение  $\pi \simeq \pi'$ , если  $\exists \beta : \pi \simeq \pi'$ .

Пусть  $\pi = (C, \varphi)$ ,  $\tilde{\pi} = (\tilde{C}, \tilde{\varphi}) \in \Pi(N)$ ,  $\hat{\pi} = (\hat{C}, \hat{\varphi}) \in \Pi(N, \varphi(C^\circ))$ . Процесс  $\pi$  — *префикс* процесса  $\tilde{\pi}$ , если  $T_C \subseteq T_{\tilde{C}}$  — замкнутое влево множество в  $\tilde{C}$ . Процесс  $\hat{\pi}$  — *суффикс* процесса  $\tilde{\pi}$ , если  $T_{\hat{C}} = T_{\tilde{C}} \setminus T_C$ . Тогда  $\tilde{\pi}$  — *расширение*  $\pi$  на процесс  $\hat{\pi}$ , а  $\hat{\pi}$  — *расширяющий* процесс для  $\pi$ , запись  $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$ . Пишем  $\pi \rightarrow \tilde{\pi}$ , если  $\exists \hat{\pi} \pi \xrightarrow{\hat{\pi}} \tilde{\pi}$ .

Заметим, что для любого  $\pi \in \Pi(N)$   $\pi_N \xrightarrow{\pi} \pi$ .

Процесс  $\pi$  сети  $N$  — *максимальный*, если  $\forall \tilde{\pi} = (\tilde{C}, \tilde{\varphi}) \in \Pi(N)$  таких, что  $\pi \rightarrow \tilde{\pi}$  выполняется  $T_{\tilde{C}} \setminus T_C = \emptyset$ . Множество всех максимальных процессов сети  $N$  обозначим через  $\Pi_{max}(N)$ .

**Пример 1.1.10** На рисунке 1.5 представлены C-сети процессов сети, изображенной рисунком 1.2. Процесс с C-сетью  $C_1$  — не максимальный, так как его можно расширить (например, до процессов с C-сетями  $C_3$  и  $C_4$ ). Процессы с C-сетями  $C_2$  и  $C_3$  — максимальные, а процесс с C-сетью  $C_4$  — бесконечный.

**Пример 1.1.11** На рисунке 1.6 представлены дополнительные примеры C-сетей максимальных процессов.

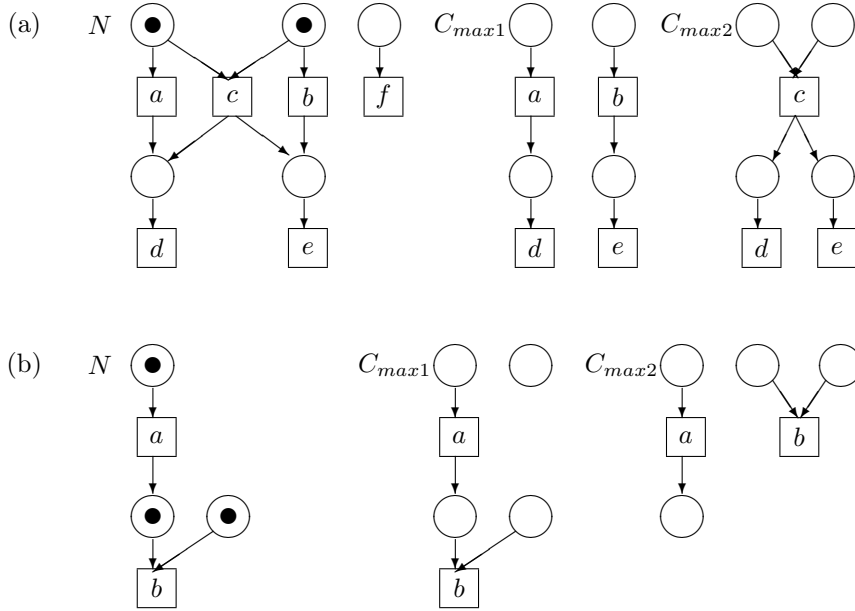


Рис. 1.6: С-сети максимальных процессов

Процесс  $\tilde{\pi}$  — расширение процесса  $\pi$  на одно действие, запись  $\pi \xrightarrow{v} \tilde{\pi}$  или  $\pi \xrightarrow{a} \tilde{\pi}$ , если  $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$ ,  $T_{\hat{C}} = \{v\}$  и  $l_{\hat{C}}(v) = a$ .

Таким образом, полное выполнение  $C$  в “контексте” сети  $N$  можно записать с помощью процессов как  $\pi_0 \xrightarrow{v_1} \dots \xrightarrow{v_n} \pi_n$ , где  $\pi_i = (C_i, \varphi_i)$ ,  $C_i^\circ = L_i$ ,  $(0 \leq i \leq n)$  и  ${}^\circ C = L_0 \xrightarrow{v_1} \dots \xrightarrow{v_n} L_n = C^\circ$  — полное выполнение  $C$ .

Процесс  $\tilde{\pi}$  — расширение процесса  $\pi$  на последовательность действий, запись  $\pi \xrightarrow{\sigma} \tilde{\pi}$  или  $\pi \xrightarrow{\omega} \tilde{\pi}$ , если  $\exists \pi_i \in \Pi(N)$ ,  $\exists v_i \in T_{\hat{C}} (1 \leq i \leq n)$   $\pi \xrightarrow{v_1} \pi_1 \xrightarrow{v_2} \dots \xrightarrow{v_n} \pi_n = \tilde{\pi}$ ,  $\sigma = v_1 \dots v_n$  и  $l_{\hat{C}}(\sigma) = \omega$ .

Процесс  $\tilde{\pi}$  — расширение  $\pi$  на множество действий или шаг, запись  $\pi \xrightarrow{V} \tilde{\pi}$  или  $\pi \xrightarrow{A} \tilde{\pi}$ , если  $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$ ,  $\prec_{\hat{C}} = \emptyset$ ,  $T_{\hat{C}} = V$  и  $l_{\hat{C}}(V) = A$ .

### 1.1.7 О-процессы

О-процессы (процессы с О-сетями, occurrence nets [159, 215, 233]) были введены в [134, 176] под именем “ветвистых” процессов. О-процессы являются недетерминированными процессами, так как во время выполнения О-процесса может происходить недетерминированный выбор между несколькими альтернативными действиями.

Заметим, что в [216] определена стандартная конструкция развертки сетей в О-сети, функция встраивания для определения развертки была введена в [154, 159], а процессы на основе “декорированных” О-сетей были исследованы в [205, 206, 207, 252, 208]. В [170] содержатся последние результаты по определению семантики сетей Петри на основе структур событий.

**Определение 1.1.13** О-сеть — ациклическая ординарная помеченная сеть  $O = \langle P_O, T_O, F_O, l_O \rangle$ , для которой выполняются следующие условия.

1.  $\forall r \in P_O \ | \bullet r | \leq 1$ , то есть нет обратных конфликтов;
2.  $\forall x \in P_O \cup T_O \ | \downarrow_O x | < \infty$ , то есть множество причин конечно;
3.  $\forall x \in P_O \cup T_O \ \neg(x \#_O x)$ , то есть отношение конфликта иррефлексивно.

Заметим, что любой О-сети  $O = \langle P_O, T_O, F_O, l_O \rangle$  можно сопоставить ПСС  $\xi_O = \langle T_O, \prec_O \cap (T_O \times T_O), \#_O \cap (T_O \times T_O), l_O \rangle$ .

Пусть  $O = \langle P_O, T_O, F_O, l_O \rangle$  — О-сеть и  $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$  — сеть.

**Определение 1.1.14** *Отображение  $\psi : P_O \cup T_O \rightarrow P_N \cup T_N$  — встраивание  $O$  в  $N$ , запись  $\psi : O \rightarrow N$ , если выполняются следующие условия.*

1.  $\psi(P_O) \in \mathcal{M}(P_N)$  и  $\psi(T_O) \in \mathcal{M}(T_N)$ , то есть сохраняются сорта;
2.  $\forall v \in T_O l_O(v) = l_N(\psi(v))$ , то есть сохраняется пометка;
3.  $\forall v \in T_O \bullet\psi(v) = \psi(\bullet v)$  и  $\psi(v)\bullet = \psi(v\bullet)$ , то есть учитывается отношение инцидентности;
4.  $\forall v, w \in T_O (\bullet v = \bullet w) \wedge (\psi(v) = \psi(w)) \Rightarrow v = w$ , то есть нет “миших” конфликтов.

**Определение 1.1.15** *О-процесс (ветвистый процесс) сети  $N$  — пара  $\varpi = (O, \psi)$ , где  $O$  — О-сеть, а  $\psi : O \rightarrow N$  — встраивание такое, что  $M_N = \psi(\circ O)$ .*

Обозначим множество всех О-процессов сети  $N$  через  $\wp(N)$ . Начальный О-процесс сети  $N$  совпадает с начальным С-процессом, то есть  $\varpi_N = \pi_N$ .

Определение изоморфизма О-процессов аналогично соответствующему определению для С-процессов.

Пусть  $\varpi = (O, \psi)$ ,  $\tilde{\varpi} = (\tilde{O}, \tilde{\psi}) \in \wp(N)$ . О-процесс  $\varpi$  — префикс О-процесса  $\tilde{\varpi}$ , если  $T_O \subseteq T_{\tilde{O}}$  — замкнутое влево множество в  $\tilde{O}$ . Тогда  $\tilde{\varpi}$  — расширение О-процесса  $\varpi$ , обозначение  $\varpi \rightarrow \tilde{\varpi}$ .

Пусть  $\pi = (C, \varphi) \in \Pi(N)$ ,  $\varpi = (O, \psi) \in \wp(N)$ . С-процесс  $\varpi$  — вычисление О-процесса  $\tilde{\varpi}$ , если  $T_C \subseteq T_O$  — конфигурация в  $O$ .

Заметим, что  $\forall \varpi \in \wp(N) \varpi_N \xrightarrow{\varpi} \varpi$ .

О-процесс  $\varpi$  сети  $N$  — максимальный, если  $\forall \tilde{\varpi} = (\tilde{O}, \tilde{\psi}) \in \wp(N)$  таких, что  $\varpi \rightarrow \tilde{\varpi}$  выполняется  $T_{\tilde{O}} \setminus T_O = \emptyset$ .

Обозначим множество всех максимальных О-процессов сети  $N$  через  $\wp_{max}(N)$ . Заметим, что  $\wp_{max}(N)$  состоит из единственного О-процесса вида  $\varpi_{max} = (O_{max}, \psi_{max})$ . В этом случае класс изоморфизма О-сети  $O_{max}$  — развертка сети  $N$ , обозначение  $\mathcal{U}(N)$ . Развертке  $\mathcal{U}(N)$  соответствует МСС  $\mathcal{E}(N) = \xi_{\mathcal{U}(N)}$  — класс изоморфизма ПСС  $\xi_O$  для  $O \in \mathcal{U}(N)$ .

**Пример 1.1.12** *На рисунке 1.7 представлены О-сети ветвистых процессов сети, изображенной рисунком 1.2. Ветвистый процесс с О-сетью  $O_1$  — не максимальный, так как его можно расширить (например, до процесса с О-сетью  $O_2$ ). Ветвистый процесс с О-сетью  $O_2$  — бесконечный.*

**Пример 1.1.13** *На рисунке 1.8 представлены дополнительные примеры О-сетей максимальных ветвистых процессов.*

## 1.2 Сети Петри с видимыми переходами

Рассмотрим эквивалентности для сетей Петри с видимыми переходами, моделирующих системы, все действия которых считаются видимыми для внешнего наблюдателя.

### 1.2.1 Базисные эквивалентности

Посмотрим, какие упомянутые в литературе базисные эквивалентности заполняют координатную плоскость на рисунке 0.1.

- Следовые эквивалентности (учитывают только протоколы функционирования систем).
  - Интерливинговая следовая эквивалентность ( $\equiv_i$ ) введена в [172, 173, 174] на языках. Определение этой эквивалентности дано в [220, 240, 245] на сетях Петри, в [94, 137, 138, 143, 148, 149, 267, 270] — на структурах событий и в [133] — на общих моделях параллельных систем, а также в [248].
  - Шаговая следовая эквивалентность ( $\equiv_s$ ). Определение на сетях Петри можно найти в [240, 245], а на структурах событий — в [137, 138, 143, 148, 149, 267, 270].
  - ЧС следовая эквивалентность ( $\equiv_{pw}$ ). Эта эквивалентность определена в [160] на ЧУММ и в [278] — на сетях Петри.

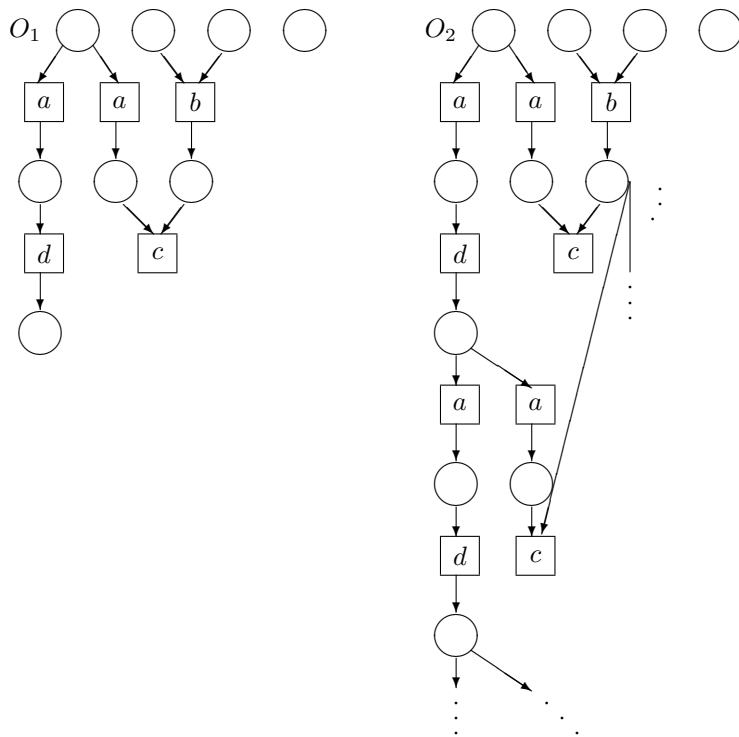


Рис. 1.7: O-сети ветвистых процессов сети на рисунке 1.2

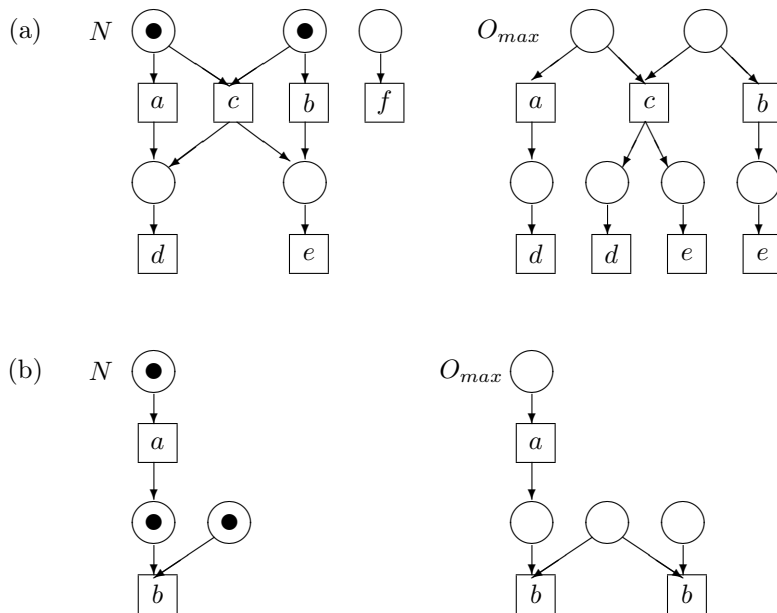


Рис. 1.8: O-сети максимальных ветвистых процессов

- ЧУММ следовая эквивалентность ( $\equiv_{pom}$ ). Семантика введена в [242, 243, 244]. Эквивалентность определена в [161, 245] для сетей Петри и в [137, 138, 143, 148, 149, 267, 270] для структур событий.
- *Обычные бисимуляционные эквивалентности* (учитывают ветвистую структуру функционирования систем).
  - Интерливинговая бисимуляционная эквивалентность ( $\leftrightarrow_i$ ) была введена в [227] на автоматах на основе наблюдаемой эквивалентности [198, 1]. Определение можно найти также в [228] для автоматов, в [8, 9, 23, 24, 33, 45, 46, 122, 161, 220, 245] для сетей Петри, в [94, 137, 138, 143, 148, 149, 267, 270, 275, 276, 280, 282] для структур событий и в [133] для общих моделей параллельных систем.
  - Шаговая бисимуляционная эквивалентность ( $\leftrightarrow_s$ ) введена в [218]. Определение на сетях Петри имеется в [8, 9, 23, 24, 35, 161, 240, 245], а на структурах событий — в [137, 138, 143, 148, 149, 275, 276, 280, 282].
  - ЧС бисимуляционная эквивалентность ( $\leftrightarrow_{pw}$ ) введена в [275, 276] на структурах событий. Определение для сетей Петри можно найти в [23, 24], а для структур событий — также в [280, 282].
  - ЧУММ бисимуляционная эквивалентность ( $\leftrightarrow_{pom}$ ) была введена в [34, 161]. Определения для сетей Петри даны в [8, 9, 23, 24, 33, 45, 46, 245], а для структур событий — в [137, 138, 143, 148, 149, 275, 276, 280, 282].
  - Процессная бисимуляционная эквивалентность ( $\leftrightarrow_{pr}$ ) введена в [23] на сетях Петри. Определение для сетей Петри есть также в [24].
- *ST-бисимуляционные эквивалентности* (учитывают продолжительность срабатывания действий в функционирования систем).
  - Интерливинговая ST-бисимуляционная эквивалентность ( $\leftrightarrow_{iST}$ ) была введена на сетях Петри в [161]. Определение для сетей Петри можно также найти в [33, 122], а для структур событий — в [148, 149, 275, 276, 280, 282].
  - ЧС ST-бисимуляционная эквивалентность ( $\leftrightarrow_{pwST}$ ) введена в [275, 276] на структурах событий. Это определение можно также найти в [280, 282].
  - ЧУММ ST-бисимуляционная эквивалентность ( $\leftrightarrow_{pomST}$ ) также введена в [275, 276] на структурах событий. Определение есть также в [280, 282].
- *Сохраняющие историю бисимуляционные эквивалентности* (учитывают “прошлое” или “историю” функционирования систем).
  - ЧУММ сохраняющая историю бисимуляционная эквивалентность ( $\leftrightarrow_{pomh}$ ) введена в [251] на поведенческих структурах под именем “бисимуляционная эквивалентность поведенческих структур”.  
 В [137] эта эквивалентность была определена на структурах событий и названа “сохраняющая историю бисимуляционная эквивалентность”.  
 В [116] в рамках структур событий была введена “наблюдаемая эквивалентность на причинных деревьях”, совпадение которой с сохраняющей историю бисимуляционной эквивалентностью было доказано в [268, 117, 118].  
 В [127] определена NMS- (Nondeterministic Measurement System) эквивалентность, также совпадающая с сохраняющей историю бисимуляционной эквивалентностью.  
 В [11, 237, 90] показано, что сохраняющая историю бисимуляционная эквивалентность совпадает с “эквивалентностью на смешанных порядках” для структур событий из [129].  
 В [102] была введена “прямая-обратная ЧУММ бисимуляционная эквивалентность” на структурах событий, также совпадающая с сохраняющей историю бисимуляционной эквивалентностью.  
 В [45] определение было введено на сетях Петри. В этой работе эквивалентности было дано имя “полностью параллельная бисимуляционная эквивалентность”.  
 Определение для сетей Петри можно также найти в [33, 46, 120, 121, 122, 139, 245], для структур событий — в [138, 140, 141, 142, 143, 148, 149, 275, 276, 277, 280, 282], а для поведенческих структур — в [265].

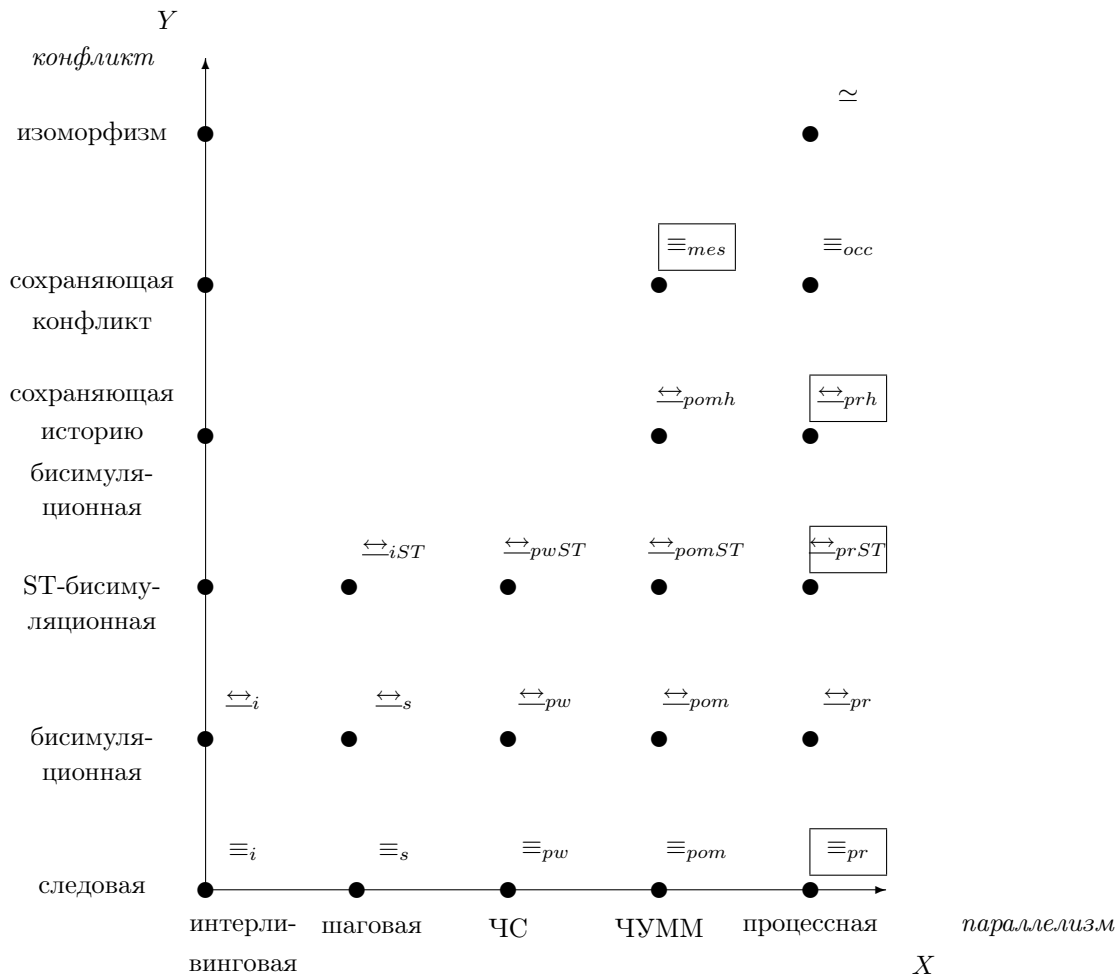


Рис. 1.9: Классификация базисных эквивалентностей

- *Сохраняющие конфликт эквивалентности* (полностью учитывают конфликт).
  - О-процессная сохраняющая конфликт эквивалентность ( $\equiv_{occ}$ ) была рассмотрена в [215, 161] на сетях Петри.
- *Изоморфизм* (совпадение систем с точностью до переименования их компонент,  $\simeq$ ) — введен в [232].

В работах [80, 150, 151, 152, 167, 266, 272, 273, 274, 278, 280, 281, 283] рассмотрена группа эквивалентностей, находящихся между следовыми и бисимуляционными эквивалентностями, то есть в пределах от семантики “линейного времени” до семантики “ветвистого времени” [153].

Распределенная бисимуляция вводится в [101, 169].

Эквивалентности, основанные на “местонахождениях”, изучены в [39, 40, 89, 187, 209, 114]. Обзор группы эквивалентностей, основанных на “местонахождениях” и “причинах”, дан в [90, 188]. Такие эквивалентности исследованы также в [90].

Обзоры эквивалентностей были сделаны также в [88, 131, 239]. Эквивалентности ЧУММ исследованы в [249].

В этом разделе мы переносим все базисные эквивалентности в рамки сетей Петри, а также вводим ряд новых базисных эквивалентностей, дополняющих уже известные в указанных семантиках. Это процессные следовая ( $\equiv_{pr}$ ), ST-бисимуляционная ( $\leftrightarrow_{prST}$ ), сохраняющая историю бисимуляционная ( $\leftrightarrow_{prh}$ ) и МСС сохраняющая конфликт ( $\equiv_{mes}$ ) эквивалентности. Они помещены в прямоугольники на рисунке 1.9.

Проводится сравнение всех упомянутых эквивалентностей, в результате чего получена диаграмма их взаимосвязей в виде ориентированного графа с дугами, соответствующими отношению “сильнее чем”.

Заметим, что исследование такого широкого набора базисных эквивалентностных понятий выполняется впервые и является важным результатом.

### Следовые эквивалентности

Следовые эквивалентности учитывают только протоколы функционирования сетей, и не принимают во внимание момент недетерминированного выбора между несколькими расширениями процесса. Поэтому их называют эквивалентностями *линейного времени*.

**Определение 1.2.1** Интерливинговый след *сети*  $N$  — это последовательность  $a_1 \cdots a_n \in Act^*$  такая, что  $\pi_N \xrightarrow{a_1} \pi_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} \pi_n$ , где  $\pi_i \in \Pi(N)$  ( $1 \leq i \leq n$ ). Обозначим множество всех интерливинговых следов *сети*  $N$  через  $IntTraces(N)$ . *Сети*  $N$  и  $N'$  интерливингово следово эквивалентны, запись  $N \equiv_i N'$ , если  $IntTraces(N) = IntTraces(N')$ .

**Определение 1.2.2** Шаговый след *сети*  $N$  — это последовательность  $A_1 \cdots A_n \in (\mathcal{M}(Act))^*$  такая, что  $\pi_N \xrightarrow{A_1} \pi_1 \xrightarrow{A_2} \dots \xrightarrow{A_n} \pi_n$ , где  $\pi_i \in \Pi(N)$  ( $1 \leq i \leq n$ ). Обозначим множество всех шаговых следов *сети*  $N$  через  $StepTraces(N)$ . *Сети*  $N$  и  $N'$  шагово следово эквивалентны, запись  $N \equiv_s N'$ , если  $StepTraces(N) = StepTraces(N')$ .

**Определение 1.2.3** ЧУММ-след *сети*  $N$  — это ЧУММ  $\rho$  — класс изоморфизма ПЧУМ  $\rho_C$  для  $\pi = (C, \varphi) \in \Pi(N)$ . Пишем  $\rho \sqsubseteq \rho'$ , если  $\rho_C \sqsubseteq \rho_{C'}$  для  $\rho_C \in \rho$  и  $\rho_{C'} \in \rho'$ . В этом случае говорим, что ЧУММ  $\rho$  менее последователен или более параллелен, чем  $\rho'$ . Обозначим через  $Pomsets(N)$  множество всех следов ЧУММ *сети*  $N$ . *Сети*  $N$  и  $N'$  ЧС следово эквивалентны, запись  $N \equiv_{pw} N'$ , если  $Pomsets(N) \sqsubseteq Pomsets(N')$  и  $Pomsets(N') \sqsubseteq Pomsets(N)$ , то есть для любого  $\rho' \in Pomsets(N')$  существует  $\rho \in Pomsets(N)$  такой, что  $\rho \sqsubseteq \rho'$  и наоборот.

**Определение 1.2.4** *Сети*  $N$  и  $N'$  ЧУММ следово эквивалентны, запись  $N \equiv_{pot} N'$ , если  $Pomsets(N) = Pomsets(N')$ .

**Определение 1.2.5** (C-)процессный след *сети*  $N$  — это класс изоморфизма  $C$  для  $\pi = (C, \varphi) \in \Pi(N)$ .  $ProcessNets(N)$  обозначает множество всех (C-)процессных следов *сети*  $N$ . *Сети*  $N$  и  $N'$  (C-)процессно следово эквивалентны, запись  $N \equiv_{pr} N'$ , если  $ProcessNets(N) = ProcessNets(N')$ .

### Бисимуляционные эквивалентности

Бисимуляция — это фундаментальная поведенческая эквивалентность. Чтобы две сети были бисимуляционно эквивалентны, должно существовать некоторое отношение  $\mathcal{R}$  (бисимуляция) на их состояниях такое, что:

- Начальные состояния обеих сетей связаны отношением  $\mathcal{R}$ .
- Если сети находятся в состояниях, связанных отношением  $\mathcal{R}$ , и одна из них перешла в новое состояние, то другая сеть может смоделировать поведение первой сети, также перейдя в новое состояние. При этом новые состояния сетей должны быть связаны отношением  $\mathcal{R}$ .

Состояниями сетей могут быть маркировки или процессы. Существуют и другие виды состояний (например, ST-процессы, которые мы рассмотрим далее).

Бисимуляционные эквивалентности учитывают момент недетерминированного выбора между несколькими расширениями процесса (“ветвления”). Поэтому их называют эквивалентностями *ветвистого времени*.

**Обычные бисимуляционные эквивалентности** Обычные бисимуляционные эквивалентности — самые слабые из рассматриваемых в данной работе эквивалентностных понятий, учитывающих ветвистую структуру поведения сетей.

**Определение 1.2.6** Пусть  $N$  и  $N'$  — некоторые сети. Отношение  $\mathcal{R} \subseteq \Pi(N) \times \Pi(N')$  —  $\star$ -бисимуляция между  $N$  и  $N'$ ,  $\star \in \{\text{интерливинговая, шаговая, ЧС, ЧУММ, процессная}\}$ , запись  $\mathcal{R} : N \leftrightarrow_{\star} N'$ ,  $\star \in \{i, s, pw, pot, pr\}$ , если:

1.  $(\pi_N, \pi_{N'}) \in \mathcal{R}$ .

2.  $(\pi, \pi') \in \mathcal{R}$ ,  $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$ ,

(a)  $|T_{\hat{C}}| = 1$ , если  $\star = i$ ;

(b)  $\prec_{\hat{C}} = \emptyset$ , если  $\star = s$ ;

$\Rightarrow \exists \tilde{\pi}' : \pi' \xrightarrow{\hat{\pi}'} \tilde{\pi}'$ ,  $(\tilde{\pi}, \tilde{\pi}') \in \mathcal{R}$  и

(a)  $\rho_{\hat{C}'} \sqsubseteq \rho_{\hat{C}}$ , если  $\star = pw$ ;

(b)  $\rho_{\hat{C}} \simeq \rho_{\hat{C}'}$ , если  $\star \in \{i, s, rot\}$ ;

(c)  $\hat{C} \simeq \hat{C}'$ , если  $\star = pr$ .

3. Как пункт 2, но роли  $N$  и  $N'$  меняются.

Сети  $N$  и  $N'$   $\star$ -бисимуляционно эквивалентны,  $\star \in \{\text{интерливингивая, шаговая, ЧС, ЧУММ, процессная}\}$ , запись  $N \leftrightarrow_{\star} N'$ , если  $\exists \mathcal{R} : N \leftrightarrow_{\star} N'$ ,  $\star \in \{i, s, pw, rot, pr\}$ .

**ST-бисимуляционные эквивалентности** ST-маркировки и ST-процессы вводятся для описания состояний сетей, в предположении, что переходы могут иметь внутреннюю структуру или срабатывают не мгновенно, а за некоторый промежуток времени.

**Определение 1.2.7** ST-маркировка сети  $N$  [33, 280] — пара  $(M, U)$ , где  $M \in \mathcal{M}(P_N)$  и  $U \in \mathcal{M}(T_N)$ .

Переходы из  $U$  считаются *работающими*. Начальная ST-маркировка сети  $N$  — пара  $(M_N, \emptyset)$ . Определим  $T_N^{\pm} = \{t^+, t^- \mid t \in T_N\}$  — множество частей переходов.  $t^+$  называется началом, а  $t^-$  — концом перехода  $t$ . Часть перехода  $q \in T_N^{\pm}$  допустима в ST-маркировке  $Q = (M, U)$ , запись  $Q \xrightarrow{q}$  если выполняется следующее:

1.  $M \xrightarrow{t}$ , если  $q = t^+$  или

2.  $t \in U$ , если  $q = t^-$ .

Если  $q$  допустима в  $M$ , то срабатывание этой части перехода изменяет маркировку  $Q$  на  $\tilde{Q}$  следующим образом:

1.  $\tilde{M} = M - \bullet t$  и  $\tilde{U} = U + t$ , если  $q = t^+$  или

2.  $\tilde{M} = M + t \bullet$  и  $\tilde{U} = U - t$ , если  $q = t^-$ ,

запись  $Q \xrightarrow{q} \tilde{Q}$ . Пишем  $Q \rightarrow \tilde{Q}$ , если  $Q \xrightarrow{q} \tilde{Q}$  для некоторой части перехода  $q$ . ST-маркировка  $\tilde{Q}$  сети  $N$  достижима из ST-маркировки  $Q$  этой сети, если:

1.  $\tilde{Q} = Q$  или

2. существуют достижимая из  $Q$  ST-маркировка  $\hat{Q}$  сети  $N$  такая, что  $\hat{Q} \rightarrow \tilde{Q}$ .

ST-маркировка  $Q$  сети  $N$  *достижимая*, если она достижима из  $M_N$ . Обозначим через  $ST - Mark(N)$  множество всех достижимых ST-маркировок сети  $N$ .

**Определение 1.2.8** ST-процесс сети  $N$  — пара  $(\pi_E, \pi_P)$  такая, что  $\pi_E, \pi_P \in \Pi(N)$ ,  $\pi_P \xrightarrow{\pi_W} \pi_E$  и  $\forall v, w \in T_{C_E} v \prec_{C_E} w \Rightarrow v \in T_{C_P}$ .

В этом случае  $\pi_E$  — процесс, начавший работать, то есть все действия  $\pi_E$  начали выполняться. Процесс  $\pi_P$  соответствует той части  $\pi_E$ , которая уже завершилась, а  $\pi_W$  — той части, которая еще работает. Очевидно, что  $\prec_{C_W} = \emptyset$ .  $ST - \Pi(N)$  обозначает множество всех ST-процессов сети  $N$ .

$(\pi_N, \pi_N)$  будет начальным ST-процессом  $N$ . Пусть  $(\pi_E, \pi_P), (\tilde{\pi}_E, \tilde{\pi}_P) \in ST - \Pi(N)$ . Пишем  $(\pi_E, \pi_P) \rightarrow (\tilde{\pi}_E, \tilde{\pi}_P)$ , если  $\pi_E \rightarrow \tilde{\pi}_E$  и  $\pi_P \rightarrow \tilde{\pi}_P$ .

ST-маркировки и ST-процессы сети  $N$  связаны следующим соотношением:  $ST - Mark(N) = \{(\varphi_E(C_E^{\circ} \cap C_P^{\circ}), \varphi_E(T_{C_E} \setminus T_{C_P})) \mid (\pi_E, \pi_P) \in ST - \Pi(N)\}$ .

ST-бисимуляционные эквивалентности учитывают длительность срабатывания переходов при функционировании сетей, в предположении, что они не являются мгновенными, а имеют начало и конец. Заметим, что в первом определении ST-бисимуляционной эквивалентности [161] считалось, что работающие в данный момент переходы забирают фишки из своих входных мест, но еще не выдают их в выходные места.



**Определение 1.2.9** Пусть  $N$  и  $N'$  — некоторые сети. Отношение  $\mathcal{R} \subseteq ST - \Pi(N) \times ST - \Pi(N') \times \mathcal{B}$ , где  $\mathcal{B} = \{\beta \mid \beta : T_C \rightarrow T_{C'}\}$ ,  $\pi = (C, \varphi) \in \Pi(N)$ ,  $\pi' = (C', \varphi') \in \Pi(N')$  —  $\star$ -ST-бисимуляция между  $N$  и  $N'$ ,  $\star \in \{\text{интерливинговая, ЧС, ЧУММ, процессная}\}$ , запись  $\mathcal{R} : N \leftrightarrow_{\star ST} N'$ ,  $\star \in \{i, pw, pot, pr\}$ , если:

1.  $((\pi_N, \pi_N), (\pi_{N'}, \pi_{N'}), \emptyset) \in \mathcal{R}$ .
2.  $((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{R} \Rightarrow \beta : \rho_{C_E} \simeq \rho_{C'_E}$  и  $\beta(T_{C_P}) = T_{C'_P}$ .
3.  $((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{R}$ ,  $(\pi_E, \pi_P) \rightarrow (\tilde{\pi}_E, \tilde{\pi}_P) \Rightarrow \exists \tilde{\beta}, (\tilde{\pi}'_E, \tilde{\pi}'_P) : (\pi'_E, \pi'_P) \rightarrow (\tilde{\pi}'_E, \tilde{\pi}'_P)$ ,  $\tilde{\beta}|_{T_{C_E}} = \beta$ ,  $((\tilde{\pi}_E, \tilde{\pi}_P), (\tilde{\pi}'_E, \tilde{\pi}'_P), \tilde{\beta}) \in \mathcal{R}$ , и если  $\pi_P \xrightarrow{\pi} \tilde{\pi}_E$ ,  $\pi'_P \xrightarrow{\pi'} \tilde{\pi}'_E$ ,  $\gamma = \tilde{\beta}|_{T_C}$ , то:
  - (a)  $\gamma^{-1} : \rho_{C'} \sqsubseteq \rho_C$ , если  $\star = pw$ ;
  - (b)  $\gamma : \rho_C \simeq \rho_{C'}$ , если  $\star \in \{pot, pr\}$ ;
  - (c)  $C \simeq C'$ , если  $\star = pr$ .
4. Как пункт 3, но роли  $N$  и  $N'$  меняются.

Сети  $N$  и  $N'$   $\star$ -ST-бисимуляционно эквивалентны,  $\star \in \{\text{интерливинговая, ЧС, ЧУММ, процессная}\}$ , запись  $N \leftrightarrow_{\star ST} N'$ , если  $\exists \mathcal{R} : N \leftrightarrow_{\star ST} N'$ ,  $\star \in \{i, pw, pot, pr\}$ .

Заметим, что мы не определяем шаговую ST-бисимуляционную эквивалентность, так как она совпадает с  $\leftrightarrow_{iST}$ , что упомянуто, например, в [276].

**Сохраняющие историю бисимуляционные эквивалентности** Сохраняющие историю бисимуляционные эквивалентности учитывают “прошлое” (историю) функционирования сетей, то есть при моделировании учитывается та часть процесса, выполнение которой приводит из начального в текущее состояние.

**Определение 1.2.10** Пусть  $N$  и  $N'$  — некоторые сети. Отношение  $\mathcal{R} \subseteq \Pi(N) \times \Pi(N') \times \mathcal{B}$ , где  $\mathcal{B} = \{\beta \mid \beta : T_C \rightarrow T_{C'}\}$ ,  $\pi = (C, \varphi) \in \Pi(N)$ ,  $\pi' = (C', \varphi') \in \Pi(N')$  —  $\star$ -сохраняющая историю бисимуляция между  $N$  и  $N'$ ,  $\star \in \{\text{ЧУММ, процессная}\}$ , запись  $N \leftrightarrow_{\star h} N'$ ,  $\star \in \{pot, pr\}$ , если:

1.  $(\pi_N, \pi_{N'}, \emptyset) \in \mathcal{R}$ .
2.  $(\pi, \pi', \beta) \in \mathcal{R} \Rightarrow$ 
  - (a)  $\beta : \rho_C \simeq \rho_{C'}$ , если  $\star \in \{pot, pr\}$ ;
  - (b)  $C \simeq C'$ , если  $\star = pr$ .
3.  $(\pi, \pi', \beta) \in \mathcal{R}$ ,  $\pi \rightarrow \tilde{\pi} \Rightarrow \exists \tilde{\beta}, \tilde{\pi}' : \pi' \rightarrow \tilde{\pi}'$ ,  $\tilde{\beta}|_{T_C} = \beta$ ,  $(\tilde{\pi}, \tilde{\pi}', \tilde{\beta}) \in \mathcal{R}$ .
4. Как пункт 3, но роли  $N$  и  $N'$  меняются.

Сети  $N$  и  $N'$   $\star$ -сохраняюще историю бисимуляционно эквивалентны,  $\star \in \{\text{ЧУММ, процессная}\}$ , запись  $N \leftrightarrow_{\star h} N'$ , если  $\exists \mathcal{R} : N \leftrightarrow_{\star h} N'$ ,  $\star \in \{pot, pr\}$ .

Очевидно, что можно упростить эти определения, рассматривая расширения процессов только на одно действие. Итерация таких расширений дает расширение на процесс.

Заметим, что мы не приводим определений интерливинговой и шаговой сохраняющих историю бисимуляционных эквивалентностей, так как они совпадают с уже введенными понятиями  $\leftrightarrow_i$  и  $\leftrightarrow_s$  соответственно. Действительно, без ограничения общности, можно считать, что отношение бисимуляции для этих новых определений включает пары процессов, соответствующих одинаковым интерливинговым (шаговым) следам. Если расширяющие процессы будут также соответствовать одинаковым следам, то получившиеся в результате процессы также будут обладать этим свойством.

Определение ЧС сохраняющей историю бисимуляционной эквивалентности не дается, так как попытка определить такое отношение по аналогии с только что введенными понятиями  $\leftrightarrow_{pomh}$  и  $\leftrightarrow_{prh}$  приводит к несимметричному отношению, так как ЧУММ С-сети одного из процессов, связанных этим отношением должно быть более параллельно, чем ЧУММ С-сети другого процесса. Для симметричности этого отношения можно потребовать, чтобы каждое из таких ЧУММ было более параллельно, чем другое, а это означает их изоморфизм, и тогда ЧС сохраняющая историю бисимуляционная эквивалентность совпадает с  $\leftrightarrow_{pomh}$ .

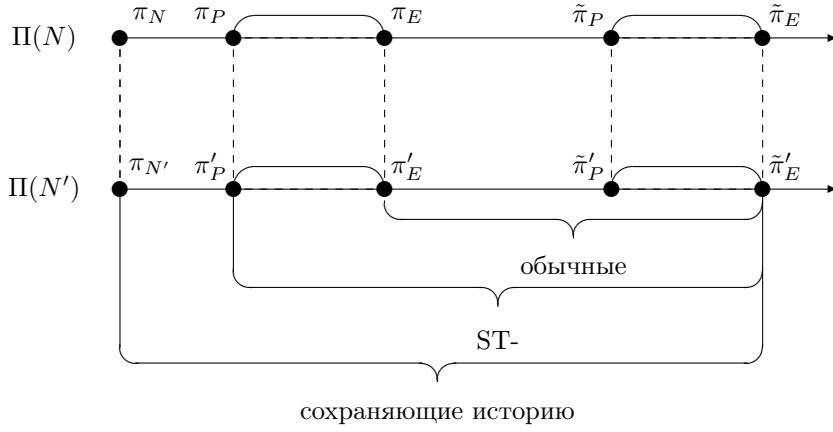


Рис. 1.10: Различающая способность бисимуляционных эквивалентностей

### Сохраняющие конфликт эквивалентности

Сохраняющие конфликт эквивалентности полностью учитывают конфликты в сетях.

**Определение 1.2.11** МСС-след сети  $N$  — МСС  $\xi$  — класс изоморфизма ПСС  $\xi_O$  для  $\varpi = (O, \psi) \in \wp(N)$ . Обозначим через  $MEStructs(N)$  множество всех МСС-следов сети  $N$ . Две сети  $N$  и  $N'$  МСС сохраняющие конфликт эквивалентны, запись  $N \equiv_{mes} N'$ , если  $MEStructs(N) = MESTructs(N')$ . Заметим, что, в силу единственности максимального  $O$ -процесса, это равносильно требованию  $\mathcal{E}(N) = \mathcal{E}(N')$ .

**Определение 1.2.12**  $O$ -процессный след сети  $N$  — это класс изоморфизма  $O$ -сети  $O$  для  $\varpi = (O, \psi) \in \wp(N)$ . Обозначим через  $OccNets(N)$  множество всех  $O$ -процессных следов сети  $N$ . Две сети  $N$  и  $N'$   $O$ -процессно сохраняющие конфликт эквивалентны, запись  $N \equiv_{occ} N'$ , если  $OccNets(N) = OccNets(N')$ . Заметим, что, в силу единственности максимального  $O$ -процесса, это равносильно требованию  $\mathcal{U}(N) = \mathcal{U}(N')$ .

Заметим, что определения сохраняющих конфликт эквивалентностей напоминают определения следовых эквивалентностей. Так,  $\equiv_{mes}$  соответствует  $\equiv_{pom}$ , если вместо ЧУММ рассматривать МСС, а  $\equiv_{occ}$  напоминает  $\equiv_{pr}$ , если С-сети заменить на  $O$ -сети. Можно попытаться ввести определения, соответствующие интерливинговому, шаговому следовым эквивалентностям и ЧС следовым эквивалентностям. При этом легко можно понять, что первые две эквивалентности можно определить только так, что они совпадают, соответственно, с  $\equiv_i$  и  $\equiv_s$ . Попробуем определить третью из них, ЧС сохраняющую конфликт эквивалентность в предположении, что понятие *большой параллельности* для МСС дается также, как и для ЧУММ, за исключением того, что требуется полное сохранение отношения конфликта. Однако, в силу единственности МСС, соответствующей данной сети, мы получим совпадение этой эквивалентности с  $\equiv_{mes}$ .

### Сравнение базисных эквивалентностей

На рисунке 1.10 показана различающая способность бисимуляционных эквивалентностей. Расширению  $ST$ -процессов  $(\pi_E, \pi_P) \rightarrow (\tilde{\pi}_E, \tilde{\pi}_P)$  в сети  $N$  соответствует расширение  $(\pi'_E, \pi'_P) \rightarrow (\tilde{\pi}'_E, \tilde{\pi}'_P)$  в сети  $N'$ . Обычные бисимуляционные эквивалентности требуют моделирования только расширяющей части процесса, то есть части от  $\pi_E$  до  $\tilde{\pi}_E$  (от  $\pi'_E$  до  $\tilde{\pi}'_E$ ).  $ST$ -бисимуляционные эквивалентности, кроме того, при моделировании учитывают множество работающих переходов между  $\pi_P$  и  $\pi_E$  ( $\pi'_P$  и  $\pi'_E$ ). Сохраняющие историю бисимуляции, кроме расширяющей, требуют учета также той части процесса, которая привела из начального в данное состояние, то есть части  $\pi_N$  до  $\pi_E$  (от  $\pi_{N'}$  до  $\pi'_E$ ). Таким образом,  $ST$ -бисимуляционные эквивалентности сильнее обычных и слабее сохраняющих историю.

Кроме того, очевидно, что бисимуляционные эквивалентности сильнее следовых, так как они учитывают ветвистую структуру сетей.

Очевидно, что сохраняющие конфликт эквивалентности сильнее бисимуляционных (так как они полностью учитывают конфликты в сетях, следовательно, и их ветвистую структуру) и слабее изоморфизма.

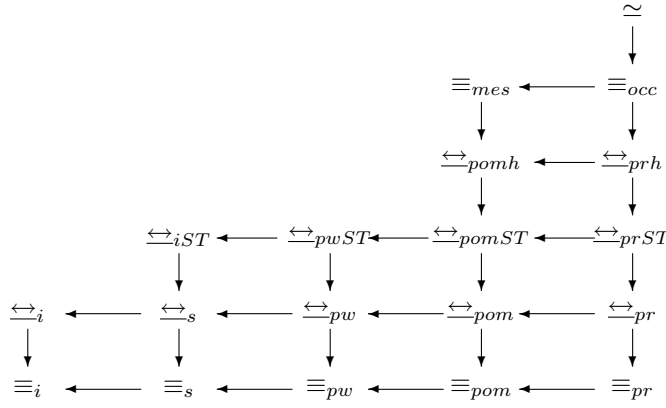


Рис. 1.11: Взаимосвязь базисных эквивалентностей

Следующая теорема, устанавливающая взаимосвязи между всеми базисными эквивалентностями, формализует эти рассуждения. (В этой теореме и далее символом ‘ $\_$ ’ будем обозначать пустую альтернативу. То есть знак, подписанный или надписанный этим символом, будем считать знаком без подписей и надписей.)

**Теорема 1.2.1** Пусть  $\leftrightarrow, \Leftrightarrow \in \{\equiv, \leftrightarrow, \simeq\}$  и  $\star, \star\star \in \{\_, i, s, pw, pom, pr, iST, pwST, pomST, prST, pomh, prh, mes, occ\}$ . Для сетей  $N$  и  $N'$   $N \leftrightarrow_\star N' \Rightarrow N \Leftrightarrow_{\star\star} N'$  тогда и только тогда, когда в графе на рисунке 1.11 существует направленный путь от  $\leftrightarrow_\star$  к  $\Leftrightarrow_{\star\star}$ .

*Доказательство.* ( $\Leftarrow$ ) Проверим, что истинность всех импликаций на рисунке 1.11.

- Связи между следовыми и интерливинговыми эквивалентностями — следствия того, что изоморфизм ПЧУМ с пустым отношением предшествования является изоморфизмом одноэлементных ПЧУМ.
- Связи между эквивалентностями ЧС и шаговыми — следствия того, что гомоморфизм ПЧУМ является изоморфизмом ПЧУМ с пустым отношением предшествования.
- Связь  $\leftrightarrow_{pwST} \rightarrow \leftrightarrow_{iST}$  — следствие того, что гомоморфизм ПЧУМ является сохраняющей пометку биекцией.
- Связи между эквивалентностями ЧУММ и ЧС — следствия того, что изоморфизм ПЧУМ является гомоморфизмом.
- Связи между процессными эквивалентностями и ЧУММ эквивалентностями — следствия того, что ПЧУМ на основе изоморфных С-сетей также изоморфны.
- Связь  $\equiv_{occ} \rightarrow \equiv_{mes}$  следует из того, что изоморфные О-сети имеют изоморфные ПСС.
- Связь  $\leftrightarrow_i \rightarrow \equiv_i$  устанавливается следующим образом. Пусть  $\mathcal{R} : N \leftrightarrow_i N'$ . Если  $\pi_N \xrightarrow{a_1} \pi_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} \pi_n$ , то существует последовательность  $(\pi_N, \pi_{N'}), (\pi_1, \pi'_1), \dots, (\pi_n, \pi'_n) \in \mathcal{R}$  такая, что  $\pi_{N'} \xrightarrow{a_1} \pi'_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} \pi'_n$ , и наоборот, в силу симметричности бисимуляции.
- Связь  $\leftrightarrow_s \rightarrow \equiv_s$  доказывается как в предыдущем случае, но с использованием  $A_1, \dots, A_n \in \mathcal{M}(Act)$  вместо  $a_1, \dots, a_n \in Act$ .
- Связь  $\leftrightarrow_{pw} \rightarrow \equiv_{pw}$  доказывается следующим образом. Пусть  $\mathcal{R} : N \leftrightarrow_{pw} N'$  и  $\rho$  — класс изоморфизма  $\rho_C$  для  $\pi = (C, \varphi) \in \Pi(N)$ . Так как  $\pi_N \xrightarrow{\pi} \pi$ , то существует пара  $(\pi, \pi') \in \mathcal{R}$  такая, что  $\pi' = (C', \varphi')$  и  $\rho_{C'} \subseteq \rho_C$ . Если  $\rho'$  — класс изоморфизма  $\rho_{C'}$ , то  $\rho' \subseteq \rho$ . Следовательно,  $Pomsets(N') \subseteq Pomsets(N)$ . Включение  $Pomsets(N) \subseteq Pomsets(N')$ , доказывается аналогично, в силу симметричности бисимуляции.
- Связь  $\leftrightarrow_{pom} \rightarrow \equiv_{pom}$  доказывается как в предыдущем случае, только с использованием изоморфизма вместо гомоморфизма ПЧУМ.

- Связь  $\leftrightarrow_{pr} \rightarrow \equiv_{pr}$  устанавливается аналогично предыдущему пункту с использованием процессных следов вместо ЧУММ следов и изоморфизма С-сетей вместо изоморфизма их ПЧУМ.
- Связь  $\leftrightarrow_{iST} \rightarrow \leftrightarrow_s$  проверяется с использованием того, что шагу  $\pi \xrightarrow{A} \tilde{\pi}$ , где  $A = \{a_1, \dots, a_n\} \in \mathcal{M}(Act)$ , соответствует последовательность ST-процессов  $(\pi_0, \pi_0), \dots, (\pi_n, \pi_0), \dots, (\pi_n, \pi_n)$  такая, что  $\pi = \pi_0 \xrightarrow{a_1} \dots \xrightarrow{a_n} \pi_n = \tilde{\pi}$ .
- Связи  $\leftrightarrow_{*ST} \rightarrow \leftrightarrow_*$ ,  $\star \in \{pw, pom, pr\}$ , доказываются с построением на основе отношения  $\mathcal{R} : N \leftrightarrow_{*ST} N'$  нового отношения  $\mathcal{S} : N \leftrightarrow_* N'$ , определяемого следующим образом:  
 $\mathcal{S} = \{(\pi, \pi') \mid \exists \beta ((\pi, \pi), (\pi', \pi'), \beta) \in \mathcal{R}\}$ .
- Связи  $\leftrightarrow_{*h} \rightarrow \leftrightarrow_{*ST}$ ,  $\star \in \{pom, pr\}$ , доказываются построением на основе  $\mathcal{R} : N \leftrightarrow_{*h} N'$  отношения  $\mathcal{S} : N \leftrightarrow_{*ST} N'$ , определяемого следующим образом:  $\mathcal{S} = \{((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \mid (\pi_E, \pi'_E, \beta) \in \mathcal{R}, (\pi_P, \pi'_P) \in ST - \Pi(N), (\pi'_E, \pi'_P) \in ST - \Pi(N'), \beta(T_{C_P}) = T_{C'_P}\}$ .
- Связь  $\equiv_{mes} \rightarrow \leftrightarrow_{pomh}$  доказывается следующим образом. Пусть  $\varpi = (O, \psi) \in \wp_{max}(N)$ ,  $\varpi' = (O', \psi') \in \wp_{max}(N')$ ,  $\gamma : \xi_O \simeq \xi_{O'}$ . Тогда  $\mathcal{R} : N \leftrightarrow_{pomh} N'$ , где отношение  $\mathcal{R}$  определяется следующим образом:  $\mathcal{R} = \{(\pi, \pi', \beta) \mid \pi - \text{вычисление } \varpi, \pi' - \text{вычисление } \varpi', \text{ такие, что } \gamma|_{T_C} : \rho_C \simeq \rho_{C'}, \beta = \gamma|_{T_C}\}$ .
- Связь  $\equiv_{occ} \rightarrow \leftrightarrow_{prh}$  доказывается следующим образом. Пусть  $\varpi = (O, \psi) \in \wp_{max}(N)$ ,  $\varpi' = (O', \psi') \in \wp_{max}(N')$ ,  $\gamma : O \simeq O'$ . Тогда  $\mathcal{R} : N \leftrightarrow_{prh} N'$ , где отношение  $\mathcal{R}$  определяется следующим образом:  $\mathcal{R} = \{(\pi, \pi', \beta) \mid \pi - \text{вычисление } \varpi, \pi' - \text{вычисление } \varpi', \text{ такие, что } \gamma|_{(P_C \cup T_C)} : C \simeq C', \beta = \gamma|_{T_C}\}$ .
- Связь  $\simeq \rightarrow \equiv_{occ}$  следует из того факта, что изоморфные сети имеют одинаковые развертки.

( $\Rightarrow$ ) Докажем, что на рисунке 1.11 от одной эквивалентности к другой нельзя провести ни одной стрелки такой, что не существует направленного пути от первой эквивалентности ко второй в графе на этом рисунке, используя следующие примеры на сетях.

- На рисунке 1.12(a)  $N \leftrightarrow_i N'$ , но  $N \not\equiv_s N'$ , так как только в сети  $N'$  действия  $a$  и  $b$  не могут сработать параллельно.
- На рисунке 1.12(c)  $N \leftrightarrow_{iST} N'$ , но  $N \not\equiv_{pw} N'$ , так как сети  $N$  соответствует ЧУММ такое, что даже более параллельное ЧУММ не может быть выполнено в  $N'$ .
- На рисунке 1.12(b)  $N \leftrightarrow_{pwST} N'$ , но  $N \not\equiv_{pom} N'$ , так как только в сети  $N'$  действие  $b$  может зависеть от  $a$ .
- На рисунке 1.12(d)  $N \equiv_{mes} N'$ , но  $N \not\equiv_{pr} N'$ , так как  $N'$  — С-сеть, не изоморфная С-сети  $N$  (из-за дополнительного выходного места).
- На рисунке 1.12(e)  $N \equiv_{pr} N'$ , но  $N \not\leftrightarrow_i N'$ , так как только в сети  $N'$  можно выполнить действие  $a$  так, что после него нельзя выполнить  $b$ .
- На рисунке 1.13(a)  $N \leftrightarrow_{pr} N'$ , но  $N \not\leftrightarrow_{iST} N'$ , так как в сети  $N'$  действие  $a$  может так начать работать, что никакое  $b$  уже не сможет стартовать до завершения  $a$ .
- На рисунке 1.13(b)  $N \leftrightarrow_{prST} N'$ , но  $N \not\leftrightarrow_{pomh} N'$ , так как только в сети  $N'$  после действия  $a$  действие  $b$  может сработать так, что действие  $c$  должно обязательно зависеть от  $a$ .
- На рисунке 1.13(c)  $N \leftrightarrow_{prh} N'$ , но  $N \not\equiv_{mes} N'$ , так как только сети  $N'$  соответствует МСС с двумя конфликтными между собой действиями  $a$ .
- На рисунке 1.13(d)  $N \equiv_{occ} N'$ , но  $N \not\equiv N'$ , так как никогда не срабатывающие переходы сетей  $N$  и  $N'$  помечены разными действиями ( $a$  и  $b$ ).  $\square$

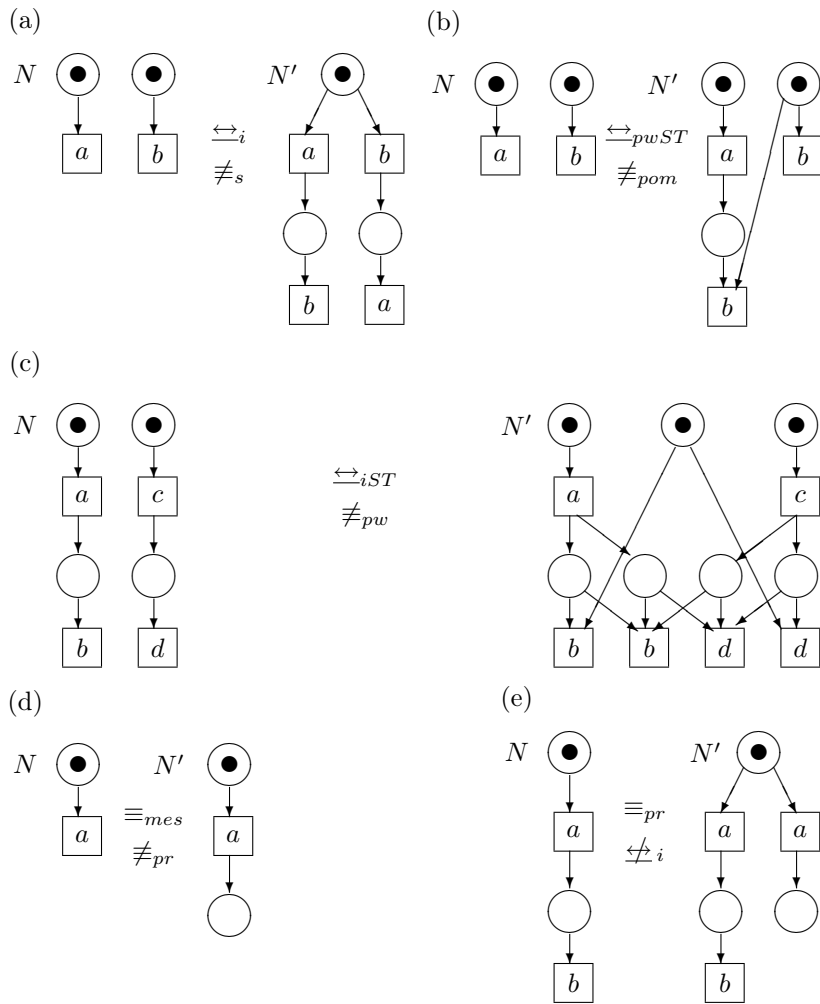


Рис. 1.12: Примеры базисных эквивалентностей

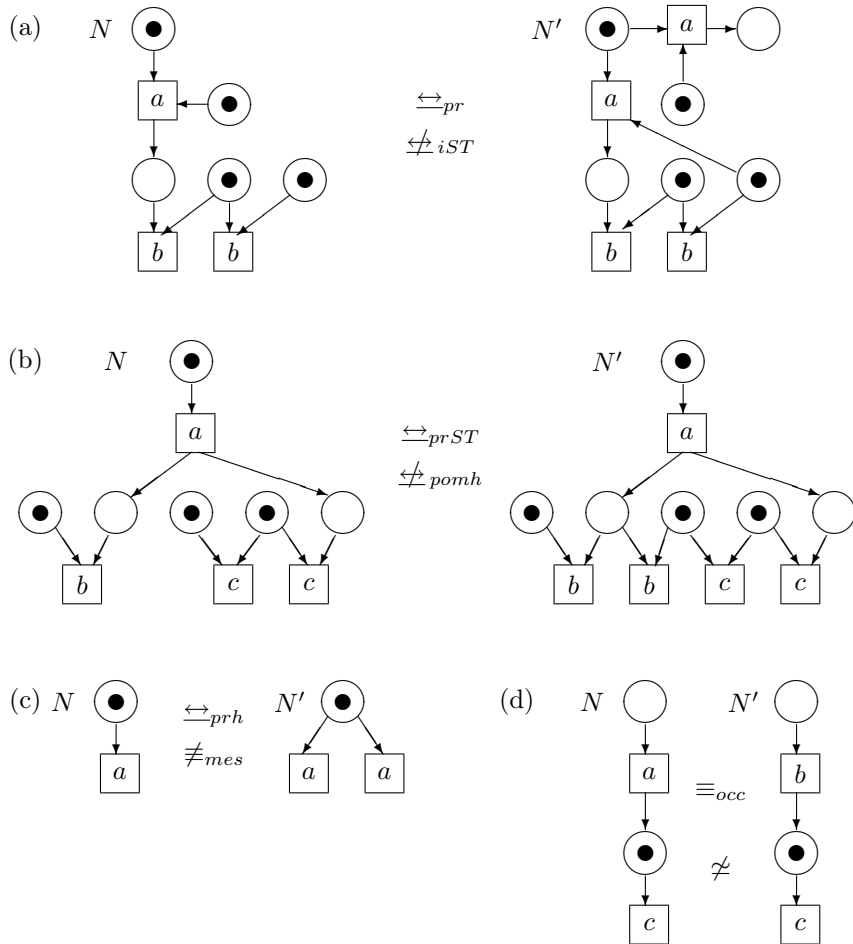


Рис. 1.13: Примеры базисных эквивалентностей (продолжение)

## 1.2.2 Обратные-прямые эквивалентности

Обратные-прямые бисимуляционные эквивалентности основаны на идее, что отношение бисимуляции должно требовать взаимного моделирования систем не только в прямом направлении (как обычно), но и при движении в обратном направлении “вдоль” истории своих выполнений. Эти эквивалентности тесно связаны с эквивалентностями логик с модальностями “прошлого” [95, 146, 191, 193] и логик “истинного параллелизма” [229, 230, 231].

Такие эквивалентностные понятия были впервые введены в [214] в рамках систем переходов. Было показано, что обратный-прямой вариант ( $\leftrightarrow_{ibif}$ ) интерливинговой бисимуляционной эквивалентности совпадает с обычной  $\leftrightarrow_i$ .

В [112, 102, 103, 104] были определены новые варианты шаговой ( $\leftrightarrow_{sbsf}$ ), ЧС ( $\leftrightarrow_{pwbprwf}$ ) и ЧУММ ( $\leftrightarrow_{pombpomf}$ ) обратных-прямых бисимуляционных эквивалентностей мест в рамках структур событий. Было проведено сравнение этих эквивалентностных понятий с обычными, ST- и сохраняющими историю бисимуляционными. Было показано, что  $\leftrightarrow_{pomST}$  влечет  $\leftrightarrow_{sbsf}$ . Установлено также совпадение  $\leftrightarrow_{pombpomf}$  и  $\leftrightarrow_{pomh}$ , дающее новую, логическую характеристику последней эквивалентности.

В [237] появилась новая идея о том, что типы прямого и обратного моделирования могут различаться (в соответствии с этой идеей можно, например, определить шаговую обратную ЧУММ прямую бисимуляционную эквивалентность ( $\leftrightarrow_{sbpomf}$ )). Было введено множество всех возможных обратных-прямых эквивалентностных понятий в интерливинговой, шаговой, ЧС и ЧУММ семантиках. Получены два новых понятия, не совпадающих с уже известными: шаговая обратная ЧС прямая ( $\leftrightarrow_{sbprf}$ ) и шаговая обратная ЧУММ прямая ( $\leftrightarrow_{sbpomf}$ ) бисимуляционные эквивалентности.

Обратные-прямые бисимуляционные эквивалентности рассматривались также в [47, 177, 178, 211, 219].

В данном разделе в рамках сетей Петри мы расширяем множество обратных-прямых эквивалентностей из [237] новыми отношениями для процессной семантики и получаем в результате два новых понятия, которые не могут быть сведены к уже известным: шаговая обратная процессная прямая ( $\leftrightarrow_{sbprf}$ ) и ЧУММ обратная процессная прямая ( $\leftrightarrow_{pombprf}$ ) бисимуляционные эквивалентности.

Мы дополняем результаты [104, 237] и устанавливаем новые взаимосвязи между обратными-прямыми бисимуляционными и базисными эквивалентностями. В частности, мы доказываем, что  $\leftrightarrow_{pomST}$  влечет  $\leftrightarrow_{sbpomf}$  и  $\leftrightarrow_{prST}$  влечет  $\leftrightarrow_{sbprf}$ . Мы также демонстрируем совпадение  $\leftrightarrow_{prbprf}$  и  $\leftrightarrow_{prh}$ .

### Последовательные выполнения

В соответствии с идеей обратных-прямых бисимуляций, при обратном моделировании можно двигаться только “вдоль” истории выполнения, приведшей в данное состояние. Поэтому в [214] в рамках модели систем переходов отношения обратной-прямой бисимуляции связывали цепочки переходов, а в [102, 103, 104, 237] в рамках структур событий — цепочки событий, называемые “историями” или “выполнениями”. Такие цепочки содержали информацию о порядке, в котором происходили события (переходы), приведшие в данное состояние, а в рамках модели структур событий эти цепочки задавали также конфигурации, содержащие информацию о причинной зависимости событий.

В рамках сетей Петри, очевидно, недостаточно рассматривать только цепочки переходов С-сетей их процессов, так как, в зависимости от функции встраивания, разные цепочки могут соответствовать одному процессу, и наоборот, разным процессам может соответствовать одна цепочка. Для того, чтобы учесть информацию о причинно-следственных связях событий и о структуре С-сетей процессов мы вводим понятие последовательных выполнений как пары, состоящей из процесса и цепочки переходов его С-сети, расширивших начальный процесс до данного.

**Определение 1.2.13** Последовательное выполнение сети  $N$  — пара  $(\pi, \sigma)$ , где:

- процесс  $\pi = (C, \varphi) \in \Pi(N)$  содержит информацию о причинно-следственных связях событий, приведших в данное состояние;
- цепочка  $\sigma \in T_C^*$  такая, что  $\pi_N \xrightarrow{\sigma} \pi$ , содержит информацию о последовательности выполнения событий, приведших в данное состояние.

Обозначим множество всех последовательных выполнений сети  $N$  через  $Runs(N)$ .

Начальное последовательное выполнение сети  $N$  — пара  $(\pi_N, \varepsilon)$ , где  $\varepsilon$  — пустая цепочка. Обозначим через  $|\sigma|$  длину цепочки  $\sigma$ .

Пусть  $(\pi, \sigma) \in Runs(N)$  и  $(\pi', \sigma') \in Runs(N')$ . Отображение  $\beta : P_C \cup T_C \rightarrow P_{C'} \cup T_{C'}$  — изоморфизм между  $(\pi, \sigma)$  и  $(\pi', \sigma')$ , обозначение  $\beta : (\pi, \sigma) \simeq (\pi', \sigma')$ , если:

1.  $\beta : \pi \simeq \pi'$ ;
2.  $\beta(\sigma) = \sigma'$  ( $\beta$  расширяется на цепочки стандартным образом).

Последовательные выполнения  $(\pi, \sigma)$  и  $(\pi', \sigma')$  *изоморфны*, обозначение  $(\pi, \sigma) \simeq (\pi', \sigma')$ , если  $\exists \beta : (\pi, \sigma) \simeq (\pi', \sigma')$ .

Пусть  $(\pi, \sigma), (\tilde{\pi}, \tilde{\sigma}) \in \text{Runs}(N)$ . Пишем  $(\pi, \sigma) \xrightarrow{\hat{\pi}} (\tilde{\pi}, \tilde{\sigma})$ , если  $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$ ,  $\exists \hat{\sigma} \in T_C^* \pi \xrightarrow{\hat{\sigma}} \tilde{\pi}$  и  $\tilde{\sigma} = \sigma \hat{\sigma}$ . Будем писать  $(\pi, \sigma) \rightarrow (\tilde{\pi}, \tilde{\sigma})$ , если  $\exists \hat{\pi} (\pi, \sigma) \xrightarrow{\hat{\pi}} (\tilde{\pi}, \tilde{\sigma})$ .

Пусть  $(\pi, \sigma) \in \text{Runs}(N)$ ,  $(\pi', \sigma') \in \text{Runs}(N')$  и  $\sigma = v_1 \cdots v_n$ ,  $\sigma' = v'_1 \cdots v'_n$ . Определим функцию  $\beta_{\sigma}^{\sigma'} : T_C \rightarrow T_{C'}$  следующим образом:  $\beta_{\sigma}^{\sigma'} = \{(v_i, v'_i) \mid 1 \leq i \leq n\}$ . Положим  $\beta_{\varepsilon}^{\varepsilon} = \emptyset$ .

Пусть  $(\pi, \sigma) \in \text{Runs}(N)$  и  $\sigma = v_1 \cdots v_n$ ,  $\pi_N \xrightarrow{v_1} \dots \xrightarrow{v_i} \pi_i$  ( $1 \leq i \leq n$ ).

Введем следующие обозначения:

- $\pi(0) = \pi_N$ ,  
 $\pi(i) = \pi_i$  ( $1 \leq i \leq n$ );
- $\sigma(0) = \varepsilon$ ,  
 $\sigma(i) = v_1 \cdots v_i$  ( $1 \leq i \leq n$ ).

Пусть  $(\pi, \sigma) \in \text{Runs}(N)$ . Обозначим  $\text{Past}(\pi, \sigma) = \pi(j)$ , где  $j = \min\{i \mid (\pi, \pi(i)) \in ST - \Pi(N)\}$ .

$ST$ -процесс последовательного выполнения  $(\pi, \sigma)$  определяется следующим образом:

$ST(\pi, \sigma) = (\pi, \text{Past}(\pi, \sigma))$ .

### Обратные-прямые бисимуляционные эквивалентности

**Определение 1.2.14** Пусть  $N$  и  $N'$  — некоторые сети. Отношение  $\mathcal{R} \subseteq \text{Runs}(N) \times \text{Runs}(N')$  —  $\star$ -обратная  $\star\star$ -прямая бисимуляция между  $N$  и  $N'$ ,  $\star, \star\star \in \{\text{интерливингивая, шаговая, ЧС, ЧУММ, процессная}\}$ , запись  $\mathcal{R} : N \xrightarrow{\star\star} N'$ ,  $\star, \star\star \in \{i, s, pw, rot, pr\}$ , если:

1.  $((\pi_N, \varepsilon), (\pi_{N'}, \varepsilon)) \in \mathcal{R}$ .

2.  $((\pi, \sigma), (\pi', \sigma')) \in \mathcal{R}$

- (обратно)

$(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\hat{\pi}} (\pi, \sigma)$ ,

(a)  $|T_{\hat{C}}| = 1$ , если  $\star = i$ ;

(b)  $\prec_{\hat{C}} = \emptyset$ , если  $\star = s$ ;

$\Rightarrow \exists (\tilde{\pi}', \tilde{\sigma}') : (\tilde{\pi}', \tilde{\sigma}') \xrightarrow{\hat{\pi}'} (\pi', \sigma')$ ,  $((\tilde{\pi}, \tilde{\sigma}), (\tilde{\pi}', \tilde{\sigma}')) \in \mathcal{R}$  и

(a)  $\rho_{\hat{C}'} \sqsubseteq \rho_{\hat{C}}$ , если  $\star = pw$ ;

(b)  $\rho_{\hat{C}} \simeq \rho_{\hat{C}'}$ , если  $\star \in \{i, s, rot\}$ ;

(c)  $\hat{C} \simeq \hat{C}'$ , если  $\star = pr$ ;

- (прямо)

$(\pi, \sigma) \xrightarrow{\hat{\pi}} (\tilde{\pi}, \tilde{\sigma})$ ,

(a)  $|T_{\hat{C}}| = 1$ , если  $\star\star = i$ ;

(b)  $\prec_{\hat{C}} = \emptyset$ , если  $\star\star = s$ ;

$\Rightarrow \exists (\tilde{\pi}', \tilde{\sigma}') : (\pi', \sigma') \xrightarrow{\hat{\pi}'} (\tilde{\pi}', \tilde{\sigma}')$ ,  $((\tilde{\pi}, \tilde{\sigma}), (\tilde{\pi}', \tilde{\sigma}')) \in \mathcal{R}$  и

(a)  $\rho_{\hat{C}'} \sqsubseteq \rho_{\hat{C}}$ , если  $\star\star = pw$ ;

(b)  $\rho_{\hat{C}} \simeq \rho_{\hat{C}'}$ , если  $\star\star \in \{i, s, rot\}$ ;

(c)  $\hat{C} \simeq \hat{C}'$ , если  $\star\star = pr$ .

3. Как пункт 2, но роли  $N$  и  $N'$  меняются.



Сети  $N$  и  $N'$   $\star$ -обратно  $\star\star$ -прямо бисимуляционно эквивалентны,  $\star, \star\star \in \{\text{интерливингивая, шаговая, ЧС, ЧУММ, процессная}\}$ , запись  $N \xleftrightarrow{\star b \star\star f} N'$ , если  $\exists \mathcal{R} : N \xleftrightarrow{\star b \star\star f} N'$ ,  $\star, \star\star \in \{i, s, pw, pot, pr\}$ .

Заметим, что обратные расширения последовательных выполнений *детерминированы*, то есть для  $(\pi, \sigma) \in \text{Runs}(N)$  существует единственное последовательное выполнение  $(\tilde{\pi}, \tilde{\sigma}) \in \text{Runs}(N)$  такое, что  $(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\hat{\pi}} (\pi, \sigma)$  и  $|\tilde{\sigma}| = i$  ( $0 \leq i \leq |\sigma|$ ). В данном случае  $(\tilde{\pi}, \tilde{\sigma}) = (\pi(i), \sigma(i))$ .

### Сравнение обратных-прямых бисимуляционных эквивалентностей

Рассмотрим взаимосвязь обратных-прямых бисимуляционных эквивалентностей.

**Предложение 1.2.1** Пусть  $\star \in \{i, s, pw, pot, pr\}$ . Для сетей  $N$  и  $N'$   $N \xleftrightarrow{pw b \star f} N' \Leftrightarrow N \xleftrightarrow{pot b \star f} N'$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие того, что изоморфизм ПЧУМ является гомоморфизмом.

( $\Rightarrow$ ) Пусть  $\mathcal{R} : N \xleftrightarrow{pw b \star f} N'$ . Докажем, что  $\mathcal{R} : N \xleftrightarrow{pot b \star f} N'$ .

1. Очевидно,  $((\pi_N, \varepsilon), (\pi_{N'}, \varepsilon)) \in \mathcal{R}$ .

2. Пусть  $((\pi, \sigma), (\pi', \sigma')) \in \mathcal{R}$ .

- (обратно)

Пусть  $(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\hat{\pi}} (\pi, \sigma)$ . Тогда  $\exists(\tilde{\pi}', \tilde{\sigma}') : (\tilde{\pi}', \tilde{\sigma}') \xrightarrow{\hat{\pi}'} (\pi', \sigma')$ ,  $((\tilde{\pi}, \tilde{\sigma}), (\tilde{\pi}', \tilde{\sigma}')) \in \mathcal{R}$  и  $\rho_{\hat{C}} \sqsubseteq \rho_{\hat{C}'}$ .

В силу симметричности бисимуляции, обратное расширение  $(\tilde{\pi}', \tilde{\sigma}') \xrightarrow{\hat{\pi}'} (\pi', \sigma')$  должно имитироваться расширением  $(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\hat{\pi}} (\pi, \sigma)$ , и только им, из-за детерминированности обратных расширений. Тогда  $\rho_{\hat{C}} \sqsubseteq \rho_{\hat{C}'}$ . Следовательно,  $\rho_{\hat{C}} \simeq \rho_{\hat{C}'}$ .

- (прямо)

Очевидно.

3. Как пункт 2, но роли  $N$  и  $N'$  меняются. □

**Предложение 1.2.2** Пусть  $\star \in \{i, s, pw, pot, pr\}$ . Для сетей  $N$  и  $N'$   $N \xleftrightarrow{\star b i f} N' \Leftrightarrow N \xleftrightarrow{\star b \star f} N'$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие того, что изоморфизм С-сетей, изоморфизм и гомоморфизм ПЧУМ С-сетей, изоморфизм ПЧУМ С-сетей с пустым отношением предшествования являются сохраняющей ометку биекцией.

( $\Rightarrow$ ) Пусть  $\mathcal{R} : N \xleftrightarrow{\star b i f} N'$ . Докажем, что  $\mathcal{R} : N \xleftrightarrow{\star b \star f} N'$ .

1. Очевидно,  $((\pi_N, \varepsilon), (\pi_{N'}, \varepsilon)) \in \mathcal{R}$ .

2. Пусть  $((\pi, \sigma), (\pi', \sigma')) \in \mathcal{R}$ .

- (обратно)

Очевидно.

- (прямо)

Пусть  $(\pi, \sigma) \xrightarrow{\hat{\pi}} (\tilde{\pi}, \tilde{\sigma})$ . Расширению на  $\hat{\pi}$  соответствует расширение на некоторую последовательность переходов. Тогда  $\exists(\tilde{\pi}', \tilde{\sigma}') : (\pi', \sigma') \xrightarrow{\hat{\pi}'} (\tilde{\pi}', \tilde{\sigma}')$ ,  $((\tilde{\pi}, \tilde{\sigma}), (\tilde{\pi}', \tilde{\sigma}')) \in \mathcal{R}$ , где расширение на  $\hat{\pi}'$  соответствует расширению на последовательность переходов, моделирующему соответствующее расширение в сети  $N$ .

В силу симметричности бисимуляции, обратное расширение  $(\pi, \sigma) \xrightarrow{\hat{\pi}} (\tilde{\pi}, \tilde{\sigma})$  должно имитироваться расширением  $(\pi', \sigma') \xrightarrow{\hat{\pi}'} (\tilde{\pi}', \tilde{\sigma}')$ , и только им, из-за детерминированности обратных расширений. Тогда

- (a)  $\rho_{\hat{C}'} \sqsubseteq \rho_{\hat{C}}$ , если  $\star = pw$ ;
- (b)  $\rho_{\hat{C}} \simeq \rho_{\hat{C}'}$ , если  $\star \in \{i, s, pot\}$ ;
- (c)  $\hat{C} \simeq \hat{C}'$ , если  $\star = pr$ .

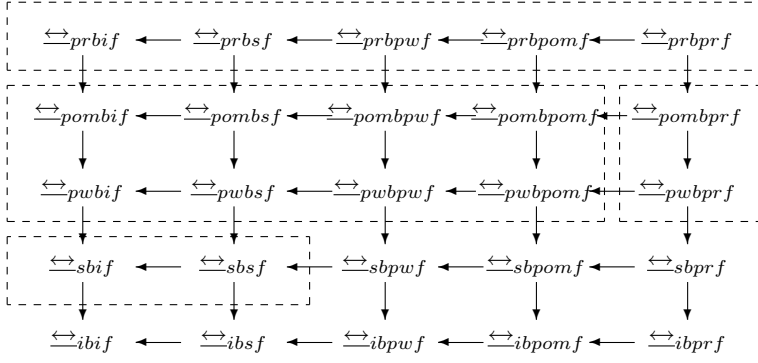


Рис. 1.14: Совпадение обратных-прямых бисимуляционных эквивалентностей

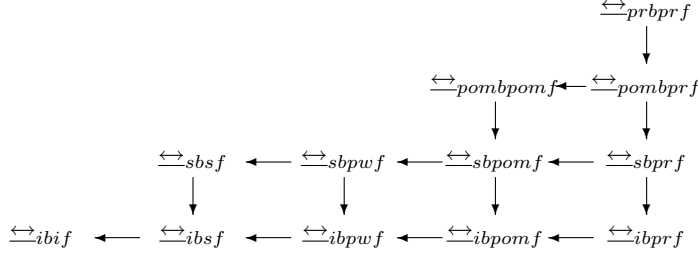


Рис. 1.15: Взаимосвязь обратных-прямых бисимуляционных эквивалентностей

3. Как пункт 2, но роли  $N$  и  $N'$  меняются. □

На рисунке 1.14 пунктирной линией обведены совпадающие обратные-прямые бисимуляционные эквивалентности.

Таким образом, взаимосвязь обратных-прямых бисимуляционных эквивалентностей можно представить рисунком 1.15.

### Сравнение обратных-прямых бисимуляционных эквивалентностей с базисными эквивалентностями

Рассмотрим взаимосвязь обратных-прямых бисимуляционных эквивалентности с базисными эквивалентностями.

**Предложение 1.2.3** Пусть  $\star \in \{i, s, pw, pom, pr\}$ . Для сетей  $N$  и  $N'$   $N \leftrightarrow_{ib\star f} N' \Leftrightarrow N \leftrightarrow_{\star} N'$ .

*Доказательство.* ( $\Leftarrow$ ) Пусть  $\mathcal{R} : N \leftrightarrow_{\star} N'$ . Определим отношение  $\mathcal{S}$  следующим образом:

$\mathcal{S} = \{((\pi, \sigma), (\pi', \sigma')) \mid (\pi, \sigma) \in Runs(N), (\pi', \sigma') \in Runs(N'), |\sigma| = |\sigma'|, l_C(\sigma) = l_C(\sigma'), \forall i (0 \leq i \leq |\sigma|) (\pi(i), \pi'(i)) \in \mathcal{R}\}$ . Очевидно,  $\mathcal{S} : N \leftrightarrow_{ib\star f} N'$ .

( $\Rightarrow$ ) Пусть  $\mathcal{R} : N \leftrightarrow_{ib\star f} N'$ . Определим отношение  $\mathcal{S}$  следующим образом:  $\mathcal{S} = \{(\pi, \pi') \mid ((\pi, \sigma), (\pi', \sigma')) \in \mathcal{R}\}$ . Очевидно,  $\mathcal{S} : N \leftrightarrow_{\star} N'$ . □

Следующее предложение дает характеристику сохраняющих историю бисимуляционных эквивалентностей посредством обратных-прямых эквивалентностей.

**Предложение 1.2.4** Пусть  $\star \in \{pom, pr\}$ . Для сетей  $N$  и  $N'$   $N \leftrightarrow_{b\star f} N' \Leftrightarrow N \leftrightarrow_{\star h} N'$ .

*Доказательство.* См. приложение А. □

Следующее предложение устанавливает новые взаимосвязи ST- и обратных-прямых бисимуляционных эквивалентностей.

**Предложение 1.2.5** Пусть  $\star \in \{pom, pr\}$ . Для сетей  $N$  и  $N'$   $N \leftrightarrow_{\star ST} N' \Rightarrow N \leftrightarrow_{sb\star f} N'$ .

*Доказательство.* Пусть  $\mathcal{R} : N \leftrightarrow_{*ST} N'$ . Определим отношение  $\mathcal{S}$  следующим образом:  
 $\mathcal{S} = \{((\pi, \sigma), (\pi', \sigma')) \mid (\pi, \sigma) \in \text{Runs}(N), (\pi', \sigma') \in \text{Runs}(N'), |\sigma| = |\sigma'|, \forall i (0 \leq i \leq |\sigma|) (ST(\pi(i), \sigma(i)), ST(\pi'(i), \sigma'(i)), \beta_{\sigma(i)}^{\sigma'(i)}) \in \mathcal{R})\}$ . Докажем, что  $\mathcal{S} : N \leftrightarrow_{sb*f} N'$ .

1.  $((\pi_N, \varepsilon), (\pi_{N'}, \varepsilon)) \in \mathcal{S}$ , так как  $ST(\pi_N, \varepsilon) = (\pi_N, \pi_N)$ ,  $ST(\pi_{N'}, \varepsilon) = (\pi_{N'}, \pi_{N'})$ ,  $\beta_\varepsilon^\varepsilon = \emptyset$  и  $((\pi_N, \pi_N), (\pi_{N'}, \pi_{N'}), \emptyset) \in \mathcal{R}$ .

2. Пусть  $((\pi, \sigma), (\pi', \sigma')) \in \mathcal{R}$ .

• (обратно)

Пусть  $(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\hat{\pi}} (\pi, \sigma)$  и  $\prec_{\hat{C}} = \emptyset$ . По определению  $\mathcal{S}$ ,  $\exists (\tilde{\pi}', \tilde{\sigma}')$  такое, что  $|\tilde{\sigma}| = |\tilde{\sigma}'|$ ,  $(\tilde{\pi}', \tilde{\sigma}') \xrightarrow{\hat{\pi}'} (\pi', \sigma')$  и  $((\tilde{\pi}, \tilde{\sigma}), (\tilde{\pi}', \tilde{\sigma}')) \in \mathcal{S}$ .

Пусть  $ST(\pi, \sigma) = (\pi, \bar{\pi})$ ,  $ST(\pi', \sigma') = (\pi', \bar{\pi}')$ . Тогда  $((\pi, \bar{\pi}), (\pi', \bar{\pi}'), \beta_\sigma^{\sigma'}) \in \mathcal{R}$ . Если  $\bar{\pi} \xrightarrow{\pi_W} \pi$  и  $\bar{\pi}' \xrightarrow{\pi'_W} \pi'$ , то  $\prec_{C_W} = \emptyset$  и  $\beta_\sigma^{\sigma'}|_{T_{C_W}} : \rho_{C_W} \simeq \rho_{C'_W}$ . Так как  $\bar{\pi} \rightarrow \tilde{\pi} \xrightarrow{\hat{\pi}} \pi$ ,  $\bar{\pi}' \rightarrow \tilde{\pi}' \xrightarrow{\hat{\pi}'} \pi'$ , то  $\prec_{\hat{C}} = \emptyset$  и  $\rho_{\hat{C}} \simeq \rho_{\hat{C}'}$ .

• (прямо)

Пусть  $(\pi, \sigma) \xrightarrow{\hat{\pi}} (\tilde{\pi}, \sigma v_1 \cdots v_n)$ . Тогда  $\exists \pi_i (1 \leq i \leq n) : (\pi, \sigma) \rightarrow (\pi_1, \sigma v_1) \rightarrow \dots \rightarrow (\pi_n, \sigma v_1 \cdots v_n) = (\tilde{\pi}, \sigma v_1 \cdots v_n)$ . Пусть  $ST(\pi_i, \sigma v_1 \cdots v_i) = (\pi_i, \bar{\pi}_i)$ ,  $(1 \leq i \leq n)$ ,  $ST(\pi, \sigma) = (\pi, \bar{\pi})$ ,  $ST(\pi', \sigma') = (\pi', \bar{\pi}')$ . Тогда  $((\pi, \bar{\pi}), (\pi', \bar{\pi}'), \beta_\sigma^{\sigma'}) \in \mathcal{R}$ . Так как  $\mathcal{R} : N \leftrightarrow_{*ST} N'$ ,  $\exists v'_i$ ,  $(\pi'_i, \bar{\pi}'_i) : (\pi', \bar{\pi}') \rightarrow (\pi'_1, \bar{\pi}'_1) \rightarrow \dots \rightarrow (\pi'_n, \bar{\pi}'_n) = (\tilde{\pi}', \bar{\pi}'_n)$ , где  $(\pi', \sigma') \rightarrow (\pi'_1, \sigma' v'_1) \rightarrow \dots \rightarrow (\pi'_n, \sigma' v'_1 \cdots v'_n) = (\tilde{\pi}', \sigma' v'_1 \cdots v'_n)$  и  $((\pi_i, \bar{\pi}_i), (\pi'_i, \bar{\pi}'_i), \beta_{\sigma v_1 \cdots v_i}^{\sigma' v'_1 \cdots v'_i}) \in \mathcal{R} (1 \leq i \leq n)$ . Следовательно, для некоторого  $\hat{\pi}'$   $(\pi', \sigma') \xrightarrow{\hat{\pi}'} (\tilde{\pi}', \sigma' v'_1 \cdots v'_n)$ .

Докажем, что  $\bar{\pi}'_i = \text{Past}(\tilde{\pi}', \sigma' v'_1 \cdots v'_n)$   $(1 \leq i \leq n)$ . Пусть  $\bar{\pi}_{i-1} \xrightarrow{\pi_i} \bar{\pi}_i$ ,  $\bar{\pi}'_{i-1} \xrightarrow{\pi'_i} \bar{\pi}'_i$   $(1 \leq i \leq n)$ . Тогда, так как  $\mathcal{R} : N \leftrightarrow_{*ST} N'$ , имеем  $\beta_{\sigma v_1 \cdots v_n}^{\sigma' v'_1 \cdots v'_n}|_{T_{\bar{C}_i}} : \rho_{\bar{C}_i} \simeq \rho_{\bar{C}'_i}$  и  $\beta_{\sigma v_1 \cdots v_n}^{\sigma' v'_1 \cdots v'_n}(T_{C_{i-1}}) = T_{C'_{i-1}}$   $(1 \leq i \leq n)$ . Отсюда следует требуемое равенство.

Заметим, что тогда  $((\tilde{\pi}, \sigma v_1 \cdots v_n), (\tilde{\pi}', \sigma' v'_1 \cdots v'_n)) \in \mathcal{S}$ .

Пусть  $\bar{\pi} \xrightarrow{\pi} \tilde{\pi}$ ,  $\bar{\pi}' \xrightarrow{\pi'} \tilde{\pi}'$ . Так как  $\mathcal{R} : N \leftrightarrow_{*ST} N'$ , имеем:

- $\beta_{\sigma v_1 \cdots v_n}^{\sigma' v'_1 \cdots v'_n}|_{T_{\bar{C}}} : \rho_{\bar{C}} \simeq \rho_{\bar{C}'}$ , если  $\star \in \{pot, pr\}$ ;
- $\bar{C} \simeq \bar{C}'$ , если  $\star = pr$ .

Так как  $\bar{\pi} \rightarrow \pi \xrightarrow{\pi} \tilde{\pi}$ ,  $\bar{\pi}' \rightarrow \pi' \xrightarrow{\pi'} \tilde{\pi}'$ , то

- $\rho_{\hat{C}} \simeq \rho_{\hat{C}'}$ , если  $\star = pot$ ;
- $\hat{C} \simeq \hat{C}'$ , если  $\star = pr$ .

3. Как пункт 2, но роли  $N$  и  $N'$  меняются. □

**Теорема 1.2.2** Пусть  $\leftrightarrow, \leftrightarrow^* \in \{\equiv, \leftrightarrow, \simeq\}$  и  $\star, \star\star \in \{\_, i, s, pw, pot, pr, iST, pwST, potST, prST, pomh, prh, mes, occ, sbsf, sbpwf, sbpomf, sbprf, pombrf\}$ . Для сетей  $N$  и  $N'$   $N \leftrightarrow_{\star} N' \Rightarrow N \leftrightarrow_{\star\star} N'$  тогда и только тогда, когда в графе на рисунке 1.16 существует направленный путь от  $\leftrightarrow_{\star}$  к  $\leftrightarrow_{\star\star}$ .

*Доказательство.*  $(\Leftarrow)$  Следствие теоремы 1.2.1 и следующих замечаний.

- Связь  $\leftrightarrow_{sbpwf} \rightarrow \leftrightarrow_{sbsf}$  — следствие того, что гомоморфизм является изоморфизмом ПЧУМ с пустым отношением предшествования.
- Связь  $\leftrightarrow_{sbpomf} \rightarrow \leftrightarrow_{sbpwf}$  — следствие того, что изоморфизм ПЧУМ является гомоморфизмом.
- Связь  $\leftrightarrow_{sbprf} \rightarrow \leftrightarrow_{sbpomf}$  — следствие того, что ПЧУМ на основе изоморфных С-сетей также изоморфны.
- Связи  $\leftrightarrow_{sb\star f} \rightarrow \leftrightarrow_{\star}$ ,  $\star \in \{s, pw, pot, pr\}$  следуют из определений обратных-прямых бисимуляционных эквивалентностей.
- Связи  $\leftrightarrow_{*ST} \rightarrow \leftrightarrow_{sb\star f}$ ,  $\star \in \{pot, pr\}$  — следствия предложения 1.2.5.

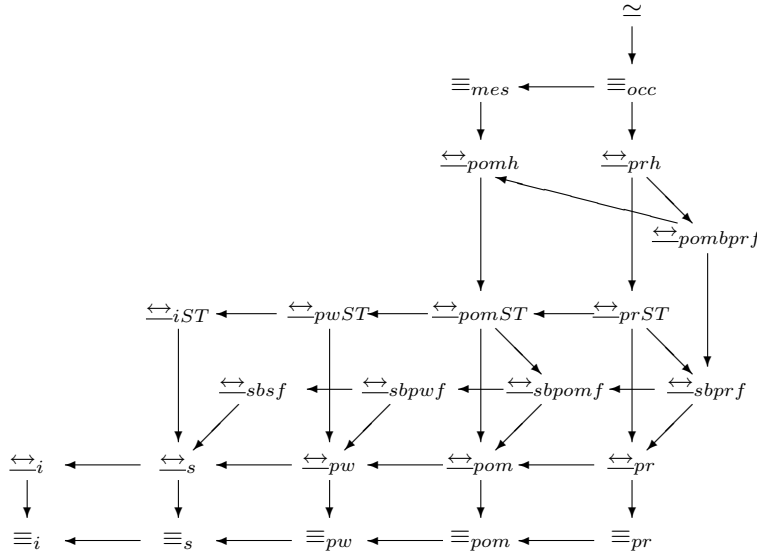


Рис. 1.16: Взаимосвязь обратных-прямых бисимуляционных эквивалентностей с базисными эквивалентностями

- Связь  $\leftrightarrow_{pombprf} \rightarrow \leftrightarrow_{sbprf}$  следует из определения изоморфизма ПЧУМ.
- Связь  $\leftrightarrow_{prh} \rightarrow \leftrightarrow_{pombprf}$  — следствие того, что по предложению 1.2.4  $\leftrightarrow_{prh} = \leftrightarrow_{prbprf}$  и того, что ПЧУМ на основе изоморфных С-сетей также изоморфны.
- Связь  $\leftrightarrow_{pombprf} \rightarrow \leftrightarrow_{pomh}$  — следствие того, что по предложению 1.2.4  $\leftrightarrow_{pomh} = \leftrightarrow_{pombpomf}$  и того, что ПЧУМ на основе изоморфных С-сетей также изоморфны.

( $\Rightarrow$ ) Отсутствие дополнительных нетривиальных стрелок в графе на рисунке 1.16 доказывается теоремой 1.2.1 и следующими примерами на сетях.

- На рисунке 1.12(c)  $N \leftrightarrow_{sbsf} N'$ , но  $N \not\equiv_{pw} N'$ .
- На рисунке 1.17(a)  $N \leftrightarrow_{sbpwf} N'$ , но  $N \not\equiv_{pom} N'$ , так как только в сети  $N'$  действие  $s$  может зависеть от  $a$  и  $b$ .
- На рисунке 1.17(b)  $N \leftrightarrow_{sbprf} N'$ , но  $N \not\equiv_{iST} N'$ , так как только в сети  $N'$  действие  $a$  может начать работать так, что:
  1. до окончания  $a$  не может сработать последовательность действий  $bc$  и
  2. после окончания  $a$  не может сработать действие  $c$ .
- На рисунке 1.17(c)  $N \leftrightarrow_{pombprf} N'$ , но  $N \not\equiv_{prST} N'$ , так как только в сети  $N'$  можно начать выполнять процесс с действием  $a$  так, чтобы его можно было расширить на действие  $b$  только одним способом (то есть так, чтобы расширенный процесс был только один).
- На рисунке 1.12(b)  $N \leftrightarrow_{pwST} N'$ , но  $N \not\equiv_{sbsf} N'$ , так как только в сети  $N'$  можно выполнить последовательность действий  $ab$  так, чтобы  $b$  обязательно зависело от  $a$ .
- На рисунке 1.13(a)  $N \leftrightarrow_{pr} N'$ , но  $N \not\equiv_{sbsf} N'$ , так как только в сети  $N'$  действие  $a$  можно выполнить так, что действие  $b$  обязательно зависит от  $a$ .  $\square$

Заметим, что пример на рисунке 1.17(b) — модификация менее сильного примера, представленного рисунком 1.18, где  $N \leftrightarrow_{sbpomf} N'$ , но  $N \not\equiv_{iST} N'$ . Операция суммирования сетей “+”, определенная, например, в [161], “перемножает” входные места сетей с сохранением всех дуг, выходящих из перемножаемых мест, и объединяет остальные части исходных сетей.

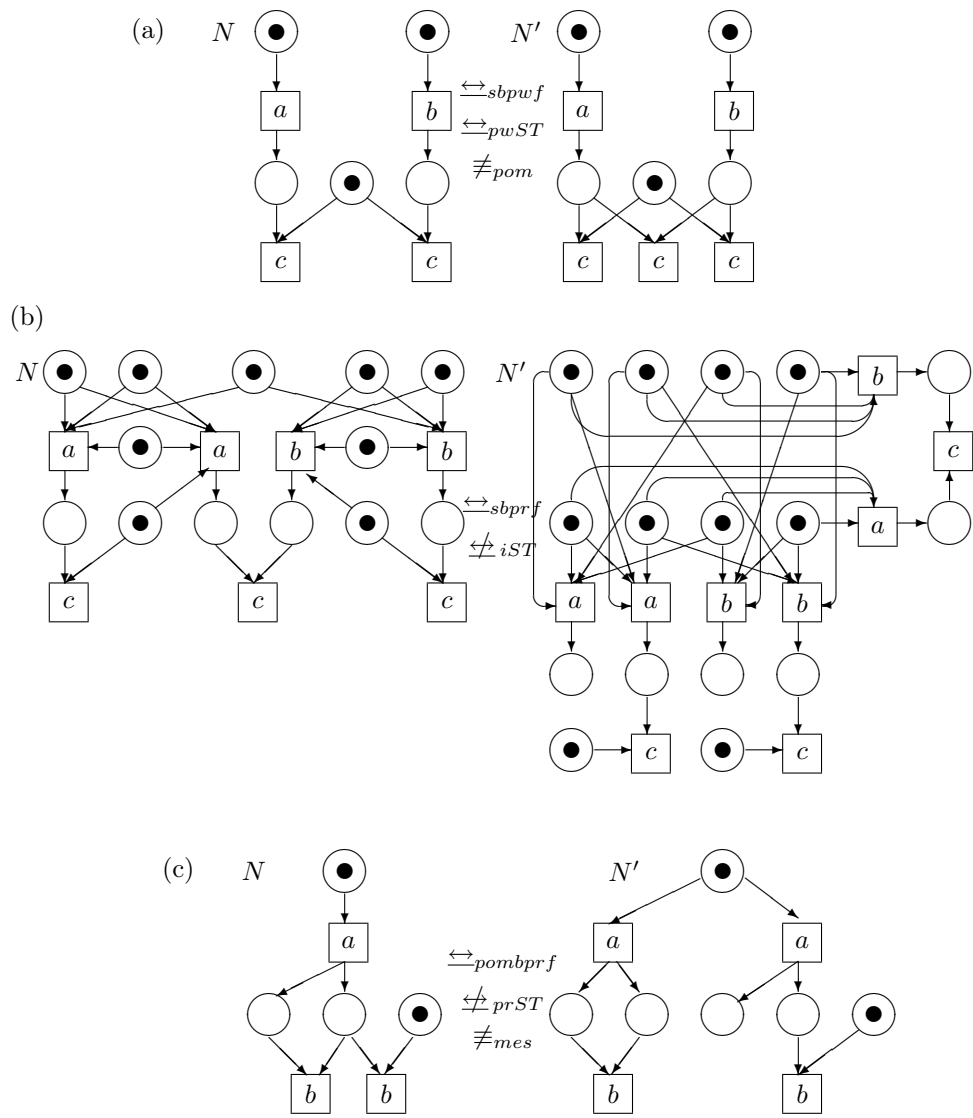


Рис. 1.17: Примеры обратных-прямых бисимуляционных эквивалентностей

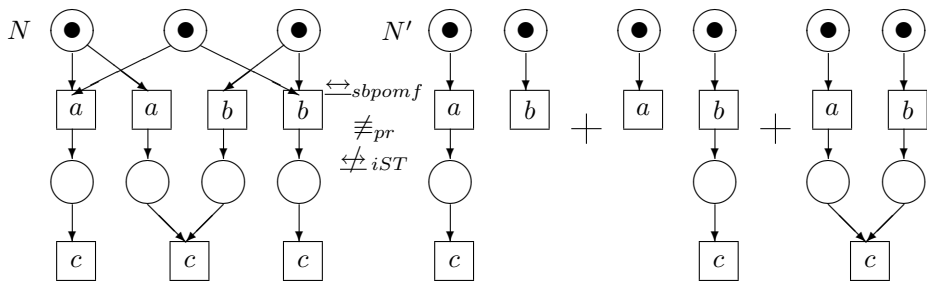


Рис. 1.18: Более ясный, но менее сильный пример обратных-прямых бисимуляционных эквивалентностей

## Логическая характеристика

**Логика HML** Логика Хеннеси-Милнера (HML) была введена в [171] на модели систем переходов для логического описания интерливинговой бисимуляционной эквивалентности.

**Определение 1.2.15** Пусть символ  $\top$  обозначает тождественно истинную формулу и  $a \in Act$ . Формула HML определяется следующим образом:

$$\Phi ::= \top \mid \neg\Phi \mid \Phi \wedge \Psi \mid \langle a \rangle \Phi$$

Определим  $[a]\Phi = \neg\langle a \rangle\neg\Phi$ .

Обозначим через **HML** множество всех формул логики HML.

**Определение 1.2.16** Пусть  $N$  — некоторая сеть и  $\pi \in \Pi(N)$ . Отношение выполнимости  $\models_N \in \Pi(N) \times \mathbf{HML}$  определяется следующим образом:

1.  $\pi \models_N \top$  — всегда;
2.  $\pi \models_N \neg\Phi$ , если  $\pi \not\models_N \Phi$ ;
3.  $\pi \models_N \Phi \wedge \Psi$ , если  $\pi \models_N \Phi$  и  $\pi \models_N \Psi$ ;
4.  $\pi \models_N \langle a \rangle \Phi$ , если  $\exists \tilde{\pi} \in \Pi(N)$   $\pi \xrightarrow{a} \tilde{\pi}$  и  $\tilde{\pi} \models_N \Phi$ .

**Определение 1.2.17** Пишем  $N \models_N \Phi$ , если  $\pi_N \models_N \Phi$ . Сети  $N$  и  $N'$  логически эквивалентны в HML, запись  $N =_{HML} N'$ , если  $\forall \Phi \in \mathbf{HML}$   $N \models_N \Phi \Leftrightarrow N' \models_{N'} \Phi$ .

Пусть  $N$  — некоторая сеть и  $\pi \in \Pi(N)$ ,  $a \in Act$ . Множество расширений процесса  $\pi$  на действие  $a$  (образов) определяется следующим образом:  $Image(\pi, a) = \{\tilde{\pi} \mid \pi \xrightarrow{a} \tilde{\pi}\}$ . Сеть  $N$  — конечно-образная, если  $\forall \pi \in \Pi(N) \forall a \in Act \ |Image(\pi, a)| < \infty$ .

**Теорема 1.2.3** [171] Для конечно-образных сетей  $N$  и  $N'$   $N \leftrightarrow_i N' \Leftrightarrow N \leftrightarrow_{ibif} N' \Leftrightarrow N =_{HML} N'$ .

**Пример 1.2.1** На рисунке 1.12(e)  $N \equiv_{pr} N'$ , но  $N \neq_{HML} N'$ , так как для  $\Phi = [a]\langle b \rangle \top$   $N \models_N \Phi$ , но  $N' \not\models_{N'} \Phi$  из-за того, что только в сети  $N'$  можно выполнить действие  $a$  так, что после него нельзя выполнить  $b$ .

**Логика PBFL** ЧУММ обратная-прямая логика (PBFL) была введена в [95] на модели структур событий для логического описания ЧУММ обратной ЧУММ прямой бисимуляционной эквивалентности.

**Определение 1.2.18** Пусть символ  $\top$  обозначает тождественно истинную формулу и  $a \in Act$ ,  $\rho$  — ЧУММ с пометкой в Act. Формула PBFL определяется следующим образом:

$$\Phi ::= \top \mid \neg\Phi \mid \Phi \wedge \Psi \mid \langle \leftarrow \rho \rangle \Phi \mid \langle a \rangle \Phi$$

Определим  $[a]\Phi = \neg\langle a \rangle\neg\Phi$  и  $[\leftarrow \rho]\Phi = \neg\langle \leftarrow \rho \rangle\neg\Phi$ .

Обозначим через **PBFL** множество всех формул логики PBFL.

Заметим, что в формуле  $\langle a \rangle \Phi$ , соответствующей случаю прямого расширения, стоит действие  $a$ , а не ЧУММ  $\rho$ , так как  $\leftrightarrow_{prombif} = \leftrightarrow_{prombrmf}$ . Поэтому достаточно рассматривать прямые расширения только на одно действие.

**Определение 1.2.19** Пусть  $N$  — некоторая сеть и  $(\pi, \sigma) \in Runs(N)$ . Отношение выполнимости  $\models_N \in Runs(N) \times \mathbf{PBFL}$  определяется следующим образом:

1.  $(\pi, \sigma) \models_N \top$  — всегда;
2.  $(\pi, \sigma) \models_N \neg\Phi$ , если  $(\pi, \sigma) \not\models_N \Phi$ ;
3.  $(\pi, \sigma) \models_N \Phi \wedge \Psi$ , если  $(\pi, \sigma) \models_N \Phi$  и  $(\pi, \sigma) \models_N \Psi$ ;
4.  $(\pi, \sigma) \models_N \langle \leftarrow \rho \rangle \Phi$ , если  $\exists (\tilde{\pi}, \tilde{\sigma}) \in Runs(N)$   $(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\hat{\pi}} (\pi, \sigma)$ , где  $\hat{\pi} = (\hat{C}, \hat{\varphi})$ ,  $\rho_{\hat{C}} \in \rho$  и  $(\tilde{\pi}, \tilde{\sigma}) \models_N \Phi$ ;

5.  $(\pi, \sigma) \models_N \langle a \rangle \Phi$ , если  $\exists(\tilde{\pi}, \tilde{\sigma}) \in \text{Runs}(N)$   $(\pi, \sigma) \xrightarrow{\hat{\pi}} (\tilde{\pi}, \tilde{\sigma})$ , где  $\hat{\pi} = (\hat{C}, \hat{\varphi})$ ,  $l_{\hat{C}}(T_{\hat{C}}) = a$  и  $(\tilde{\pi}, \tilde{\sigma}) \models_N \Phi$ .

**Определение 1.2.20** Пишем  $N \models_N \Phi$ , если  $(\pi_N, \varepsilon) \models_N \Phi$ . Сети  $N$  и  $N'$  логически эквивалентны в  $PBFL$ , запись  $N =_{PBFL} N'$ , если  $\forall \Phi \in \mathbf{PBFL} N \models_N \Phi \Leftrightarrow N' \models_{N'} \Phi$ .

**Теорема 1.2.4** [95] Для конечно-образных сетей  $N$  и  $N'$   $N \xleftrightarrow{\text{pomh}} N' \Leftrightarrow N \xleftrightarrow{\text{pomh}} N' \Leftrightarrow N =_{PBFL} N'$ .

**Пример 1.2.2** На рисунке 1.13(b)  $N =_{HML} N'$ , но  $N \neq_{PBFL} N'$ , так как для  $\Phi = [a][b]\langle c \rangle \langle \leftarrow (a; b) \parallel c \rangle \top$  ( $(a; b) \parallel c$  обозначает ЧУММ, в котором действие  $b$  зависит от  $a$ , и  $a, b$  независимы с действием  $c$ ),  $N \models_N \Phi$ , но  $N' \not\models_{N'} \Phi$  из-за того, что только в сети  $N'$  после действия  $a$  действие  $b$  может сработать так, что действие  $c$  должно обязательно зависеть от  $a$ .

**Логика PrBFL** Введем новую процессную обратную-прямую логику ( $PrBFL$ ) для логического описания процессной обратной процессной прямой бисимуляционной эквивалентности.

**Определение 1.2.21** Пусть символ  $\top$  обозначает тождественно истинную формулу и  $a \in \text{Act}$ , а  $\mathbf{C}$  — класс изоморфизма  $\mathbf{C}$ -сети  $\mathbf{C}$ . Формула  $PrBFL$  определяется следующим образом:

$$\Phi ::= \top \mid \neg \Phi \mid \Phi \wedge \Psi \mid \langle \leftarrow \mathbf{C} \rangle \Phi \mid \langle a \rangle \Phi$$

Определим  $[a]\Phi = \neg \langle a \rangle \neg \Phi$  и  $[\leftarrow \mathbf{C}]\Phi = \neg \langle \leftarrow \mathbf{C} \rangle \neg \Phi$ .

Обозначим через  $\mathbf{PrBFL}$  множество всех формул логики  $PrBFL$ .

Заметим, что в формуле  $\langle a \rangle \Phi$ , соответствующей случаю прямого расширения, стоит действие  $a$ , а не класс изоморфизма  $\mathbf{C}$ , так как  $\xleftrightarrow{\text{prbif}} = \xleftrightarrow{\text{prbprf}}$ . Поэтому достаточно рассматривать прямые расширения только на одно действие.

**Определение 1.2.22** Пусть  $N$  — некоторая сеть и  $(\pi, \sigma) \in \text{Runs}(N)$ . Отношение выполнимости  $\models_N \subseteq \text{Runs}(N) \times \mathbf{PrBFL}$  определяется следующим образом:

1.  $(\pi, \sigma) \models_N \top$  — всегда;
2.  $(\pi, \sigma) \models_N \neg \Phi$ , если  $(\pi, \sigma) \not\models_N \Phi$ ;
3.  $(\pi, \sigma) \models_N \Phi \wedge \Psi$ , если  $(\pi, \sigma) \models_N \Phi$  и  $(\pi, \sigma) \models_N \Psi$ ;
4.  $(\pi, \sigma) \models_N \langle \leftarrow \mathbf{C} \rangle \Phi$ , если  $\exists(\tilde{\pi}, \tilde{\sigma}) \in \text{Runs}(N)$   $(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\hat{\pi}} (\pi, \sigma)$ , где  $\hat{\pi} = (\hat{C}, \hat{\varphi})$ ,  $\hat{C} \in \mathbf{C}$  и  $(\tilde{\pi}, \tilde{\sigma}) \models_N \Phi$ ;
5.  $(\pi, \sigma) \models_N \langle a \rangle \Phi$ , если  $\exists(\tilde{\pi}, \tilde{\sigma}) \in \text{Runs}(N)$   $(\pi, \sigma) \xrightarrow{\hat{\pi}} (\tilde{\pi}, \tilde{\sigma})$ , где  $\hat{\pi} = (\hat{C}, \hat{\varphi})$ ,  $l_{\hat{C}}(T_{\hat{C}}) = a$  и  $(\tilde{\pi}, \tilde{\sigma}) \models_N \Phi$ .

**Определение 1.2.23** Пишем  $N \models_N \Phi$ , если  $(\pi_N, \varepsilon) \models_N \Phi$ . Сети  $N$  и  $N'$  логически эквивалентны в  $PrBFL$ , запись  $N =_{PrBFL} N'$ , если  $\forall \Phi \in \mathbf{PrBFL} N \models_N \Phi \Leftrightarrow N' \models_{N'} \Phi$ .

**Теорема 1.2.5** Для конечно-образных сетей  $N$  и  $N'$   $N \xleftrightarrow{\text{prh}} N' \Leftrightarrow N \xleftrightarrow{\text{prbprf}} N' \Leftrightarrow N =_{PrBFL} N'$ .

*Доказательство.* Так как по предложению 1.2.4  $\xleftrightarrow{\text{prh}} = \xleftrightarrow{\text{prbprf}}$ , остается доказать совпадение  $\xleftrightarrow{\text{prbprf}}$  и  $=_{PrBFL}$ .

( $\Leftarrow$ ) Пусть  $N =_{PrBFL} N'$ . Так как по предложению 1.2.2  $\xleftrightarrow{\text{prbprf}} = \xleftrightarrow{\text{prbif}}$ , достаточно доказать  $N \xleftrightarrow{\text{prbif}} N'$ . Определим отношение  $\mathcal{R} = \{((\pi, \sigma), (\pi', \sigma')) \mid (\pi, \sigma) \in \text{Runs}(N), (\pi', \sigma') \in \text{Runs}(N'), (\pi, \sigma) =_{PrBFL} (\pi', \sigma')\}$ . Докажем  $\mathcal{R} : N \xleftrightarrow{\text{prbif}} N'$ .

1.  $((\pi_N, \varepsilon), (\pi_{N'}, \varepsilon)) \in \mathcal{R}$ , так как  $N =_{PrBFL} N'$ .
2. Пусть  $((\pi, \sigma), (\pi', \sigma')) \in \mathcal{R}$ .
  - (обратно) Аналогично (и более просто) случаю прямого расширения.
  - (прямо) Пусть  $(\pi, \sigma) \xrightarrow{a} (\tilde{\pi}, \tilde{\sigma})$ . Пусть  $\Phi_1, \Phi_2, \dots$  — перечисление формул  $PrBFL$ . Определим формулы  $\Psi_i$  ( $i \in \mathbf{N}$ ) следующим образом:
    - (а)  $\Psi_0 = \top$ ;

$$(b) \Psi_{i+1} = \begin{cases} \Psi_i \wedge \Phi_i, & \text{если } (\tilde{\pi}, \tilde{\sigma}) \models_N \Phi_i; \\ \Psi_i \wedge \neg \Phi_i, & \text{если } (\tilde{\pi}, \tilde{\sigma}) \models_N \neg \Phi_i. \end{cases}$$

По конструкции,  $\forall i \in \mathbf{N} (\pi, \sigma) \models_N \langle a \rangle \Psi_i$ . Так как  $((\pi, \sigma), (\pi', \sigma')) \in \mathcal{R}$ , имеем  $\forall i \in \mathbf{N} \exists (\pi'_i, \sigma'_i)$   $(\pi', \sigma') \xrightarrow{a} (\pi'_i, \sigma'_i)$  и  $\forall i \in \mathbf{N} (\pi'_i, \sigma'_i) \models_{N'} \Psi_i$ .

Так как  $N$  — конечно-образная сеть, число ее прямых расширений (образов) конечно. Следовательно, конечно и число  $(\pi'_i, \sigma'_i)$ . Поэтому  $\exists (\tilde{\pi}', \tilde{\sigma}')$   $(\pi', \sigma') \xrightarrow{a} (\tilde{\pi}', \tilde{\sigma}')$  и  $\forall i \in \mathbf{N} (\tilde{\pi}', \tilde{\sigma}') \models_{N'} \Psi_i$ . Следовательно,  $((\tilde{\pi}, \tilde{\sigma}), (\tilde{\pi}', \tilde{\sigma}')) \in \mathcal{R}$ .

3. Как пункт 2, но роли  $N$  и  $N'$  меняются.

( $\Rightarrow$ ) Пусть  $\mathcal{R} : N \xleftrightarrow{prbprf} N'$ . По предложению 1.2.2,  $\mathcal{R} : N \xleftrightarrow{prbif} N'$ . Докажем индукцией по структуре формул  $PrBFL$ , что  $((\pi, \sigma), (\pi', \sigma')) \in \mathcal{R}$  влечет  $(\pi, \sigma) =_{PrBFL} (\pi', \sigma')$ .

Случаи  $\top, \neg \Phi, \Phi \wedge \Psi$  — тривиальны.

Предположим,  $(\pi, \sigma) \models_N \langle \leftarrow \mathbf{C} \rangle \Phi$ . Тогда  $\exists (\tilde{\pi}', \tilde{\sigma}') (\tilde{\pi}', \tilde{\sigma}') \xrightarrow{\hat{\pi}'} (\pi', \sigma') \hat{\pi}' = (\hat{C}', \hat{\varphi}')$ ,  $\hat{C}' \in \mathbf{C}$  и  $((\tilde{\pi}, \tilde{\sigma}), (\tilde{\pi}', \tilde{\sigma}')) \in \mathcal{R}$ . По индукционному предположению,  $(\tilde{\pi}', \tilde{\sigma}') \models_{N'} \Phi$ . Следовательно,  $(\pi', \sigma') \models_{N'} \langle \leftarrow \mathbf{C} \rangle \Phi$ . Наконец,  $((\pi_N, \varepsilon), (\pi_{N'}, \varepsilon)) \in \mathcal{R}$  влечет  $N =_{PrBFL} N'$ .

Случай  $\langle a \rangle \Phi$  рассматривается аналогично.  $\square$

**Пример 1.2.3** На рисунке 1.17(c)  $N =_{PBFL} N'$ , но  $N \neq_{PrBFL} N'$ , так как для  $\Phi = [a][b]\langle c \rangle \langle \leftarrow \mathbf{C} \rangle \top$  ( $\mathbf{C}$  обозначает класс изоморфизма  $C$ -сети, в которой оба выходных места  $a$ -помеченного перехода являются входными для перехода с пометкой  $b$ ),  $N \models_N \Phi$ , но  $N' \not\models_{N'} \Phi$  из-за того, что только в сети  $N'$  можно начать выполнять процесс с действием  $a$  так, чтобы его можно было расширить на действие  $b$  только одним способом (то есть так, чтобы расширенный процесс был только один).

### 1.2.3 Эквивалентности мест

Бисимуляционные эквивалентности мест были введены в [8] на основе определения из [223, 224, 225]. Бисимуляции мест — отношения не на маркировках или процессах, а на местах сетей. Отношение на маркировках получается с помощью операции “поднятия” отношения на местах. В [8, 9] введена интерливинговая бисимуляционная эквивалентность мест ( $\sim_i$ ). В этой работе определяется также интерливинговая строгая бисимуляционная эквивалентность мест ( $\approx_i$ ) с помощью введения дополнительного требования о том, что соответствующие переходы сетей должны быть также (как и маркировки) связаны отношением бисимуляции. Ставится вопрос о возможности введения сохраняющей историю бисимуляционной эквивалентности мест.

В [23, 24, 21, 22] были введены шаговая ( $\sim_s$ ), ЧС ( $\sim_{pw}$ ), ЧУММ ( $\sim_{pom}$ ), процессная ( $\sim_{pr}$ ) бисимуляционные эквивалентности мест и их строгие аналоги ( $\approx_s, \approx_{pw}, \approx_{pom}, \approx_{pr}$ ). Установлено, что  $\sim_i, \sim_s$  и  $\sim_{pw}$  совпадают. Также показано, что все строгие бисимуляционные эквивалентности мест совпадают с  $\sim_{pr}$ . Таким образом, остаются три не совпадающих между собой отношения:  $\sim_i, \sim_{pom}$  и  $\sim_{pr}$ . В этих работах предложен также полиномиальный алгоритм упрощения сетей с сохранением их поведения (то есть исходная и редуцированная сети бисимуляционно эквивалентны).

В данном разделе проводится сравнение бисимуляционных эквивалентностей мест с обратными-прямыми бисимуляционными и базисными эквивалентностями. Таким образом, мы расширяем результаты [23, 24].

В частности, мы доказываем, что  $\sim_{pr}$  влечет  $\xleftrightarrow{prh}$  и отвечаем на вопрос из [8]:  $\sim_{pr}$  достаточно сильна, чтобы сохранять “истории” выполнения сетей, поэтому нет смысла определять сохраняющую историю бисимуляционную эквивалентность мест.

Так как ST- и сохраняющие историю бисимуляционные эквивалентности являются следствиями  $\sim_{pr}$ , алгоритм упрощения сетей из [23, 24], основанный на этой эквивалентности, сохраняет *временные следы* [161] исходной сети (так как ST-бисимуляционные эквивалентности *соответствуют реальному времени* [161]) и “истории” ее функционирования (так как сохраняющие историю эквивалентности учитывают “прошлое” процессов).

### Бисимуляционные эквивалентности мест

Обычные бисимуляции можно определить на основе маркировок (вместо процессов), заменив в определениях процессы на соответствующие им маркировки.



**Определение 1.2.24** Пусть  $N$  и  $N'$  — некоторые сети. Отношение  $\mathcal{R} \subseteq \mathcal{M}(N) \times \mathcal{M}(N')$  —  $\star$ -бисимуляция между  $N$  и  $N'$ ,  $\star \in \{\text{интерливингивая, шаговая, ЧС, ЧУММ, процессная}\}$ , запись  $\mathcal{R} : N \xleftrightarrow{\star} N'$ ,  $\star \in \{i, s, pw, pot, pr\}$ , если:

1.  $(M_N, M_{N'}) \in \mathcal{R}$ .
2.  $(M, M') \in \mathcal{R}$ ,  $M \xrightarrow{\hat{\pi}} \widetilde{M}$ ,
  - (a)  $|T_{\widehat{C}}| = 1$ , если  $\star = i$ ;
  - (b)  $\prec_{\widehat{C}} = \emptyset$ , если  $\star = s$ ; $\Rightarrow \exists \widetilde{M}' : M' \xrightarrow{\hat{\pi}'} \widetilde{M}'$ ,  $(\widetilde{M}, \widetilde{M}') \in \mathcal{R}$  и
  - (a)  $\rho_{\widehat{C}} \sqsubseteq \rho_{\widehat{C}'}$ , если  $\star = pw$ ;
  - (b)  $\rho_{\widehat{C}} \simeq \rho_{\widehat{C}'}$ , если  $\star \in \{i, s, pot\}$ ;
  - (c)  $\widehat{C} \simeq \widehat{C}'$ , если  $\star = pr$ .

3. Как пункт 2, но роли  $N$  и  $N'$  меняются.

Сети  $N$  и  $N'$   $\star$ -бисимуляционно эквивалентны,  $\star \in \{\text{интерливингивая, шаговая, ЧС, ЧУММ, процессная}\}$ , запись  $N \xleftrightarrow{\star} N'$ , если  $\exists \mathcal{R} : N \xleftrightarrow{\star} N'$ ,  $\star \in \{i, s, pw, pot, pr\}$ .

Бисимуляции мест связывают не маркировки, а места сетей. Отношение на маркировках получается с помощью операции “поднятия” отношения бисимуляции на местах.

Перед введением операции поднятия заметим, что в определениях бисимуляций на основе маркировок будут использоваться любые маркировки сетей, не обязательно достижимые. Как замечено в [23, 24], это не изменяет отношения бисимуляционной эквивалентности.

**Определение 1.2.25** Пусть  $N$  и  $N'$  — некоторые сети и  $\mathcal{R} \subseteq P_N \times P_{N'}$  — отношение на местах. Поднятие отношения  $\mathcal{R}$  — отношение  $\overline{\mathcal{R}} \subseteq \mathcal{M}(P_N) \times \mathcal{M}(P_{N'})$ , определяемое следующим образом:

$$(M, M') \in \overline{\mathcal{R}} \Leftrightarrow \begin{cases} \exists \{(p_1, p'_1), \dots, (p_n, p'_n)\} \in \mathcal{M}(\mathcal{R}) : \\ M = \{p_1, \dots, p_n\}, M' = \{p'_1, \dots, p'_n\} \end{cases}$$

**Определение 1.2.26** Пусть  $N$  и  $N'$  — некоторые сети. Отношение  $\mathcal{R} \subseteq P_N \times P_{N'}$  —  $\star$ -бисимуляция мест между  $N$  и  $N'$ ,  $\star \in \{\text{интерливингивая, шаговая, ЧС, ЧУММ, процессная}\}$ , запись  $\mathcal{R} : N \sim_{\star} N'$ , если  $\overline{\mathcal{R}} : N \xleftrightarrow{\star} N'$ ,  $\star \in \{i, s, pw, pot, pr\}$ .

Сети  $N$  и  $N'$   $\star$ -бисимуляционно эквивалентны на местах,  $\star \in \{\text{интерливингивая, шаговая, ЧС, ЧУММ, процессная}\}$ , запись  $N \sim_{\star} N'$ , если  $\exists \mathcal{R} : N \sim_{\star} N'$ ,  $\star \in \{i, s, pw, pot, pr\}$ .

Строгие бисимуляционные эквивалентности мест определяются с помощью введения дополнительного требования о том, что соответствующие переходы сетей должны быть также (как и маркировки) связаны отношением  $\overline{\mathcal{R}}$ . Это отношение определяется на переходах следующим образом.

**Определение 1.2.27** Пусть  $N$  и  $N'$  — некоторые сети и  $t \in T_N$ ,  $t' \in T_{N'}$ . Тогда

$$(t, t') \in \overline{\mathcal{R}} \Leftrightarrow \begin{cases} (\bullet t, \bullet t') \in \overline{\mathcal{R}} \wedge \\ (t \bullet, t' \bullet) \in \overline{\mathcal{R}} \wedge \\ l_N(t) = l_{N'}(t') \end{cases}$$

**Определение 1.2.28** Отношение  $\mathcal{R} \subseteq P_N \times P_{N'}$  — строгая  $\star$ -бисимуляция мест между  $N$  и  $N'$ ,  $\star \in \{\text{интерливингивая, шаговая, ЧС, ЧУММ, процессная}\}$ , запись  $\mathcal{R} : N \approx_{\star} N'$ ,  $\star \in \{i, s, pw, pot, pr\}$ , если:

1.  $\overline{\mathcal{R}} : N \xleftrightarrow{\star} N'$ .
2. В определении  $\star$ -бисимуляции в пункт 2 (и симметрично в пункт 3) добавляется новое требование:  $\forall v \in T_{\widehat{C}} (\hat{\varphi}(v), \hat{\varphi}'(\beta(v))) \in \overline{\mathcal{R}}$ , где:
  - (a)  $\beta : \rho_{\widehat{C}} \sqsubseteq \rho_{\widehat{C}'}$ , если  $\star = pw$ ;

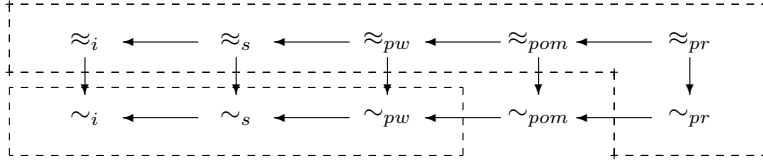


Рис. 1.19: Совпадение бисимуляционных эквивалентностей мест

$$\sim_i \longleftarrow \sim_{pot} \longleftarrow \sim_{pr}$$

Рис. 1.20: Взаимосвязь бисимуляционных эквивалентностей мест

(b)  $\beta : \rho_{\hat{C}} \simeq \rho_{\hat{C}'},$  если  $\star \in \{i, s, pot\};$

(c)  $\beta : \hat{C} \simeq \hat{C}'$ , если  $\star = pr.$

Сети  $N$  и  $N'$  строго  $\star$ -бисимуляционно эквивалентны на местах,  $\star \in \{\text{интерливингивая, шаговая, ЧС, ЧУММ, процессная}\}$ , запись  $N \approx_{\star} N'$ , если  $\exists \mathcal{R} : N \approx_{\star} N'$ ,  $\star \in \{i, s, pw, pot, pr\}$ .

Важное свойство бисимуляций мест — *аддитивность*. Пусть для сетей  $N$  и  $N'$   $\mathcal{R} : N \sim_{\star} N'$ . Тогда из того, что  $(M_1, M'_1) \in \overline{\mathcal{R}}$  и  $(M_2, M'_2) \in \overline{\mathcal{R}}$  следует  $((M_1 + M_2), (M'_1 + M'_2)) \in \overline{\mathcal{R}}$ . В частности, если в места  $p \in P_N$  и  $p' \in P_{N'}$  такие, что  $(p, p') \in \mathcal{R}$ , добавить по  $n$  фишек, то сети, полученные в результате такого изменения начальных маркировок, должны остаться бисимуляционно эквивалентными на местах.

### Сравнение бисимуляционных эквивалентностей мест

Рассмотрим взаимосвязь бисимуляционных эквивалентностей мест.

**Предложение 1.2.6** [23, 24] Для сетей  $N$  и  $N'$ :

1.  $N \sim_i N' \Leftrightarrow N \sim_{pw} N';$
2.  $N \sim_{pr} N' \Leftrightarrow N \sim_i N' \Leftrightarrow N \approx_{pr} N'.$

На рисунке 1.19 пунктирной линией обведены совпадающие бисимуляционные эквивалентности мест.

Таким образом, взаимосвязь бисимуляционных эквивалентностей мест можно представить рисунком 1.20.

### Сравнение бисимуляционных эквивалентностей мест с базисными эквивалентностями и обратными-прямыми бисимуляционными эквивалентностями

Рассмотрим взаимосвязь бисимуляционных эквивалентностей мест с базисными эквивалентностями и обратными-прямыми бисимуляционными эквивалентностями.

Следующее предложение является фундаментальным и представляет один из важнейших новых результатов данного раздела. Оно показывает, что процессная бисимуляционная эквивалентность достаточно сильна, чтобы сохранять историю выполнений.

**Предложение 1.2.7** Для сетей  $N$  и  $N'$   $N \sim_{pr} N' \Rightarrow N \stackrel{\leftrightarrow}{prh} N'.$

*Доказательство.* По предложению 1.2.6,  $\exists \mathcal{R} : N \approx_{pr} N'$ . Тогда  $\overline{\mathcal{R}} : N \stackrel{\leftrightarrow}{pr} N'$  и переходы  $N$  и  $N'$  связаны отношением  $\overline{\mathcal{R}}$ . Определим отношение  $\mathcal{S}$  следующим образом:  $\mathcal{S} = \{(\pi, \pi', \beta) \mid \pi = (C, \varphi) \in \Pi(N), \pi' = (C', \varphi') \in \Pi(N'), \beta = id_{T_C}, \forall r \in P_C (\varphi(r), \varphi'(r)) \in \mathcal{R}, \forall v \in T_C (\varphi(v), \varphi'(v)) \in \overline{\mathcal{R}}\}$ . Докажем, что  $\mathcal{S} : N \stackrel{\leftrightarrow}{prh} N'.$

1. Очевидно,  $(\pi_N, \pi_{N'}, \emptyset) \in \mathcal{S}.$
2. По определению  $\mathcal{S}$ ,  $(\pi, \pi', \beta) \in \mathcal{S} \Rightarrow \beta : \rho_C \simeq \rho_{C'}$  и  $C \simeq C';$

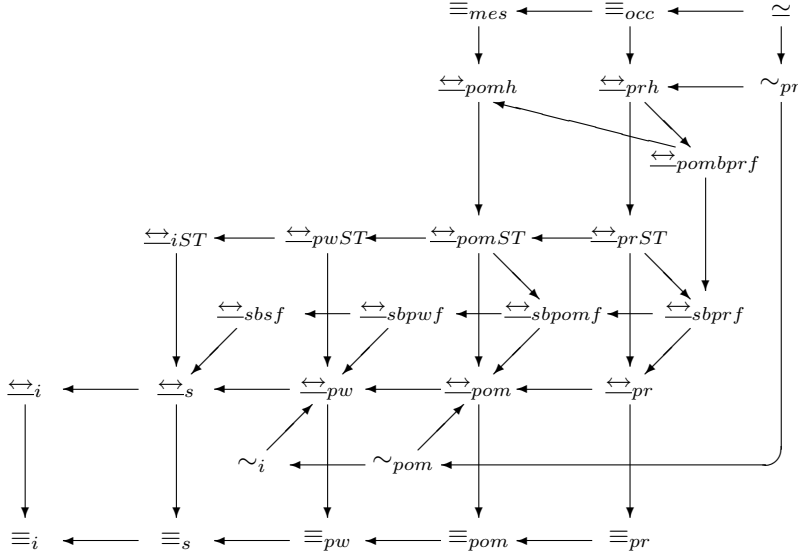


Рис. 1.21: Взаимосвязь бисимуляционных эквивалентностей мест с базисными эквивалентностями и обратными-прямыми бисимуляционными эквивалентностями

3. Пусть  $(\pi, \pi', \beta) \in \mathcal{S}$ ,  $\pi = (C, \varphi)$ ,  $\pi' = (C, \varphi')$  и  $\pi \xrightarrow{v} \tilde{\pi}$ ,  $\tilde{\pi} = (\tilde{C}, \tilde{\varphi})$ .

Рассмотрим срабатывание  $\tilde{\varphi}(\bullet v) \xrightarrow{\tilde{\varphi}(v)} \tilde{\varphi}(v \bullet)$  сети  $N$ . По определению  $\mathcal{S}$ ,  $(\varphi(\bullet v), \varphi'(v \bullet)) \in \overline{\mathcal{R}}$ . Так как  $\varphi(\bullet v) = \tilde{\varphi}(\bullet v)$ , имеем  $(\tilde{\varphi}(\bullet v), \varphi'(v \bullet)) \in \overline{\mathcal{R}}$ .

Так как  $\mathcal{R} : N \approx_{pr} N'$ , имеем  $\exists u', \tilde{M}' : \varphi'(\bullet v) \xrightarrow{u'} \tilde{M}'$ ,  $(\tilde{\varphi}(v), u') \in \overline{\mathcal{R}}$  и  $(\tilde{\varphi}(v \bullet), \tilde{M}') \in \overline{\mathcal{R}}$ .

Пусть  $v \bullet = \{r_1, \dots, r_n\}$ ,  $\tilde{M}' = \{p'_1, \dots, p'_n\}$ ,  $\forall i (1 \leq i \leq n) (\tilde{\varphi}(r_i), p'_i) \in \mathcal{R}$ . Определим отображение  $\tilde{\varphi}'$  следующим образом:

- (a)  $\tilde{\varphi}'|_{(P_C \cup T_C)} = \varphi'$ ;
- (b)  $\tilde{\varphi}'(v) = u'$ ;
- (c)  $\forall i (1 \leq i \leq n) \tilde{\varphi}'(r_i) = p'_i$ .

Так как по определению  $\tilde{\varphi}'$  имеем  $u' = \tilde{\varphi}'(v)$ ,  $\tilde{M}' = \tilde{\varphi}'(v \bullet)$ ,  $\varphi'(\bullet v) = \tilde{\varphi}'(\bullet v)$ , то  $\tilde{\varphi}'(\bullet v) \xrightarrow{\tilde{\varphi}'(v)} \tilde{\varphi}'(v \bullet)$  — срабатывание сети  $N'$  и  $(\tilde{\varphi}(v), \tilde{\varphi}'(v)) \in \overline{\mathcal{R}}$ ,  $(\tilde{\varphi}(v \bullet), \tilde{\varphi}'(v \bullet)) \in \overline{\mathcal{R}}$ .

Следовательно,  $\tilde{\varphi}(\bullet v) - \bullet \tilde{\varphi}(v) = \tilde{\varphi}(v \bullet) - \tilde{\varphi}(v) \bullet$  и  $\tilde{\varphi}'(\bullet v) - \bullet \tilde{\varphi}'(v) = \tilde{\varphi}'(v \bullet) - \tilde{\varphi}'(v) \bullet$ . В силу аддитивности бисимуляций мест и так как  $\tilde{\varphi}$  — встраивание, имеем  $(\emptyset, \tilde{\varphi}'(\bullet v) - \bullet \tilde{\varphi}'(v)) \in \overline{\mathcal{R}}$  и  $(\emptyset, \tilde{\varphi}'(v \bullet) - \tilde{\varphi}'(v) \bullet) \in \overline{\mathcal{R}}$ . Следовательно,  $\tilde{\varphi}'(\bullet v) = \bullet \tilde{\varphi}'(v)$  и  $\tilde{\varphi}'(v \bullet) = \tilde{\varphi}'(v) \bullet$ . Поэтому  $\tilde{\varphi}'$  — встраивание и  $\tilde{\pi}' = (\tilde{C}, \tilde{\varphi}') \in \Pi(N')$ .

Кроме того, имеем  $\pi' \xrightarrow{v} \tilde{\pi}'$ . Определим  $\tilde{\beta} = id_{T_{\tilde{C}}}$ . Тогда  $\tilde{\beta}|_{T_C} = \beta$ . Легко проверить также, что  $(\tilde{\pi}, \tilde{\pi}', \tilde{\beta}) \in \mathcal{S}$ .

4. Как пункт 3, но роли  $N$  и  $N'$  меняются. □

**Теорема 1.2.6** Пусть  $\leftrightarrow, \leftrightarrow^* \in \{\equiv, \leftrightarrow, \sim, \simeq\}$ ,  $\star, \star\star \in \{\_, i, s, pw, pom, pr, iST, pwST, pomST, prST, pomh, prh, mes, occ, sbsf, sbpwf, sbpomf, sbprf, pombprf\}$ . Для сетей  $N$  и  $N'$   $N \leftrightarrow_\star N' \Rightarrow N \leftrightarrow_{\star\star} N'$  тогда и только тогда, когда в графе на рисунке 1.21 существует направленный путь от  $\leftrightarrow_\star$  к  $\leftrightarrow_{\star\star}$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие теоремы 1.2.2 и следующих замечаний.

- Связи  $\sim_\star \rightarrow \leftrightarrow_\star$ ,  $\star \in \{i, pom, pr\}$  следуют из определений.
- Связь  $\sim_{pr} \rightarrow \leftrightarrow_{prh}$  следует из предложения 1.2.7.
- Связь  $\sim_{pom} \rightarrow \sim_i$  следует из определений.

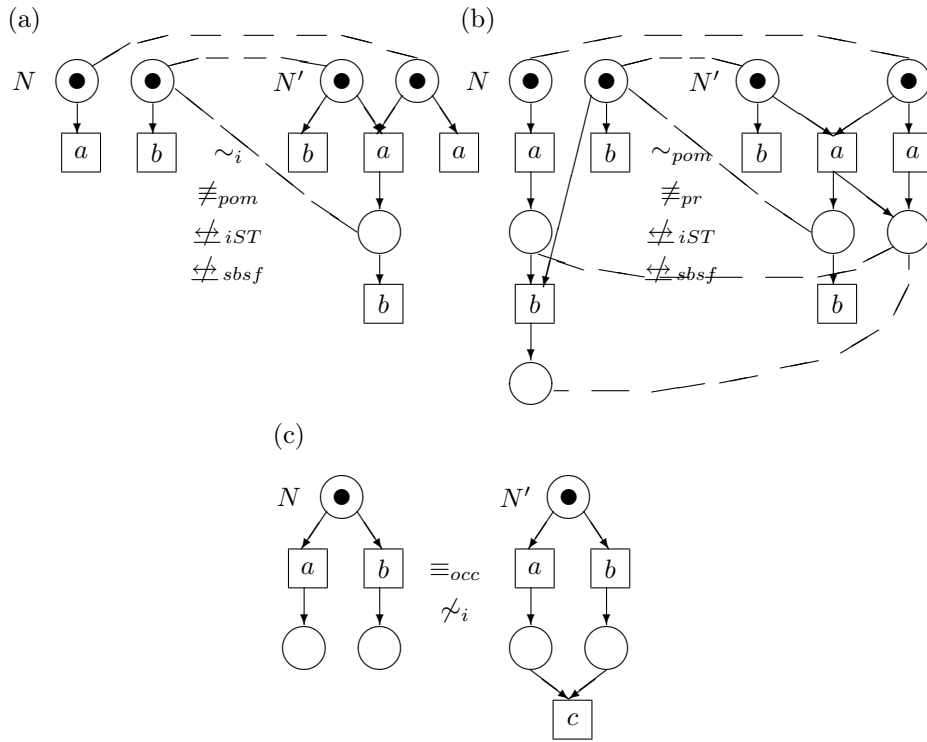


Рис. 1.22: Примеры бисимуляционных эквивалентностей мест

- Связь  $\sim_{pr} \rightarrow \sim_{pom}$  — следствие того, что ПЧУМ на основе изоморфных С-сетей также изоморфны.
- Связь  $\simeq \rightarrow \sim_{pr}$  очевидна.

( $\Rightarrow$ ) Отсутствие дополнительных нетривиальных стрелок в графе на рисунке 1.21 доказывается теоремой 1.2.2 и следующими примерами. Заметим, что пунктирные линии на рисунке 1.22 соединяют места сетей, связанные отношением бисиммуляции мест.

- На рисунке 1.22(a)  $N \sim_i N'$ , но  $N \not\sim_{pom} N'$ , так как только в сети  $N'$  действие  $b$  может зависеть от  $a$ .
- На рисунке 1.22(b)  $N \sim_{pom} N'$ , но  $N \not\sim_{pr} N'$ , так как только в сети  $N'$  у перехода, помеченного действием  $a$ , может быть два входных места (и два выходных места).
- На рисунке 1.22(c)  $N \equiv_{occ} N'$ , но  $N \not\sim_i N'$ , так как любая бисиммуляция мест должна связывать входные места сетей  $N$  и  $N'$ , а если мы добавим в эти места еще по одной фишке, только в  $N'$  может выполняться действие  $c$ .
- На рисунке 1.22(b)  $N \sim_{pom} N'$ , но  $N \not\sim_{iST} N'$ , так как только в сети  $N'$  действие  $a$  может стартовать так, что никакое  $b$  не сможет начать работать, пока  $a$  не завершится.
- На рисунке 1.13(c)  $N \sim_{pr} N'$ , но  $N \not\sim_{mes} N'$ , так только сети  $N'$  соответствует МСС с двумя конфликтными действиями  $a$ .
- На рисунке 1.22(b)  $N \sim_{pom} N'$ , но  $N \not\sim_{sbsf} N'$ , так как только в сети  $N'$  действие  $a$  может выполняться так, что  $b$  обязательно зависит от  $a$ .  $\square$

### Редукция сетей с использованием бисимуляционных эквивалентностей мест

В литературе было предложено много методов редукции сетей с сохранением эквивалентности (например, в [213]). Но многие из них — локальные, не всегда эффективные, или сохраняют достаточно небольшое количество свойств поведения [23, 24, 22].

Бисимуляции мест дают возможность эффективного, глобального и семантически корректного упрощения сетей [23, 24]. Основная идея состоит в рассмотрении так называемых автобисимуляционных эквивалентностей мест, то есть эквивалентностей между сетью и ей же самой, и в последующем слиянии эквивалентных мест заодно с редукцией излишних промежуточных переходов и дуг. В данном разделе мы перечислим основные результаты, полученные в данной области, а также покажем, чем может быть здесь полезна установленная нами взаимосвязь эквивалентностей.

*Автобисимуляция* — бисимуляция между сетью и ей самой.

В некоторых случаях, как, например, для бисимуляции мест, не всякая автобисимуляция является эквивалентностью из-за нарушения рефлексивности.

*Эквивисимуляция* — автобисимуляция, являющаяся эквивалентностью.

Другая проблема состоит в том, что не для всех бисимуляций мест их объединение является таковой, а для упрощения сетей нам хотелось бы использовать максимальную бисимуляцию, объединение всех бисимуляций. Следующее предложение доказывает, что, при условии рефлексивности исходных отношений, удается разрешить эти проблемы.

**Предложение 1.2.8** [23, 24] Если  $\mathcal{R}_1$  и  $\mathcal{R}_2$  — рефлексивные интерливинговые автобисимуляции мест сети  $N$ , то  $(\mathcal{R}_1 \cup \mathcal{R}_2)^*$  (транзитивное замыкание  $(\mathcal{R}_1 \cup \mathcal{R}_2)$ ) — интерливинговая эквивисимуляция мест.

**Определение 1.2.29** Для сети  $N$  отношение  $\mathcal{R}_i(N) = \bigcup \{ \mathcal{R} \mid \mathcal{R} : N \sim_i N \text{ и } \mathcal{R} \text{ рефлексивно} \}$  — каноническая интерливинговая автобисимуляция мест.

**Определение 1.2.30** Пусть для сети  $N$  отношение  $\mathcal{E} \subseteq P_N \times P_N$  — эквивалентность.

Для  $p \in P_N$  определим  $[p]_{\mathcal{E}} = \{ q \mid (p, q) \in \mathcal{E} \}$  — класс эквивалентности  $p$  относительно  $\mathcal{E}$ .

Для  $M \in \mathcal{M}(P_N)$  определим  $M/\mathcal{E} = \sum_{p \in P_N} [p]_{\mathcal{E}}$  — категоризация (разбиение)  $M$  относительно  $\mathcal{E}$ .

Определим  $N/\mathcal{E} = \langle P_N/\mathcal{E}, T_N, F_N/\mathcal{E}, l_N, M_N/\mathcal{E} \rangle$ , где  $F_N/\mathcal{E}$  вводится следующим образом:

1.  $\bullet t = M$  в сети  $N \Rightarrow \bullet t = M/\mathcal{E}$  в сети  $N/\mathcal{E}$ ;
2.  $t \bullet = M$  в сети  $N \Rightarrow t \bullet = M/\mathcal{E}$  в сети  $N/\mathcal{E}$ .

Очевидно, если  $M \xrightarrow{t} \widetilde{M}$  в сети  $N$ , то  $M/\mathcal{E} \xrightarrow{t} \widetilde{M}/\mathcal{E}$  в сети  $N/\mathcal{E}$ .

**Предложение 1.2.9** [23, 24] Если  $\mathcal{R} : N \sim_i N$  — эквивалентность, то  $[ ]_{\mathcal{R}} : N \sim_i N/\mathcal{E}$

**Определение 1.2.31** Каноническая интерливинговая категоризация сети  $N$  есть сеть  $N/\sim_i = N/\mathcal{R}_i(N)$ .

**Определение 1.2.32** Пусть  $N$  — некоторая сеть. Отношение  $\mathcal{R} \subseteq P_N \times P_N$  имеет свойство переноса, если  $\forall t \in T_N \forall p \in \bullet t \forall q : (p, q) \in \mathcal{R}$  выполняется  $\exists u \in T_N : l_N(t) = l_N(u), \bullet t - p + q \xrightarrow{u} \widetilde{M}$  и  $(t \bullet, \widetilde{M}) \in \mathcal{R}$ .

**Теорема 1.2.7** [23, 24] Пусть  $N$  — некоторая сеть. Если  $\mathcal{R} \subseteq P_N \times P_N$  — рефлексивное и симметричное отношение, обладающее свойством переноса, то  $\mathcal{R}^*$  (транзитивное замыкание  $\mathcal{R}$ ) — интерливинговая бисимуляция мест сети  $N$ .

**Следствие 1.2.1** [23, 24] Для сети  $N$  максимальное отношение  $\mathcal{R} \subseteq P_N \times P_N$ , обладающее свойством переноса, есть  $\mathcal{R}_i(N)$ .

На этом следствии основан эффективный алгоритм нахождения  $\mathcal{R}_i(N)$ , описанный в [23, 24].

1. Определяем начальное отношение как  $\mathcal{R} = P_N \times P_N$ .
2. Просматриваем все пары  $(p, q) \in \mathcal{R}$  и проверяем свойство переноса.
  - (а) Если свойство переноса выполняется для всех таких пар, то  $\mathcal{R} = \mathcal{R}_i(N)$ .
  - (б) Иначе, существует пара  $(p, q)$ , для которой свойство не имеет места. Тогда удаляем из  $\mathcal{R}$  пары  $(p, q)$  и  $(q, p)$  и возвращаемся к пункту 2.

Если сеть конечна, таких пар (и проверок) — конечное число. Сложность алгоритма —  $\mathcal{O}(|P_N|^2 \times |T_N|^2)$ , если  $\forall t \in T_N |\bullet t| + |t \bullet| \leq d$  (константа зависит от  $d$ ) [236]. Алгоритм был реализован как система CAESAR на языке LOTOS [236].

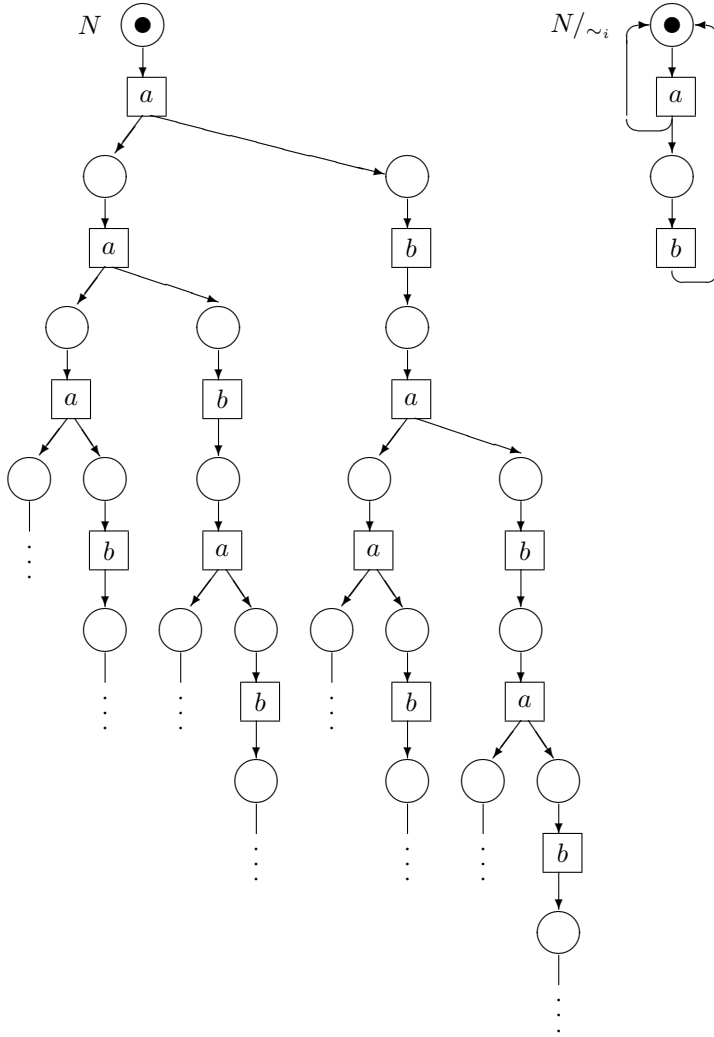


Рис. 1.23: Пример редукции сети по модулю  $\sim_i$

**Пример 1.2.4** На рисунке 1.23 показана редукция сети с использованием  $\sim_i$ . Сеть  $N$  — часть бесконечной сети, соответствующей формуле Petri Box Calculus [43] вида  $E = \mu X.(a; (X \parallel (b; X)))$ . На рисунке представлена развертка уровня 2 с 14 переходами. Очевидно, последующие развертки дают сколь угодно большое число переходов. Сеть  $N/\sim_i$  представляет из себя упрощение  $N$  категоризацией по модулю  $\mathcal{R}_i$  (и удалением излишних переходов).

Что касается  $\sim_{pot}$  и  $\sim_{pr}$ , известны следующие результаты.

Мы не можем использовать  $\sim_{pot}$  для упрощения сетей, так как в [23, 24] приведен пример, когда  $N \not\sim_{pot} N/\sim_{pot}$ .

Так как  $\sim_{pr} = \sim_i$ , мы можем слегка модифицировать алгоритм для  $\mathcal{R}_i$ , чтобы получить  $\mathcal{R}_{pr}$ : будем проверять на бисимуляцию все пары переходов, возникающие во время установления свойства переноса. Сложность алгоритма остается той же. Таким образом, можно эффективно упрощать сети по модулю  $\sim_{pr}$ .

Мы установили взаимосвязи  $\sim_{pr}$  с другими эквивалентностями, и это дает нам следующие важные результаты.

- Так как  $\sim_{pr}$  влечет  $\leftrightarrow_{prh}$  и  $\leftrightarrow_{prST}$ , редуцированная сеть имеет те же истории выполнений и *временные следы* [161], что и исходная сеть.
- Так как  $\leftrightarrow_{prh} = =_{PrBFL}$ , то все свойства, описанные формулами логики **PrBFL**, сохраняются и у редуцированной сети.

## 1.2.4 Сохранение эквивалентностей при детализациях

При рассмотрении сетей, связанных каким-либо отношением эквивалентности, возникает вопрос, сохраняется ли оно при некотором одинаковом их преобразовании. При разработке распределенных систем по методу “сверху вниз” применяется преобразование, называемое *детализацией* [113]. В соответствии с концепцией детализации, действия или состояния, которые считались атомарными на одном уровне абстракции, на более низком уровне считаются имеющими некоторую внутреннюю структуру. В последние годы детализация стала предметом углубленных исследований как в рамках сетевых моделей [64, 117, 119, 137, 138, 139, 140, 141, 142, 143, 144, 145, 148, 149, 150, 164, 217, 241, 271, 272, 273, 274, 278, 282], так и в процессных алгебрах [13, 14, 15, 212].

В этом разделе рассматривается детализация переходов, при которой переходы сети, помеченные некоторым действием, заменяются на подсеть определенного вида: SM-сеть, особый подкласс автоматных или S-сетей [46].

Заметим, что в литературе сохранение при SM-детализациях доказано только для  $\leftrightarrow_{iST}$  [121, 122] и  $\leftrightarrow_{pomh}$  [46]. Для  $\simeq$  данное свойство очевидно. В [113] показано сохранение при детализациях  $\equiv_{pom}$ , но в рамках другой модели и для другого вида детализации. Аналогично, сохранение при детализациях  $\leftrightarrow_{pwST}$  и  $\leftrightarrow_{pomST}$  установлено в [276, 280], но это было сделано в рамках структур событий, и использовался другой оператор детализации (не допускающий, например, конфликты в подставляемой структуре событий).

Новыми результатами, представленными в данном разделе, являются доказательства сохранения при SM-детализациях  $\equiv_*$ ,  $\leftrightarrow_{*ST}$ ,  $\star \in \{pw, pom, pr\}$ , а также  $\equiv_{mes}$  и  $\equiv_{occ}$ . Особенно важный результат — сохранение при SM-детализациях  $\sim_{pr}$ , который влечет возможность использования метода редукции сетей на основе данной эквивалентности в процессе нисходящей разработки систем. Кроме того, с помощью контрпримеров мы демонстрируем, что никакие другие эквивалентности из тех, что были рассмотрены, не обладают данным свойством.

**Определение 1.2.33** SM-сеть — сеть  $D = \langle P_D, T_D, F_D, l_D, M_D \rangle$  такая, что:

1.  $\forall t \in T_D \ | \bullet t | = | t \bullet | = 1$ , то есть каждый имеет ровно одно входное и ровно одно выходное места;
2.  $\exists p_{in}, p_{out} \in P_D$  такие, что  $p_{in} \neq p_{out}$  и  ${}^\circ D = \{p_{in}\}$ ,  $D^\circ = \{p_{out}\}$ , то есть сеть  $D$  имеет единственное входное и единственное выходное места.
3.  $M_D = \{p_{in}\}$ , то есть сначала имеется единственная фишка в  $p_{in}$ .

**Определение 1.2.34** Пусть  $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$  — некоторая сеть,  $a \in l_N(T_N)$  и  $D = \langle P_D, T_D, F_D, l_D, M_D \rangle$  — SM-сеть. SM-детализация, обозначение  $ref(N, a, D)$ , — это (с точностью до изоморфизма) сеть  $\bar{N} = \langle P_{\bar{N}}, T_{\bar{N}}, F_{\bar{N}}, l_{\bar{N}}, M_{\bar{N}} \rangle$ , такая, что:

- $P_{\bar{N}} = P_N \cup \{ \langle p, u \rangle \mid p \in P_D \setminus \{p_{in}, p_{out}\}, u \in l_N^{-1}(a) \}$ ;
- $T_{\bar{N}} = (T_N \setminus l_N^{-1}(a)) \cup \{ \langle t, u \rangle \mid t \in T_D, u \in l_N^{-1}(a) \}$ ;
- $F_{\bar{N}}(\bar{x}, \bar{y}) = \begin{cases} F_N(\bar{x}, \bar{y}), & \bar{x}, \bar{y} \in P_N \cup (T_N \setminus l_N^{-1}(a)); \\ F_D(x, y), & \bar{x} = \langle x, u \rangle, \bar{y} = \langle y, u \rangle, u \in l_N^{-1}(a); \\ F_N(\bar{x}, u), & \bar{y} = \langle y, u \rangle, \bar{x} \in \bullet u, u \in l_N^{-1}(a), y \in p_{in}^\bullet; \\ F_N(u, \bar{y}), & \bar{x} = \langle x, u \rangle, \bar{y} \in \bullet u, u \in l_N^{-1}(a), x \in \bullet p_{out}; \\ 0, & \text{иначе;} \end{cases}$
- $l_{\bar{N}}(\bar{u}) = \begin{cases} l_N(\bar{u}), & \bar{u} \in T_N \setminus l_N^{-1}(a); \\ l_D(t), & \bar{u} = \langle t, u \rangle, t \in T_D, u \in l_N^{-1}(a); \end{cases}$
- $M_{\bar{N}}(p) = \begin{cases} M_N(p), & p \in P_N; \\ 0, & \text{иначе.} \end{cases}$

**Пример 1.2.5** На рисунке 1.24 показано действие SM-детализации.

Некоторая сетевая эквивалентность сохраняется при детализациях, если эквивалентные сети остаются таковыми после одновременного применения к ним любого оператора детализации.

**Предложение 1.2.10** Пусть  $\star \in \{i, s\}$ ,  $\star\star \in \{i, s, pw, pom, pr, sbsf, sbpwf, sbpomf, sbprf, pombprf\}$ ,  $\star\star\star \in \{i, pom\}$ . Эквивалентности  $\equiv_*$ ,  $\leftrightarrow_{\star\star}$ ,  $\sim_{\star\star\star}$  не сохраняются при SM-детализациях.

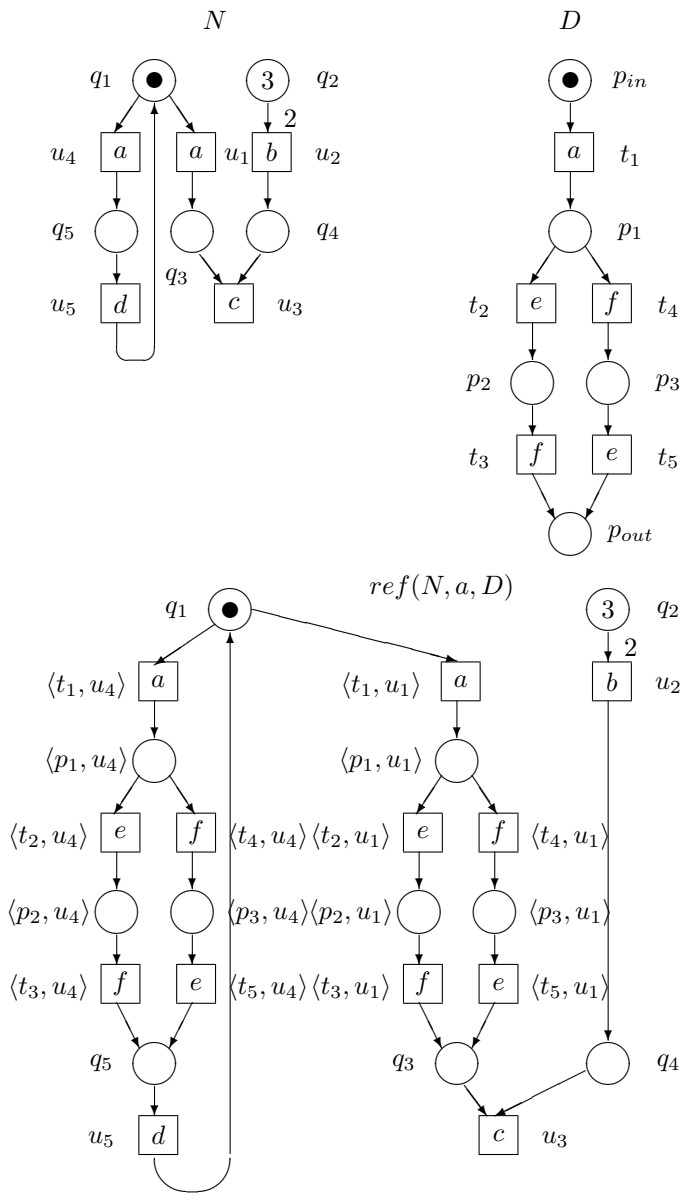


Рис. 1.24: Пример SM-детализации



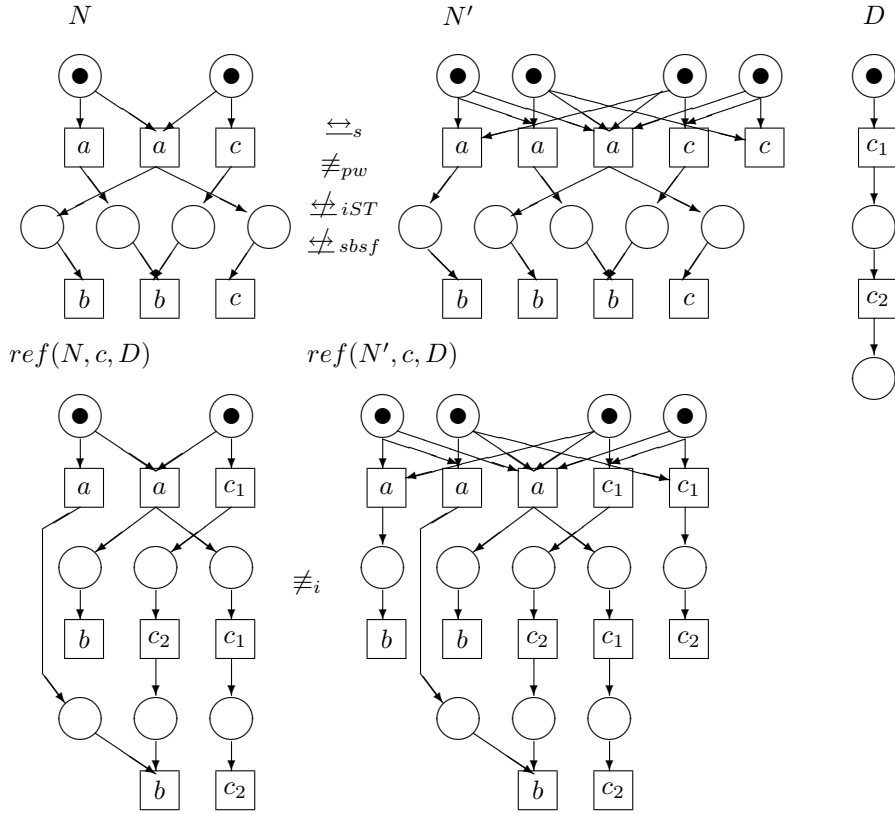


Рис. 1.25: Эквивалентности от  $\equiv_i$  до  $\leftrightarrow_s$  не сохраняются при SM-детализациях

*Доказательство.*

- На рисунке 1.25  $N \leftrightarrow_s N'$ , но  $ref(N, c, D) \not\equiv_i ref(N', c, D)$ , так как только в сети  $ref(N', c, D)$  можно выполнить последовательность действий  $c_1abc_2$ . Следовательно, эквивалентности от  $\equiv_i$  до  $\leftrightarrow_s$  не сохраняются при SM-детализациях.
- На рисунке 1.26  $N \leftrightarrow_{sbprf} N'$ , но  $ref(N, a, D) \not\equiv_i ref(N', a, D)$ , так как только в сети  $ref(N', a, D)$  действие  $a_1$  может сработать так, что сразу после него не могут сработать:
  1. ни последовательность действий  $bc$ ,
  2. ни последовательность действий  $a_2c$ .

Следовательно, эквивалентности от  $\leftrightarrow_i$  до  $\leftrightarrow_{sbprf}$  не сохраняются при SM-детализациях.

- На рисунке 1.27  $N \leftrightarrow_{pombrf} N'$ , но  $ref(N, a, D) \not\equiv_{pr} ref(N', a, D)$ , так как только в сети  $ref(N', a, D)$  действие  $a_1$  может сработать так, что после него может сработать последовательность действий  $a_2b$ , которой соответствует лишь один процесс (переход с действием  $b$  присоединяется к переходу с действием  $a_2$  лишь одним способом). Следовательно, эквивалентности от  $\leftrightarrow_{pr}$  до  $\leftrightarrow_{pombrf}$  не сохраняются при SM-детализациях.
- На рисунке 1.28  $N \sim_{pom} N'$ , но  $ref(N, a, D) \not\equiv_i ref(N', a, D)$ , так как только в  $ref(N', a, D)$  после срабатывания действия  $a_1$  не сможет выполняться  $b$ . Следовательно, эквивалентности от  $\leftrightarrow_i$  до  $\sim_{pom}$  не сохраняются при SM-детализациях.

На рисунке 1.29 линиями обведены эквивалентности, отсутствие сохранения которых при SM-детализациях доказывается каждым из рисунков 1.25–1.28.  $\square$

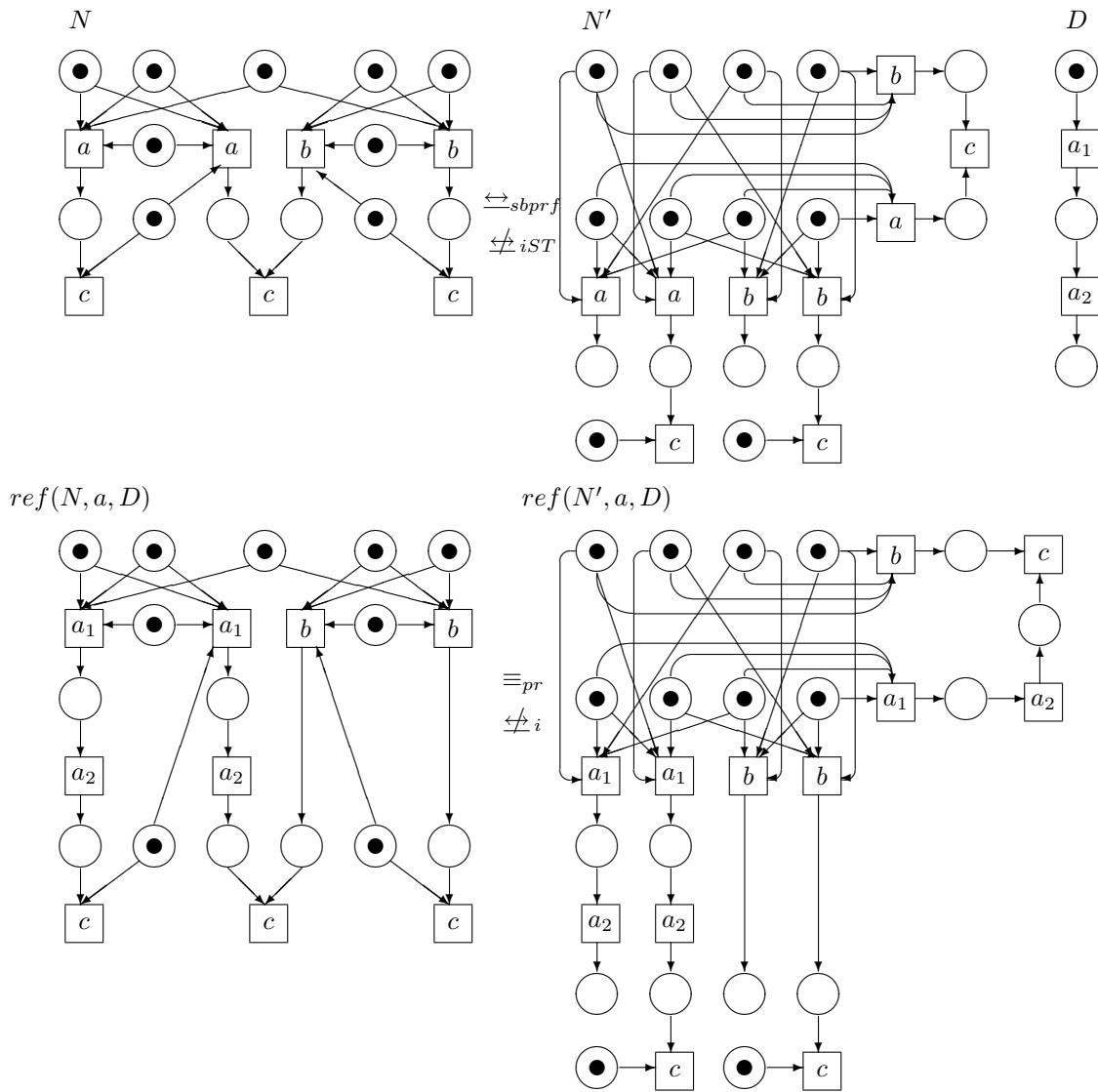


Рис. 1.26: Эквивалентности от  $\equiv_{iST}$  до  $\xrightarrow{\text{sbprf}}$  не сохраняются при SM-детализациях

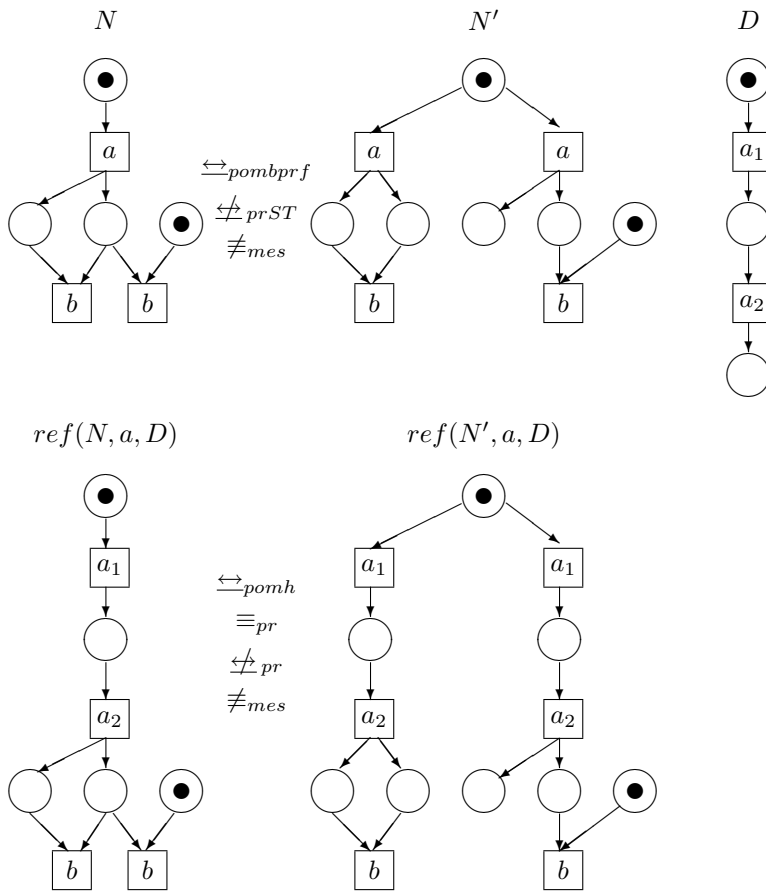


Рис. 1.27: Эквивалентности от  $\xleftrightarrow{\text{pr}}$  до  $\xleftrightarrow{\text{pombrf}}$  не сохраняются при SM-детализациях

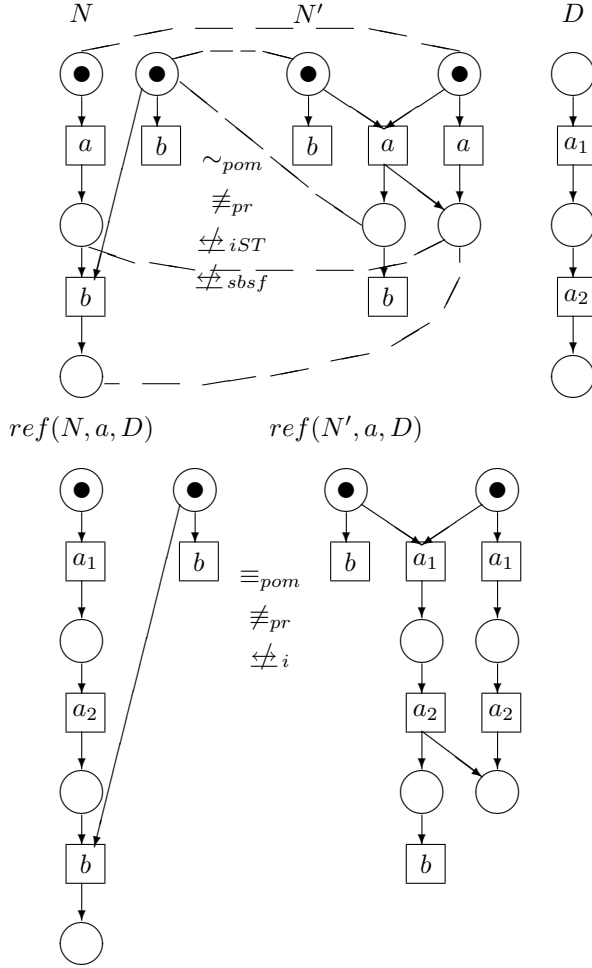


Рис. 1.28: Эквивалентности от  $\leftrightarrow_i$  до  $\sim_{pom}$  не сохраняются при SM-детализациях

Рассмотрим, какие из сетевых эквивалентностей сохраняются при SM-детализациях.

**Предложение 1.2.11** Пусть  $\star \in \{pw, pom, pr\}$ . Для сетей  $N$  и  $N'$  таких, что  $a \in l_N(T_N) \cap l_{N'}(T_{N'})$  и SM-сети  $D$   $N \equiv_\star N' \Rightarrow ref(N, a, D) \equiv_\star ref(N', a, D)$ .

*Доказательство.* См. приложение А. □

**Предложение 1.2.12** Пусть  $\star \in \{i, pw, pom, pr\}$ . Для сетей  $N$  и  $N'$  таких, что  $a \in l_N(T_N) \cap l_{N'}(T_{N'})$  и SM-сети  $D$   $N \leftrightarrow_{\star ST} N' \Rightarrow ref(N, a, D) \leftrightarrow_{\star ST} ref(N', a, D)$ .

*Доказательство.* См. приложение А. □

Заметим, что сохранение при детализациях  $\leftrightarrow_{\star ST}$ ,  $\star \in \{i, pw, pom\}$  установлено в [276], но это было сделано в рамках структур событий, и использовался другой оператор уточнения (не допускающий, например, конфликты в подставляемой структуре событий).

**Предложение 1.2.13** [46] Для сетей  $N$  и  $N'$  таких, что  $a \in l_N(T_N) \cap l_{N'}(T_{N'})$  и SM-сети  $D$   $N \leftrightarrow_{pomh} N' \Rightarrow ref(N, a, D) \leftrightarrow_{pomh} ref(N', a, D)$ .

**Предложение 1.2.14** Для сетей  $N$  и  $N'$  таких, что  $a \in l_N(T_N) \cap l_{N'}(T_{N'})$  и SM-сети  $D$   $N \leftrightarrow_{prh} N' \Rightarrow ref(N, a, D) \leftrightarrow_{prh} ref(N', a, D)$ .

*Доказательство.* В доказательстве предложения 8.5 в [46] вводится конструкция (используемая в предложениях этого раздела), преобразующая отношение бисимуляции на исходных сетях в отношение на детализованных сетях, и затем проверяется, что сконструированное таким образом отношение — ЧУММ

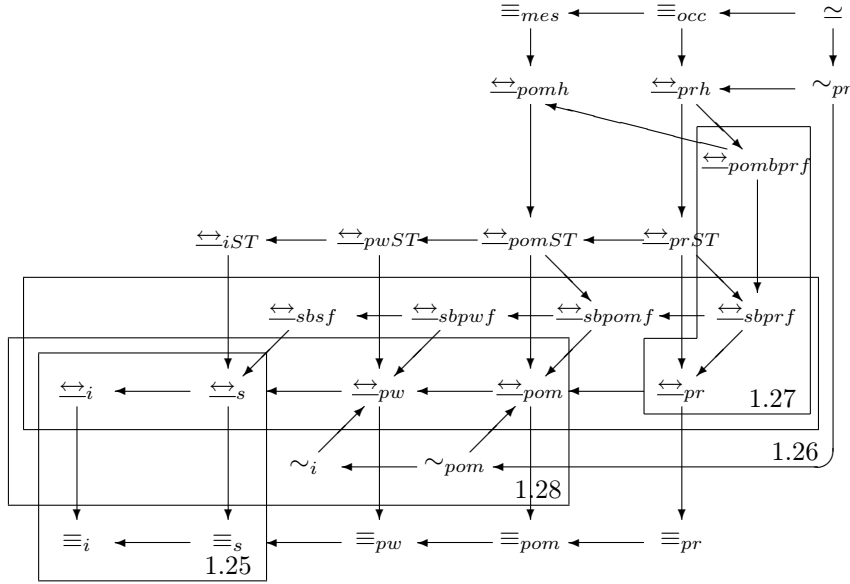


Рис. 1.29: Эквивалентности, не сохраняющиеся при SM-детализациях

сохраняющая историю бисимуляция. Для доказательства того, что в условиях данного предложения это отношение — процессная сохраняющая историю бисимуляция, достаточно заметить, что приведенная в [46] конструкция сохраняет изоморфизм С-сетей процессов.  $\square$

Следующее предложение демонстрирует новый результат для сохраняющих конфликт эквивалентностей.

**Предложение 1.2.15** Пусть  $\star \in \{mes, occ\}$ . Для сетей  $N$  и  $N'$  таких, что  $a \in l_N(T_N) \cap l_{N'}(T_{N'})$  и SM-сети  $D$   $N \equiv_\star N' \Rightarrow ref(N, a, D) \equiv_\star ref(N', a, D)$ .

*Доказательство.* См. приложение А.  $\square$

Следующее предложение является особенно важным. В нем доказывается сохранение при SM-детализациях процессной бисимуляционной эквивалентности мест. В литературе для эквивалентностей мест этот вопрос никогда не исследовался.

**Предложение 1.2.16** Для сетей  $N$  и  $N'$  таких, что  $a \in l_N(T_N) \cap l_{N'}(T_{N'})$  и SM-сети  $D$   $N \sim_{pr} N' \Rightarrow ref(N, a, D) \sim_{pr} ref(N', a, D)$ .

*Доказательство.* Пусть  $\bar{N} = ref(N, a, D)$ ,  $\bar{N}' = ref(N', a, D)$  и  $\mathcal{R} : N \sim_{pr} N'$ . По предложению 1.2.6,  $\mathcal{R} : N \approx_i N'$ . Достаточно доказать, что  $\bar{N} \approx_i \bar{N}'$ . Определим отношение  $\mathcal{S}$  следующим образом:  $\mathcal{S} = \mathcal{R} \cup \{(\langle p, u \rangle, \langle p, u' \rangle) \mid p \in P_D \setminus \{p_{in}, p_{out}\}, (u, u') \in \bar{\mathcal{R}}\}$ . Докажем, что  $\mathcal{S} : \bar{N} \approx_i \bar{N}'$ .

1.  $(M_{\bar{N}}, M_{\bar{N}'}) \in \mathcal{S}$ , так как  $(M_N, M_{N'}) \in \mathcal{R}$ .
2. Пусть  $(M, M') \in \mathcal{S}$  и  $M \xrightarrow{\bar{u}} \tilde{M}$ . Возможны два случая:
  - (а)  $\bar{u} = u \in T_N$ ;
  - (б)  $\bar{u} = \langle t, u \rangle$ ,  $t \in T_D$ ,  $u \in T_N$ ,  $l_N(u) = a$ .

Рассмотрим случай (б), так как в случае (а) все очевидно. Пусть  $\bullet t = \{p\}$ ,  $t^\bullet = \{q\}$ . Тогда имеем следующее:

$$\bullet \langle t, u \rangle = \begin{cases} \bullet u, & t \in p_{in}^\bullet; \\ \langle p, u \rangle, & \text{иначе.} \end{cases}$$

$$\langle t, u \rangle^\bullet = \begin{cases} u^\bullet, & t \in p_{out}^\bullet; \\ \langle q, u \rangle, & \text{иначе.} \end{cases}$$

Возможны четыре случая:

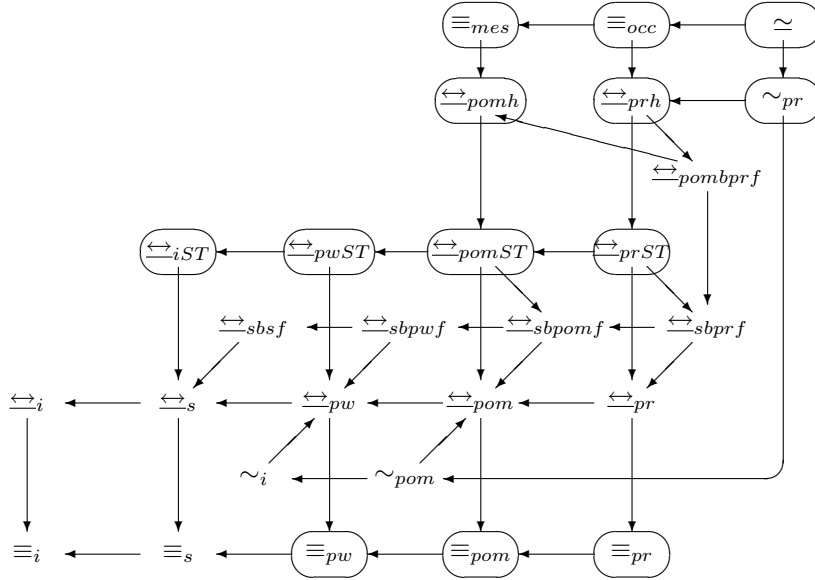


Рис. 1.30: Сохранение эквивалентностей при SM-детализациях

- (a)  $t \in p_{in}^\bullet \cap \bullet p_{out}$ ;
- (b)  $t \in p_{in}^\bullet \setminus \bullet p_{out}$ ;
- (c)  $t \in \bullet p_{out} \setminus p_{in}^\bullet$ ;
- (d)  $t \notin p_{in}^\bullet \cup \bullet p_{out}$ .

Рассмотрим случай (d), так как случаи (a)–(c) — более простые. Имеем  $\bullet \langle t, u \rangle = \langle p, u \rangle \in M$ . Так как  $(M, M') \in \overline{\mathcal{S}}$ , по определению  $\mathcal{S}$  имеем:  $\exists u' \in T_{N'} : (u, u') \in \overline{\mathcal{R}}$  и  $(\langle p, u \rangle, \langle p, u' \rangle) \in \mathcal{S}$ ,  $\langle p, u' \rangle \in M'$ . Так как  $\bullet \langle t, u' \rangle = \langle p, u' \rangle$ , то  $(\bullet \langle t, u \rangle, \bullet \langle t, u' \rangle) \in \overline{\mathcal{S}}$ ,  $\bullet \langle t, u' \rangle \in M'$ .

Тогда  $\exists \widetilde{M}' : M' \xrightarrow{\langle t, u' \rangle} \widetilde{M}'$ . Имеем:  $l_{\overline{N}}(\langle t, u \rangle) = l_D(t) = l_{\overline{N}'}(\langle t, u' \rangle)$ . Так как  $\langle t, u \rangle^\bullet = \langle q, u \rangle$ , по определению  $\mathcal{S}$  имеем  $(\langle q, u \rangle, \langle q, u' \rangle) \in \mathcal{S}$ . Так как  $\langle t, u' \rangle^\bullet = \langle q, u' \rangle$ , то  $(\langle t, u \rangle^\bullet, \langle t, u' \rangle^\bullet) \in \overline{\mathcal{S}}$ .

Таким образом,  $(\langle t, u \rangle, \langle t, u' \rangle) \in \overline{\mathcal{S}}$  и  $(\widetilde{M}, \widetilde{M}') \in \overline{\mathcal{S}}$ .

3. Как пункт 2, но роли  $\overline{N}$  и  $\overline{N}'$  меняются. □

**Предложение 1.2.17** Для сетей  $N$  и  $N'$  таких, что  $a \in l_N(T_N) \cap l_{N'}(T_{N'})$  и SM-сети  $D \ N \simeq N' \Rightarrow ref(N, a, D) \simeq ref(N', a, D)$ .

*Доказательство.* Очевидно. □

**Теорема 1.2.8** Пусть  $\leftrightarrow \in \{\equiv, \leftrightarrow, \sim, \simeq\}$  и  $\star \in \{\_, i, s, pw, pom, pr, iST, pwST, pomST, prST, pomh, prh, mes, occ, sbsf, sbprf, sbpomf, sbprf, pombprf\}$ . Для сетей  $N$  и  $N'$  таких, что  $a \in l_N(T_N) \cap l_{N'}(T_{N'})$  и SM-сети  $D \ N \leftrightarrow_\star N' \Rightarrow ref(N, a, D) \leftrightarrow_\star ref(N', a, D)$  тогда и только тогда, когда эквивалентность  $\leftrightarrow_\star$  заключена в овал на рисунке 1.30.

*Доказательство.* С использованием предложений 1.2.10–1.2.17. □

## 1.2.5 Эквивалентности на подклассах сетей Петри

В литературе был предложен ряд подклассов сетей Петри посредством введения некоторых ограничений на их определение. При этом рассмотренные ранее эквивалентности могут стать более зависимыми или даже слиться на таких сетях.

Результаты для подклассов сетей можно найти в [8, 96, 245, 292], а на ограничениях других формальных параллельных моделей — в [19, 133, 267, 269, 270]. Такого рода результаты могут быть использованы

как для упрощения проверки сетей на эквивалентность, так и для восстановления некоторой сложной информации по более простой. Например, в [186] описано, как восстановить частичные языки P/T-систем по множеству их шаговых цепочек.

Взаимосвязь параллелизма и недетерминизма значительно усложняет анализ систем. Поэтому в теории сетей Петри были введены подклассы сетей, в которых присутствует только один из этих аспектов.

Мы рассмотрим такие подклассы сетей Петри, как последовательные сети [46], в которых невозможно параллельное выполнение переходов и T-сети [250], в которых нет конфликтных переходов.

Известно, что помеченные сети Петри имеют большую моделирующую силу, чем непомеченные. Поэтому интересно рассмотреть еще один подкласс сетей Петри: сети со строгой пометкой [46], в которых никакие два различных перехода не могут помечаться одним и тем же действием, и поэтому их можно рассматривать как непомеченные, идентифицируя имена переходов с действиями, помечающими эти переходы.

Заметим, что на последовательных сетях в литературе исследовались лишь  $\leftrightarrow_i$ ,  $\leftrightarrow_{pom}$  и  $\leftrightarrow_{pomh}$  [46]. Сравнение эквивалентностей на T-сетях не проводилось. В [245] на строго помеченных сетях проведено сравнение лишь небольшой группы эквивалентностей, состоящей из  $\equiv_*$ ,  $\leftrightarrow_*$ ,  $\star \in \{i, s, pom\}$  и  $\leftrightarrow_{pomh}$ .

Таким образом, мы впервые исследуем такой широкий набор эквивалентностей для трех указанных подклассов сетей.

### Эквивалентности на последовательных сетях

**Определение 1.2.35** Сеть  $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$  — последовательная, если  $\forall M \in Mark(N)$   $\neg \exists t, u \in T_N : \bullet t + \bullet u \subseteq M$ , то есть никакие два перехода не допустимы вместе ни в одной из достижимых маркировок.

**Лемма 1.2.1** В полном выполнении C-сети C переход  $v \in T_C$  может сработать тогда и только тогда, когда уже сработали все переходы из  $(\downarrow_C v) \cap T_C$ .

*Доказательство.* ( $\Rightarrow$ ) Если переход  $v \in T_C$  может сработать, в местах из  $\bullet v$  есть фишки. Эти места можно разделить на две группы.

1.  $r \in \bullet v$  и  $\bullet r = \emptyset$ . Тогда  $r \in {}^\circ C$  и, по определению полного выполнения C-сети, в  $r$  есть фишка.
2.  $r \in \bullet v$  и  $\bullet r \neq \emptyset$ . Так как в C-сетях места не разветвлены, фишка в  $r$  может появиться лишь после срабатывания единственного перехода  $w \in T_C$ , где  $\bullet r = \{w\}$ .

Таким образом, для срабатывания перехода  $v$  необходимо, чтобы сработали все переходы из  $\bullet(\bullet v) = {}^2\bullet v$ , а для их выполнения необходимо срабатывание переходов из  ${}^4\bullet v$ , и так далее. Так как, по определению C-сетей, в  $F_C$  нет бесконечных в обратную сторону цепочек, существует  $n \in \mathbf{N}$  такое, что  ${}^{2n}\bullet v = \emptyset$ . Следовательно, для срабатывания  $v$  необходимо, чтобы выполнились все переходы из  $\cup_{i=1}^{n-1} {}^{2i}\bullet v = (\downarrow_C v) \cap T_C$ .

( $\Leftarrow$ ) Если в полном выполнении C уже сработали все переходы из  $(\downarrow_C v) \cap T_C$  для  $v \in T_C$ , то выполнились и переходы из  ${}^2\bullet v$ . Поэтому все места из  $\bullet v$  имеют фишки, и  $v$  может сработать.  $\square$

**Лемма 1.2.2** Пусть C — C-сеть и  $v, w \in T_C$ . В этом случае  $v \prec_C w$  тогда и только тогда, когда в любом полном выполнении C  $v$  срабатывает раньше  $w$ .

*Доказательство.* ( $\Rightarrow$ ) По лемме 1.2.1 переход  $w \in T_C$  может сработать только тогда, когда уже сработали все переходы из  $(\downarrow_C w) \cap T_C$ , а  $v \in (\downarrow_C w) \cap T_C$ .

( $\Leftarrow$ ) Докажем от противного. Пусть  $v \not\prec_C w$ . Возможны два варианта.

1.  $w \prec_C v$ . Тогда, по доказанному ранее, в любом полном выполнении C  $w$  срабатывает перед  $v$ , что противоречит исходному предположению.
2.  $w \not\prec_C v$ , то есть  $v$  и  $w$  не сравнимы по отношению  $\prec_C$ . Тогда существует полное выполнение C, в котором с самого начала срабатывают все переходы из  $(\downarrow_C w) \cap T_C$ , а затем  $w$ . Так как  $v \notin (\downarrow_C w) \cap T_C$ , в этом выполнении  $w$  срабатывает раньше  $v$ .

Итак, оба варианта должны быть отвергнуты. Следовательно,  $v \prec_C w$ .  $\square$

**Предложение 1.2.18** Для последовательных сетей  $N$  и  $N'$   $N \equiv_i N' \Leftrightarrow N \equiv_{pom} N'$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие теоремы 1.2.1.

( $\Rightarrow$ ) Пусть  $N \equiv_i N'$ . Тогда  $IntTraces(N) = IntTraces(N')$ . Для доказательства того, что  $N \equiv_{pom} N'$ , достаточно установить равенство  $Pomsets(N) = Pomsets(N')$ . Оно очевидно, так как  $Pomsets(N)$  и  $Pomsets(N')$  — линейно упорядоченные мультимножества (цепочки), и существует взаимно однозначное отображение между  $IntTraces(N)$  и  $Pomsets(N)$  ( $IntTraces(N')$  и  $Pomsets(N')$  соответственно).  $\square$

**Предложение 1.2.19** Для последовательных сетей  $N$  и  $N'$   $N \leftrightarrow_i N' \Leftrightarrow N \leftrightarrow_{pomh} N'$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие теоремы 1.2.1.

( $\Rightarrow$ ) Пусть  $\mathcal{R} : N \leftrightarrow_i N'$ . Докажем, что  $\mathcal{S} : N \leftrightarrow_{pomh} N'$ , где  $\mathcal{S}$  определяется так:  $\mathcal{S} = \{(\pi, \pi', \beta) \mid (\pi, \pi') \in \mathcal{R}, \beta : \rho_C \simeq \rho_{C'}, \pi = (C, \varphi), \pi' = (C', \varphi')\}$ .

1. Очевидно,  $(\pi_N, \pi_{N'}, \emptyset) \in \mathcal{S}$ .

2. По определению  $\mathcal{S}$ ,  $(\pi, \pi', \beta) \in \mathcal{S} \Rightarrow \beta : \rho_C \simeq \rho_{C'}$ ;

3. Пусть  $(\pi, \pi', \beta) \in \mathcal{S}$ ,  $\pi \xrightarrow{v} \tilde{\pi}$ ,  $\tilde{\pi} = (\tilde{C}, \tilde{\varphi})$ . По определению последовательных сетей, все переходы в C-сетях их процессов линейно упорядочены отношением предшествования. Поэтому, по лемме 1.2.2 полные выполнения таких C-сетей единственны, и существует взаимно однозначное соответствие между ЧУММ этих C-сетей и последовательностями действий, выписанных в порядке срабатывания помечаемых ими переходов.

Рассмотрим полное выполнение  $\tilde{C}$  в  $N$   $\pi_N = \pi_0 \xrightarrow{v_1} \dots \xrightarrow{v_n} \pi_n = \pi \xrightarrow{v} \tilde{\pi}$ . По определению интерливинговой бисимуляции  $\exists \tilde{\pi}', v'$  такие, что  $\pi_{N'} = \pi'_0 \xrightarrow{\beta(v_1)} \dots \xrightarrow{\beta(v_n)} \pi'_n = \pi' \xrightarrow{v'} \tilde{\pi}'$ ,  $\tilde{\pi}' = (\tilde{C}', \tilde{\varphi}')$ ,  $l_{\tilde{C}}(v) = l_{\tilde{C}'}(v')$ ,  $(\tilde{\pi}, \tilde{\pi}') \in \mathcal{R}$ . Это — единственное полное выполнение  $\tilde{C}'$  в  $N'$ .

Рассмотрим отображение  $\tilde{\beta}$ , где  $\tilde{\beta}|_{T_C} = \beta$  и  $\tilde{\beta}(v) = v'$ . Очевидно, что  $\tilde{\beta} : \rho_{\tilde{C}} \simeq \rho_{\tilde{C}'}$ , так как существует взаимно однозначное соответствие между  $\rho_{\tilde{C}}$ ,  $\rho_{\tilde{C}'}$  и последовательностью действий  $l_{\tilde{C}}(v_1) \dots l_{\tilde{C}}(v_n) l_{\tilde{C}}(v)$ .

4. Как пункт 3, но роли  $N$  и  $N'$  меняются.  $\square$

**Предложение 1.2.20** Для последовательных сетей  $N$  и  $N'$   $N \leftrightarrow_{pr} N' \Leftrightarrow N \leftrightarrow_{pombprf} N'$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие теоремы 1.2.2.

( $\Rightarrow$ ) Имеем:  $N \leftrightarrow_{pr} N'$ . По предложению 1.2.3,  $N \leftrightarrow_{ibprf} N'$ . Так как для последовательных сетей ЧУММ C-сетей процессов являются линейно упорядоченными, существует взаимно однозначное соответствие между последовательностями действий, выписанных в порядке их срабатывания. Следовательно,  $N \leftrightarrow_{pombprf} N'$ .  $\square$

**Предложение 1.2.21** Для последовательных сетей  $N$  и  $N'$   $N \sim_i N' \Leftrightarrow N \sim_{pom} N'$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие теоремы 1.2.6.

( $\Rightarrow$ ) Имеем:  $\exists \mathcal{R} : N \sim_i N'$ . По определению бисимуляций мест,  $\overline{\mathcal{R}} : N \leftrightarrow_i N'$ . Так как по предложению 1.2.19 на последовательных сетях сливаются все эквивалентности от  $\leftrightarrow_i$  до  $\leftrightarrow_{pomh}$ , имеем  $\overline{\mathcal{R}} : N \leftrightarrow_{pom} N'$ . По определению бисимуляций мест,  $\mathcal{R} : N \sim_{pom} N'$ .  $\square$

На рисунке 1.31 пунктирными линиями обведены эквивалентности, совпадающие на последовательных сетях.

**Теорема 1.2.9** Пусть  $\leftrightarrow, \leftrightarrow \in \{\equiv, \leftrightarrow, \sim, \simeq\}$ ,  $\star, \star \in \{\_, i, pr, prST, prh, mes, occ\}$ . Для последовательных сетей  $N$  и  $N'$   $N \leftrightarrow_{\star} N' \Rightarrow N \leftrightarrow_{\star\star} N'$  тогда и только тогда, когда в графе на рисунке 1.32 существует направленный путь от  $\leftrightarrow_{\star}$  к  $\leftrightarrow_{\star\star}$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие теоремы 1.2.6.

( $\Rightarrow$ ) Отсутствие дополнительных нетривиальных стрелок в графе на рисунке 1.32 доказывается следующими примерами на последовательных сетях.

- На рисунке 1.12(d)  $N \equiv_{mes} N'$ , но  $N \not\equiv_{pr} N'$ .
- На рисунке 1.12(e)  $N \equiv_{pr} N'$ , но  $N \not\equiv_i N'$ .



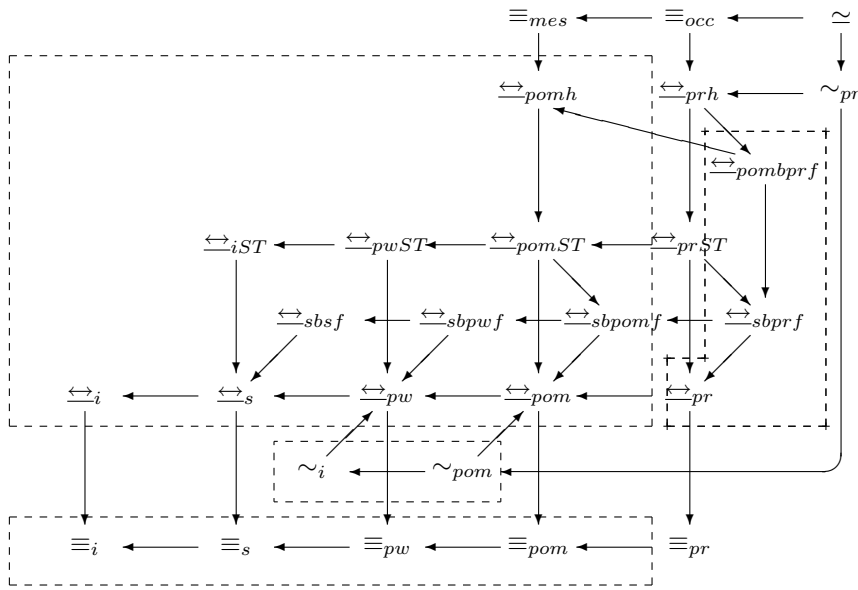


Рис. 1.31: Совпадение эквивалентностей на последовательных сетях

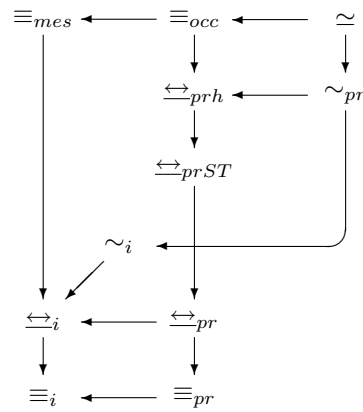


Рис. 1.32: Взаимосвязь эквивалентностей на последовательных сетях

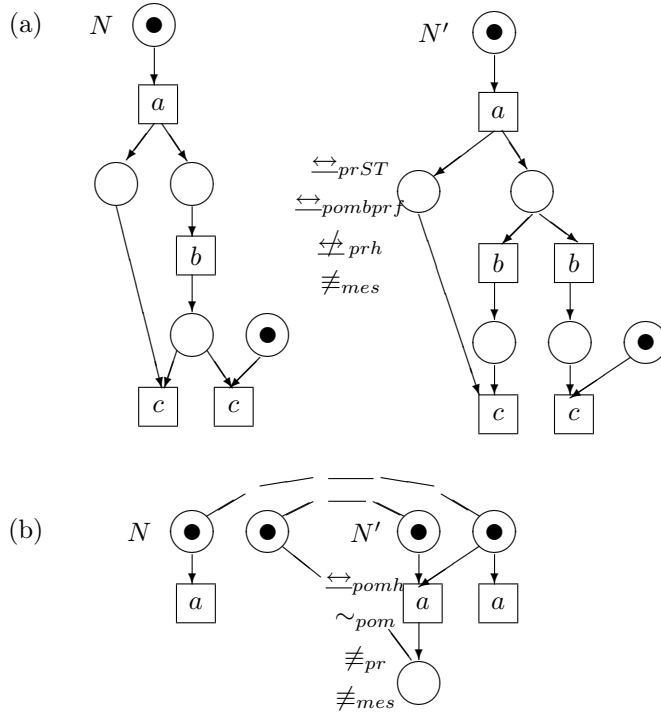


Рис. 1.33: Примеры эквивалентностей на последовательных сетях

- На рисунке 1.17(c)  $N \xleftrightarrow{pr} N'$ , но  $N \not\xleftrightarrow{pr.ST} N'$ , так как только в сети  $N'$  можно начать выполнять процесс с действием  $a$  так, чтобы его можно было расширить на действие  $b$  только одним способом (то есть так, чтобы расширенный процесс был только один).
- На рисунке 1.33(a)  $N \xleftrightarrow{pr.ST} N'$ , но  $N \not\xleftrightarrow{pr.h} N'$ , так как только в сети  $N'$  может выполняться процесс, в котором срабатывают последовательно действия  $a$  и  $b$ , так, чтобы следующее действие,  $c$ , могло расширить этот процесс только одним способом (то есть С-сеть с действием  $c$ , расширяя С-сеть, соответствующую последовательности  $ab$ , сцепляется с ее подсетью, содержащей  $a$ , только одним способом).
- На рисунке 1.13(c)  $N \xleftrightarrow{pr.h} N'$ , но  $N \not\equiv_{mes} N'$ .
- На рисунке 1.13(d)  $N \equiv_{occ} N'$ , но  $N \not\cong N'$ .
- На рисунке 1.33(b)  $N \sim_i N'$ , но  $N \not\equiv_{pr} N'$ , так как только в сети  $N'$  переход с пометкой  $a$  имеет 2 входных места.
- На рисунке 1.22(c)  $N \equiv_{occ} N'$ , но  $N \not\sim_i N'$ .
- На рисунке 1.13(c)  $N \sim_{pr} N'$ , но  $N \not\equiv_{mes} N'$ . □

### Эквивалентности на строго помеченных сетях

**Определение 1.2.36** Сеть  $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$  — строго помеченная, если ее помечающая функция  $l_N$  инъективна, то есть  $\forall t, u \in T_N \ t \neq u \Rightarrow l_N(t) \neq l_N(u)$ .

Рассмотрим эквивалентности на сетях со строгой пометкой.

**Предложение 1.2.22** Пусть  $\star \in \{i, s, pw, pom, pr\}$ . Для строго помеченных сетей  $N$  и  $N'$   $N \equiv_\star N' \Leftrightarrow N \xleftrightarrow{\star} N'$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие теоремы 1.2.1.

( $\Rightarrow$ ) Определим отношение  $\mathcal{R}$  следующим образом: образом:  $\mathcal{R} = \{(\pi, \pi') \mid \pi = (C, \varphi) \in \Pi(N), \pi' = (C', \varphi') \in \Pi(N'), \rho_C \asymp \rho_{C'}\}$ . Докажем, что  $\mathcal{R} : N \xleftrightarrow{\star} N'$ .

1. Очевидно,  $(\pi_N, \pi_{N'}, \emptyset) \in \mathcal{R}$ .
2. Пусть  $(\pi, \pi') \in \mathcal{R}$ ,  $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$  и
  - (a)  $|T_{\hat{C}}| = 1$ , если  $\star = i$ ;
  - (b)  $\prec_{\hat{C}} = \emptyset$ , если  $\star = s$ ;

Так как  $N \equiv_{\star} N'$ , то  $\exists \tilde{\pi}' \in \Pi(N')$  такой, что

- (a)  $\rho_{\tilde{C}'} \sqsubseteq \rho_{\tilde{C}}$ , если  $\star = pw$ ;
- (b)  $\rho_{\tilde{C}} \simeq \rho_{\tilde{C}'}$ , если  $\star \in \{i, s, pom\}$ ;
- (c)  $\tilde{C} \simeq \tilde{C}'$ , если  $\star = pr$ .

Следовательно,  $\exists \tilde{\pi}', \hat{\pi}' : \tilde{\pi}' \xrightarrow{\hat{\pi}'} \tilde{\pi}'$ ,  $\rho_C \asymp \rho_{C'}$  и

- (a)  $\rho_{\hat{C}'} \sqsubseteq \rho_{\hat{C}}$ , если  $\star = pw$ ;
- (b)  $\rho_{\hat{C}} \simeq \rho_{\hat{C}'}$ , если  $\star \in \{i, s, pom\}$ ;
- (c)  $\hat{C} \simeq \hat{C}'$ , если  $\star = pr$ .

Так как  $\rho_C \asymp \rho_{C'}$ , имеем  $\rho_{C'} \asymp \rho_{C'}$ . Так как  $N'$  — строго помеченная сеть, маркировки  $M'$  и  $\bar{M}'$ , соответствующие процессам  $\pi'$  и  $\tilde{\pi}'$ , получаются в результате срабатывания одного и того же множества переходов  $N'$ . Следовательно,  $M' = \bar{M}'$ , и процесс  $\pi'$  можно расширить на  $\hat{\pi}'$ . Тогда  $\exists \tilde{\pi}' : \pi' \xrightarrow{\hat{\pi}'} \tilde{\pi}'$  и  $(\tilde{\pi}, \tilde{\pi}') \in \mathcal{R}$ .

3. Как пункт 2, но роли  $N$  и  $N'$  меняются. □

**Предложение 1.2.23** Для строго помеченных сетей  $N$  и  $N'$   $N \leftrightarrow_s N' \Leftrightarrow N \leftrightarrow_{iST} N'$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие теоремы 1.2.1.

( $\Rightarrow$ ) Пусть  $\mathcal{R} : N \leftrightarrow_s N'$ . Определим отношение  $\mathcal{S}$  следующим образом:  $\mathcal{S} = \{((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \mid (\pi_E, \pi'_E), (\pi_P, \pi'_P) \in \mathcal{R}, (\pi_E, \pi_P) \in ST - \Pi(N), (\pi'_E, \pi'_P) \in ST - \Pi(N'), \beta : \rho_{C_E} \asymp \rho_{C'_E}, \beta(T_{C_P}) = T_{C'_P}\}$ . Докажем, что  $\mathcal{S} : N \leftrightarrow_{iST} N'$ .

Достаточно рассматривать только расширения ST-процессов вида  $(\pi_E, \pi_P) \rightarrow (\tilde{\pi}_E, \pi_P)$ , где  $\pi_E \rightarrow \tilde{\pi}_E$  — расширение на одно действие.

1. Очевидно,  $((\pi_N, \pi_N), (\pi_{N'}, \pi_{N'}), \emptyset) \in \mathcal{S}$ .
2. По определению  $\mathcal{S}$ ,  $((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{S} \Rightarrow \beta : \rho_{C_E} \asymp \rho_{C'_E}$  и  $\beta(T_{C_P}) = T_{C'_P}$ ;
3. Пусть  $((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{S}$  и  $(\pi_E, \pi_P) \rightarrow (\tilde{\pi}_E, \pi_P)$ , где  $(\tilde{\pi}_E, \pi_P) \in ST - \Pi(N)$ ,  $\pi_E \xrightarrow{v} \tilde{\pi}_E$  и  $v$  — переход с пометкой  $a$ . Так как  $(\pi_E, \pi_P) \in ST - \Pi(N)$ , по определению ST-процесса,  $\exists V \subseteq T_{C_E} \pi_P \xrightarrow{V} \pi_E$ ,  $l_{C_E}(V) = A \in \mathcal{M}(Act)$ . Тогда  $\pi_P \xrightarrow{V \cup v} \tilde{\pi}_E$ . Так как  $(\pi'_E, \pi'_P) \in ST - \Pi(N')$ ,  $\exists V' \subseteq T_{C'_E} \pi'_P \xrightarrow{V'} \pi'_E$ . По определению  $\mathcal{S}$ ,  $l_{C'_E}(V') = A$ . Имеем следующие два случая.
  - (a) Существуют процесс  $\tilde{\pi}'_E$  и переход  $v'$  с пометкой  $a$  такие, что  $\pi'_P \xrightarrow{V' \cup v'} \tilde{\pi}'_E$ . Тогда  $(\tilde{\pi}'_E, \pi'_P) \in ST - \Pi(N')$ . Определим отображение  $\tilde{\beta}$  следующим образом:  $\tilde{\beta}|_{T_{C_E}} = \beta$ ,  $\tilde{\beta}(v) = v'$ . Тогда  $((\tilde{\pi}_E, \pi_P), (\tilde{\pi}'_E, \pi'_P), \tilde{\beta}) \in \mathcal{S}$  и выполняются все требования пункта 3 определения интерливинговой ST-бисимуляции.
  - (b) Не существует  $a$ -помеченного перехода  $v'$  такого, что  $\pi'_P$  можно расширить на  $V' \cup v'$ . По определению  $\mathcal{S}$ ,  $(\pi_P, \pi'_P) \in \mathcal{R}$ . Тогда по определению шаговой бисимуляции  $\pi_P$  можно расширить на  $A \cup a$ . Пусть  $M'_P$  — маркировка сети  $N'$ , соответствующая процессу  $\pi'_P$  и  $\varphi'_E(V') = U'$ . Тогда  $\exists \bar{U}' \in \mathcal{M}(T_{N'}) : l_{N'}(\bar{U}') = A$ ,  $\exists u' \in T_{N'} : l_{N'}(u') = a$  и  $\bullet \bar{U}' + \bullet u' \subseteq M'_P$ . Тогда  $l_{N'}(U') = A = l_{N'}(\bar{U}')$ . Так как  $N'$  — строго помеченная сеть, то  $U' = \bar{U}'$  и процесс  $\pi_P$  можно расширить на  $V' \cup v'$ , где  $v'$  — переход с пометкой  $a$  такой, что  $\varphi'_E(v') = u'$ . Это противоречит исходному предположению.

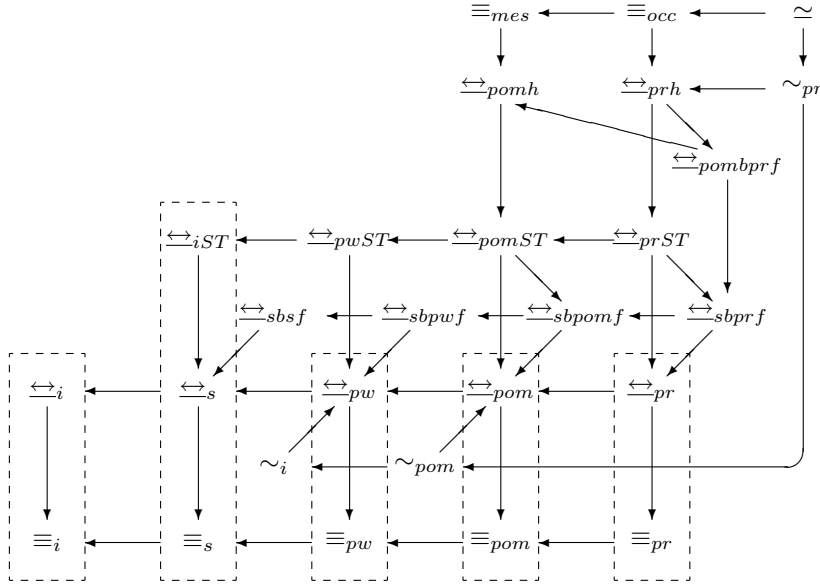


Рис. 1.34: Совпадение эквивалентностей на строго помеченных сетях

4. Как пункт 3, но роли  $N$  и  $N'$  меняются. □

На рисунке 1.34 пунктирной линией обведены эквивалентности, совпадающие на строго помеченных сетях.

Рассмотрим примеры, показывающие, какие эквивалентности не сливаются на строго помеченных сетях.

**Пример 1.2.6** • На рисунке 1.35(a)  $N \leftrightarrow_i N'$ , но  $N \not\equiv_s N'$ , так как только в сети  $N$  действия  $a$  и  $b$  могут выполняться параллельно.

- На рисунке 1.35(b)  $N \leftrightarrow_{pwST} N'$ , но  $N \not\equiv_{pom} N'$ , так как только в сети  $N'$  действие  $b$  может зависеть от  $a$ .
- На рисунке 1.12(d)  $N \equiv_{mes} N'$ , но  $N \not\equiv_{pr} N'$ .
- На рисунке 1.35(c)  $N \leftrightarrow_{pomST} N'$ , но  $N \not\leftrightarrow_{pomh} N'$ , так как только в сети  $N'$  может сработать последовательность из двух действий  $a$  и  $b$  так, что следующее действие,  $c$ , должно обязательно зависеть от  $a$ .
- На рисунке 1.35(d)  $N \leftrightarrow_{prh} N'$ , но  $N \not\equiv_{mes} N'$ , так как только сеть  $N'$  имеет развертку в  $O$ -сеть, в которой переходы с пометками  $a$  и  $b$  имеют общее входное место. Этой  $O$ -сети соответствует ПСС с конфликтными действиями  $a$  и  $b$ .
- На рисунке 1.13(d)  $N \equiv_{occ} N'$ , но  $N \not\sim N'$ .
- На рисунке 1.22(c)  $N \equiv_{occ} N'$ , но  $N \not\sim_i N'$ .

### Эквивалентности на T-сетях

**Определение 1.2.37** T-сеть — сеть  $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$  такая, что  $\forall p \in P_N \ | \bullet p | \leq 1$  и  $| p \bullet | \leq 1$ .

Рассмотрим, как ведут себя ранее введенные эквивалентности на T-сетях.

**Предложение 1.2.24** Для T-сетей без автопараллелизма  $N$  и  $N'$   $N \equiv_i N' \Leftrightarrow N \leftrightarrow_i N'$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие теоремы 1.2.1.

( $\Rightarrow$ ) Пусть  $N \equiv_i N'$ . Определим отношение  $\mathcal{R}$  следующим образом:  $\mathcal{R} = \{(\pi, \pi') \mid \pi_N = \pi_0 \xrightarrow{a_1} \dots \xrightarrow{a_n} \pi_n = \pi, \pi_{N'} = \pi'_0 \xrightarrow{a_1} \dots \xrightarrow{a_n} \pi'_n = \pi', a_i \in Act \ (1 \leq i \leq n)\}$ . Докажем, что  $\mathcal{R} : N \leftrightarrow_i N'$ . Проведем доказательство от противного.

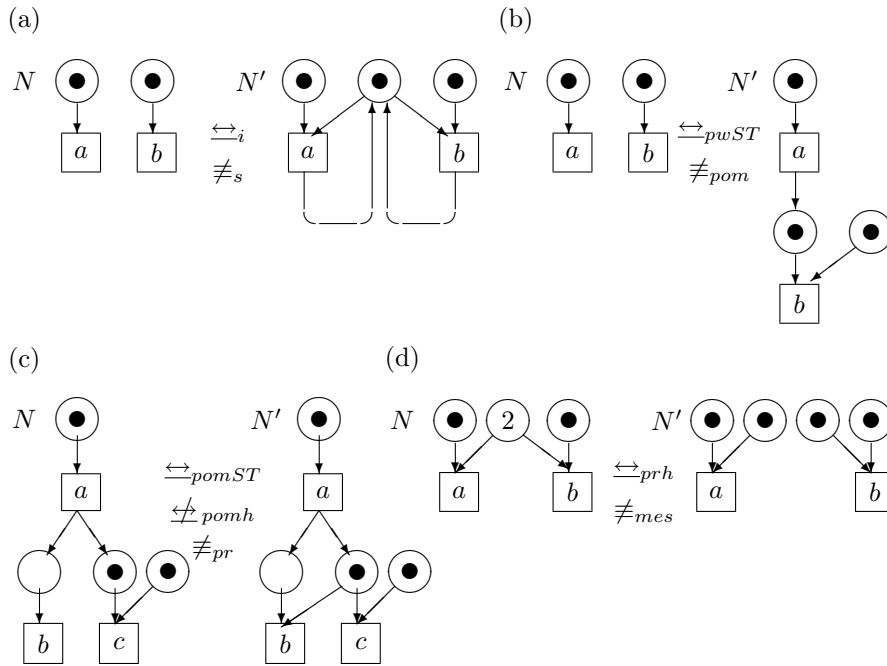


Рис. 1.35: Примеры эквивалентностей на сетях со строгой пометкой

1. Очевидно,  $(\pi_N, \pi_{N'}, \emptyset) \in \mathcal{R}$ .

2. Пусть  $(\pi, \pi') \in \mathcal{R}$ ,  $\pi \xrightarrow{a} \tilde{\pi}$ ,  $a \in Act$ , но  $\pi'$  невозможно расширить на действие  $a$ . По определению отношения  $\mathcal{R}$  имеем последовательности расширений процессов  $\pi_N = \pi_0 \xrightarrow{a_1} \dots \xrightarrow{a_n} \pi_n = \pi$  и  $\pi_{N'} = \pi'_0 \xrightarrow{a_1} \dots \xrightarrow{a_n} \pi'_n = \pi'$ . Так как  $N \equiv_i N'$ , то существуют  $\tilde{\pi}'_i \in \Pi(N')$  ( $0 \leq i \leq n$ ) такие, что  $\pi_{N'} = \tilde{\pi}'_0 \xrightarrow{a_1} \dots \xrightarrow{a_n} \tilde{\pi}'_n = \pi'$  и  $\tilde{\pi}'$  можно расширить на действие  $a$ . Пусть маркировки  $M'_i$  соответствуют процессам  $\pi'_i$ , а  $\bar{M}'_i - \tilde{\pi}'_i$  ( $0 \leq i \leq n$ ) и  $l_{N'}(t'_i) = l_{N'}(\tilde{t}'_i) = a_i$  ( $1 \leq i \leq n$ ). Тогда  $M_{N'} = M'_0 \xrightarrow{t'_1} \dots \xrightarrow{t'_n} M'_n = M'$  и  $M_{N'} = \bar{M}'_0 \xrightarrow{\tilde{t}'_1} \dots \xrightarrow{\tilde{t}'_n} \bar{M}'_n = \bar{M}'$  — последовательности срабатываний переходов в  $N'$ . Так как в  $M'$ , в отличие от  $\bar{M}'$ , не допустим ни один  $a$ -помеченный переход, то  $M' \neq \bar{M}'$ . Выберем наименьшее  $i$  ( $1 \leq i \leq n$ ) такое, что  $M'_{i-1} = \bar{M}'_{i-1}$ , но  $M'_i \neq \bar{M}'_i$ . Имеем  $M'_{i-1} \xrightarrow{t'_i} M'_i$  и  $M'_{i-1} \xrightarrow{\tilde{t}'_i} \bar{M}'_i$ . Очевидно, что  $t'_i \neq \tilde{t}'_i$  и  $\bullet t'_i \subseteq M'_{i-1}$ ,  $\bullet \tilde{t}'_i \subseteq M'_{i-1}$ . Имеем два случая.

(а)  $\bullet t'_i + \bullet \tilde{t}'_i \subseteq M'_{i-1}$ . Так как  $l_{N'}(t'_i) = l_{N'}(\tilde{t}'_i) = a_i$ , то действие  $a_i$  автопараллельно в  $N'$ , что противоречит предположению об отсутствии автопараллелизма.

(б)  $\bullet t'_i + \bullet \tilde{t}'_i \not\subseteq M'_{i-1}$ . Тогда  $\bullet t'_i \cap \bullet \tilde{t}'_i \neq \emptyset$ , что противоречит определению T-сетей.

Таким образом, оба варианта должны быть отвергнуты и  $\mathcal{R} : N \not\leftrightarrow_i N'$ .

3. Как пункт 2, но роли  $N$  и  $N'$  меняются. □

**Предложение 1.2.25** Для T-сетей без автопараллелизма  $N$  и  $N'$   $N \leftrightarrow_i N' \Leftrightarrow N \leftrightarrow_{iST} N'$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие теоремы 1.2.1.

( $\Rightarrow$ ) Пусть  $\mathcal{R} : N \leftrightarrow_i N'$ . Определим отношение  $\mathcal{S}$  следующим образом:  $\mathcal{S} = \{((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \mid (\pi_E, \pi'_E), (\pi_P, \pi'_P) \in \mathcal{R}, (\pi_E, \pi_P) \in ST - \Pi(N), (\pi'_E, \pi'_P) \in ST - \Pi(N'), \beta : \rho_{C_E} \simeq \rho_{C'_E}, \beta(T_{C_P}) = T_{C'_P}\}$ . Докажем, что  $\mathcal{S} : N \leftrightarrow_{iST} N'$ .

Достаточно рассматривать только расширения ST-процессов вида  $(\pi_E, \pi_P) \rightarrow (\tilde{\pi}_E, \tilde{\pi}_P)$ , где  $\pi_E \rightarrow \tilde{\pi}_E$  — расширение на одно действие.

1. Очевидно,  $((\pi_N, \pi_N), (\pi_{N'}, \pi_{N'}), \emptyset) \in \mathcal{S}$ .

2. По определению  $\mathcal{S}$ ,  $((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{S} \Rightarrow \beta : \rho_{C_E} \simeq \rho_{C'_E}$  и  $\beta(T_{C_P}) = T_{C'_P}$ ;

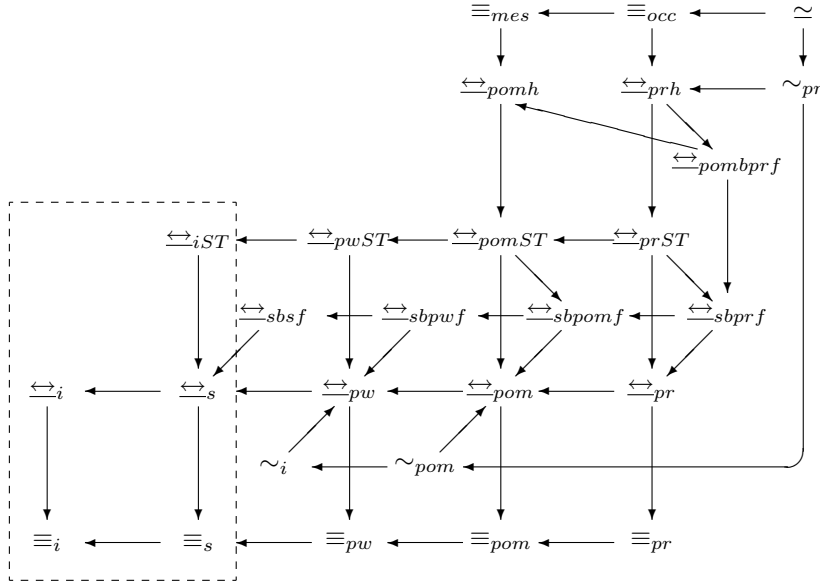


Рис. 1.36: Совпадение эквивалентностей на Т-сетях без автопараллелизма

3. Пусть  $((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{S}$  и  $(\pi_E, \pi_P) \rightarrow (\tilde{\pi}_E, \pi_P)$ , где  $(\tilde{\pi}_E, \pi_P) \in ST - \Pi(N)$ ,  $\pi_E \xrightarrow{v} \tilde{\pi}_E$  и  $v$  — переход с пометкой  $a$ . Так как  $(\pi_E, \pi_P) \in ST - \Pi(N)$ , по определению ST-процесса,  $\exists V \subseteq T_{C_E} \pi_P \xrightarrow{V} \pi_E$ ,  $l_{C_E}(V) = A \in \mathcal{M}(Act)$ . Тогда  $\pi_P \xrightarrow{V \cup v} \tilde{\pi}_E$ , и все действия в  $A \cup a$  различны из-за отсутствия автопараллелизма в  $N$ . Так как  $(\pi'_E, \pi'_P) \in ST - \Pi(N')$ ,  $\exists V' \subseteq T_{C'_E} \pi'_P \xrightarrow{V'} \pi'_E$ . По определению  $\mathcal{S}$ ,  $l_{C'_E}(V') = A$ . Имеем следующие два случая.

- Существуют процесс  $\tilde{\pi}'_E$  и переход  $v'$  с пометкой  $a$  такие, что  $\pi'_P \xrightarrow{V' \cup v'} \tilde{\pi}'_E$ . Тогда  $(\tilde{\pi}'_E, \pi'_P) \in ST - \Pi(N')$ . Определим отображение  $\tilde{\beta}$  следующим образом:  $\tilde{\beta}|_{T_{C_E}} = \beta$ ,  $\tilde{\beta}(v) = v'$ . Тогда  $((\tilde{\pi}_E, \pi_P), (\tilde{\pi}'_E, \pi'_P), \tilde{\beta}) \in \mathcal{S}$  и выполняются все требования пункта 3 определения интерливинговой ST-бисимуляции.
- Не существует  $a$ -помеченного перехода  $v'$  такого, что  $\pi'_P$  можно расширить на  $V' \cup v'$ . По определению  $\mathcal{S}$ ,  $(\pi_P, \pi'_P) \in \mathcal{R}$ . Тогда по определению интерливинговой бисимуляции существуют процесс  $\tilde{\pi}'_P$  и  $a$ -помеченный переход  $w'$  такие, что  $\pi'_P \xrightarrow{w'} \tilde{\pi}'_P$ . Пусть  $M'_P$  — маркировка сети  $N'$ , соответствующая процессу  $\pi'_P$ ,  $U' = \varphi'_E(V')$ ,  $u' = \tilde{\varphi}'_P(w')$ . Тогда  $\bullet U' \subseteq M'_P$ ,  $\bullet u' \subseteq M'_P$ , но  $\bullet U' + \bullet u' \not\subseteq M'_P$ . Заметим, что все переходы в  $U' \cup u'$  различны, так как различны действия в  $A \cup a$ . Тогда  $\exists t' \in U' \bullet t' \cap \bullet u' \neq \emptyset$  и  $t' \neq u'$ . Имеем противоречие с определением Т-сетей.

4. Как пункт 3, но роли  $N$  и  $N'$  меняются. □

**Следствие 1.2.2** Для Т-сетей без автопараллелизма  $N$  и  $N'$   $N \equiv_i N' \Leftrightarrow N \leftrightarrow_{iST} N'$ .

На рисунке 1.36 пунктирной линией обведены эквивалентности, совпадающие на Т-сетях без автопараллелизма.

Заметим, что к безопасным Т-сетям без автопараллелизма можно применить результаты [270] и получить совпадение шаговой и ЧУММ следовых эквивалентностей.

Рассмотрим примеры, показывающие, какие эквивалентности не сливаются на Т-сетях.

**Пример 1.2.7** • На рисунке 1.37(a)  $N \leftrightarrow_i N'$ , но  $N \not\equiv_s N'$ , так как только в сети  $N'$  действие  $a$  не может сработать параллельно с самим собой.

- На рисунке 1.37(b)  $N \equiv_s N'$ , но  $N \not\equiv_{pw} N'$ , так как сеть  $N$  является представлением такого ЧУММ, что даже менее последовательный ЧУММ нельзя выполнить в сети  $N'$ .
- На рисунке 1.35(b)  $N \leftrightarrow_{pwST} N'$ , но  $N \not\equiv_{pom} N'$ .
- На рисунке 1.12(d)  $N \equiv_{mes} N'$ , но  $N \not\equiv_{pr} N'$ .

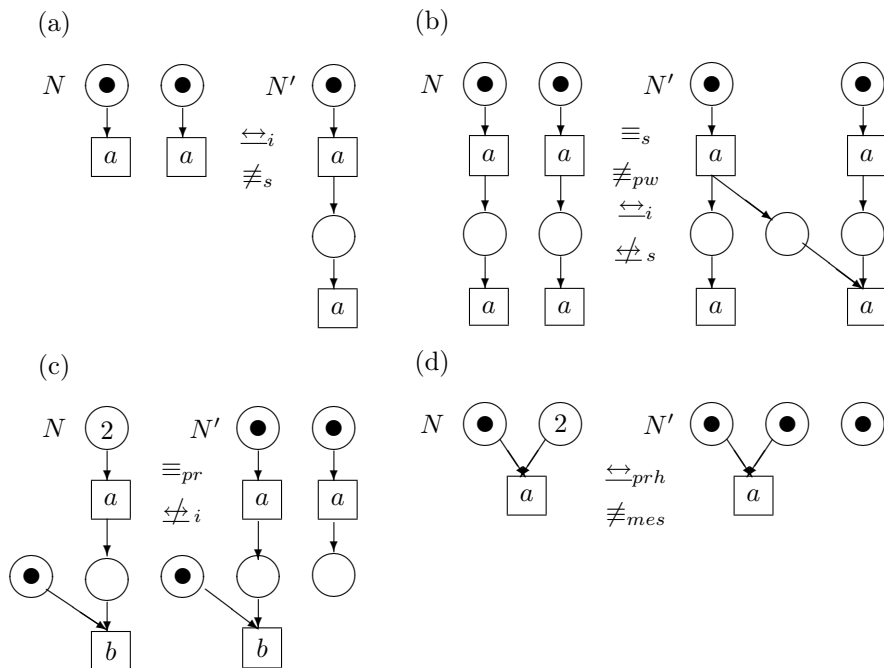


Рис. 1.37: Примеры эквивалентностей на T-сетях

- На рисунке 1.37(c)  $N \equiv_{pr} N'$ , но  $N \not\equiv_i N'$ , так как только в сети  $N'$  можно выполнить действие  $a$  так, что после него не может выполняться  $b$ .
- На рисунке 1.37(d)  $N \leftrightarrow_{prh} N'$ , но  $N \not\equiv_{mes} N'$ , так как только сети  $N'$  соответствует ПСС с двумя конфликтными действиями  $a$ .
- На рисунке 1.13(d)  $N \equiv_{occ} N'$ , но  $N \neq N'$ .

### 1.3 Сети Петри с невидимыми переходами

В этом разделе мы рассмотрим расширение сетей Петри — сети Петри с невидимыми переходами. Переходы сетей этого класса могут быть помечены специальным символом  $\tau \notin Act$ , обозначающим *невидимое, внутреннее* или “*тихое*” действие, которое не воспринимается внешним наблюдателем, следящим за поведением моделируемой системы, но может, тем не менее, оказывать косвенное влияние на это поведение. Обозначим  $Act_\tau = Act \cup \{\tau\}$ .

**Определение 1.3.1** Сеть с невидимыми переходами — сеть  $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$  такая, что  $l_N : T_N \rightarrow Act_\tau$ , то есть переходы сети могут помечаться символом  $\tau$ .

На сетях Петри с невидимыми переходами вводятся  $\tau$ -эквивалентности, которые абстрагируются от невидимых  $\tau$ .  $\tau$ -эквивалентности обозначаются символами, надписанными знаком  $\tau$ , чтобы отличить их от обычных эквивалентностей.

$\tau$ -эквивалентности совпадают с эквивалентностями на сетях Петри с видимыми переходами (обычных сетях). Так как обычные сети Петри — подкласс сетей Петри с невидимыми переходами, то, возможно, некоторые  $\tau$ -эквивалентности сливаются на них. Кроме того, некоторые связи, которые существовали между эквивалентностями на обычных сетях Петри, могут перестать существовать между их аналогами —  $\tau$ -эквивалентностями на сетях Петри с невидимыми переходами.

В [275, 276, 280, 282] рассмотрен пример на структурах событий с невидимыми действиями, демонстрирующий независимость ST- и сохраняющих историю  $\tau$ -бисимуляционных эквивалентностей.

Таким образом, в рамках сетей с невидимыми переходами взаимосвязи  $\tau$ -эквивалентностей резко меняются по сравнению с соответствующими им эквивалентностными отношениями на обычных сетях.

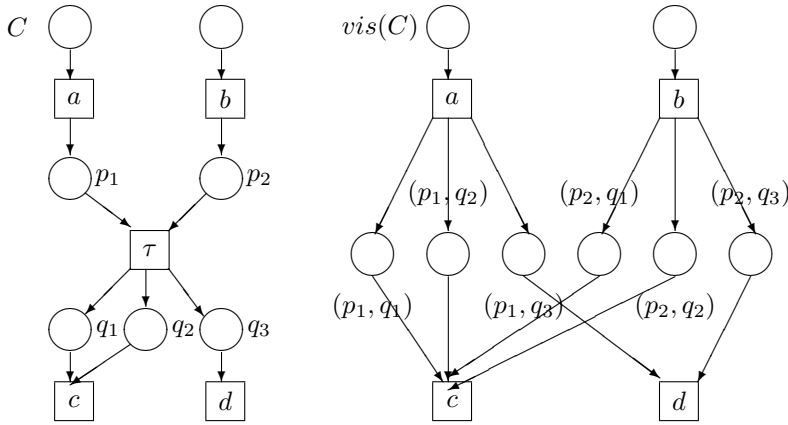


Рис. 1.38: Действие функции  $vis$  на С-сеть

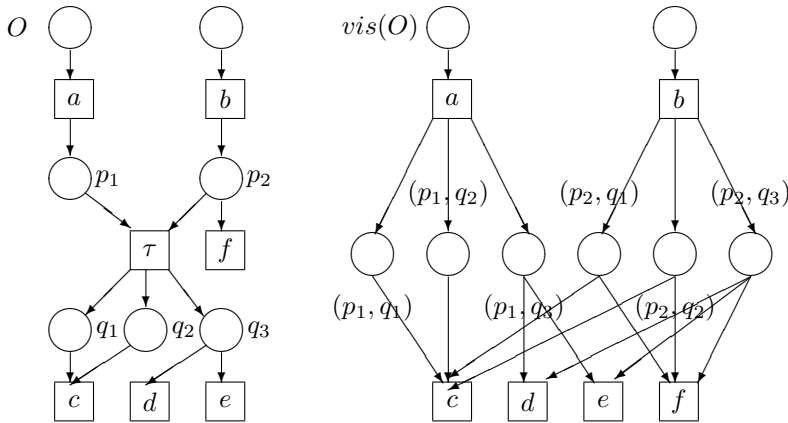


Рис. 1.39: Действие функции  $vis$  на О-сеть

Попытаемся определить  $\tau$ -эквивалентности в процессной семантике. Мы абстрагируемся от невидимых переходов С-сетей и О-сетей удалением таких переходов и “перемножением” их входных и выходных мест с сохранением всех дуг, связанных с этими местами. После такого преобразования, очевидно, сохраняются все причинные зависимости видимых переходов. Последовательно применяя данное преобразование к сети с невидимыми переходами, мы в конечном итоге получим сеть, имеющую только видимые переходы. Пусть такое полное абстрагирование от невидимых переходов осуществляет функция  $vis$ .

**Пример 1.3.1** Действие функции  $vis$  на С-сеть показано на рисунке 1.38, а на О-сеть — на рисунке 1.39.

Так как все причинно-следственные связи сохранены, ЧУММ  $\tau$ -эквивалентности будут следовать из процессных. Однако во время такого абстрагирования может измениться количество входных и выходных мест некоторых видимых переходов. В результате этого возникают различные проблемы.

В частности из процессной сохраняющей историю  $\tau$ -бисимуляционной эквивалентности ( $\Leftrightarrow_{prh}^\tau$ ) не будет следовать процессная  $\tau$ -бисимуляционная ( $\Leftrightarrow_{pr}^\tau$ ), что демонстрируется следующим примером.

**Пример 1.3.2** На рисунке 1.40  $N \Leftrightarrow_{prh}^\tau N'$ , но  $N \not\Leftrightarrow_{pr}^\tau N'$ , так как абстракцией С-сетей процессов, расширяющих начальные процессы  $N$  и  $N'$ , всегда будет либо сеть, состоящая всего из двух мест, либо сеть, образованная  $a$ -помеченным переходом с двумя входными местами, но только сеть  $N'$  после срабатывания невидимого перехода может перейти в состояние, из которого возможен процесс с С-сетью из  $a$ -помеченного перехода с одним входным местом.

Поэтому, нет смысла вводить процессные  $\tau$ -эквивалентности.



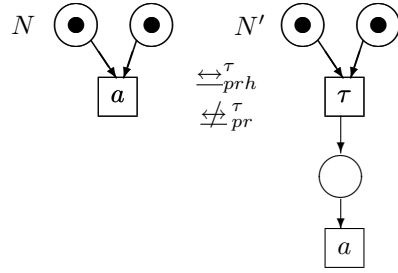


Рис. 1.40: Нарушение взаимосвязей процессных  $\tau$ -бисимуляционных эквивалентностей по сравнению с процессными бисимуляционными эквивалентностями

В качестве еще одного примера сильного изменения взаимосвязей эквивалентностей на сетях с невидимыми переходами можно привести следующий факт: МСС  $\tau$ -сохраняющая история эквивалентность не влечет даже обычные  $\tau$ -бисимуляционные отношения, а только  $\tau$ -следовые.

### 1.3.1 Базисные $\tau$ -эквивалентности

Следующие базисные  $\tau$ -эквивалентности были рассмотрены в литературе.

- *$\tau$ -следовые эквивалентности* (учитывают только протоколы функционирования систем).
  - Интерливинговая  $\tau$ -следовая эквивалентность ( $\equiv_i^\tau$ ) введена в [240] на сетях Петри. Определение для систем переходов имеется в [152].
  - Шаговая  $\tau$ -следовая эквивалентность ( $\equiv_s^\tau$ ) введена в [240] на сетях Петри.
  - ЧС  $\tau$ -следовая эквивалентность ( $\equiv_{pw}^\tau$ ) введена в [277] на сетях Петри.
  - ЧУММ  $\tau$ -следовая эквивалентность ( $\equiv_{pom}^\tau$ ) введена в [240] на сетях Петри.
- *Обычные  $\tau$ -бисимуляционные эквивалентности* (учитывают ветвистую структуру функционирования систем).
  - Интерливинговая  $\tau$ -бисимуляционная эквивалентность ( $\leftrightarrow_i^\tau$ ) введена в [198] на системах переходов и в [240, 245] на сетях Петри. Определение для систем переходов имеется в [152].
  - Шаговая  $\tau$ -бисимуляционная эквивалентность ( $\leftrightarrow_s^\tau$ ) введена в [240] на сетях Петри.
  - ЧС  $\tau$ -бисимуляционная эквивалентность ( $\leftrightarrow_{pw}^\tau$ ) введена в [276] на сетях Петри.
  - ЧУММ  $\tau$ -бисимуляционная эквивалентность ( $\leftrightarrow_{pom}^\tau$ ) введена в [240] на сетях Петри.
- *ST- $\tau$ -бисимуляционные эквивалентности* (учитывают продолжительность срабатывания переходов в функционировании систем).
  - Интерливинговая ST- $\tau$ -бисимуляционная эквивалентность ( $\leftrightarrow_{iST}^\tau$ ) введена в [276] на сетях Петри.
  - ЧС ST- $\tau$ -бисимуляционная эквивалентность ( $\leftrightarrow_{pwST}^\tau$ ) введена в [276] на сетях Петри.
  - ЧУММ ST- $\tau$ -бисимуляционная эквивалентность ( $\leftrightarrow_{pomST}^\tau$ ) введена в [276] на сетях Петри.
- *Сохраняющие историю  $\tau$ -бисимуляционные эквивалентности* (учитывают “прошрое” или “историю” функционирования систем).
  - ЧУММ сохраняющая историю  $\tau$ -бисимуляционная эквивалентность ( $\leftrightarrow_{pomh}^\tau$ ) введена в [120, 121, 122] на сетях Петри.
- *Сохраняющие историю ST- $\tau$ -бисимуляционные эквивалентности* (учитывают “историю” и продолжительность срабатывания переходов в функционировании систем).

- ЧУММ сохраняющая историю ST- $\tau$ -бисимуляционная эквивалентность ( $\leftrightarrow_{pomhST}^\tau$ ) введена в [120, 121, 122] на сетях Петри.
- *Обычные ветвистые  $\tau$ -бисимуляционные эквивалентности* (учитывают ветвистую структуру функционирования систем и влияние на нее невидимых действий).
  - Интерливинговая ветвистая  $\tau$ -бисимуляционная эквивалентность ( $\leftrightarrow_{ibr}^\tau$ ) введена в [162, 163, 151, 165, 152, 153, 166] на системах переходов и в [245] на сетях Петри.
- *Сохраняющие историю ветвистые  $\tau$ -бисимуляционные эквивалентности* (учитывают “историю” и ветвистую структуру функционирования систем и влияние на нее невидимых действий).
  - ЧУММ сохраняющая историю ветвистая  $\tau$ -бисимуляционная эквивалентность ( $\leftrightarrow_{pomhbr}^\tau$ ) введена в [120] на сетях Петри.

Кроме того, был определен ряд других  $\tau$ -эквивалентностей, обзор большей части которых дан в [152] на системах переходов. Среди них — следующие понятия.

Интерливинговая семи-ветвистая  $\tau$ -бисимуляционная эквивалентность ( $\leftrightarrow_{isbr}^\tau$ ) была введена в [162, 163] на системах переходов.

Интерливинговая квази-ветвистая  $\tau$ -бисимуляционная эквивалентность ( $\leftrightarrow_{iqbr}^\tau$ ) была введена в [103] на структурах событий. Определение на системах переходов дано в [52], где эквивалентность названа простой ветвистой бисимуляционной.

Интерливинговая  $\eta$ - $\tau$ -бисимуляционная эквивалентность ( $\leftrightarrow_{in}^\tau$ ) была введена в [60] на системах переходов.

Интерливинговая  $\Delta$ - $\tau$ -бисимуляционная эквивалентность ( $\leftrightarrow_{i\Delta}^\tau$ ) была введена в [199, 285] и в [286], где она названа бисимуляционной с задержкой, на системах переходов.

Мы переносим указанные  $\tau$ -эквивалентности, определенные на других моделях, в рамки сетей Петри, а также расширяем множество базисных понятий  $\tau$ -эквивалентностей  $\tau$ -сохраняющими конфликт (полностью учитывают конфликт): вводится МСС  $\tau$ -сохраняющая конфликт эквивалентность ( $\equiv_{mes}^\tau$ ).

Проводится сравнение всех базисных  $\tau$ -эквивалентностей и устанавливаются их взаимосвязи, что является новым результатом для такого широкого их набора.

### $\tau$ -следовые эквивалентности

$\tau$ -следовые эквивалентности получаются из следовых эквивалентностей абстрагированием от невидимых действий.

Обозначим через  $\varepsilon$  пустую строку.

Пусть  $\sigma = a_1 \cdots a_n \in Act_\tau^*$ . Определим  $vis(\sigma)$  следующим образом (в следующем определении  $a \in Act_\tau$ ).

1.  $vis(\varepsilon) = \varepsilon$ ;
2.  $vis(\sigma a) \begin{cases} vis(\sigma)a, & a \in Act; \\ vis(\sigma), & \text{иначе.} \end{cases}$

**Определение 1.3.2** Видимый интерливинговый след *сетей*  $N$  — это последовательность  $vis(a_1 \cdots a_n) \in Act^*$  такая, что  $\pi_N \xrightarrow{a_1} \pi_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} \pi_n$ , где  $\pi_i \in \Pi(N)$  ( $1 \leq i \leq n$ ). Обозначим множество всех видимых интерливинговых следов *сетей*  $N$  через  $VisIntTraces(N)$ . Две *сети*  $N$  и  $N'$  интерливингово  $\tau$ -следово эквивалентны, запись  $N \equiv_i^\tau N'$ , если  $VisIntTraces(N) = VisIntTraces(N')$ .

Пусть  $\Sigma = A_1 \cdots A_n \in (\mathcal{M}(Act_\tau))^*$ . Определим  $vis(\Sigma)$  следующим образом (в следующем определении  $A \in \mathcal{M}(Act_\tau)$ ).

1.  $vis(\varepsilon) = \varepsilon$ ;
2.  $vis(\Sigma A) \begin{cases} vis(\Sigma)(A \cap Act), & A \cap Act \neq \emptyset; \\ vis(\Sigma), & \text{иначе.} \end{cases}$

**Определение 1.3.3** Видимый шаговый след *сетей*  $N$  — это последовательность  $vis(A_1 \cdots A_n) \in (\mathcal{M}(Act))^*$  такая, что  $\pi_N \xrightarrow{A_1} \pi_1 \xrightarrow{A_2} \dots \xrightarrow{A_n} \pi_n$ , где  $\pi_i \in \Pi(N)$  ( $1 \leq i \leq n$ ). Обозначим множество всех видимых шаговых следов *сетей*  $N$  через  $VisStepTraces(N)$ . Две *сети*  $N$  и  $N'$  шагово  $\tau$ -следово эквивалентны, запись  $N \equiv_s^\tau N'$ , если  $VisStepTraces(N) = VisStepTraces(N')$ .

Пусть  $\rho = \langle X, \prec, l \rangle$  — ПЧУМ такой, что  $l : X \rightarrow Act_\tau$ . Обозначим  $vis(X) = \{x \in X \mid l(x) \in Act\}$  и  $vis(\rho) = \rho|_{vis(X)}$ .

**Определение 1.3.4** Видимый ЧУММ след сети  $N$  — это ЧУММ  $vis(\rho)$  — класс изоморфизма ПЧУМ  $vis(\rho_C)$  для  $\pi = (C, \varphi) \in \Pi(N)$ . Обозначим через  $VisPomsets(N)$  множество всех видимых ЧУММ следов сети  $N$ . Две сети  $N$  и  $N'$   $\tau$ -следово эквивалентны ЧС (ЧС), запись  $N \equiv_{pw}^\tau N'$ , если  $VisPomsets(N) \subseteq VisPomsets(N')$  и  $VisPomsets(N') \subseteq VisPomsets(N)$ .

**Определение 1.3.5** Две сети  $N$  и  $N'$  ЧУММ  $\tau$ -следово эквивалентны, запись  $N \equiv_{pom}^\tau N'$ , если  $VisPomsets(N) = VisPomsets(N')$ .

### $\tau$ -бисимуляционные эквивалентности

$\tau$ -бисимуляционные эквивалентности получаются из бисимуляционных эквивалентностей абстрагированием от невидимых действий.

Пусть  $C = \langle P_C, T_C, F_C, l_C \rangle$  — С-сеть с невидимыми переходами. Обозначим  $vis(T_C) = \{v \in T_C \mid l_C(v) \in Act\}$  и  $vis(\prec_C) = \prec_C \cap (vis(T_C) \times vis(T_C))$ .

### Обычные $\tau$ -бисимуляционные эквивалентности

**Определение 1.3.6** Пусть  $N$  и  $N'$  — некоторые сети с невидимыми переходами. Отношение  $\mathcal{R} \subseteq \Pi(N) \times \Pi(N')$  —  $\star$ - $\tau$ -бисимуляция между  $N$  и  $N'$ ,  $\star \in \{\text{интерливингивая, шаговая, ЧС, ЧУММ}\}$ , запись  $\mathcal{R} : N \leftrightarrow_\star^\tau N'$ ,  $\star \in \{i, s, pw, pom\}$ , если:

1.  $(\pi_N, \pi_{N'}) \in \mathcal{R}$ .

2.  $(\pi, \pi') \in \mathcal{R}$ ,  $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$ ,

- (a)  $|vis(T_{\hat{C}})| = 1$ , если  $\star = i$ ;

- (b)  $vis(\prec_{\hat{C}}) = \emptyset$ , если  $\star = s$ ;

$$\Rightarrow \exists \tilde{\pi}' : \pi' \xrightarrow{\tilde{\pi}'} \tilde{\pi}', (\tilde{\pi}, \tilde{\pi}') \in \mathcal{R} \text{ и}$$

- (a)  $vis(\rho_{\hat{C}}) \subseteq vis(\rho_{\tilde{C}})$ , если  $\star = pw$ ;

- (b)  $vis(\rho_{\hat{C}}) \simeq vis(\rho_{\tilde{C}})$ , если  $\star \in \{i, s, pom\}$ .

3. Как пункт 2, но роли  $N$  и  $N'$  меняются.

Сети  $N$  и  $N'$   $\star$ - $\tau$ -бисимуляционно эквивалентны,  $\star \in \{\text{интерливингивая, шаговая, ЧС, ЧУММ}\}$ , запись  $N \leftrightarrow_\star^\tau N'$ , если  $\exists \mathcal{R} : N \leftrightarrow_\star^\tau N'$ ,  $\star \in \{i, s, pw, pom\}$ .

**ST- $\tau$ -бисимуляционные эквивалентности** При введении понятия ST- $\tau$ -процессов на сетях Петри с  $\tau$ -действиями предполагается, что действие  $\tau$  срабатывает мгновенно. Поэтому в срабатывании перехода, помеченного таким действием, невозможно различить начало и конец, и он не может находиться в множестве работающих переходов.

**Определение 1.3.7** ST- $\tau$ -процесс сети  $N$  — пара  $(\pi_E, \pi_P)$  такая, что  $\pi_E, \pi_P \in \Pi(N)$ ,  $\pi_P \xrightarrow{\pi_W} \pi_E$  и  $\forall v, w \in T_{C_E} (v \prec_{C_E} w) \vee (l_{C_E}(v) = \tau) \Rightarrow v \in T_{C_P}$ .

В этом случае  $\pi_E$  — процесс, начавший работать, то есть все действия  $\pi_E$  начали выполняться. Процесс  $\pi_P$  соответствует той части  $\pi_E$ , которая уже завершилась, а  $\pi_W$  — той части, которая еще работает. Очевидно, что  $\prec_{C_W} = \emptyset$ .  $ST^\tau - \Pi(N)$  обозначает множество всех ST- $\tau$ -процессов сети  $N$ .

$(\pi_N, \pi_{N'})$  будет начальным ST- $\tau$ -процессом  $N$ . Пусть  $(\pi_E, \pi_P), (\tilde{\pi}_E, \tilde{\pi}_P) \in ST^\tau - \Pi(N)$ . Пишем  $(\pi_E, \pi_P) \rightarrow (\tilde{\pi}_E, \tilde{\pi}_P)$ , если  $\pi_E \rightarrow \tilde{\pi}_E$  и  $\pi_P \rightarrow \tilde{\pi}_P$ .

**Определение 1.3.8** Пусть  $N$  и  $N'$  — некоторые сети с невидимыми переходами. Отношение  $\mathcal{R} \subseteq ST^\tau - \Pi(N) \times ST^\tau - \Pi(N') \times \mathcal{B}$ , где  $\mathcal{B} = \{\beta \mid \beta : vis(T_C) \rightarrow vis(T_{C'})\}$ ,  $\pi = (C, \varphi) \in \Pi(N)$ ,  $\pi' = (C', \varphi') \in \Pi(N')$  —  $\star$ -ST- $\tau$ -бисимуляция между  $N$  и  $N'$ ,  $\star \in \{\text{интерливингивая, ЧС, ЧУММ}\}$ , запись  $\mathcal{R} : N \leftrightarrow_{\star ST}^\tau N'$ ,  $\star \in \{i, pw, pom\}$ , если:

1.  $((\pi_N, \pi_N), (\pi_{N'}, \pi_{N'}), \emptyset) \in \mathcal{R}$ .
2.  $((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{R} \Rightarrow \beta : vis(\rho_{C_E}) \asymp vis(\rho_{C'_E})$  и  $\beta(vis(T_{C_P})) = vis(T_{C'_P})$ .
3.  $((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{R}, (\pi_E, \pi_P) \rightarrow (\tilde{\pi}_E, \tilde{\pi}_P) \Rightarrow \exists \tilde{\beta}, (\tilde{\pi}'_E, \tilde{\pi}'_P) : (\pi'_E, \pi'_P) \rightarrow (\tilde{\pi}'_E, \tilde{\pi}'_P), \tilde{\beta}|_{vis(T_{C_E})} = \beta, ((\tilde{\pi}_E, \tilde{\pi}_P), (\tilde{\pi}'_E, \tilde{\pi}'_P), \tilde{\beta}) \in \mathcal{R}$ , и если  $\pi_P \xrightarrow{\pi} \tilde{\pi}_P, \pi'_P \xrightarrow{\pi'} \tilde{\pi}'_P, \gamma = \tilde{\beta}|_{T_C}$ , то:
  - (a)  $\gamma^{-1} : vis(\rho_{C'}) \sqsubseteq vis(\rho_C)$ , если  $\star = pw$ ;
  - (b)  $\gamma : vis(\rho_C) \simeq vis(\rho_{C'})$ , если  $\star = pot$ .
4. Как пункт 3, но роли  $N$  и  $N'$  меняются.

Сети  $N$  и  $N'$   $\star$ -ST- $\tau$ -бисимуляционно эквивалентны,  $\star \in \{\text{интерливингивая, ЧС, ЧУММ}\}$ , запись  $N \xleftrightarrow{\star}_{ST}^{\tau} N'$ , если  $\exists \mathcal{R} : N \xleftrightarrow{\star}_{ST}^{\tau} N', \star \in \{i, pw, pot\}$ .

### Сохраняющие историю $\tau$ -бисимуляционные эквивалентности

**Определение 1.3.9** Пусть  $N$  и  $N'$  — некоторые сети с невидимыми переходами. Отношение  $\mathcal{R} \subseteq \Pi(N) \times \Pi(N') \times \mathcal{B}$ , где  $\mathcal{B} = \{\beta \mid \beta : vis(T_C) \rightarrow vis(T_{C'}), \pi = (C, \varphi) \in \Pi(N), \pi' = (C', \varphi') \in \Pi(N')\}$  — ЧУММ  $\star$ -сохраняющая историю  $\tau$ -бисимуляция между  $N$  и  $N'$ , запись  $N \xleftrightarrow{\star}_{pot}^{\tau} N'$ , если:

1.  $(\pi_N, \pi_{N'}, \emptyset) \in \mathcal{R}$ .
2.  $(\pi, \pi', \beta) \in \mathcal{R} \Rightarrow \beta : vis(\rho_C) \simeq vis(\rho_{C'})$ .
3.  $(\pi, \pi', \beta) \in \mathcal{R}, \pi \rightarrow \tilde{\pi} \Rightarrow \exists \tilde{\beta}, \tilde{\pi}' : \pi' \rightarrow \tilde{\pi}', \tilde{\beta}|_{vis(T_C)} = \beta, (\tilde{\pi}, \tilde{\pi}', \tilde{\beta}) \in \mathcal{R}$ .
4. Как пункт 3, но роли  $N$  и  $N'$  меняются.

Сети  $N$  и  $N'$  ЧУММ  $\star$ -сохраняюще историю  $\tau$ -бисимуляционно эквивалентны, запись  $N \xleftrightarrow{\star}_{pot}^{\tau} N'$ , если  $\exists \mathcal{R} : N \xleftrightarrow{\star}_{pot}^{\tau} N'$ .

### Сохраняющие историю ST- $\tau$ -бисимуляционные эквивалентности

**Определение 1.3.10** Пусть  $N$  и  $N'$  — некоторые сети с невидимыми переходами. Отношение  $\mathcal{R} \subseteq ST^{\tau} - \Pi(N) \times ST^{\tau} - \Pi(N') \times \mathcal{B}$ , где  $\mathcal{B} = \{\beta \mid \beta : vis(T_C) \rightarrow vis(T_{C'}), \pi = (C, \varphi) \in \Pi(N), \pi' = (C', \varphi') \in \Pi(N')\}$  — ЧУММ  $\star$ -сохраняющая историю ST- $\tau$ -бисимуляция между  $N$  и  $N'$ , запись  $\mathcal{R} : N \xleftrightarrow{\star}_{pot}^{\tau} N'$ , если:

1.  $((\pi_N, \pi_N), (\pi_{N'}, \pi_{N'}), \emptyset) \in \mathcal{R}$ .
2.  $((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{R} \Rightarrow \beta : vis(\rho_{C_E}) \simeq vis(\rho_{C'_E})$  и  $\beta(vis(T_{C_P})) = vis(T_{C'_P})$ .
3.  $((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{R}, (\pi_E, \pi_P) \rightarrow (\tilde{\pi}_E, \tilde{\pi}_P) \Rightarrow \exists \tilde{\beta}, (\tilde{\pi}'_E, \tilde{\pi}'_P) : (\pi'_E, \pi'_P) \rightarrow (\tilde{\pi}'_E, \tilde{\pi}'_P), \tilde{\beta}|_{vis(T_{C_E})} = \beta, ((\tilde{\pi}_E, \tilde{\pi}_P), (\tilde{\pi}'_E, \tilde{\pi}'_P), \tilde{\beta}) \in \mathcal{R}$ .
4. Как пункт 3, но роли  $N$  и  $N'$  меняются.

Сети  $N$  и  $N'$  ЧУММ  $\star$ -сохраняюще историю-ST- $\tau$ -бисимуляционно эквивалентны, запись  $N \xleftrightarrow{\star}_{pot}^{\tau} N'$ , если  $\exists \mathcal{R} : N \xleftrightarrow{\star}_{pot}^{\tau} N'$ .

**Обычные ветвистые  $\tau$ -бисимуляционные эквивалентности** Для некоторой сети  $N$  и  $\pi, \tilde{\pi} \in \Pi(N)$  пишем  $\pi \Rightarrow \tilde{\pi}$ , когда  $\exists \hat{\pi} = (\hat{C}, \hat{\varphi})$  такой, что  $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$  и  $vis(T_{\hat{C}}) = \emptyset$ .

**Определение 1.3.11** Пусть  $N$  и  $N'$  — некоторые сети с невидимыми переходами. Отношение  $\mathcal{R} \subseteq \Pi(N) \times \Pi(N')$  — интерливинговая ветвистая  $\tau$ -бисимуляция между  $N$  и  $N'$ , запись  $N \xleftrightarrow{\tau}_{ibr} N'$ , если:

1.  $(\pi_N, \pi_{N'}) \in \mathcal{R}$ .
2.  $(\pi, \pi') \in \mathcal{R}, \pi \xrightarrow{a} \tilde{\pi} \Rightarrow$ 
  - (a)  $a = \tau$  и  $(\tilde{\pi}, \pi') \in \mathcal{R}$  или

(b)  $a \neq \tau$  и  $\exists \tilde{\pi}', \tilde{\pi}' : \pi' \Rightarrow \tilde{\pi}' \xrightarrow{a} \tilde{\pi}'$ ,  $(\pi, \tilde{\pi}') \in \mathcal{R}$ ,  $(\tilde{\pi}, \tilde{\pi}') \in \mathcal{R}$ .

3. Как пункт 2, но роли  $N$  и  $N'$  меняются.

Сети  $N$  и  $N'$  интерливингово ветвисто  $\tau$ -бисимуляционно эквивалентны, запись  $N \xleftrightarrow{\tau}_{ibr} N'$ , если  $\exists \mathcal{R} : N \xleftrightarrow{\tau}_{ibr} N'$ .

### Сохраняющие историю ветвистые $\tau$ -бисимуляционные эквивалентности

**Определение 1.3.12** Пусть  $N$  и  $N'$  — некоторые сети с невидимыми переходами. Отношение  $\mathcal{R} \subseteq \Pi(N) \times \Pi(N') \times \mathcal{B}$ , где  $\mathcal{B} = \{\beta \mid \beta : T_C \rightarrow T_{C'}, \pi = (C, \varphi) \in \Pi(N), \pi' = (C', \varphi') \in \Pi(N')\}$ , — ЧУММ сохраняющая историю ветвистая  $\tau$ -бисимуляция между  $N$  и  $N'$ , запись  $N \xleftrightarrow{\tau}_{pomhbr} N'$ , если:

1.  $(\pi_N, \pi_{N'}, \emptyset) \in \mathcal{R}$ .

2.  $(\pi, \pi', \beta) \in \mathcal{R} \Rightarrow \beta : vis(\rho_C) \simeq vis(\rho_{C'})$ .

3.  $(\pi, \pi', \beta) \in \mathcal{R}$ ,  $\pi \rightarrow \tilde{\pi} \Rightarrow$

(a)  $(\tilde{\pi}, \pi', \beta) \in \mathcal{R}$  или

(b)  $\exists \tilde{\beta}, \tilde{\pi}', \tilde{\pi}' : \pi' \Rightarrow \tilde{\pi}' \rightarrow \tilde{\pi}'$ ,  $\tilde{\beta}|_{vis(T_C)} = \beta$ ,  $(\pi, \tilde{\pi}', \beta) \in \mathcal{R}$ ,  $(\tilde{\pi}, \tilde{\pi}', \tilde{\beta}) \in \mathcal{R}$ .

4. Как пункт 3, но роли  $N$  и  $N'$  меняются.

Сети  $N$  и  $N'$  ЧУММ сохраняющие историю ветвисто  $\tau$ -бисимуляционно эквивалентны, запись  $N \xleftrightarrow{\tau}_{pomhbr} N'$ , если  $\exists \mathcal{R} : N \xleftrightarrow{\tau}_{pomhbr} N'$ .

### $\tau$ -сохраняющие конфликт эквивалентности

$\tau$ -сохраняющие конфликт эквивалентности получаются из сохраняющих конфликт эквивалентностей абстрагированием от невидимых действий.

Пусть  $\xi = \langle X, \prec, \#, l \rangle$  — ПСС такая, что  $l : X \rightarrow Act_\tau$ . Обозначим  $vis(X) = \{x \in X \mid l(x) \in Act\}$  и  $vis(\xi) = \xi|_{vis(X)}$ .

**Определение 1.3.13** Видимый МСС-след сети с невидимыми переходами  $N$  — МСС  $vis(\xi)$  — класс изоморфизма ПСС  $vis(\xi_O)$  для  $\varpi = (O, \psi) \in \wp(N)$ . Обозначим через  $VisMEStructs(N)$  множество всех видимых МСС-следов сети с невидимыми переходами  $N$ . Две сети с невидимыми переходами  $N$  и  $N'$  сохраняющие конфликт  $\tau$ -эквивалентны на МСС, запись  $N \equiv_{mes}^\tau N'$ , если  $VisMEStructs(N) = VisMEStructs(N')$ . Заметим, что, в силу единственности максимального  $O$ -процесса, это равносильно требованию  $vis(\mathcal{E}(N)) = vis(\mathcal{E}(N'))$ .

### Сравнение базисных $\tau$ -эквивалентностей

Исследуем взаимосвязь базисных  $\tau$ -эквивалентностей.

На рисунке 1.41 показана различающая способность обычных и ветвистых  $\tau$ -бисимуляционных эквивалентностей. И те, и другие эквивалентности требуют, чтобы отношением бисимуляции были связаны начальные процессы сетей  $\pi_N$  и  $\pi_{N'}$ . Кроме того, если отношением бисимуляции связаны текущие процессы  $\pi$  и  $\pi'$ , и один из них был расширен, то другой тоже можно расширить так, чтобы вторая сеть моделировала поведение первой при абстрагировании от невидимых действий. При этом новые, расширенные процессы  $\tilde{\pi}$  и  $\tilde{\pi}'$  должны быть также связаны бисимуляцией.

Ветвистые  $\tau$ -бисимуляционные эквивалентности сильнее обычных, так как отношением ветвистой  $\tau$ -бисимуляции должны быть связаны еще некоторые промежуточные процессы. Расширение на невидимое действие  $\tau$ , представленное на рисунке 1.41(a), моделируется расширением на последовательность невидимых действий. При этом новый процесс  $\tilde{\pi}$  первой сети должен связываться с текущим процессом  $\pi'$  второй. Расширение на видимое действие  $a$ , представленное на рисунке 1.41(b), моделируется расширением на последовательность действий, лишь одно из которых (а именно  $a$ ) видимое. При этом текущий процесс  $\pi$  должен быть связан с процессом  $\pi_1$ , который достигается непосредственно перед расширением на действие  $a$ , а новый процесс  $\tilde{\pi}$  — с процессом  $\pi_2$ , который получается непосредственно после этого расширения. Эти дополнительные связи представлены на рисунке наклонными линиями.

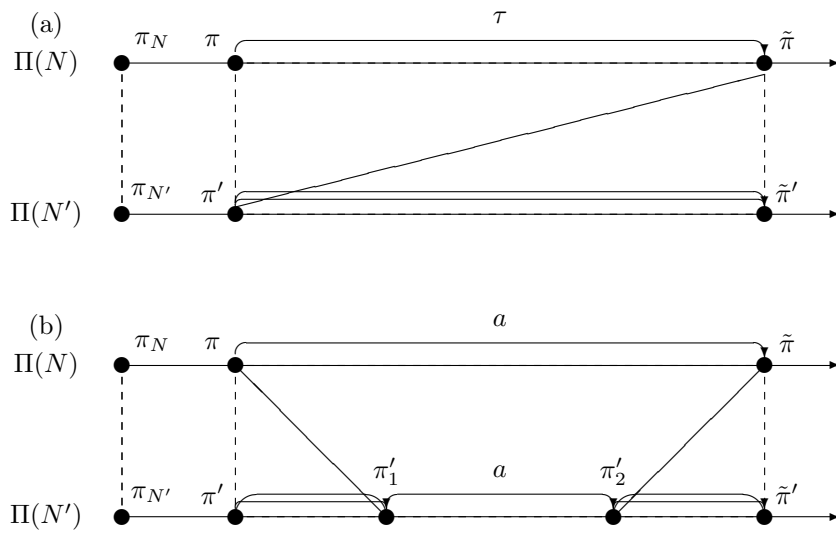


Рис. 1.41: Различающая способность обычных и ветвистых  $\tau$ -бисимуляционных эквивалентностей

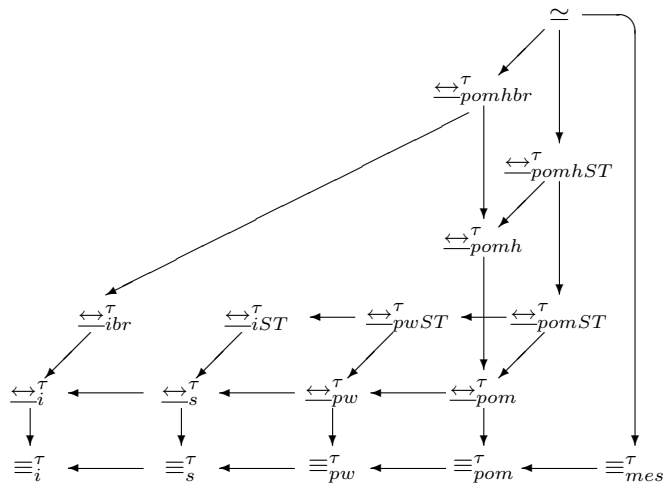


Рис. 1.42: Взаимосвязь базисных  $\tau$ -эквивалентностей

**Теорема 1.3.1** Пусть  $\leftrightarrow, \leftrightarrow^{\tau} \in \{\equiv^{\tau}, \leftrightarrow^{\tau}, \simeq\}$ ,  $\star, \star\star \in \{\_, i, s, pw, pom, iST, pwST, pomST, pomh, pomhST, ibr, pomhbr, mes\}$ . Для сетей с невидимыми переходами  $N$  и  $N'$   $N \leftrightarrow_{\star} N' \Rightarrow N \leftrightarrow_{\star\star} N'$  тогда и только тогда, когда в графе на рисунке 1.42 существует направленный путь от  $\leftrightarrow_{\star}$  к  $\leftrightarrow_{\star\star}$ .

*Доказательство.* ( $\Leftarrow$ ) Проверим истинность импликаций в графе на рисунке 1.42.

- Связи  $\leftrightarrow_s^{\tau} \rightarrow \leftrightarrow_i^{\tau}$ ,  $\leftrightarrow \in \{\equiv^{\tau}, \leftrightarrow^{\tau}\}$ , — следствия того, что изоморфизм ПЧУМ с пустым отношением предшествования является изоморфизмом одноэлементных ПЧУМ.
- Связи  $\leftrightarrow_{pw}^{\tau} \rightarrow \leftrightarrow_s^{\tau}$ ,  $\leftrightarrow \in \{\equiv^{\tau}, \leftrightarrow^{\tau}\}$ , — следствия того, что гомоморфизм ПЧУМ является изоморфизмом ПЧУМ с пустым отношением предшествования.
- Связь  $\leftrightarrow_{pwST}^{\tau} \rightarrow \leftrightarrow_{iST}^{\tau}$  — следствие того, что гомоморфизм ПЧУМ является изоморфизмом одноэлементных ПЧУМ.
- Связи  $\leftrightarrow_{pom}^{\tau} \rightarrow \leftrightarrow_{pw}^{\tau}$ ,  $\leftrightarrow \in \{\equiv^{\tau}, \leftrightarrow^{\tau}\}$ , — следствия того, что изоморфизм ПЧУМ является гомоморфизмом.
- Связь  $\equiv_{mes}^{\tau} \rightarrow \equiv_{pom}^{\tau}$  — следствие того, что множества ЧУММ изоморфных ПСС также изоморфны.
- Связь  $\leftrightarrow_i^{\tau} \rightarrow \equiv_i^{\tau}$  доказывается следующим образом. Пусть  $\mathcal{R} : N \leftrightarrow_i^{\tau} N'$ . Если  $\pi_N \xrightarrow{a_1} \pi_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} \pi_n$ , то существует последовательность  $(\pi_N, \pi_{N'}), \dots, (\pi_n, \pi'_m) \in \mathcal{R}$  такая, что  $\pi_{N'} \xrightarrow{a'_1} \pi'_1 \xrightarrow{a'_2} \dots \xrightarrow{a'_m} \pi'_m$ ,  $vis(a_1 \dots a_n) = vis(a'_1 \dots a'_m)$ , и наоборот, в силу симметричности бисимуляции.
- Связь  $\leftrightarrow_s^{\tau} \rightarrow \equiv_s^{\tau}$  доказывается, как в предыдущем случае, но с использованием  $A_1, \dots, A_n \in \mathcal{M}(Act_{\tau})$  вместо  $a_1, \dots, a_n \in Act_{\tau}$ .
- Связь  $\leftrightarrow_{pw}^{\tau} \rightarrow \equiv_{pw}^{\tau}$  доказывается следующим образом. Пусть  $\mathcal{R} : N \leftrightarrow_{pw}^{\tau} N'$  и  $\pi = (C, \varphi) \in \Pi(N)$ . Так как  $\pi_N \xrightarrow{\pi} \pi$ , то существует пара  $(\pi, \pi') \in \mathcal{R}$  такая, что  $\pi' = (C', \varphi')$  и  $vis(\rho_{C'}) \sqsubseteq vis(\rho_C)$ . Следовательно,  $VisPomsets(N') \sqsubseteq VisPomsets(N)$ . Включение  $VisPomsets(N) \sqsubseteq VisPomsets(N')$  доказывается аналогично, в силу симметричности бисимуляции.
- Связь  $\leftrightarrow_{pom}^{\tau} \rightarrow \equiv_{pom}^{\tau}$  доказывается как в предыдущем случае, но с использованием изоморфизма вместо гомоморфизма ПЧУМ.
- Связь  $\leftrightarrow_{iST}^{\tau} \rightarrow \leftrightarrow_s^{\tau}$  доказывается с использованием того, что шагу  $\pi \xrightarrow{A} \tilde{\pi}$ , где  $A = \{a_1, \dots, a_n\} \in \mathcal{M}(Act)$ , соответствует последовательность ST- $\tau$ -процессов  $(\pi_0, \pi_0), \dots, (\pi_n, \pi_0), \dots, (\pi_n, \pi_n)$  такая, что  $\pi_0 \xrightarrow{a_1} \dots \xrightarrow{a_n} \pi_n$ .
- Связи  $\leftrightarrow_{\star ST}^{\tau} \rightarrow \leftrightarrow_{\star}^{\tau}$ ,  $\star \in \{pw, pom\}$  доказываются построением на основе отношения  $\mathcal{R} : N \leftrightarrow_{\star ST}^{\tau} N'$  нового отношения  $\mathcal{S} : N \leftrightarrow_{\star}^{\tau} N'$ , определяемого следующим образом:  $\mathcal{S} = \{(\pi, \pi') \mid \exists \beta ((\pi, \pi), (\pi', \pi'), \beta) \in \mathcal{R}\}$ .
- Связь  $\leftrightarrow_{pomhST}^{\tau} \rightarrow \leftrightarrow_{pomh}^{\tau}$  доказываются построением на основе отношения  $\mathcal{R} : N \leftrightarrow_{pomhST}^{\tau} N'$  нового отношения  $\mathcal{S} : N \leftrightarrow_{pomh}^{\tau} N'$ , определяемого следующим образом:  $\mathcal{S} = \{(\pi, \pi', \beta) \mid ((\pi, \pi), (\pi', \pi'), \beta) \in \mathcal{R}\}$ .
- Связь  $\leftrightarrow_{pomh}^{\tau} \rightarrow \leftrightarrow_{pom}^{\tau}$  доказываются построением на основе отношения  $\mathcal{R} : N \leftrightarrow_{pomh}^{\tau} N'$  нового отношения  $\mathcal{S} : N \leftrightarrow_{pom}^{\tau} N'$ , определяемого следующим образом:  $\mathcal{S} = \{(\pi, \pi') \mid \exists \beta ((\pi, \pi), (\pi', \pi'), \beta) \in \mathcal{R}\}$ .
- Связь  $\leftrightarrow_{pomhST}^{\tau} \rightarrow \leftrightarrow_{pomST}^{\tau}$  следует из определений.
- Связь  $\leftrightarrow_{ibr}^{\tau} \rightarrow \leftrightarrow_i^{\tau}$  следует из определений.
- Связь  $\leftrightarrow_{pomhbr}^{\tau} \rightarrow \leftrightarrow_{pomh}^{\tau}$  следует из определений.
- Связь  $\leftrightarrow_{pomhbr}^{\tau} \rightarrow \leftrightarrow_{ibr}^{\tau}$  доказываются построением на основе отношения  $\mathcal{R} : N \leftrightarrow_{pomhbr}^{\tau} N'$  нового отношения  $\mathcal{S} : N \leftrightarrow_{ibr}^{\tau} N'$ , определяемого следующим образом:  $\mathcal{S} = \{(\pi, \pi') \mid \exists \beta (\pi, \pi', \beta) \in \mathcal{R}\}$ .
- Связь  $\simeq \rightarrow \leftrightarrow_{pomhbr}^{\tau}$  очевидна.
- Связь  $\simeq \rightarrow \leftrightarrow_{pomhST}^{\tau}$  очевидна.

- Связь  $\simeq \rightarrow \equiv_{mes}^{\tau}$  очевидна.

( $\Rightarrow$ ) Отсутствие дополнительных нетривиальных стрелок в графе на рисунке 1.42 доказывается следующими примерами на сетях с невидимыми переходами.

- На рисунке 1.12(a)  $N \xleftrightarrow{ibr}^{\tau} N'$ , но  $N \not\equiv_s^{\tau} N'$ , так как только в сети  $N'$  действия  $a$  и  $b$  не могут работать параллельно.
- На рисунке 1.12(c)  $N \xleftrightarrow{iST}^{\tau} N'$ , но  $N \not\equiv_{pw}^{\tau} N'$ , так как сети  $N$  соответствует ЧУММ такое, что даже менее последовательного ЧУММ не может быть выполнено в сети  $N'$ .
- На рисунке 1.12(b)  $N \xleftrightarrow{pwST}^{\tau} N'$ , но  $N \not\equiv_{pom}^{\tau} N'$ , так как только в сети  $N'$  действие  $b$  может зависеть от  $a$ .
- На рисунке 1.43(a)  $N \equiv_{mes}^{\tau} N'$ , но  $N \not\equiv_i^{\tau} N'$ , так как только в сети  $N'$  действие  $\tau$  может работать так, что в соответствующем ему начальном состоянии сети  $N$  действие  $a$  не может выполняться.
- На рисунке 1.13(a)  $N \xleftrightarrow{pom}^{\tau} N'$ , но  $N \not\equiv_{iST}^{\tau} N'$ , так как только в сети  $N'$  действие  $a$  может начать работать так, что никакое действие  $b$  уже не сможет стартовать до завершения  $a$ .
- На рисунке 1.13(b)  $N \xleftrightarrow{pomST}^{\tau} N'$ , но  $N \not\equiv_{pomh}^{\tau} N'$ , так как только в сети  $N'$  после действия  $a$  действие  $b$  может работать так, что действие  $c$  должно обязательно зависеть от  $a$ .
- На рисунке 1.43(b)  $N \xleftrightarrow{pomh}^{\tau} N'$ , но  $N \not\equiv_{iST}^{\tau} N'$ , так как только в сети  $N'$  действие  $a$  может стартовать так, что  $b$  никогда не случится.
- На рисунке 1.43(c)  $N \xleftrightarrow{pomhST}^{\tau} N'$ , но  $N \not\equiv_{ibr}^{\tau} N'$ , так как в сети  $N'$  действие  $a$  может произойти так, что оно будет симитировано последовательностью действий  $\tau a$  в  $N$ . Тогда состояние сети  $N$ , достигаемое после  $\tau$ , должно быть связано с начальным состоянием сети  $N$ , но в этом случае выполнение действия  $b$  из начального состояния  $N'$  не может быть смоделировано из соответствующего состояния  $N$ .
- На рисунке 1.43(d)  $N \xleftrightarrow{pomhbr}^{\tau} N'$ , но  $N \not\equiv_{iST}^{\tau} N'$ , так как в сети  $N'$  действие  $c$  может стартовать так, что в течение работы соответствующего действия  $c$  сети  $N$  действие  $a$  может работать так, что  $b$  никогда не случится.
- На рисунке 1.13(c)  $N \xleftrightarrow{pomhST}^{\tau} N'$ , но  $N \not\equiv_{mes}^{\tau} N'$ , так как только сети  $N'$  соответствует МСС с двумя конфликтными действиями  $a$ .
- На рисунке 1.13(c)  $N \xleftrightarrow{pomhbr}^{\tau} N'$ , но  $N \not\equiv_{mes}^{\tau} N'$ .
- На рисунке 1.13(d)  $N \equiv_{mes}^{\tau} N'$ , но  $N \not\approx N'$ , так как никогда не срабатывающие переходы сетей  $N$  и  $N'$  помечены разными действиями ( $a$  и  $b$ ).  $\square$



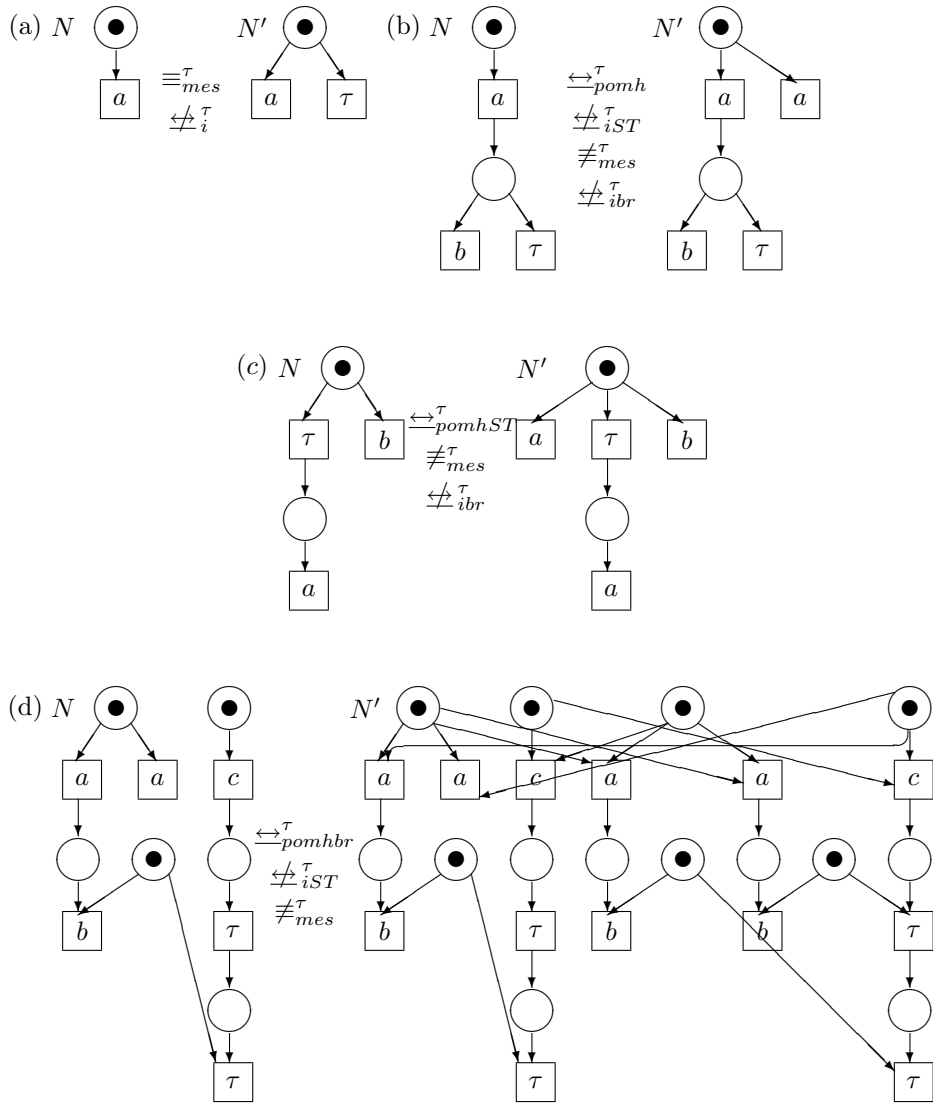


Рис. 1.43: Примеры базисных  $\tau$ -эквивалентностей

### 1.3.2 Обратные-прямые $\tau$ -эквивалентности

В этом разделе мы рассмотрим обратные-прямые  $\tau$ -бисимуляционные эквивалентности.

Следующие понятия обратных-прямых  $\tau$ -бисимуляционных эквивалентностей были определены в литературе.

В [214] на системах переходов с невидимыми действиями введена интерливинговая обратная интерливинговая прямая  $\tau$ -бисимуляционная эквивалентность  $(\leftrightarrow_{ibif}^\tau)$ . Там же показано, что она совпадает с  $\leftrightarrow_{ibr}^\tau$ .

В [237] на структурах событий с  $\tau$ -действиями введена ЧУММ обратная ЧУММ прямая  $(\leftrightarrow_{pombpomf}^\tau)$   $\tau$ -бисимуляционная эквивалентность и доказано ее совпадение с  $\leftrightarrow_{pomhbr}^\tau$ .

Мы дополняем эти  $\tau$ -эквивалентности шестью новыми понятиями: интерливинговая обратная шаговая прямая  $(\leftrightarrow_{ibsf}^\tau)$ , интерливинговая обратная ЧС прямая  $(\leftrightarrow_{ibpwf}^\tau)$ , интерливинговая обратная ЧУММ прямая  $(\leftrightarrow_{ibpomf}^\tau)$ , шаговая обратная шаговая прямая  $(\leftrightarrow_{sbsf}^\tau)$ , шаговая обратная ЧС прямая  $(\leftrightarrow_{sbpwf}^\tau)$  и шаговая обратная ЧУММ прямая  $(\leftrightarrow_{sbpomf}^\tau)$   $\tau$ -бисимуляционные эквивалентности.

Мы также проводим сравнение всех обратных-прямых  $\tau$ -бисимуляционных эквивалентностей с базисными  $\tau$ -эквивалентностями.

#### Обратные-прямые $\tau$ -бисимуляционные эквивалентности

**Определение 1.3.14** Пусть  $N$  и  $N'$  — некоторые сети с невидимыми переходами. Отношение  $\mathcal{R} \subseteq Runs(N) \times Runs(N')$  —  $\star$ -обратная  $\star\star$ -прямая  $\tau$ -бисимуляция между  $N$  и  $N'$ ,  $\star, \star\star \in \{\text{интерливинговая, шаговая, ЧС, ЧУММ}\}$ , запись  $\mathcal{R} : N \leftrightarrow_{\star b \star \star f}^\tau N'$ ,  $\star, \star\star \in \{i, s, pw, pom\}$ , если:

1.  $((\pi_N, \varepsilon), (\pi_{N'}, \varepsilon)) \in \mathcal{R}$ .

2.  $((\pi, \sigma), (\pi', \sigma')) \in \mathcal{R}$

- (обратно)

$$(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\hat{\pi}} (\pi, \sigma),$$

- (a)  $|\text{vis}(T_{\hat{C}})| = 1$ , если  $\star = i$ ;

- (b)  $\text{vis}(\prec_{\hat{C}}) = \emptyset$ , если  $\star = s$ ;

$$\Rightarrow \exists (\tilde{\pi}', \tilde{\sigma}') : (\tilde{\pi}', \tilde{\sigma}') \xrightarrow{\hat{\pi}'} (\pi', \sigma'), ((\tilde{\pi}, \tilde{\sigma}), (\tilde{\pi}', \tilde{\sigma}')) \in \mathcal{R} \text{ и}$$

- (a)  $\text{vis}(\rho_{\hat{C}'}) \sqsubseteq \text{vis}(\rho_{\hat{C}})$ , если  $\star = pw$ ;

- (b)  $\text{vis}(\rho_{\hat{C}}) \simeq \text{vis}(\rho_{\hat{C}'})$ , если  $\star \in \{i, s, pom\}$ ;

- (прямо)

$$(\pi, \sigma) \xrightarrow{\hat{\pi}} (\tilde{\pi}, \tilde{\sigma}),$$

- (a)  $|\text{vis}(T_{\hat{C}})| = 1$ , если  $\star\star = i$ ;

- (b)  $\text{vis}(\prec_{\hat{C}}) = \emptyset$ , если  $\star\star = s$ ;

$$\Rightarrow \exists (\tilde{\pi}', \tilde{\sigma}') : (\pi', \sigma') \xrightarrow{\hat{\pi}'} (\tilde{\pi}', \tilde{\sigma}'), ((\tilde{\pi}, \tilde{\sigma}), (\tilde{\pi}', \tilde{\sigma}')) \in \mathcal{R} \text{ и}$$

- (a)  $\text{vis}(\rho_{\hat{C}'}) \sqsubseteq \text{vis}(\rho_{\hat{C}})$ , если  $\star\star = pw$ ;

- (b)  $\text{vis}(\rho_{\hat{C}}) \simeq \text{vis}(\rho_{\hat{C}'})$ , если  $\star\star \in \{i, s, pom\}$ .

3. Как пункт 2, но роли  $N$  и  $N'$  меняются.

Сети с невидимыми переходами  $N$  и  $N'$   $\star$ -обратно  $\star\star$ -прямо  $\tau$ -бисимуляционно эквивалентны,

$\star, \star\star \in \{\text{интерливинговая, шаговая, ЧС, ЧУММ}\}$ , запись  $N \leftrightarrow_{\star b \star \star f}^\tau N'$ , если  $\exists \mathcal{R} : N \leftrightarrow_{\star b \star \star f}^\tau N'$ ,  $\star, \star\star \in \{i, s, pw, pom\}$ .

#### Сравнение обратных-прямых $\tau$ -бисимуляционных эквивалентностей

Исследуем взаимосвязь обратных-прямых  $\tau$ -бисимуляционных эквивалентностей.

**Предложение 1.3.1** Пусть  $\star \in \{i, s, pw, pom\}$ . Для сетей с невидимыми переходами  $N$  и  $N'$   $N \leftrightarrow_{pw b \star f}^\tau N' \Leftrightarrow N \leftrightarrow_{pom b \star f}^\tau N'$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие того, что изоморфизм ПЧУМ является гомоморфизмом.

( $\Rightarrow$ ) Пусть  $\mathcal{R} : N \xleftrightarrow{\tau}_{pw\star f} N'$ . Докажем, что  $\mathcal{R} : N \xleftrightarrow{\tau}_{pom\star f} N'$ .

1. Очевидно,  $((\pi_N, \varepsilon), (\pi_{N'}, \varepsilon)) \in \mathcal{R}$ .

2. Пусть  $((\pi, \sigma), (\pi', \sigma')) \in \mathcal{R}$ .

- (обратно)

Пусть  $(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\hat{\pi}} (\pi, \sigma)$ . Тогда  $\exists(\tilde{\pi}', \tilde{\sigma}') : (\tilde{\pi}', \tilde{\sigma}') \xrightarrow{\hat{\pi}' } (\pi', \sigma')$ ,  $((\tilde{\pi}, \tilde{\sigma}), (\tilde{\pi}', \tilde{\sigma}')) \in \mathcal{R}$  и  $\rho_{\hat{C}'} \sqsubseteq \rho_{\hat{C}}$ .

В силу симметричности бисимуляции, обратное расширение  $(\tilde{\pi}', \tilde{\sigma}') \xrightarrow{\hat{\pi}' } (\pi', \sigma')$  должно имитироваться некоторым расширением  $(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\hat{\pi}} (\pi, \sigma)$  таким, что  $vis(\rho_{\hat{C}}) \sqsubseteq vis(\rho_{\hat{C}'})$ . Из-за детерминированности обратных расширений,  $vis(T_{\hat{C}}) = vis(T_{\hat{C}'})$ . Тогда  $vis(\rho_{\hat{C}}) = vis(\rho_{\hat{C}'})$ . Следовательно,  $vis(\rho_{\hat{C}}) \simeq vis(\rho_{\hat{C}'})$ .

- (прямо)

Очевидно.

3. Как пункт 2, но роли  $N$  и  $N'$  меняются. □

**Предложение 1.3.2** Пусть  $\star \in \{i, s, pw, pom\}$ . Для сетей с невидимыми переходами  $N$  и  $N'$   $N \xleftrightarrow{\tau}_{\star bif} N' \Leftrightarrow N \xleftrightarrow{\tau}_{\star b\star f} N'$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие того, что изоморфизм С-сетей, изоморфизм и гомоморфизм ПЧУМ С-сетей, изоморфизм ПЧУМ С-сетей с пустым отношением предшествования являются сохраняющими пометку биекциями.

( $\Rightarrow$ ) Пусть  $\mathcal{R} : N \xleftrightarrow{\tau}_{\star bif} N'$ . Докажем, что  $\mathcal{R} : N \xleftrightarrow{\tau}_{\star b\star f} N'$ .

1. Очевидно,  $((\pi_N, \varepsilon), (\pi_{N'}, \varepsilon)) \in \mathcal{R}$ .

2. Пусть  $((\pi, \sigma), (\pi', \sigma')) \in \mathcal{R}$ .

- (обратно)

Очевидно.

- (прямо)

Пусть  $(\pi, \sigma) \xrightarrow{\hat{\pi}} (\tilde{\pi}, \tilde{\sigma})$ . Расширению на  $\hat{\pi}$  соответствует расширение на некоторую последовательность переходов. Тогда  $\exists(\tilde{\pi}', \tilde{\sigma}') : (\pi', \sigma') \xrightarrow{\hat{\pi}' } (\tilde{\pi}', \tilde{\sigma}')$ ,  $((\tilde{\pi}, \tilde{\sigma}), (\tilde{\pi}', \tilde{\sigma}')) \in \mathcal{R}$ , где расширение на  $\hat{\pi}'$  соответствует расширению на последовательность переходов, моделирующему соответствующее расширение в сети  $N$ .

В силу симметричности бисимуляции, расширение  $(\pi, \sigma) \xrightarrow{\hat{\pi}} (\tilde{\pi}, \tilde{\sigma})$  должно имитироваться некоторым расширением  $(\pi', \sigma') \xrightarrow{\hat{\pi}' } (\tilde{\pi}', \tilde{\sigma}')$ , таким, что

- (a)  $vis(\rho_{\hat{C}'}) \sqsubseteq vis(\rho_{\hat{C}})$ , если  $\star = pw$ ;
- (b)  $vis(\rho_{\hat{C}}) \simeq vis(\rho_{\hat{C}'})$ , если  $\star \in \{i, s, pom\}$ .

Из-за детерминированности обратных расширений,  $vis(T_{\hat{C}'}) = vis(T_{\hat{C}})$ . Тогда  $vis(\rho_{\hat{C}}) = vis(\rho_{\hat{C}'})$ .

3. Как пункт 2, но роли  $N$  и  $N'$  меняются. □

На рисунке 1.44 пунктирной линией обведены совпадающие обратные-прямые  $\tau$ -бисимуляционные эквивалентности.

Следовательно, взаимосвязь обратных-прямых  $\tau$ -бисимуляционных эквивалентностей можно представить рисунком 1.45.

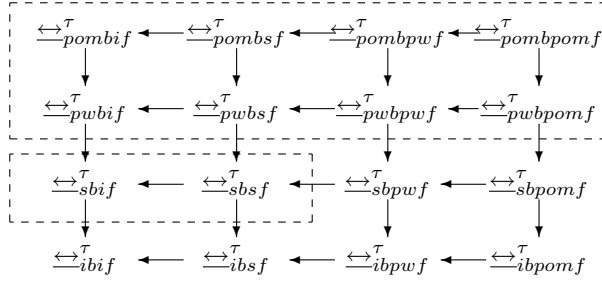


Рис. 1.44: Совпадение обратных-прямых  $\tau$ -бисимуляционных эквивалентностей

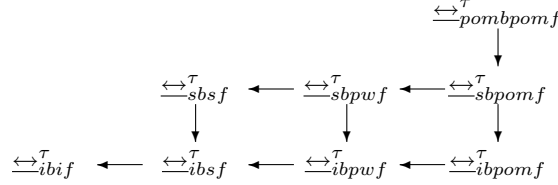


Рис. 1.45: Взаимосвязь обратных-прямых  $\tau$ -бисимуляционных эквивалентностей

### Сравнение обратных-прямых $\tau$ -бисимуляционных эквивалентностей с базисными $\tau$ -эквивалентностями

Исследуем взаимосвязь обратных-прямых  $\tau$ -бисимуляционных эквивалентностей с базисными  $\tau$ -эквивалентностями.

Для некоторой сети с невидимыми переходами  $N$  и  $(\pi, \sigma), (\tilde{\pi}, \tilde{\sigma}) \in Runs(N)$  пишем  $(\pi, \sigma) \Rightarrow (\tilde{\pi}, \tilde{\sigma})$ , когда  $(\pi, \sigma) \rightarrow (\tilde{\pi}, \tilde{\sigma})$  и  $\pi \Rightarrow \tilde{\pi}$ .

Пусть для некоторых сетей с невидимыми переходами  $N$  и  $N'$   $(\pi, \sigma) \in Runs(N)$ ,  $(\pi', \sigma') \in Runs(N')$ .

Пишем  $(\pi, \sigma) \stackrel{\tau}{\Leftrightarrow}_{ibif} (\pi', \sigma')$ , если  $\exists \mathcal{R} : N \stackrel{\tau}{\Leftrightarrow}_{ibif} N'$  такое, что  $((\pi, \sigma), (\pi', \sigma')) \in \mathcal{R}$  и аналогично для  $\stackrel{\tau}{\Leftrightarrow}_{pombpomf}$ .

Пишем  $\pi \stackrel{\tau}{\Leftrightarrow}_{ibr} \pi'$ , если  $\exists \mathcal{R} : N \stackrel{\tau}{\Leftrightarrow}_{ibr} N'$  такое, что  $(\pi, \pi') \in \mathcal{R}$ .

Пишем  $\pi \stackrel{\tau}{\Leftrightarrow}_{pomhbr} \pi'$ , если  $\exists \mathcal{R} : N \stackrel{\tau}{\Leftrightarrow}_{pomhbr} N' \exists \beta$  такое, что  $(\pi, \pi', \beta) \in \mathcal{R}$ .

**Лемма 1.3.1 (X-лемма 1)** Пусть для сетей с невидимыми переходами  $N$  и  $N'$   $N \stackrel{\tau}{\Leftrightarrow}_{ibif} N'$  и  $(\pi, \sigma), (\tilde{\pi}, \tilde{\sigma}) \in Runs(N)$ ,  $(\pi', \sigma'), (\tilde{\pi}', \tilde{\sigma}') \in Runs(N')$  такие, что  $(\pi, \sigma) \Rightarrow (\tilde{\pi}, \tilde{\sigma})$ ,  $(\pi', \sigma') \Rightarrow (\tilde{\pi}', \tilde{\sigma}')$ . Тогда  $(\pi, \sigma) \stackrel{\tau}{\Leftrightarrow}_{ibif} (\tilde{\pi}', \tilde{\sigma}')$  и  $(\tilde{\pi}, \tilde{\sigma}) \stackrel{\tau}{\Leftrightarrow}_{ibif} (\pi', \sigma')$  влечет  $(\pi, \sigma) \stackrel{\tau}{\Leftrightarrow}_{ibif} (\pi', \sigma')$  и  $(\tilde{\pi}, \tilde{\sigma}) \stackrel{\tau}{\Leftrightarrow}_{ibif} (\tilde{\pi}', \tilde{\sigma}')$ .

*Доказательство.* Как доказательство следующей леммы, но используя процессные расширения на одно действие.  $\square$

**Лемма 1.3.2 (X-лемма 2)** Пусть для сетей с невидимыми переходами  $N$  и  $N'$   $N \stackrel{\tau}{\Leftrightarrow}_{pombpomf} N'$  и  $(\pi, \sigma), (\tilde{\pi}, \tilde{\sigma}) \in Runs(N)$ ,  $(\pi', \sigma'), (\tilde{\pi}', \tilde{\sigma}') \in Runs(N')$  такие, что  $(\pi, \sigma) \Rightarrow (\tilde{\pi}, \tilde{\sigma})$ ,  $(\pi', \sigma') \Rightarrow (\tilde{\pi}', \tilde{\sigma}')$ . Тогда  $(\pi, \sigma) \stackrel{\tau}{\Leftrightarrow}_{pombpomf} (\tilde{\pi}', \tilde{\sigma}')$  и  $(\tilde{\pi}, \tilde{\sigma}) \stackrel{\tau}{\Leftrightarrow}_{pombpomf} (\pi', \sigma')$  влечет  $(\pi, \sigma) \stackrel{\tau}{\Leftrightarrow}_{pombpomf} (\pi', \sigma')$  и  $(\tilde{\pi}, \tilde{\sigma}) \stackrel{\tau}{\Leftrightarrow}_{pombpomf} (\tilde{\pi}', \tilde{\sigma}')$ .

*Доказательство.* Достаточно доказать  $(\tilde{\pi}, \tilde{\sigma}) \stackrel{\tau}{\Leftrightarrow}_{pombpomf} (\tilde{\pi}', \tilde{\sigma}')$ , так как  $(\pi, \sigma) \stackrel{\tau}{\Leftrightarrow}_{pombpomf} (\pi', \sigma')$  доказывается аналогично.

Пусть  $(\pi, \sigma) \Rightarrow (\tilde{\pi}, \tilde{\sigma})$ ,  $(\pi', \sigma') \Rightarrow (\tilde{\pi}', \tilde{\sigma}')$  и  $(\pi, \sigma) \stackrel{\tau}{\Leftrightarrow}_{pombpomf} (\tilde{\pi}', \tilde{\sigma}')$ ,  $(\tilde{\pi}, \tilde{\sigma}) \stackrel{\tau}{\Leftrightarrow}_{pombpomf} (\pi', \sigma')$ . Достаточно проверить только моделирование сети  $N$  сетью  $N'$  в обратном и прямом направлениях, так как моделирование сети  $N'$  сетью  $N$  доказывается симметрично.

- (обратно)

Пусть  $(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\hat{\pi}} (\tilde{\pi}, \tilde{\sigma})$ ,  $\hat{\pi} = (\hat{C}, \hat{\varphi})$ . Тогда, так как  $(\tilde{\pi}, \tilde{\sigma}) \stackrel{\tau}{\Leftrightarrow}_{pombpomf} (\pi', \sigma')$ ,  $\exists \tilde{\pi}' = (\tilde{C}', \tilde{\varphi}')$ ,  $(\tilde{\pi}', \tilde{\sigma}')$  такие, что  $(\tilde{\pi}', \tilde{\sigma}') \xrightarrow{\tilde{\pi}'} (\pi', \sigma')$ ,  $(\tilde{\pi}, \tilde{\sigma}) \stackrel{\tau}{\Leftrightarrow}_{pombpomf} (\tilde{\pi}', \tilde{\sigma}')$  и  $vis(\rho_{\hat{C}}) \simeq vis(\rho_{\tilde{C}'})$ .

Заметим, что если  $(\tilde{\pi}', \tilde{\sigma}') \xrightarrow{\hat{\pi}'} (\tilde{\pi}', \tilde{\sigma}')$ ,  $\hat{\pi}' = (\hat{C}', \hat{\varphi}')$ , то  $vis(\rho_{\hat{C}'}) = vis(\rho_{\tilde{C}'})$ . Следовательно,  $vis(\rho_{\hat{C}}) \simeq vis(\rho_{\tilde{C}'})$ .

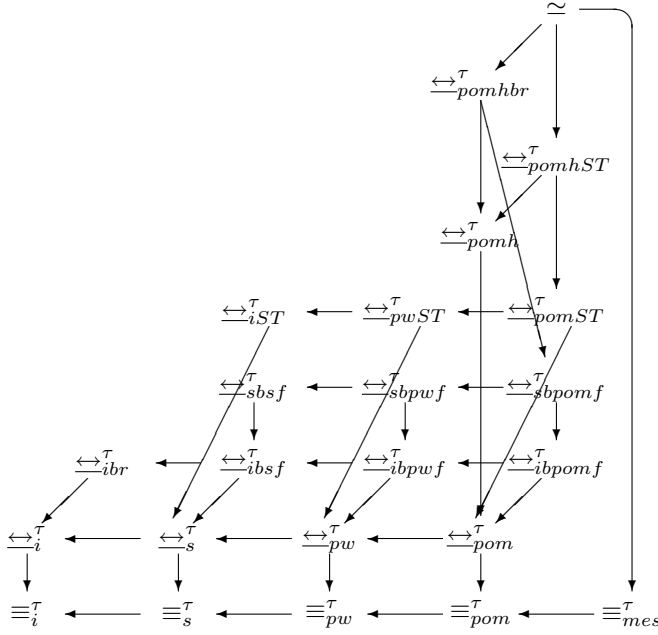


Рис. 1.46: Взаимосвязь обратных-прямых  $\tau$ -бисимуляционных эквивалентностей с базисными  $\tau$ -эквивалентностями

- (прямо)

Пусть  $(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\hat{\pi}} (\bar{\pi}, \bar{\sigma})$ ,  $\hat{\pi} = (\hat{C}, \hat{\varphi})$ . Заметим, что если  $(\pi, \sigma) \xrightarrow{\tilde{\pi}} (\bar{\pi}, \bar{\sigma})$ ,  $\tilde{\pi} = (\check{C}, \check{\varphi})$ , то  $vis(\rho_{\hat{C}}) = vis(\rho_{\check{C}})$ .

Так как  $(\pi, \sigma) \xleftrightarrow{\tau}_{pomf} (\tilde{\pi}', \tilde{\sigma}')$ ,  $\exists \hat{\pi}' = (\hat{C}', \hat{\varphi}')$ ,  $(\tilde{\pi}', \tilde{\sigma}')$  такие, что  $(\tilde{\pi}', \tilde{\sigma}') \xrightarrow{\hat{\pi}'} (\bar{\pi}', \bar{\sigma}')$ ,  $(\bar{\pi}, \bar{\sigma}) \xleftrightarrow{\tau}_{pomf} (\bar{\pi}', \bar{\sigma}')$  и  $vis(\rho_{\check{C}}) \simeq vis(\rho_{\hat{C}'})$ . Следовательно,  $vis(\rho_{\hat{C}}) \simeq vis(\rho_{\hat{C}'})$ .  $\square$

Следующие два предложения дают характеристику ветвистых  $\tau$ -бисимуляционных эквивалентностей посредством обратных-прямых  $\tau$ -эквивалентностей.

**Предложение 1.3.3** Для сетей с невидимыми переходами  $N$  и  $N'$   $N \xleftrightarrow{\tau}_{ibif} N' \Leftrightarrow N \xleftrightarrow{\tau}_{ibr} N'$ .

*Доказательство.* Как доказательство следующего предложения 1.3.4, но используя процессные расширения на одно действие.  $\square$

**Предложение 1.3.4** Для сетей с невидимыми переходами  $N$  и  $N'$   $N \xleftrightarrow{\tau}_{pomf} N' \Leftrightarrow N \xleftrightarrow{\tau}_{pomhbr} N'$ .

*Доказательство.* См. приложение А.  $\square$

**Теорема 1.3.2** Пусть  $\leftrightarrow, \Leftrightarrow \in \{\equiv^{\tau}, \xleftrightarrow{\tau}, \simeq\}$  и  $\star, \star\star \in \{\_, i, s, pw, pom, iST, pwST, pomST, pomh, pomhST, ibr, pomhbr, mes, ibsf, ibpwf, ibpmf, sbsf, sbpwf, sbpmf\}$ . Для сетей с невидимыми переходами  $N$  и  $N'$   $N \leftrightarrow_{\star} N' \Rightarrow N \Leftrightarrow_{\star\star} N'$  тогда и только тогда, когда в графе на рисунке 1.46 существует направленный путь от  $\leftrightarrow_{\star}$  к  $\Leftrightarrow_{\star\star}$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие теоремы 1.3.1 и следующих замечаний.

- Связь  $\xleftrightarrow{\tau}_{ibsf} \rightarrow \xleftrightarrow{\tau}_{ibr}$  существует, так как по предложению 1.3.3  $\xleftrightarrow{\tau}_{ibr} = \xleftrightarrow{\tau}_{ibif}$  и изоморфизм ПЧУМ с пустым отношением предшествования является изоморфизмом одноэлементных ПЧУМ.
- Связи  $\xleftrightarrow{\tau}_{\star bpwf} \rightarrow \xleftrightarrow{\tau}_{\star bsf}$ ,  $\star \in \{i, s\}$ , — следствия того, что гомоморфизм ПЧУМ является изоморфизмом ПЧУМ с пустым отношением предшествования.
- Связи  $\xleftrightarrow{\tau}_{\star bpmf} \rightarrow \xleftrightarrow{\tau}_{\star bpwf}$ ,  $\star \in \{i, s\}$ , — следствия того, что изоморфизм ПЧУМ является гомоморфизмом.

- Связи  $\xleftrightarrow{\tau}_{ib^*f} \rightarrow \xleftrightarrow{\tau}_*$ ,  $\star \in \{s, pw, pom\}$  доказываются построением на основе отношения  $\mathcal{R} : N \xleftrightarrow{\tau}_{sb^*f} N'$  нового отношения  $\mathcal{S} : N \xleftrightarrow{\tau}_* N'$ , определяемого следующим образом:  
 $\mathcal{S} = \{(\pi, \pi') \mid \exists \sigma, \sigma' ((\pi, \sigma), (\pi', \sigma')) \in \mathcal{R}\}$ .
- Связи  $\xleftrightarrow{\tau}_{sb^*f} \rightarrow \xleftrightarrow{\tau}_{ib^*f}$ ,  $\star \in \{s, pw, pom\}$ , — следствия того, что изоморфизм ПЧУМ с пустым отношением предшествования является изоморфизмом одноэлементных ПЧУМ.
- Связь  $\xleftrightarrow{\tau}_{pomhbr} \rightarrow \xleftrightarrow{\tau}_{sbpomf}$  существует, так как по предложению 1.3.4  $\xleftrightarrow{\tau}_{pomhbr} = \xleftrightarrow{\tau}_{pombpomf}$  и гомоморфизм является изоморфизмом ПЧУМ с пустым отношением предшествования.

( $\Rightarrow$ )

Отсутствие дополнительных нетривиальных стрелок в графе на рисунке 1.46 доказывается следующими примерами на сетях с невидимыми переходами.

- На рисунке 1.12(с)  $N \xleftrightarrow{\tau}_{sbsf} N'$ , но  $N \not\xleftrightarrow{\tau}_{pw} N'$ .
- На рисунке 1.17(а)  $N \xleftrightarrow{\tau}_{sbprwf} N'$ , но  $N \not\xleftrightarrow{\tau}_{pom} N'$ .
- На рисунке 1.13(а)  $N \xleftrightarrow{\tau}_{ibpomf} N'$ , но  $N \not\xleftrightarrow{\tau}_{sbsf} N'$ , так как только в сети  $N'$  действие  $a$  можно выполнить так, что действие  $b$  обязательно зависит от  $a$ .  $\square$

### Логическая характеристика

**Логика BFL** Обратная-прямая логика (BFL) была введена в [214] на модели систем переходов для логического описания интерливинговой обратной интерливинговой прямой бисимуляционной эквивалентности.

**Определение 1.3.15** Пусть символ  $\top$  обозначает тождественно истинную формулу и  $a \in Act$ . Формула BFL определяется следующим образом:

$$\Phi ::= \top \mid \neg\Phi \mid \Phi \wedge \Psi \mid \langle \leftarrow a \rangle \Phi \mid \langle a \rangle \Phi$$

Определим  $[a]\Phi = \neg\langle a \rangle\neg\Phi$  и  $\langle \leftarrow a \rangle\Phi = \neg\langle \leftarrow a \rangle\neg\Phi$ .

Обозначим через **BFL** множество всех формул логики BFL.

**Определение 1.3.16** Пусть  $N$  — некоторая сеть с невидимыми переходами и  $(\pi, \sigma) \in Runs(N)$ . Отношение выполнимости  $\models_N \subseteq Runs(N) \times \mathbf{BFL}$  определяется следующим образом:

1.  $(\pi, \sigma) \models_N \top$  — всегда;
2.  $(\pi, \sigma) \models_N \neg\Phi$ , если  $(\pi, \sigma) \not\models_N \Phi$ ;
3.  $(\pi, \sigma) \models_N \Phi \wedge \Psi$ , если  $(\pi, \sigma) \models_N \Phi$  и  $(\pi, \sigma) \models_N \Psi$ ;
4.  $(\pi, \sigma) \models_N \langle \leftarrow a \rangle \Phi$ , если  $\exists(\tilde{\pi}, \tilde{\sigma}) \in Runs(N)$   $(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\hat{\pi}} (\pi, \sigma)$ , где  $\hat{\pi} = (\hat{C}, \hat{\varphi})$ ,  $vis(l_{\hat{C}}(T_{\hat{C}})) = a$  и  $(\tilde{\pi}, \tilde{\sigma}) \models_N \Phi$ ;
5.  $(\pi, \sigma) \models_N \langle a \rangle \Phi$ , если  $\exists(\tilde{\pi}, \tilde{\sigma}) \in Runs(N)$   $(\pi, \sigma) \xrightarrow{\hat{\pi}} (\tilde{\pi}, \tilde{\sigma})$ , где  $\hat{\pi} = (\hat{C}, \hat{\varphi})$ ,  $vis(l_{\hat{C}}(T_{\hat{C}})) = a$  и  $(\tilde{\pi}, \tilde{\sigma}) \models_N \Phi$ .

**Определение 1.3.17** Пишем  $N \models_N \Phi$ , если  $(\pi_N, \varepsilon) \models_N \Phi$ . Сети с невидимыми переходами  $N$  и  $N'$  логически эквивалентны в BFL, запись  $N =_{BFL} N'$ , если  $\forall \Phi \in \mathbf{BFL}$   $N \models_N \Phi \Leftrightarrow N' \models_{N'} \Phi$ .

Пусть  $N$  — некоторая сеть с невидимыми переходами и  $\pi \in \Pi(N)$ ,  $a \in Act$ . Множество видимых расширений процесса  $\pi$  на действие  $a$  (образов) определяется следующим образом:  $VisImage(\pi, a) = \{\tilde{\pi} \mid \pi \xrightarrow{\hat{\pi}} \tilde{\pi}, \hat{\pi} = (\hat{C}, \hat{\varphi}), vis(l_{\hat{C}}(T_{\hat{C}})) = a\}$ . Сеть с невидимыми переходами  $N$  — конечно-образная, если  $\forall \pi \in \Pi(N) \forall a \in Act \ |VisImage(\pi, a)| < \infty$ .

**Теорема 1.3.3** [214] Для конечно-образных сетей с невидимыми переходами  $N$  и  $N'$   $N \xleftrightarrow{\tau}_{ib^*f} N' \Leftrightarrow N \xleftrightarrow{\tau}_{ibr} N' \Leftrightarrow N =_{BFL} N'$ .

**Пример 1.3.3** На рисунке 1.43(с)  $N \xleftrightarrow{\tau}_{pomhST} N'$ , но  $N \neq_{BFL} N'$ , так как для  $\Phi = \langle a \rangle \langle \leftarrow a \rangle \langle b \rangle \top$   $N \models_N \Phi$ , но  $N' \not\models_{N'} \Phi$  из-за того, что в сети  $N'$  действие  $a$  может произойти так, что оно будет сымитировано последовательностью действий  $\tau a$  в  $N$ . Тогда состоянием сети  $N$ , достигаемое после  $\tau$ , должно быть связано с начальным состоянием сети  $N$ , но в этом случае выполнение действия  $b$  из начального состояния  $N'$  не может быть смоделировано из соответствующего состояния  $N$ .

**Логика SPBFL** ЧУММ обратная-прямая логика с невидимыми действиями (SPBFL) была введена в [237] на модели структур событий для логического описания ЧУММ обратной ЧУММ прямой  $\tau$ -бисимуляционной эквивалентности.

**Определение 1.3.18** Пусть символ  $\top$  обозначает тождественно истинную формулу и  $a \in Act$ , а  $\rho$  — ЧУММ с пометкой в  $Act$ . Формула SPBFL определяется следующим образом:

$$\Phi ::= \top \mid \neg\Phi \mid \Phi \wedge \Psi \mid \langle \leftarrow \rho \rangle \Phi \mid \langle a \rangle \Phi$$

Определим  $[a]\Phi = \neg\langle a \rangle\neg\Phi$  и  $[\leftarrow \rho]\Phi = \neg\langle \leftarrow \rho \rangle\neg\Phi$ .

Обозначим через **SPBFL** множество всех формул логики SPBFL.

Заметим, что в формуле  $\langle a \rangle \Phi$ , соответствующей случаю прямого расширения, стоит действие  $a$ , а не ЧУММ  $\rho$ , так как  $\xrightarrow{\rho} \text{prombf} = \xrightarrow{\rho} \text{prombrmf}$ . Поэтому достаточно рассматривать прямые расширения только на одно действие.

**Определение 1.3.19** Пусть  $N$  — некоторая сеть и  $(\pi, \sigma) \in Runs(N)$ . Отношение выполнимости  $\models_N \subseteq Runs(N) \times \mathbf{SPBFL}$  определяется следующим образом:

1.  $(\pi, \sigma) \models_N \top$  — всегда;
2.  $(\pi, \sigma) \models_N \neg\Phi$ , если  $(\pi, \sigma) \not\models_N \Phi$ ;
3.  $(\pi, \sigma) \models_N \Phi \wedge \Psi$ , если  $(\pi, \sigma) \models_N \Phi$  и  $(\pi, \sigma) \models_N \Psi$ ;
4.  $(\pi, \sigma) \models_N \langle \leftarrow \rho \rangle \Phi$ , если  $\exists (\tilde{\pi}, \tilde{\sigma}) \in Runs(N)$   $(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\hat{\pi}} (\pi, \sigma)$ , где  $\hat{\pi} = (\hat{C}, \hat{\varphi})$ ,  $vis(\rho_{\hat{C}}) \in \rho$  и  $(\tilde{\pi}, \tilde{\sigma}) \models_N \Phi$ ;
5.  $(\pi, \sigma) \models_N \langle a \rangle \Phi$ , если  $\exists (\tilde{\pi}, \tilde{\sigma}) \in Runs(N)$   $(\pi, \sigma) \xrightarrow{\hat{\pi}} (\tilde{\pi}, \tilde{\sigma})$ , где  $\hat{\pi} = (\hat{C}, \hat{\varphi})$ ,  $vis(l_{\hat{C}}(T_{\hat{C}})) = a$  и  $(\tilde{\pi}, \tilde{\sigma}) \models_N \Phi$ .

**Определение 1.3.20** Пишем  $N \models_N \Phi$ , если  $(\pi_N, \varepsilon) \models_N \Phi$ . Сети с невидимыми переходами  $N$  и  $N'$  логически эквивалентны в SPBFL, запись  $N =_{SPBFL} N'$ , если  $\forall \Phi \in \mathbf{SPBFL}$   $N \models_N \Phi \Leftrightarrow N' \models_{N'} \Phi$ .

**Теорема 1.3.4** [237] Для конечно-образных сетей с невидимыми переходами  $N$  и  $N'$   $N \xrightarrow{\tau} \text{promhbr} N' \Leftrightarrow N \xrightarrow{\tau} \text{prombrmf} N' \Leftrightarrow N =_{SPBFL} N'$ .

**Пример 1.3.4** На рисунке 1.13(b)  $N =_{BFL} N'$ , но  $N \neq_{SPBFL} N'$ , так как для  $\Phi = [a][b]\langle c \rangle \langle \leftarrow (a;b) \parallel c \rangle \top$  ( $\langle (a;b) \parallel c \rangle$  обозначает ЧУММ, в котором действие  $b$  зависит от  $a$ , и  $a$ ,  $b$  независимы с действием  $c$ ),  $N \models_N \Phi$ , но  $N' \not\models_{N'} \Phi$  из-за того, что только в сети  $N'$  после действия  $a$  действие  $b$  может работать так, что действие  $c$  должно обязательно зависеть от  $a$ .

### 1.3.3 Сравнение эквивалентностей с $\tau$ -эквивалентностями

Исследуем взаимосвязь эквивалентностей, не абстрагирующихся от невидимых действий со всеми рассмотренными  $\tau$ -эквивалентностями.

**Предложение 1.3.5** Пусть  $\leftrightarrow \in \{\equiv, \xleftrightarrow{\tau}\}$ ,  $\star \in \{i, s, pw, pot, iST, pwST, potST, mes, sbsf, sbpwf, sbpomf\}$ ,  $\star\star \in \{s, pw, pot\}$ . Для сетей с невидимыми переходами  $N$  и  $N'$ :

1.  $N \leftrightarrow_{\star} N' \Rightarrow N \xleftrightarrow{\tau}_{\star} N'$ ;
2.  $N \xleftrightarrow{\tau}_{pot} N' \Rightarrow N \xleftrightarrow{\tau}_{pot} ST N'$ ;
3.  $N \xleftrightarrow{\tau}_i N' \Rightarrow N \xleftrightarrow{\tau}_{ibr} N'$ ;
4.  $N \xleftrightarrow{\tau}_{pot} N' \Rightarrow N \xleftrightarrow{\tau}_{pot} hbr N'$ ;
5.  $N \xleftrightarrow{\tau}_{\star\star} N' \Rightarrow N \xleftrightarrow{\tau}_{ib\star\star} f N'$ .

и все импликации — строгие.

*Доказательство.*

1. По определениям.

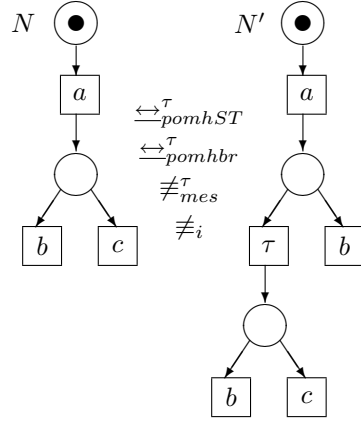


Рис. 1.47: Пример взаимосвязей эквивалентностей и  $\tau$ -эквивалентностей

2. Доказывается построением на основе отношения  $\mathcal{R} : N \stackrel{\tau}{\leftrightarrow}_{pomh} N'$  нового отношения  $\mathcal{S} : N \stackrel{\tau}{\leftrightarrow}_{pomhST} N'$ , определяемого следующим образом:  $\mathcal{S} = \{((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \mid (\pi_E, \pi'_E, \beta) \in \mathcal{R}, (\pi_E, \pi_P) \in ST^\tau - \Pi(N), (\pi'_E, \pi'_P) \in ST^\tau - \Pi(N'), \beta(T_{C_P}) = T_{C'_P}\}$ .
3. По определениям.
4. По определениям.
5. Доказывается построением на основе отношения  $\mathcal{R} : N \stackrel{\tau}{\leftrightarrow}_{**} N'$  нового отношения  $\mathcal{S} : N \stackrel{\tau}{\leftrightarrow}_{ib**f} N'$ , определяемого следующим образом:  $\mathcal{S} = \{((\pi, \sigma), (\pi', \sigma')) \mid (\pi, \sigma) \in Runs(N), (\pi', \sigma') \in Runs(N'), |\sigma| = |\sigma'|, l_C(\sigma) = l_{C'}(\sigma'), \forall i (0 \leq i \leq |\sigma|) (\pi(i), \pi'(i)) \in \mathcal{R}\}$ .

Строгость импликаций доказывается следующими примерами.

1. На рисунке 1.43(с)  $N \stackrel{\tau}{\leftrightarrow}_{pomhST} N'$ , но  $N \not\equiv_i N'$ , так как только в сети  $N'$  действие  $a$  может выполняться из начального состояния.
2. На рисунке 1.43(а)  $N \equiv_{mes}^\tau N'$ , но  $N \not\equiv_i N'$ , так как только в сети  $N'$  действие  $\tau$  может выполняться из начального состояния.
3. На рисунке 1.47  $N \stackrel{\tau}{\leftrightarrow}_{pombr} N'$ , но  $N \not\equiv_i N'$ , так как только в сети  $N'$  последовательность действий  $a\tau$  может выполняться из начального состояния.  $\square$

### 1.3.4 Сохранение $\tau$ -эквивалентностей при детализациях

Проверим рассмотренные  $\tau$ -эквивалентности на сохранение при SM-детализациях.

В литературе сохранение при SM-детализациях было доказано только для  $\stackrel{\tau}{\leftrightarrow}_i$  и  $\stackrel{\tau}{\leftrightarrow}_{pomhST}$  [121, 122]. Данное свойство очевидно для  $\simeq$ . Сохранение при детализациях  $\stackrel{\tau}{\leftrightarrow}_{pwST}$  и  $\stackrel{\tau}{\leftrightarrow}_{pomST}$  установлено в [276, 280], но это было сделано в рамках структур событий, и использовался другой оператор детализации.

Новыми результатами, представленными в данном разделе, является доказательство сохранения при SM-детализациях  $\equiv_\star^\tau$ ,  $\stackrel{\tau}{\leftrightarrow}_{\star ST}$ ,  $\star \in \{pw, pom\}$ , а также  $\equiv_{mes}^\tau$ . Кроме того, с помощью контрпримеров мы демонстрируем, что никакие другие  $\tau$ -эквивалентности из тех, что были рассмотрены, не обладают данным свойством.

Следующее предложение демонстрирует, какие из  $\tau$ -эквивалентностей не сохраняются при SM-детализациях.

**Предложение 1.3.6** Пусть  $\star \in \{i, s\}$ ,  $\star\star \in \{i, s, pw, pom, pomh, ibr, pomhbr, ibsf, ibpwf, ibpomf, sbsf, sbpwf, sbpomf\}$ . Тогда  $\tau$ -эквивалентности  $\equiv_\star^\tau$ ,  $\stackrel{\tau}{\leftrightarrow}_{\star\star}$  не сохраняются при SM-детализациях.

*Доказательство.*



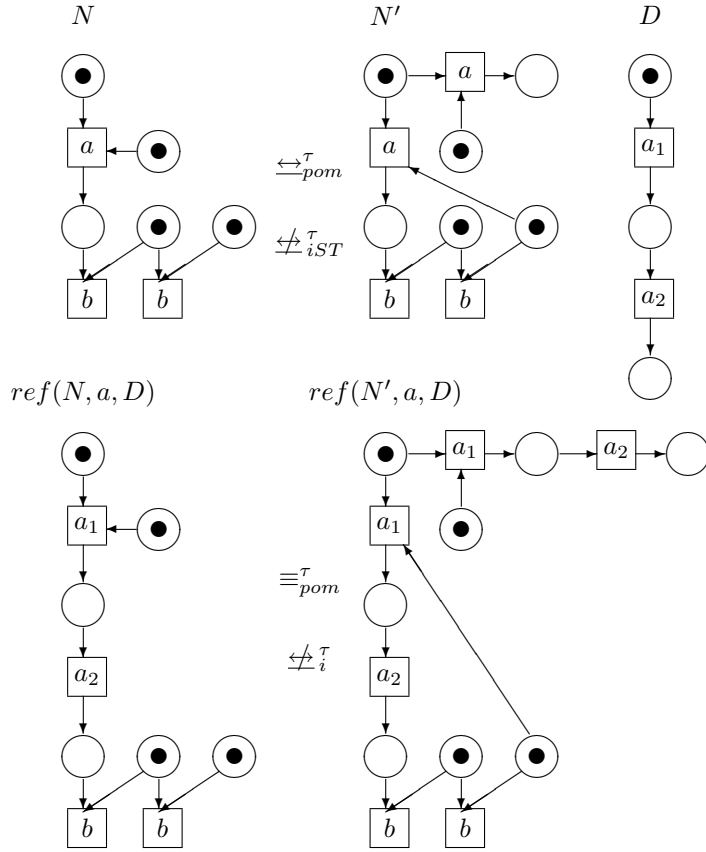


Рис. 1.48:  $\tau$ -эквивалентности от  $\leftrightarrow_i^{\tau}$  до  $\leftrightarrow_{pom}^{\tau}$  не сохраняются при SM-детализациях

- На рисунке 1.25  $N \leftrightarrow_s^{\tau} N'$ , но  $ref(N, c, D) \not\equiv_i^{\tau} ref(N', c, D)$ , так как только в  $ref(N', c, D)$  можно выполнить последовательность действий  $c_1abc_2$ . Следовательно,  $\tau$ -эквивалентности от  $\equiv_i^{\tau}$  до  $\leftrightarrow_s^{\tau}$  не сохраняются при SM-детализациях.
- На рисунке 1.48  $N \leftrightarrow_{pom}^{\tau} N'$ , но  $ref(N, a, D) \not\equiv_i^{\tau} ref(N', a, D)$ , так как только в  $ref(N', a, D)$  после срабатывания действия  $a_1$  не сможет выполняться  $b$ . Следовательно, эквивалентности от  $\leftrightarrow_i^{\tau}$  до  $\leftrightarrow_{pom}^{\tau}$  не сохраняются при SM-детализациях.
- На рисунке 1.49  $N \leftrightarrow_{pomhbr}^{\tau} N'$ , но  $ref(N, a, D) \not\equiv_i^{\tau} ref(N', a, D)$ , так как только в  $ref(N', a, D)$  действие  $c_1$  может выполняться так, что после соответствующего действия  $c_1$  в сети  $N$  действие  $a$  может случиться так, что действие  $b$  никогда не случится. Следовательно, эквивалентности от  $\leftrightarrow_i^{\tau}$  до  $\leftrightarrow_{pomhbr}^{\tau}$  не сохраняются при SM-детализациях.

На рисунке 1.50 линиями обведены  $\tau$ -эквивалентности, которые не сохраняются при SM-детализациях, что демонстрируют примеры на рисунках 1.25, 1.48 и 1.49. □

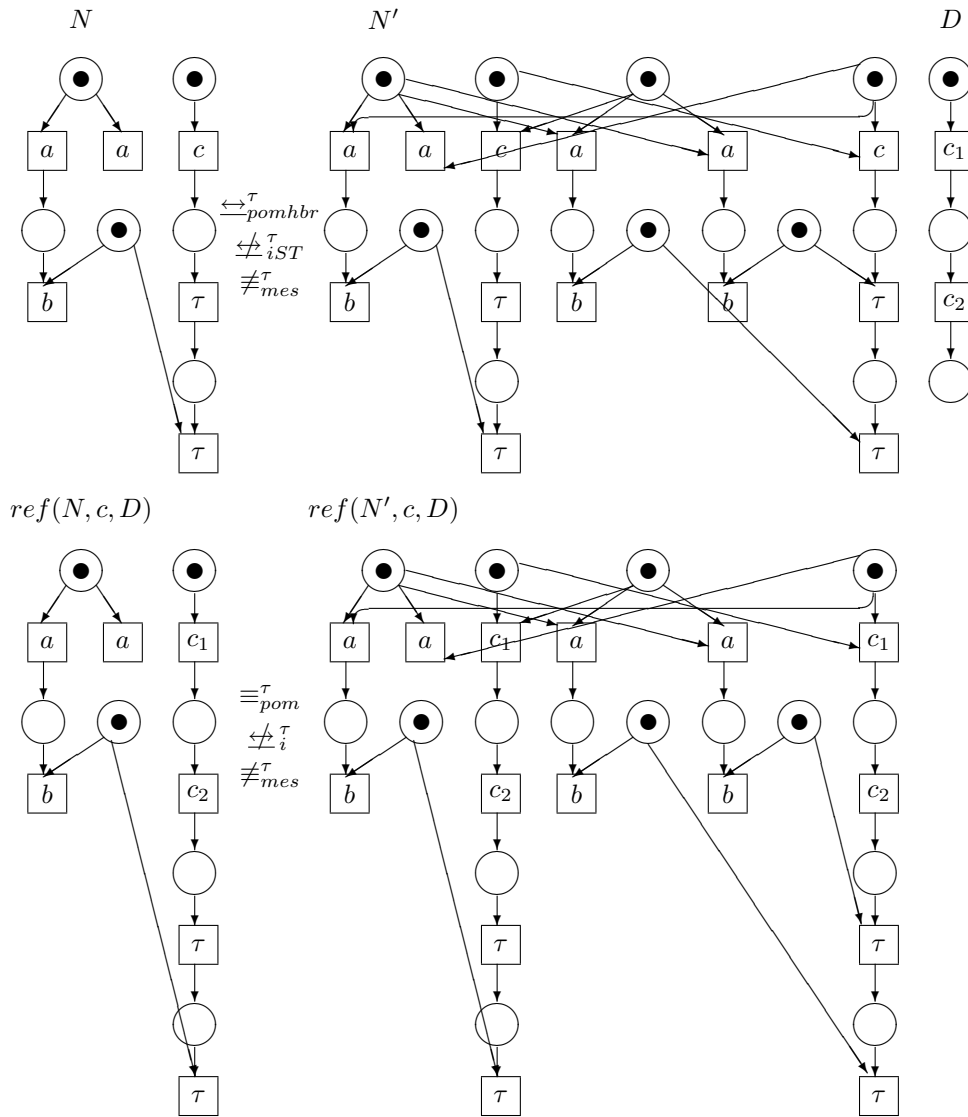


Рис. 1.49:  $\tau$ -эквивалентности от  $\xleftrightarrow{i}^{\tau}$  до  $\xleftrightarrow{pomhbr}^{\tau}$  не сохраняются при SM-детализациях

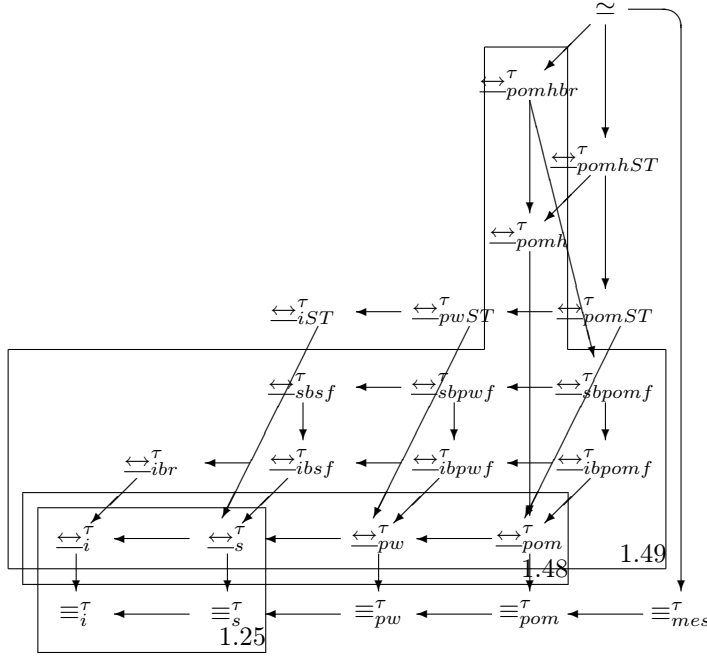


Рис. 1.50:  $\tau$ -эквивалентности, не сохраняющиеся при SM-детализациях

Рассмотрим, какие  $\tau$ -эквивалентности сохраняются при SM-детализациях.

**Предложение 1.3.7** Пусть  $\star \in \{pw, pom\}$ . Для сетей с невидимыми переходами  $N$  и  $N'$  таких, что  $a \in l_N(T_N) \cap l_{N'}(T_{N'}) \cap Act$  и SM-сети с невидимыми переходами  $D$   $N \equiv_{\star}^{\tau} N' \Rightarrow ref(N, a, D) \equiv_{\star}^{\tau} ref(N', a, D)$ .

*Доказательство.* Аналогично доказательству соответствующего предложения для эквивалентностей на сетях с видимыми переходами, с учетом невидимых действий.  $\square$

**Предложение 1.3.8** Пусть  $\star \in \{i, pw, pom\}$ . Для сетей с невидимыми переходами  $N$  и  $N'$  таких, что  $a \in l_N(T_N) \cap l_{N'}(T_{N'}) \cap Act$  и SM-сети с невидимыми переходами  $D$   $N \leftrightarrow_{\star ST}^{\tau} N' \Rightarrow ref(N, a, D) \leftrightarrow_{\star ST}^{\tau} ref(N', a, D)$ .

*Доказательство.* Аналогично доказательству соответствующего предложения для эквивалентностей на сетях с видимыми переходами, с учетом невидимых действий и того, что в  $ST$ - $\tau$ -процессах работающими (максимальными) являются только видимые действия.  $\square$

**Предложение 1.3.9** [121, 122] Для сетей с невидимыми переходами  $N$  и  $N'$  таких, что  $a \in l_N(T_N) \cap l_{N'}(T_{N'}) \cap Act$  и SM-сети с невидимыми переходами  $D$   $N \leftrightarrow_{pomhST}^{\tau} N' \Rightarrow ref(N, a, D) \leftrightarrow_{pomhST}^{\tau} ref(N', a, D)$ .

Следующее предложение демонстрирует новый результат для МСС-эквивалентности.

**Предложение 1.3.10** Для сетей с невидимыми переходами  $N$  и  $N'$  таких, что  $a \in l_N(T_N) \cap l_{N'}(T_{N'}) \cap Act$  и SM-сети с невидимыми переходами  $D$   $N \equiv_{mes}^{\tau} N' \Rightarrow ref(N, a, D) \equiv_{mes}^{\tau} ref(N', a, D)$ .

*Доказательство.* Аналогично доказательству соответствующего предложения для эквивалентностей на сетях с видимыми переходами, с учетом невидимых действий.  $\square$

**Предложение 1.3.11** Для сетей с невидимыми переходами  $N$  и  $N'$  таких, что  $a \in l_N(T_N) \cap l_{N'}(T_{N'}) \cap Act$  и SM-сети с невидимыми переходами  $D$   $N \simeq N' \Rightarrow ref(N, a, D) \simeq ref(N', a, D)$ .

*Доказательство.* Очевидно.  $\square$

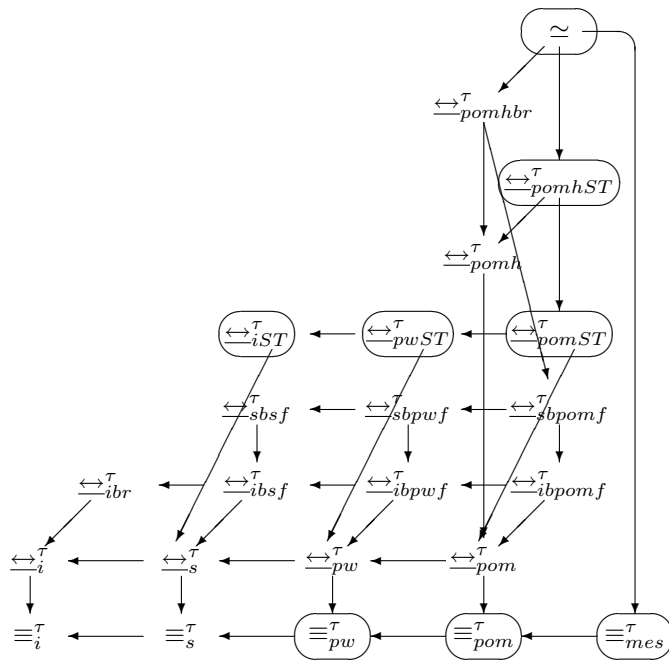


Рис. 1.51: Сохранение  $\tau$ -эквивалентностей при SM-детализациях

**Теорема 1.3.5** Пусть  $\leftrightarrow \in \{\equiv^\tau, \leftrightarrow^\tau, \simeq\}$  и  $\star \in \{\_, i, s, pw, pom, iST, pwST, pomST, pomh, pomhST, ibr, pomhbr, mes, ibr, ibpwf, ibpomf, sbsf, sbpwf, sbpomf\}$ . Для сетей с невидимыми переходами  $N$  и  $N'$  так, что  $a \in l_N(T_N) \cap l_{N'}(T_{N'}) \cap Act$  и SM-сети с невидимыми переходами  $D N \leftrightarrow_\star N' \Rightarrow ref(N, a, D) \leftrightarrow_\star ref(N', a, D)$  тогда и только тогда, когда эквивалентность  $\leftrightarrow_\star$  заключена в овал на рисунке 1.51.

*Доказательство.* По предложениям 1.3.6–1.3.10. □

### 1.3.5 $\tau$ -эквивалентности на подклассах сетей с невидимыми переходами

Рассмотрим  $\tau$ -эквивалентности на сетях с видимыми переходами и на последовательных сетях с невидимыми переходами.

Заметим, что на последовательных сетях в литературе исследовались лишь  $\leftrightarrow_i^\tau$ ,  $\leftrightarrow_{pom}^\tau$  и  $\leftrightarrow_{pomh}^\tau$  [46].

#### $\tau$ -эквивалентности на сетях с видимыми переходами

Рассмотрим  $\tau$ -эквивалентности на сетях с видимыми переходами, где ни один из переходов не помечен действием  $\tau$ .

**Предложение 1.3.12** Пусть  $\leftrightarrow \in \{\equiv, \leftrightarrow\}$ ,  $\star \in \{i, s, pw, pom, iST, pwST, pomST, mes, sbsf, sbpwf, sbpomf\}$ ,  $\star\star \in \{s, pw, pom\}$ . Для сетей с видимыми переходами  $N$  и  $N'$ :

1.  $N \leftrightarrow_\star N' \Leftrightarrow N \leftrightarrow_\star^\tau N'$ ;
2.  $N \leftrightarrow_{pomh} N' \Leftrightarrow N \leftrightarrow_{pomhST}^\tau N'$ ;
3.  $N \leftrightarrow_i N' \Leftrightarrow N \leftrightarrow_{ibr}^\tau N'$ ;
4.  $N \leftrightarrow_{pomh} N' \Leftrightarrow N \leftrightarrow_{pomhbr}^\tau N'$ ;
5.  $N \leftrightarrow_{\star\star} N' \Leftrightarrow N \leftrightarrow_{ib\star\star f}^\tau N'$ .

*Доказательство.* ( $\Leftarrow$ )

1. По определениям.
2. Доказывается построением на основе отношения  $\mathcal{R} : N \leftrightarrow_{pomhST}^\tau N'$  нового отношения  $\mathcal{S} : N \leftrightarrow_{pomh} N'$ , определяемого следующим образом:  $\mathcal{S} = \{(\pi, \pi', \beta) \mid \exists \beta ((\pi, \pi), (\pi', \pi'), \beta) \in \mathcal{R}\}$ .
3. По определениям.
4. По определениям.
5. Доказывается построением на основе отношения  $\mathcal{R} : N \leftrightarrow_{ib\star\star f}^\tau N'$  the нового отношения  $\mathcal{S} : N \leftrightarrow_{\star\star} N'$ , определяемого следующим образом:  $\mathcal{S} = \{(\pi, \pi') \mid \exists \sigma, \sigma' ((\pi, \sigma), (\pi', \sigma')) \in \mathcal{R}\}$ .

( $\Rightarrow$ ) По предложению 1.3.5, так как сети с видимыми переходами — подкласс сетей с невидимыми переходами. □

На рисунке 1.52 пунктирной линией обведены  $\tau$ -эквивалентности, совпадающие на сетях с видимыми переходами.

**Теорема 1.3.6** Пусть  $\leftrightarrow, \leftrightarrow\star \in \{\equiv, \leftrightarrow, \simeq\}$ ,  $\star, \star\star \in \{\_, i, s, pw, pom, iST, pwST, pomST, pomh, ibr, mes, sbsf, sbpwf, sbpomf\}$ . Для сетей с видимыми переходами  $N$  и  $N'$   $N \leftrightarrow_\star N' \Rightarrow N \leftrightarrow_{\star\star} N'$  тогда и только тогда, когда в графе на рисунке 1.53 существует направленный путь от  $\leftrightarrow_\star$  к  $\leftrightarrow_{\star\star}$ .

*Доказательство.* По предложению 1.3.11 и теореме 1.3.2. □

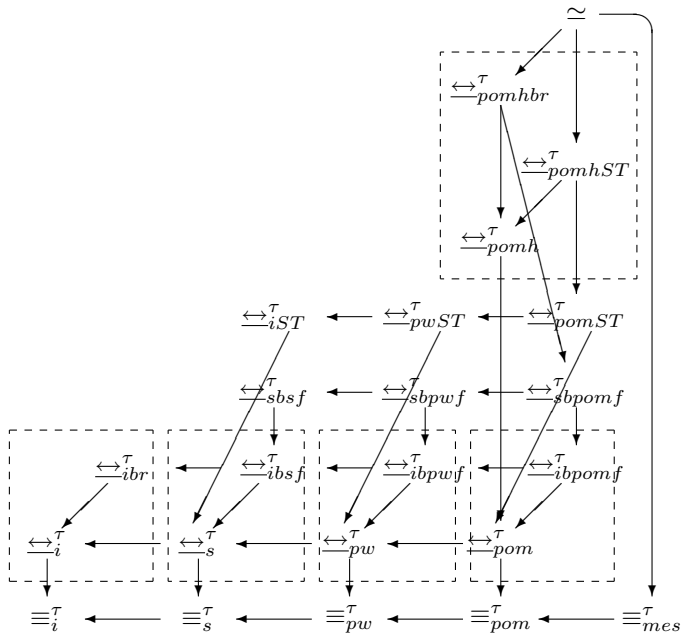


Рис. 1.52: Совпадение  $\tau$ -эквивалентностей на сетях с видимыми переходами

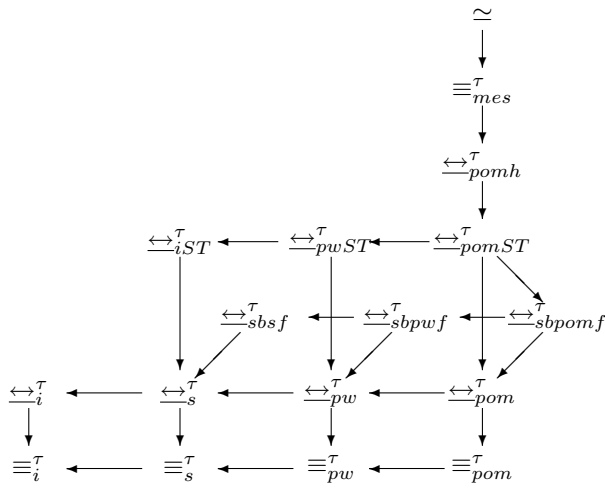


Рис. 1.53: Взаимосвязь  $\tau$ -эквивалентностей на сетях с видимыми переходами

### $\tau$ -эквивалентности на последовательных сетях с невидимыми переходами

Рассмотрим  $\tau$ -эквивалентности на последовательных сетях с невидимыми переходами, где никакие два перехода не могут сработать параллельно.

**Определение 1.3.21** *Сеть с невидимыми переходами*  $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$  — последовательная, если  $\forall M \in \text{Mark}(N) \neg \exists t, u \in T_N : \bullet t + \bullet u \subseteq M$ .

**Предложение 1.3.13** *Для последовательных сетей с невидимыми переходами*  $N$  и  $N'$ :  $N \equiv_i^\tau N' \Leftrightarrow N \equiv_{\text{pom}}^\tau N'$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие теоремы 1.3.1.

( $\Rightarrow$ ) Пусть  $N \equiv_i^\tau N'$ . Тогда  $\text{VisIntTraces}(N) = \text{VisIntTraces}(N')$ . Для доказательства того, что  $N \equiv_{\text{pom}}^\tau N'$ , достаточно установить равенство  $\text{VisPomsets}(N) = \text{VisPomsets}(N')$ . Оно очевидно, так как  $\text{VisPomsets}(N)$  и  $\text{VisPomsets}(N')$  — линейно упорядоченные мультимножества (цепочки), и существует взаимно однозначное отображение между  $\text{VisIntTraces}(N)$  и  $\text{VisPomsets}(N)$  ( $\text{VisIntTraces}(N')$  и  $\text{VisPomsets}(N')$  соответственно).  $\square$

**Предложение 1.3.14** *Для последовательных сетей с невидимыми переходами*  $N$  и  $N'$ :  $N \leftrightarrow_i^\tau N' \Leftrightarrow N \leftrightarrow_{\text{pomh}}^\tau N'$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие теоремы 1.3.1.

( $\Rightarrow$ ) Пусть  $\mathcal{R} : N \leftrightarrow_i^\tau N'$ . Докажем, что  $\mathcal{S} : N \leftrightarrow_{\text{pomh}}^\tau N'$ , где  $\mathcal{S}$  определяется так:  $\mathcal{S} = \{(\pi, \pi', \beta) \mid (\pi, \pi') \in \mathcal{R}, \beta : \text{vis}(\rho_C) \simeq \text{vis}(\rho_{C'}), \pi = (C, \varphi), \pi' = (C', \varphi')\}$ .

1. Очевидно,  $(\pi_N, \pi_{N'}, \emptyset) \in \mathcal{S}$ .

2. По определению  $\mathcal{S}$ ,  $(\pi, \pi', \beta) \in \mathcal{S} \Rightarrow \beta : \text{vis}(\rho_C) \simeq \text{vis}(\rho_{C'})$ ;

3. Пусть  $(\pi, \pi', \beta) \in \mathcal{S}$ ,  $\pi \xrightarrow{v} \tilde{\pi}$ ,  $\tilde{\pi} = (\tilde{C}, \tilde{\varphi})$ . По определению последовательных сетей, все переходы в  $C$ -сетях их процессов линейно упорядочены отношением предшествования. Поэтому, по лемме 1.2.2 полные выполнения таких  $C$ -сетей единственны, и существует взаимно однозначное соответствие между видимыми ЧУММ этих  $C$ -сетей и последовательностями видимых действий, выписанных в порядке срабатывания помечаемых ими переходов.

Рассмотрим полное выполнение  $\tilde{C}$  в  $N$   $\pi_N = \pi_0 \xrightarrow{v_1} \dots \xrightarrow{v_n} \pi_n = \pi \xrightarrow{v} \tilde{\pi}$ . По определению интерливинговой  $\tau$ -бисимуляции  $\exists \tilde{\pi}', v'$  такие, что  $\pi_{N'} = \pi'_0 \xrightarrow{\beta(v_1)} \dots \xrightarrow{\beta(v_n)} \pi'_n = \pi' \xrightarrow{v'} \tilde{\pi}'$ ,  $\tilde{\pi}' = (\tilde{C}', \tilde{\varphi}')$ ,  $l_{\tilde{C}}(v) = l_{\tilde{C}'}(v')$ ,  $(\tilde{\pi}, \tilde{\pi}') \in \mathcal{R}$ . Это — единственное полное выполнение  $\tilde{C}'$  в  $N'$ .

Рассмотрим отображение  $\tilde{\beta}$ , где  $\tilde{\beta}|_{TC} = \beta$  и  $\tilde{\beta}(v) = v'$ . Очевидно, что  $\tilde{\beta} : \rho_{\tilde{C}} \simeq \rho_{\tilde{C}'}$ , так как существует взаимно однозначное соответствие между  $\text{vis}(\rho_{\tilde{C}})$ ,  $\text{vis}(\rho_{\tilde{C}'})$  и последовательностью видимых действий  $\text{vis}(l_{\tilde{C}}(v_1) \dots l_{\tilde{C}}(v_n) l_{\tilde{C}}(v))$ .

4. Как пункт 3, но роли  $N$  и  $N'$  меняются.  $\square$

**Предложение 1.3.15** *Для последовательных сетей с невидимыми переходами*  $N$  и  $N'$ :  $N \leftrightarrow_{iST}^\tau N' \Leftrightarrow N \leftrightarrow_{\text{pomhST}}^\tau N'$ .

*Доказательство.* Аналогично доказательству предложения 1.3.13.  $\square$

**Предложение 1.3.16** *Для последовательных сетей с невидимыми переходами*  $N$  и  $N'$ :  $N \leftrightarrow_{ibr}^\tau N' \Leftrightarrow N \leftrightarrow_{\text{pomhbr}}^\tau N'$ .

*Доказательство.* Аналогично доказательству предложения 1.3.13.  $\square$

На рисунке 1.54 пунктирной линией обведены  $\tau$ -эквивалентности, совпадающие на последовательных сетях с невидимыми переходами.

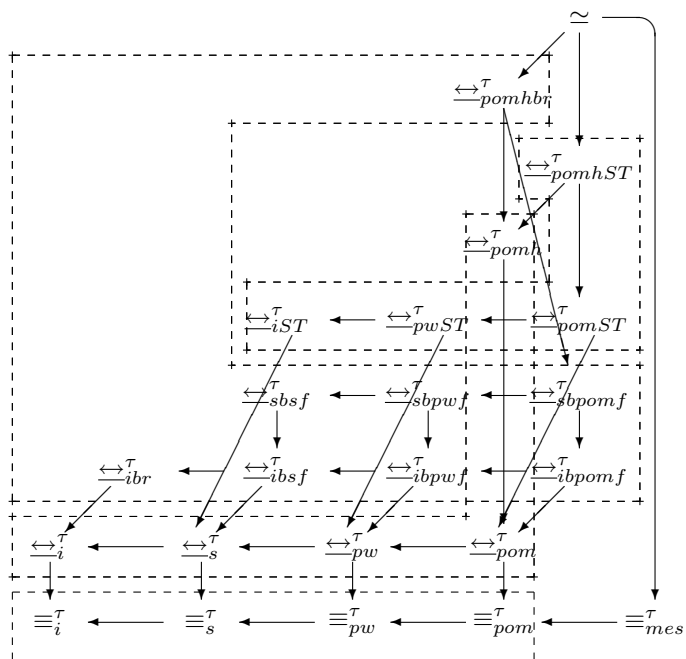


Рис. 1.54: Совпадение  $\tau$ -эквивалентностей на последовательных сетях с невидимыми переходами



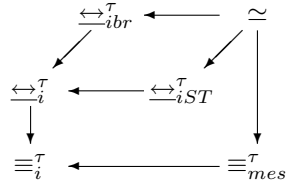


Рис. 1.55: Взаимосвязь  $\tau$ -эквивалентностей на последовательных сетях с невидимыми переходами

**Теорема 1.3.7** Пусть  $\leftrightarrow, \Leftarrow \in \{\equiv^\tau, \leftrightarrow^\tau, \simeq\}$ ,  $\star, \star\star \in \{_, i, iST, ibr, mes\}$ . Для последовательных сетей с невидимыми переходами  $N$  и  $N'$   $N \leftrightarrow_\star N'$  Rightarrow  $N \Leftarrow_{\star\star} N'$  тогда и только тогда, когда в графе на рисунке 1.55 существует направленный путь от  $\leftrightarrow_\star$  к  $\Leftarrow_{\star\star}$ .

*Доказательство.* ( $\Leftarrow$ ) По теореме 1.3.1.

( $\Rightarrow$ ) Отсутствие дополнительных нетривиальных стрелок в графе на рисунке 1.55 доказывается следующими примерами на последовательных сетях с невидимыми переходами.

- На рисунке 1.43(a)  $N \equiv_{mes}^\tau N'$ , но  $N \not\leftrightarrow_i^\tau N'$ .
- На рисунке 1.43(c)  $N \leftrightarrow_i^\tau N'$ , но  $N \not\leftrightarrow_{ibr}^\tau N'$ .
- На рисунке 1.43(b)  $N \leftrightarrow_i^\tau N'$ , но  $N \not\leftrightarrow_{iST}^\tau N'$ .
- На рисунке 1.13(c)  $N \leftrightarrow_{ibr}^\tau N'$ , но  $N \not\equiv_{mes}^\tau N'$ .
- На рисунке 1.13(c)  $N \leftrightarrow_{iST}^\tau N'$ , но  $N \not\equiv_{mes}^\tau N'$ .

□

## Глава 2

# Временные сети Петри

В последние годы наблюдается растущий интерес к системам реального времени. В связи с этим возникает необходимость ввести понятие времени в формальные модели таких систем. Разработан ряд методов спецификации и анализа, учитывающих временные аспекты функционирования.

Однако значительно меньшая часть исследований посвящена введению времени в эквивалентностные отношения. В нескольких работах [12, 92] по этой теме изучались в основном вопросы разрешимости временных эквивалентностей.

В этих работах системы реального времени моделируются параллельными таймерными процессами или временными автоматами, содержащими компоненты для измерения времени, называемые таймерами. Но параллелизм не может быть представлен явно при использовании таких моделей.

С другой стороны, в работах [196, 91] были введены временные сети Петри, расширение сетей Петри компонентами, учитывающими время. Временная сеть функционирует одним из двух способов: срабатыванием переходов или задержкой на определенное время. Использование временных сетей для верификации описано, например, в [293]. В [253] изложены идеи сохраняющей поведение редукции временных сетей.

В этой главе эквивалентностные понятия изучаются на двух классах временных сетей Петри: временные сети с видимыми переходами и временные сети с невидимыми переходами (переходы могут быть помечены символом невидимого действия  $\tau$ ).

Основные результаты данной главы следующие.

- Ряд известных эквивалентностей перенесен на временные сети из других формальных моделей (например, временных систем переходов, параллельных таймерных процессов, временных автоматов и алгебр временных процессов).
- Введены новые временные (учитывающие временные аспекты) и не-временные (абстрагирующиеся от времени) эквивалентностные отношения с целью получения их полного набора для возможности выбора наиболее подходящей семантики.
- Дана характеристика временных эквивалентностей посредством региональных эквивалентностных отношений. Эти отношения основаны на концепции регионов, объединяющих бесконечное число промежуточных состояний, получающихся только изменением временной составляющей и не различающихся по влиянию на поведение временной сети. Таким образом, за счет значительной редукции состояний упрощается проверка временных эквивалентностей.
- Проведено сравнение всех рассмотренных сетевых эквивалентностей и получена полная картина их взаимосвязей, демонстрирующая их различающую силу.
- Введена новая операция временной детализации, позволяющая менять уровень абстракции от структуры сетей и сохранять временные задержки.
- Все эквивалентности проверены на сохранение при операции временной детализации для выявления претендентов на использование в нисходящей разработке временных систем.
- Исследована взаимосвязь эквивалентностей на подклассах временных сетей, позволяющая выявить природу этих отношений и упростить их проверку.

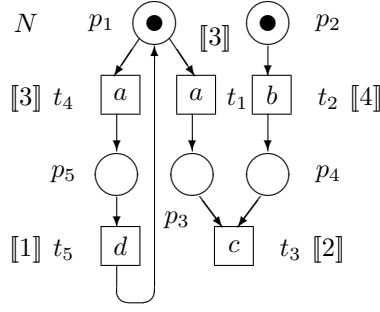


Рис. 2.1: Пример временной сети

## 2.1 Основные определения

**Определение 2.1.1** Временная сеть — шестерка  $N = \langle P_N, T_N, F_N, l_N, M_N, \Upsilon_N \rangle$ , где:

- $\langle P_N, T_N, F_N, l_N, M_N \rangle$  — безопасная сеть без петель (с пометкой над Act);
- $\Upsilon_N : T_N \rightarrow \mathbf{N}$  — начальная функция временной задержки.

**Пример 2.1.1** На рисунке 2.1 дано графическое представление временной сети. Начальные временные задержки изображаются числами в двойных квадратных скобках возле соответствующих переходов.

Пусть  $N = \langle P_N, T_N, F_N, l_N, M_N, \Upsilon_N \rangle$  и  $N' = \langle P_{N'}, T_{N'}, F_{N'}, l_{N'}, M_{N'}, \Upsilon_{N'} \rangle$  — временные сети. Отображение  $\beta : N \rightarrow N'$  — изоморфизм между  $N$  и  $N'$ , запись  $\beta : N \simeq N'$ , если:

1.  $\beta : \langle P_N, T_N, F_N, l_N, M_N \rangle \simeq \langle P_{N'}, T_{N'}, F_{N'}, l_{N'}, M_{N'} \rangle$ ;
2.  $\forall t \in T_N \ \Upsilon_N(t) = \Upsilon_{N'}(\beta(t))$ .

Временные сети  $N$  и  $N'$  изоморфны, запись  $N \simeq N'$ , если  $\exists \beta : N \simeq N'$ .

Пусть  $\mathbf{R}^+$  — множество неотрицательных действительных чисел.

Состояние временной сети  $N$  — пара  $Q = (M, \Upsilon)$ , где:

1.  $M \in 2^{P_N}$  — маркировка сети  $N$ ;
2.  $\Upsilon : T_N \rightarrow \mathbf{R}^+$  — функция временной задержки.

Начальное состояние временной сети  $N$  — пара  $Q_N = (M_N, \Upsilon_N)$ .

Временная сеть может функционировать посредством срабатывания переходов или ожиданием в течение определенного времени (временной задержки).

Пусть  $Q = (M, \Upsilon)$  — состояние временной сети  $N$ .

Переход  $t \in T_N$  — допустим в  $Q$ , запись  $Q \xrightarrow{t}$ , если  $M \xrightarrow{t}$  и  $\Upsilon(t) = 0$ . Если  $t$  допустим в  $Q$ , то срабатывание этого перехода изменяет состояние  $Q$  на  $\tilde{Q} = (\tilde{M}, \tilde{\Upsilon})$ , запись  $Q \xrightarrow{t} \tilde{Q}$ , где  $\tilde{Q}$  определяется следующим образом:

1.  $\tilde{M} = (M \setminus \bullet t) \cup t \bullet$ ;
2.  $\forall u \in T_N \ \tilde{\Upsilon}(u) \begin{cases} \Upsilon_N(u), & (\bullet u \subseteq \tilde{M}) \wedge (\bullet u \not\subseteq M); \\ \Upsilon(u), & \text{иначе.} \end{cases}$

Пишем  $Q \xrightarrow{a} \tilde{Q}$ , если  $\exists t \ Q \xrightarrow{t} \tilde{Q}$  и  $l_N(t) = a$ .

Время  $\delta \in \mathbf{R}^+$  может пройти в  $Q$ , если  $\forall u \in T_N$  такого, что  $\bullet u \subseteq M$  выполняется:  $0 < \delta \leq \Upsilon(u)$ . Если  $\delta$  может пройти в  $Q$ , истечение этого времени меняет состояние  $Q$  на  $\tilde{Q} = (\tilde{M}, \tilde{\Upsilon})$ , запись  $Q \xrightarrow{\delta} \tilde{Q}$ , где  $\tilde{Q}$  определяется следующим образом:

1.  $\tilde{M} = M$ ;
2.  $\forall u \in T_N \ \tilde{\Upsilon}(u) = \Upsilon(u) \ominus \delta$ .

Будем считать, что отношение  $\xrightarrow{\delta}$  обладает следующим свойством аддитивности:  $\exists \delta_1, \delta_2$ ,  $Q_1 \xrightarrow{\delta_1} Q_1 \xrightarrow{\delta_2} \tilde{Q}$  и  $\delta_1 + \delta_2 = \delta$  влечет  $Q \xrightarrow{\delta} \tilde{Q}$ .

Состояние  $Q$  временной сети  $N$  — *достижимое*, если  $Q = Q_N$  или существует достижимое состояние  $\hat{Q} \in N$  такое, что  $\hat{Q} \xrightarrow{x} Q$  для некоторого  $x \in Act \cup \mathbf{R}^+$ .  $States(N)$  обозначает множество *всех достижимых состояний*  $N$ .

Будем рассматривать только временные сети, удовлетворяющие *условию прогресса*, в соответствии с которым  $\sum_{i=0}^n \Upsilon_N(t_i) > 0$  для любого цикла  $\{t_0, \dots, t_n\}$  временной сети  $N$ .

## 2.2 Временные сети Петри с видимыми переходами

Рассмотрим эквивалентности для временных сетей Петри с видимыми переходами, моделирующих системы, все действия которых считаются видимыми для внешнего наблюдателя.

Следующие базисные эквивалентности для временных формальных моделей с видимыми действиями были рассмотрены в литературе.

- *Временные эквивалентности* (учитывают временные задержки в функционировании систем).
  - Временная следовая эквивалентность ( $\equiv_t$ ) введена в [12] на временных автоматах.
  - Временная бисимуляционная эквивалентность ( $\leftrightarrow_t$ ) введена в [92] на параллельных таймерных процессах и в [12] — на временных автоматах.
- *Не-временные эквивалентности* (не учитывают временные задержки в функционировании систем).
  - Не-временная следовая эквивалентность ( $\equiv_u$ ) введена в [12] на временных автоматах.
  - Не-временная бисимуляционная эквивалентность ( $\leftrightarrow_u$ ) введена в [12] на временных автоматах.
- *Изоморфизм* (совпадение временных систем с точностью до переименования их компонент). Это отношение ( $\simeq$ ) введено в [196].

Мы рассматриваем также региональные эквивалентности, разбивающие множества состояний временных систем на “регионы” [10]. Так как региональные эквивалентности совпадают с временными, проверка последних облегчается вследствие сокращения бесконечного числа промежуточных состояний, получающихся задержками на различное время. Бисимуляционная региональная эквивалентность ( $\leftrightarrow_r$ ) была рассмотрена в [92] на параллельных таймерных процессах. Мы вводим новое понятие следовой региональной эквивалентности ( $\equiv_r$ ).

Проводится сравнение всех эквивалентностных понятий. Доказывается совпадение региональных и временных эквивалентностей.

Эквивалентности проверяются на сохранение при новой операции временной SM-детализации.

Проводится сравнение всех эквивалентностных отношений на не-временных сетях.

### 2.2.1 Временные эквивалентности

#### Временная следовая эквивалентность

**Определение 2.2.1** Временной след временной сети  $N$  — последовательность  $x_1 \dots x_n \in (Act \cup \mathbf{R}^+)^*$  такая, что  $Q_N \xrightarrow{x_1} Q_1 \xrightarrow{x_2} \dots \xrightarrow{x_n} Q_n$ . Обозначим множество всех временных следов временной сети  $N$  через  $TimeTraces(N)$ . Временные сети  $N$  и  $N'$  временно следово эквивалентны, запись  $N \equiv_t N'$ , если  $TimeTraces(N) = TimeTraces(N')$ .

#### Временная бисимуляционная эквивалентность

**Определение 2.2.2** Пусть  $N$  и  $N'$  — временные сети. Отношение  $\mathcal{R} \subseteq States(N) \times States(N')$  — временная бисимуляция между  $N$  и  $N'$ , запись  $\mathcal{R} : N \leftrightarrow_t N'$ , если:

1.  $(Q_N, Q_{N'}) \in \mathcal{R}$ .
2.  $(Q, Q') \in \mathcal{R}$ ,  $Q \xrightarrow{x} \tilde{Q}$  ( $x \in Act \cup \mathbf{R}^+$ )  $\Rightarrow \exists \tilde{Q}' : Q' \xrightarrow{x} \tilde{Q}'$ ,  $(\tilde{Q}, \tilde{Q}') \in \mathcal{R}$ .
3. Как пункт 2, но роли of  $N$  и  $N'$  меняются.

Временные сети  $N$  и  $N'$  временно бисимуляционно эквивалентны, запись  $N \leftrightarrow_t N'$ , если  $\exists \mathcal{R} : N \leftrightarrow_t N'$ .

## 2.2.2 Не-временные эквивалентности

Пусть  $N$  — временная сеть и  $Q, \tilde{Q} \in States(N)$ . Введем следующие обозначения.

- $Q \xrightarrow{t} \tilde{Q} \Leftrightarrow \exists \delta_1, \delta_2, Q_1, Q_2 \quad Q \xrightarrow{\delta_1} Q_1 \xrightarrow{t} Q_2 \xrightarrow{\delta_2} \tilde{Q}$ ;
- $Q \xrightarrow{a} \tilde{Q} \Leftrightarrow \exists t \quad Q \xrightarrow{t} \tilde{Q}, l_N(t) = a$ .

### Не-временная следовая эквивалентность

**Определение 2.2.3** Не-временной след временной сети  $N$  — последовательность  $a_1 \cdots a_n \in Act^*$  такая, что  $Q_N \xrightarrow{a_1} Q_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} Q_n$ . Обозначим множество всех не-временных следов временной сети  $N$  через  $UntimeTraces(N)$ . Временные сети  $N$  и  $N'$  не-временно следово эквивалентны, запись  $N \equiv_u N'$ , если  $UntimeTraces(N) = UntimeTraces(N')$ .

### Не-временные бисимуляционные эквивалентности

**Определение 2.2.4** Пусть  $N$  и  $N'$  — временные сети. Отношение  $\mathcal{R} \subseteq States(N) \times States(N')$  — не-временная бисимуляция между  $N$  и  $N'$ , запись  $\mathcal{R} : N \xleftrightarrow{u} N'$ , если:

1.  $(Q_N, Q_{N'}) \in \mathcal{R}$ .
2.  $(Q, Q') \in \mathcal{R}, Q \xrightarrow{a} \tilde{Q} (a \in Act) \Rightarrow \exists \tilde{Q}' : Q' \xrightarrow{a} \tilde{Q}', (\tilde{Q}, \tilde{Q}') \in \mathcal{R}$ .
3. Как пункт 2, но роли of  $N$  и  $N'$  меняются.

Временные сети  $N$  и  $N'$  не-временно бисимуляционно эквивалентны, запись  $N \xleftrightarrow{u} N'$ , если  $\exists \mathcal{R} : N \xleftrightarrow{u} N'$ .

## 2.2.3 Региональные эквивалентности

Пусть  $\delta \in \mathbf{R}^+$ . Обозначим через  $[\delta]$  целую часть, а через  $\{\delta\}$  — дробную часть  $\delta$ .

**Определение 2.2.5** Пусть  $Q_1 = (M_1, \Upsilon_1)$  и  $Q_2 = (M_2, \Upsilon_2)$  — состояния временной сети  $N$ . Функции временной задержки  $\Upsilon_1$  и  $\Upsilon_2$  регионально эквивалентны, запись  $\Upsilon_1 =_{reg} \Upsilon_2$ , если:

1.  $\forall t \in T_N \quad [\Upsilon_1(t)] [\Upsilon_2(t)]$ ;
2. (a)  $\forall t \in T_N \quad \{\Upsilon_1(t)\} = 0 \Leftrightarrow \{\Upsilon_2(t)\} = 0$ ;  
(b)  $\forall t, u \in T_N \quad \{\Upsilon_1(t)\} < \{\Upsilon_1(u)\} \Leftrightarrow \{\Upsilon_2(t)\} < \{\Upsilon_2(u)\}$ .

Состояния  $Q_1$  и  $Q_2$  регионально эквивалентны, запись  $Q_1 =_{reg} Q_2$ , если  $M_1 = M_2$  и  $\Upsilon_1 =_{reg} \Upsilon_2$ .

Временной регион состояния  $Q$  определяется следующим образом:  $[Q] = [Q]_{=reg} = \{R \mid R =_{reg} Q\}$ .

Определим множество всех достижимых региональных состояний временной сети  $N$  как:  $RegStates(N) = States(N) /_{=reg} = \{[Q] \mid Q \in States(N)\}$ . Пусть  $S_N = [Q_N]$  — начальное региональное состояние.

**Пример 2.2.1** Найдем временной регион состояния  $Q = (M, \Upsilon)$  временной сети  $N$  с семью переходами. Если  $\Upsilon(t_i) = \delta_i$  ( $1 \leq i \leq 7$ ) и  $\delta_1 = 0.1, \delta_2 = 5, \delta_3 = 1.33, \delta_4 = 7.42, \delta_5 = 7.999, \delta_6 = 5, \delta_7 = 0$ , тогда соответствующий временной регион  $[Q]$  имеет задержки переходов, определяемые следующими неравенствами:

1.  $0 = \delta_7 < \delta_1 < 1 < \delta_3 < 2 < 5 = \delta_2 = \delta_6 < 7 < \delta_4 < \delta_5 < 8$ ;
2.  $0 = \{\delta_2\} = \{\delta_6\} = \{\delta_7\} < \{\delta_1\} < \{\delta_3\} < \{\delta_4\} < \{\delta_5\}$ .

Отображение  $\kappa : \mathbf{R}^+ \rightarrow \mathbf{R}^+$  — равномерное, если:

1.  $\kappa(0) = 0$ ;
2.  $\forall n \in \mathbf{N} \quad \kappa(x + n) = \kappa(x) + n$ .

Пусть  $Q = (M, \Upsilon)$  — состояние временной сети  $N$  и  $Q' = (M, \Upsilon)$  — состояние временной сети  $N'$ . Обозначим  $\kappa(Q) = (M, \kappa \circ \Upsilon)$ .

**Лемма 2.2.1** [92]

1. Пусть  $Q_1 = (M_1, \Upsilon_1), Q_2 = (M_2, \Upsilon_2)$  — состояния временной сети  $N$ . Пишем  $Q_1 =_{reg} Q_2$ , если существует равномерное отображение  $\kappa$  такое, что  $\kappa(Q_1) = (Q_2)$ .
2. Если  $\kappa$  — равномерное отображение, то  $\forall \delta \in \mathbf{R}^+$  отображение  $\kappa_\delta$ , определяемое как  $\forall x \in \mathbf{R}^+$   
 $\kappa_\delta(x) = \kappa(x + \delta) - \kappa(\delta)$ , также равномерное.

**Предложение 2.2.1** Пусть  $N$  — временная сеть,  $Q_1, Q_2 \in States(N)$  и  $\kappa$  — равномерное отображение такое, что  $\kappa(Q_1) = Q_2$ . Тогда:

1.  $Q_1 \xrightarrow{t} \tilde{Q}_1$  влечет  $Q_2 \xrightarrow{t} \kappa(\tilde{Q}_1)$  для некоторого  $t \in T_N$ ;
2.  $Q_1 \xrightarrow{\delta} \tilde{Q}_1$  влечет  $Q_2 \xrightarrow{\kappa(\delta)} \kappa_\delta(\tilde{Q}_1)$  для некоторого  $\delta \in \mathbf{R}^+$ .

*Доказательство.* Пусть  $Q_1 = (M, \Upsilon)$ . Тогда  $Q_2 = (M, \kappa \circ \Upsilon)$ .

1. Пусть  $Q_1 \xrightarrow{t} \tilde{Q}_1$ , где  $\tilde{Q}_1 = (\tilde{M}, \tilde{\Upsilon})$ . Тогда  $\Upsilon(t) = 0$ . Так как  $\kappa(\Upsilon(t)) = \kappa(0) = 0$ , имеем  $Q_2 \xrightarrow{t} \tilde{Q}_2$ , где  $\tilde{Q}_2 = (\tilde{M}, \tilde{\Upsilon}_2)$  и

$$\forall u \in T_N \quad \tilde{\Upsilon}_2(u) \begin{cases} \Upsilon_N(u), & (\bullet u \subseteq \tilde{M}) \wedge (\bullet u \not\subseteq M); \\ \kappa(\Upsilon(u)), & \text{иначе.} \end{cases}$$

Так как  $\forall u \in T_N \quad \Upsilon_N(u) \in \mathbf{N}$ , то  $\Upsilon_N(u) = \kappa(\Upsilon_N(u))$ . Следовательно,  $\tilde{\Upsilon}_2 = \kappa \circ \tilde{\Upsilon}$  и  $\tilde{Q}_2 = \kappa(\tilde{Q}_1)$ .

2. Пусть  $Q_1 \xrightarrow{\delta} \tilde{Q}_1$ , где  $\tilde{Q}_1 = (\tilde{M}, \tilde{\Upsilon})$  и  $\tilde{\Upsilon} = \Upsilon \ominus \delta$ . Тогда  $\forall u \in T_N$  такое, что  $\bullet u \subseteq M$  справедливо:  $0 < \delta \leq \Upsilon(u)$ . Так как  $\kappa$  — строго монотонное отображение, то  $0 = \kappa(0) < \kappa(\delta) \leq \kappa(\Upsilon(u))$ , и имеем  $Q_2 \xrightarrow{\kappa(\delta)} \tilde{Q}_2$ , где  $\tilde{Q}_2 = (\tilde{M}, \tilde{\Upsilon}_2)$  и  $\tilde{\Upsilon}_2 = \kappa \circ \Upsilon \ominus \kappa(\delta)$ .

Пусть  $u \in T_N$ . Возможны следующие два случая.

- (а)  $\Upsilon(u) \geq \delta$ .

Тогда  $\tilde{\Upsilon}(u) \ominus \delta = \Upsilon(u) - \delta$ .

Так как  $\kappa$  — строго монотонное отображение, имеем  $\kappa(\Upsilon(u)) \geq \kappa(\delta)$  и  $\kappa(\Upsilon(u)) \ominus \kappa(\delta) = \kappa(\Upsilon(u) - \delta) = \kappa_\delta(\Upsilon(u))$ .

- (б)  $\Upsilon(u) < \delta$ .

Тогда  $\tilde{\Upsilon}(u) = \Upsilon(u) \ominus \delta = 0$ .

Так как  $\kappa$  — строго монотонное отображение, имеем  $\kappa(\Upsilon(u)) < \kappa(\delta)$  и  $\kappa(\Upsilon(u)) \ominus \kappa(\delta) = 0 = \kappa_\delta(0) = \kappa_\delta(\tilde{\Upsilon}(u))$ .

Следовательно,  $\tilde{\Upsilon}_2 = \kappa_\delta \circ \tilde{\Upsilon}$  и  $\tilde{Q}_2 = \kappa_\delta(\tilde{Q}_1)$ . □

Пусть для временных сетей  $N$  и  $N'$   $Q \in States(N)$ ,  $Q' \in States(N')$ . Пишем  $Q \equiv_t Q'$ , если  $\exists x_i \in Act \cup \mathbf{R}^+$  ( $1 \leq i \leq n$ ) такие, что  $Q_N \xrightarrow{x_1} Q_1 \xrightarrow{x_2} \dots \xrightarrow{x_n} Q_n = Q$  и  $Q_{N'} \xrightarrow{x_1} Q'_1 \xrightarrow{x_2} \dots \xrightarrow{x_n} Q'_n = Q'$ .

**Предложение 2.2.2** Пусть  $N$  и  $N'$  — временные сети,  $Q_1, Q_2 \in States(N)$ ,  $Q'_1, Q'_2 \in States(N')$ . Тогда  $Q_1 =_{reg} Q_2$  и  $Q'_1 =_{reg} Q'_2$  влечет:  $Q_1 \equiv_t Q'_1 \Leftrightarrow Q_2 \equiv_t Q'_2$ .

*Доказательство.* Как для временной бисимуляционной эквивалентности в предложении 2.2.3, но проще. □

Пусть для временных сетей  $N$  и  $N'$   $Q \in States(N)$ ,  $Q' \in States(N')$ . Пишем  $Q \leftrightarrow_t Q'$ , если  $\exists \mathcal{R} : N \leftrightarrow_t N'$  такое, что  $(Q, Q') \in \mathcal{R}$ .

**Предложение 2.2.3** Пусть  $N$  и  $N'$  — временные сети,  $Q_1, Q_2 \in States(N)$ ,  $Q'_1, Q'_2 \in States(N')$ . Тогда  $Q_1 =_{reg} Q_2$  и  $Q'_1 =_{reg} Q'_2$  влечет:  $Q_1 \leftrightarrow_t Q'_1 \Leftrightarrow Q_2 \leftrightarrow_t Q'_2$ .

*Доказательство.* ( $\Leftarrow$ ) Определим отношение  $\mathcal{R}$  следующим образом:  $\mathcal{R} = \{(Q_1, Q'_1) \mid \exists Q_2 \in States(N) \exists Q'_2 \in States(N') (Q_1 =_{reg} Q_2) \wedge (Q'_1 =_{reg} Q'_2) \wedge (Q_2 \leftrightarrow_t Q'_2)\}$ . Нам требуется доказать, что  $\mathcal{R}$  имеет “свойство переноса”, то есть взаимного моделирования. Так как это свойство симметрично, достаточно проверить только моделирование  $N$  посредством  $N'$ .

Пусть  $(Q_1, Q'_1) \in \mathcal{R}$ . Тогда  $\exists(Q_2, Q'_2)$  такое, что  $Q_1 =_{reg} Q_2$ ,  $Q'_1 =_{reg} Q'_2$  и  $Q_2 \leftrightarrow_t Q'_2$ . Следовательно, для некоторых равномерных отображений  $\kappa$  и  $\kappa'$  имеем  $Q_2 = \kappa(Q_1)$  и  $Q'_2 = \kappa'(Q'_1)$ .

1. Пусть  $Q_1 \xrightarrow{a} \tilde{Q}_1$  ( $a \in Act$ ). По предложению 2.2.1,  $Q_2 \xrightarrow{a} \kappa(\tilde{Q}_1)$ .

Так как  $Q_2 \leftrightarrow_t Q'_2$ , то  $\exists \tilde{Q}'_2 : Q'_2 \xrightarrow{a} \tilde{Q}'_2$  и  $\kappa(\tilde{Q}_1) \leftrightarrow_t \tilde{Q}'_2$ . Имеем  $Q' = (\kappa')^{-1}(Q'_2)$ . Так как отображение, обратное к равномерному, также равномерно, по предложению 2.2.1 имеем  $Q'_1 \xrightarrow{a} (\kappa')^{-1}(\tilde{Q}'_2)$  и  $\tilde{Q}_1 \leftrightarrow_t (\kappa')^{-1}(\tilde{Q}'_2)$ .

Так как  $\tilde{Q}_1 =_{reg} \kappa(\tilde{Q}_1)$  и  $(\kappa')^{-1}(\tilde{Q}'_2) =_{reg} \tilde{Q}'_2$ , имеем  $(\tilde{Q}_1, (\kappa')^{-1}(\tilde{Q}'_2)) \in \mathcal{R}$ .

2. Случай  $Q_1 \xrightarrow{\delta} \tilde{Q}_1$  ( $\delta \in \mathbf{R}^+$ ) рассматривается аналогично.

( $\Rightarrow$ ) Симметрично предыдущему пункту. □

Пусть  $Q = (M, \Upsilon)$  — состояние временной сети  $N$ . Определим

$$\zeta(Q) = \begin{cases} 1/2, & \forall t \in T_N \{\Upsilon(t)\} = 0; \\ \min\{\{\Upsilon(t)\} \mid (t \in T_N) \wedge (\{\Upsilon(t)\} \neq 0)\}, & \text{иначе.} \end{cases}$$

Пусть  $N$  — временная сеть и  $S, \tilde{S} \in RegStates(N)$ . Введем следующие обозначения.

- $S \xrightarrow{t} \tilde{S} \Leftrightarrow \exists Q \in S, \tilde{Q} \in \tilde{S} Q \xrightarrow{t} \tilde{Q}$ ;
- $S \xrightarrow{a} \tilde{S} \Leftrightarrow \exists t S \xrightarrow{t} \tilde{S}, l_N(t) = a$ ;
- $S \xrightarrow{\vee} \tilde{S} \Leftrightarrow \exists Q \in S, \tilde{Q} \in \tilde{S} Q \xrightarrow{\zeta(Q)} \tilde{Q}$ .

### Региональная следовая эквивалентность

**Определение 2.2.6** Региональный след временной сети  $N$  — последовательность  $x_1 \cdots x_n \in (Act \cup \{\sqrt{\cdot}\})^*$  такая, что  $S_N \xrightarrow{x_1} S_1 \xrightarrow{x_2} \dots \xrightarrow{x_n} S_n$ . Обозначим множество всех региональных следов временной сети  $N$  через  $RegTraces(N)$ . Временные сети  $N$  и  $N'$  регионально следово эквивалентны, запись  $N \equiv_r N'$ , если  $RegTraces(N) = RegTraces(N')$ .

### Региональная бисимуляционная эквивалентность

**Определение 2.2.7** Пусть  $N$  и  $N'$  — временные сети. Отношение  $\mathcal{R} \subseteq RegStates(N) \times RegStates(N')$  — региональная бисимуляция между  $N$  и  $N'$ , запись  $\mathcal{R} : N \leftrightarrow_r N'$ , если:

1.  $(S_N, S_{N'}) \in \mathcal{R}$ .
2.  $(S, S') \in \mathcal{R}, S \xrightarrow{x} \tilde{S} (x \in Act \cup \{\sqrt{\cdot}\}) \Rightarrow \exists \tilde{S}' : S' \xrightarrow{x} \tilde{S}', (\tilde{S}, \tilde{S}') \in \mathcal{R}$ .
3. Как пункт 2, но роли  $N$  и  $N'$  меняются.

Временные сети  $N$  и  $N'$  регионально бисимуляционно эквивалентны, запись  $N \leftrightarrow_r N'$ , если  $\exists \mathcal{R} : N \leftrightarrow_r N'$ .

### 2.2.4 Сравнение эквивалентностей

В этом разделе проводится сравнение всех рассмотренных эквивалентностей. В частности, доказывается совпадение региональных эквивалентностных понятий с временными, что дает возможность упрощения проверки последних за счет уменьшения количества рассматриваемых состояний.

**Лемма 2.2.2** Пусть для временной сети  $N$   $Q = (M, \Upsilon) \in States(N)$ . Тогда  $\forall u_1, u_2 \in T_N (\{\Upsilon(u_1)\} \neq 0) \wedge (\{\Upsilon(u_2)\} \neq 0) \Rightarrow \{\Upsilon(u_1)\} = \{\Upsilon(u_2)\}$ .

*Доказательство.* Индукция по отношению перехода на состояниях.

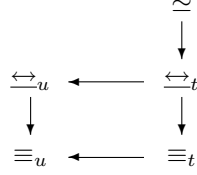


Рис. 2.2: Взаимосвязь эквивалентностей

1. Для  $Q_N = (M_N, \Upsilon_N)$  имеем  $\forall u \in T_N \{ \Upsilon_N(u) \} = 0$ .
2. Пусть  $Q = (M, \Upsilon) \in States(N)$ , и утверждение леммы выполняется для  $Q$ .
  - (a) Пусть  $\exists t \in T_N Q \xrightarrow{t} \tilde{Q}$ ,  $\tilde{Q} = (\tilde{M}, \tilde{\Upsilon})$ . Утверждение леммы выполняется также для  $\tilde{Q}$ , так как  $\forall u \in T_N (\tilde{\Upsilon}(u)\Upsilon(u)) \vee (\tilde{\Upsilon}(u) = \Upsilon_N(u))$  и  $\{ \Upsilon_N(u) \} = 0$ .
  - (b) Пусть  $\exists \delta \in \mathbf{R}^+ Q \xrightarrow{\delta} \tilde{Q}$ ,  $\tilde{Q} = (\tilde{M}, \tilde{\Upsilon})$ . Тогда  $\tilde{\Upsilon} = \Upsilon \ominus \delta$ . Рассмотрим  $u_1, u_2 \in T_N$  такие, что  $(\{\tilde{\Upsilon}(u_1)\} \neq 0) \wedge (\{\tilde{\Upsilon}(u_2)\} \neq 0)$ . Тогда  $\Upsilon(u_1) > \delta$  и  $\Upsilon(u_2) > \delta$ . Имеем  $\{ \Upsilon(u_1) \} = \{ \Upsilon(u_2) \}$  по индукционному предположению. Следовательно,  $\{ \tilde{\Upsilon}(u_1) \} = \{ \Upsilon(u_1) \ominus \delta \} \{ \Upsilon(u_1) - \delta \} = \{ 1 + \{ \Upsilon(u_1) \} - \{ \delta \} \} \{ 1 + \{ \Upsilon(u_2) \} - \{ \delta \} \} = \{ \Upsilon(u_2) - \delta \} \{ \Upsilon(u_2) \ominus \delta \} = \{ \tilde{\Upsilon}(u_2) \}$ .  $\square$

**Предложение 2.2.4** Для временных сетей  $N$  и  $N'$   $N \equiv_t N' \Leftrightarrow N \equiv_r N'$ .

*Доказательство.* Как для временной бисимуляционной эквивалентности в предложении 2.2.5, но проще.  $\square$

**Предложение 2.2.5** Для временных сетей  $N$  и  $N'$   $N \xleftrightarrow{t} N' \Leftrightarrow N \xleftrightarrow{r} N'$ .

*Доказательство.* ( $\Leftarrow$ ) Пусть  $\mathcal{R} : N \xleftrightarrow{r} N'$ . Определим отношение  $\mathcal{S}$  следующим образом:  $\mathcal{S} = \{ (Q, Q') \mid \exists (S, S') \in \mathcal{R} Q = (M, \Upsilon) \in S, Q' = (M', \Upsilon') \in S', \forall t \in T_N \forall t' \in T_{N'} (\{ \Upsilon(u_1) \} \neq 0) \wedge (\{ \Upsilon(u_2) \} \neq 0) \Rightarrow \{ \Upsilon(t) \} = \{ \Upsilon'(t') \} \}$ . Это определение корректно по лемме 2.2.2. Докажем  $\mathcal{S} : N \xleftrightarrow{t} N'$ .

1.  $(Q_N, Q_{N'}) \in \mathcal{S}$ , так как  $(S_N, S_{N'}) \in \mathcal{R}$ .
2. Пусть  $(Q, Q') \in \mathcal{S}$ . Очевидно, все срабатывания переходов  $N$  имитируются  $N'$ . Рассмотрим временные задержки. По определению региональной бисимуляционной эквивалентности имеем  $\forall (S, S') \in \mathcal{R} S \xrightarrow{\delta} \tilde{S} \Rightarrow \exists \tilde{S}' : S' \xrightarrow{\delta} \tilde{S}', (\tilde{S}, \tilde{S}') \in \mathcal{R}$ . Следовательно,  $\forall (Q, Q') \in \mathcal{S} Q \xrightarrow{\delta} \tilde{Q} \Rightarrow \exists \tilde{Q}' : Q' \xrightarrow{\delta} \tilde{Q}', (\tilde{Q}, \tilde{Q}') \in \mathcal{S}$ .

Пусть  $Q \xrightarrow{\delta} \tilde{Q}$ . Рассмотрим последовательность пар  $(Q_i, Q'_i)$  ( $i \geq 0$ ) такую, что:

- (a)  $Q_0 = Q, Q'_0 = Q'$ ;
- (b)  $Q_i \xrightarrow{\zeta(Q_i)} Q_{i+1}, Q'_i \xrightarrow{\zeta(Q_i)} Q'_{i+1}$ ;
- (c)  $(Q_i, Q'_i) \in \mathcal{S}$ .

Тогда  $\exists n \in \mathbf{N} \sum_{i=0}^n \zeta(Q_i) \geq \delta$ , и  $n$  — минимальное. Имеем  $\tilde{Q} \in [Q_n]$ . Тогда  $\exists \tilde{Q}' : \tilde{Q}' \in [Q'_n], Q' \xrightarrow{\delta} \tilde{Q}'$  и если  $\tilde{Q} = (\tilde{M}, \tilde{\Upsilon})$ ,  $\tilde{Q}' = (\tilde{M}', \tilde{\Upsilon}')$ , то  $\forall t \in T_N \forall t' \in T_{N'} \{ \Upsilon(t) \} = \{ \Upsilon'(t') \}$ . Так как  $([Q_n], [Q'_n]) \in \mathcal{R}$ , имеем  $(\tilde{Q}, \tilde{Q}') \in \mathcal{S}$ .

3. Как пункт 2, но роли  $N$  и  $N'$  меняются.

( $\Rightarrow$ ) По определениям.  $\square$

Следующая теорема устанавливает взаимосвязь эквивалентностных отношений.

**Теорема 2.2.1** Пусть  $\leftrightarrow, \leftrightarrow^* \in \{ \equiv, \leftrightarrow, \simeq \}$  и  $\star, \star\star \in \{ \_ , t, u \}$ . Для временных сетей  $N$  и  $N'$   $N \leftrightarrow_\star N' \Rightarrow N \leftrightarrow_{\star\star} N'$  тогда и только тогда, когда в графе на рисунке 2.2 существует направленный путь от  $\leftrightarrow_\star$  к  $\leftrightarrow_{\star\star}$ .

*Доказательство.* ( $\Leftarrow$ )



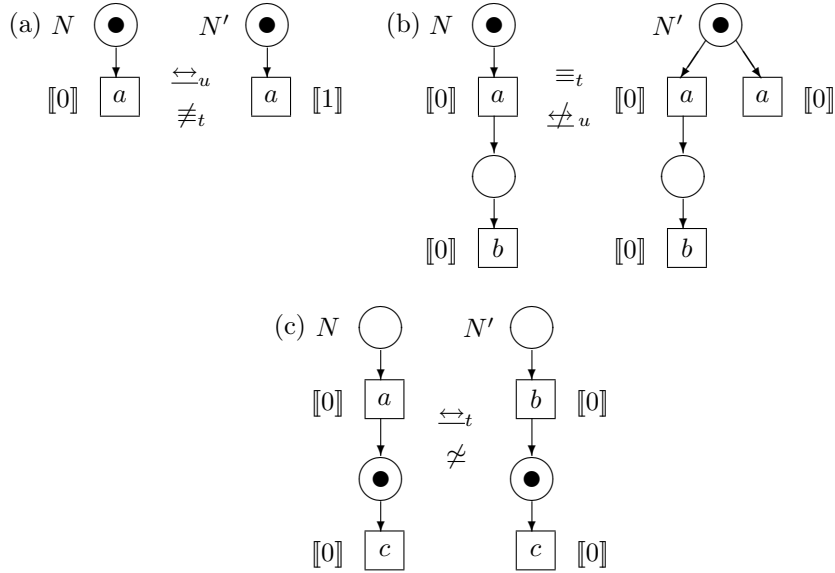


Рис. 2.3: Примеры эквивалентностей

- Связи  $\leftrightarrow_t \rightarrow \leftrightarrow_u$ ,  $\leftrightarrow \in \{\equiv, \leftrightarrow\}$ , существуют, так как в результате абстракции от времени получаются более слабые эквивалентности.
- Связи  $\leftrightarrow_\star \rightarrow \equiv_\star$ ,  $\star \in \{t, u\}$ , существуют, так как бисимуляционные эквивалентности влекут следовые.
- Связь  $\simeq \rightarrow \leftrightarrow_t$  очевидна.

( $\Rightarrow$ )

- На рисунке 2.3(a)  $N \leftrightarrow_u N'$  но  $N \not\equiv_t N'$ , так как только во временной сети  $N'$  один квант времени может пройти до срабатывания действия  $a$ .
- На рисунке 2.3(b)  $N \equiv_t N'$  но  $N \not\leftrightarrow_u N'$ , так как только во временной сети  $N'$  действие  $a$  выполнится так, что действие  $b$  не может произойти после него.
- На рисунке 2.3(c)  $N \leftrightarrow_t N'$  но  $N \not\equiv_u N'$ , так как никогда не срабатывающие переходы временных сетей  $N$  и  $N'$  помечены разными действиями ( $a$  и  $b$ ).  $\square$

## 2.2.5 Сохранение эквивалентностей при временных детализациях

В этом разделе мы вводим концепцию времени в известную операцию SM-детализации [46] и получаем новое понятие временной SM-детализации. Временная SM-детализация заменяет переходы временной сети, помеченные определенным действием, на временную SM-сеть, любое “полное” функционирование которой дает временную задержку, совпадающую с задержкой детализуемого перехода.

Идеи временной детализации изложены, например, в [93] в рамках временных процессных алгебр и в [136] — на временных сетях.

**Определение 2.2.8** Временная SM-сеть — временная сеть  $D = \langle P_D, T_D, F_D, l_D, M_D, \Upsilon_D \rangle$  такая, что:

1.  $\forall t \in T_D \ | \bullet t | = | t \bullet | = 1$ , то есть каждый переход имеет ровно одно входное и ровно одно выходное места;
2.  $\exists p_{in}, p_{out} \in P_D$  такое, что  $p_{in} \neq p_{out}$  и  ${}^\circ D = \{p_{in}\}$ ,  $D^\circ = \{p_{out}\}$ , то есть сеть  $D$  имеет единственное входное и единственное выходное места.
3.  $M_D = \{p_{in}\}$ , то есть сначала имеется единственная фишка в  $p_{in}$ .

4. для любых двух последовательностей  $(\{p_{in}\}, \Upsilon_D) \xrightarrow{t_1} (M_1, \Upsilon_1) \xrightarrow{t_2} \dots \xrightarrow{t_n} (M_n, \Upsilon_n) = (\{p_{out}\}, \Upsilon_n)$  и  $(\{p_{in}\}, \Upsilon_D) \xrightarrow{\bar{t}_1} (\bar{M}_1, \bar{\Upsilon}_1) \xrightarrow{\bar{t}_2} \dots \xrightarrow{\bar{t}_m} (\bar{M}_m, \bar{\Upsilon}_m) = (\{p_{out}\}, \bar{\Upsilon}_m)$  выполняется:  

$$\sum_{i=1}^n \Upsilon_D(t_i) = \sum_{j=1}^m \Upsilon_D(\bar{t}_j) = \Upsilon(D).$$

Действие  $a \in Act$  — конфликтное, если  $\exists Q = (M, \Upsilon) \in States(N) \exists t, u \in T_N$  такое, что  $\bullet t \subseteq M$ ,  $\bullet u \subseteq M$ ,  $\bullet t \cup \bullet u \not\subseteq M$ ,  $\Upsilon(t) = \Upsilon(u) = 0$  и  $l_N(t) = a$ .

Действие  $a \in Act$  — автопараллельное, если  $\exists Q = (M, \Upsilon) \in States(N) \exists t, u \in T_N$  такое, что  $\bullet t \cup \bullet u \subseteq M$ ,  $\Upsilon(t) = \Upsilon(u) = 0$  и  $l_N(t) = l_N(u) = a$ .

Временная SM-детализация не заменяет переходы, помеченные конфликтными или автопараллельными действиями, так как это может привести к проблемам, аналогичным рассмотренным в [94].

**Определение 2.2.9** Пусть  $N = \langle P_N, T_N, F_N, l_N, M_N, \Upsilon_N \rangle$  — временная сеть,  $a \in l_N(T_N)$  — ни конфликтно, ни автопараллельно и  $D = \langle P_D, T_D, F_D, l_D, M_D, \Upsilon_D \rangle$  — временная SM-сеть. Обозначим  $T = \{t \in T_N \mid (l_N(t) = a) \wedge (\Upsilon_N(t) = \Upsilon(D))\}$ . Временная SM-детализация, запись  $tref(N, a, D)$ , — это (с точностью до изоморфизма) временная сеть  $\bar{N} = \langle P_{\bar{N}}, T_{\bar{N}}, F_{\bar{N}}, l_{\bar{N}}, M_{\bar{N}}, \Upsilon_{\bar{N}} \rangle$ , где:

- $P_{\bar{N}} = P_N \cup \{ \langle p, u \rangle \mid p \in P_D \setminus \{p_{in}, p_{out}\}, u \in T \};$
- $T_{\bar{N}} = (T_N \setminus T) \cup \{ \langle t, u \rangle \mid t \in T_D, u \in T \};$
- $F_{\bar{N}}(\bar{x}, \bar{y}) \begin{cases} F_N(\bar{x}, \bar{y}), & \bar{x}, \bar{y} \in P_N \cup (T_N \setminus T); \\ F_D(x, y), & \bar{x} = \langle x, u \rangle, \bar{y} = \langle y, u \rangle, u \in T; \\ F_N(\bar{x}, u), & \bar{y} = \langle y, u \rangle, \bar{x} \in \bullet u, u \in T, y \in p_{in}^\bullet; \\ F_N(u, \bar{y}), & \bar{x} = \langle x, u \rangle, \bar{y} \in \bullet u, u \in T, x \in \bullet p_{out}; \\ 0, & \text{иначе;} \end{cases}$
- $l_{\bar{N}}(\bar{u}) \begin{cases} l_N(\bar{u}), & \bar{u} \in T_N \setminus T; \\ l_D(t), & \bar{u} = \langle t, u \rangle, t \in T_D, u \in T; \end{cases}$
- $M_{\bar{N}} = M_N;$
- $\Upsilon_{\bar{N}}(\bar{u}) \begin{cases} \Upsilon_N(\bar{u}), & \bar{u} \in T_N \setminus T; \\ \Upsilon_D(t), & \bar{u} = \langle t, u \rangle, t \in T_D, u \in T. \end{cases}$

**Пример 2.2.2** На рисунке 2.4 показано действие временной SM-детализации.

Следующие примеры поясняют необходимость учета времени в SM-детализации, а также необходимость заменять только переходы, не помеченные конфликтными или автопараллельными действиями.

**Пример 2.2.3** • На рисунке 2.5  $N \xleftrightarrow{t} N'$ , но  $ref(N, a, D) \not\equiv_u ref(N', a, D)$ , так как только во временной сети  $ref(N', a, D)$  может сработать действие  $b$ . Следовательно, эквивалентности от  $\equiv_u$  до  $\xleftrightarrow{t}$  (то есть все, кроме  $\simeq$ ) не сохраняются при SM-детализациях.

Причина состоит в том, что  $b$  не могло выполниться в  $N'$ , так как его всегда “опережало” действие  $a$ , то есть выбор был всегда в пользу  $a$ , а после детализации у всех переходов с пометкой  $a$  начальная временная задержка меняется с 0 на 1, и становится такой же, как и у  $b$ -помеченных переходов. Поэтому, операция детализации должна учитывать время и заменять переходы на временные SM-сети, имеющие ту же временную задержку. Таким образом, вместо  $ref$  мы должны использовать  $tref$ .

- На рисунке 2.6  $N \xleftrightarrow{t} N'$ , но  $tref(N, a, D) \not\equiv_u tref(N', a, D)$ , так как только во временной сети  $ref(N', a, D)$  невозможно выполнить последовательность действий  $a_1 b a_2$ . Следовательно, эквивалентности от  $\equiv_u$  до  $\xleftrightarrow{t}$  (то есть все, кроме  $\simeq$ ) не сохраняются при SM-детализациях.

Причина состоит в том, что переход  $N'$  с пометкой  $a$  участвует в конфликте, то есть  $a$  — конфликтное действие. Поэтому, операция временной SM-детализации не должна заменять переходы, помеченные конфликтными действиями.

- На рисунке 2.7  $N \xleftrightarrow{t} N'$ , но  $tref(N, a, D) \not\equiv_u tref(N', a, D)$ , так как только во временной сети  $ref(N', a, D)$  невозможно выполнить последовательность действий  $a_1 a_1$ . Следовательно, эквивалентности от  $\equiv_u$  до  $\xleftrightarrow{t}$  (то есть все, кроме  $\simeq$ ) не сохраняются при SM-детализациях.

Причина состоит в том, что во временной сети  $N$  из начального состояния допустимы два перехода с действием  $a$ , то есть  $a$  — автопараллельное действие. Поэтому, операция временной SM-детализации не должна заменять переходы, помеченные автопараллельными действиями.

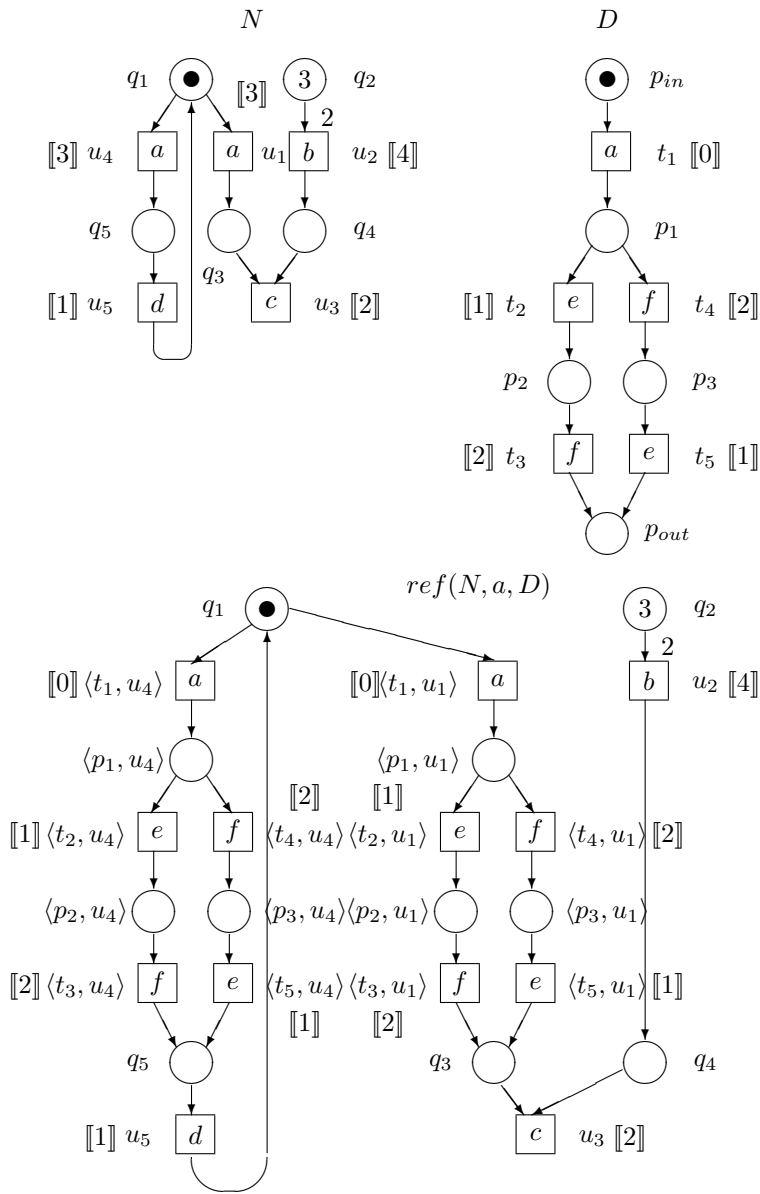


Рис. 2.4: Пример временной SM-детализации

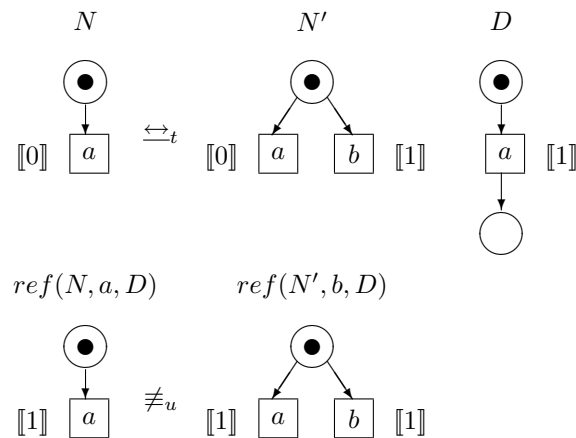


Рис. 2.5: Эквивалентности от  $\equiv_u$  до  $\leftrightarrow_t$  не сохраняются при SM-детализациях

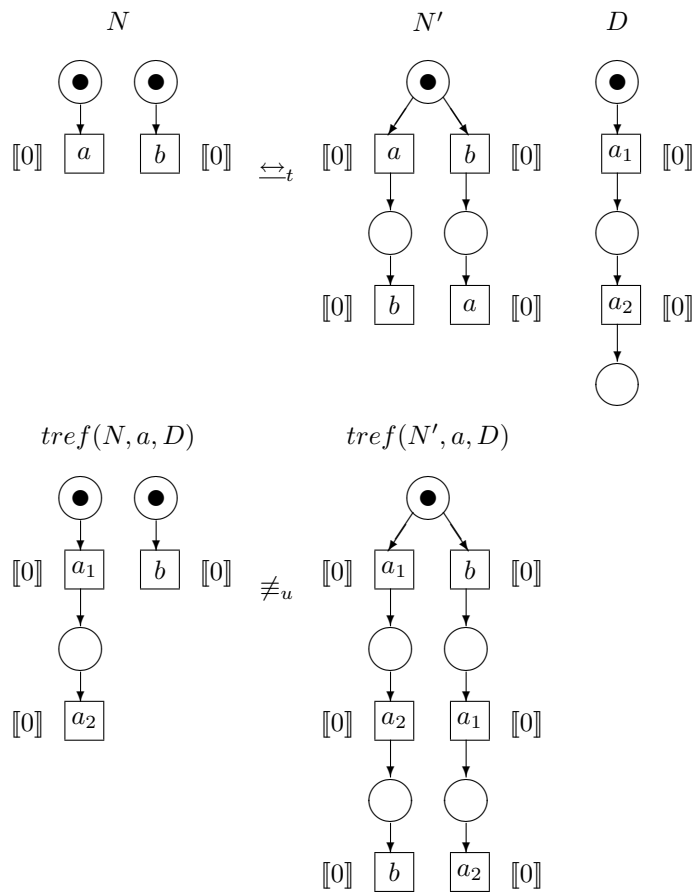


Рис. 2.6: Эквивалентности от  $\equiv_u$  до  $\leftrightarrow_t$  не сохраняются при временных SM-детализациях (конфликт)

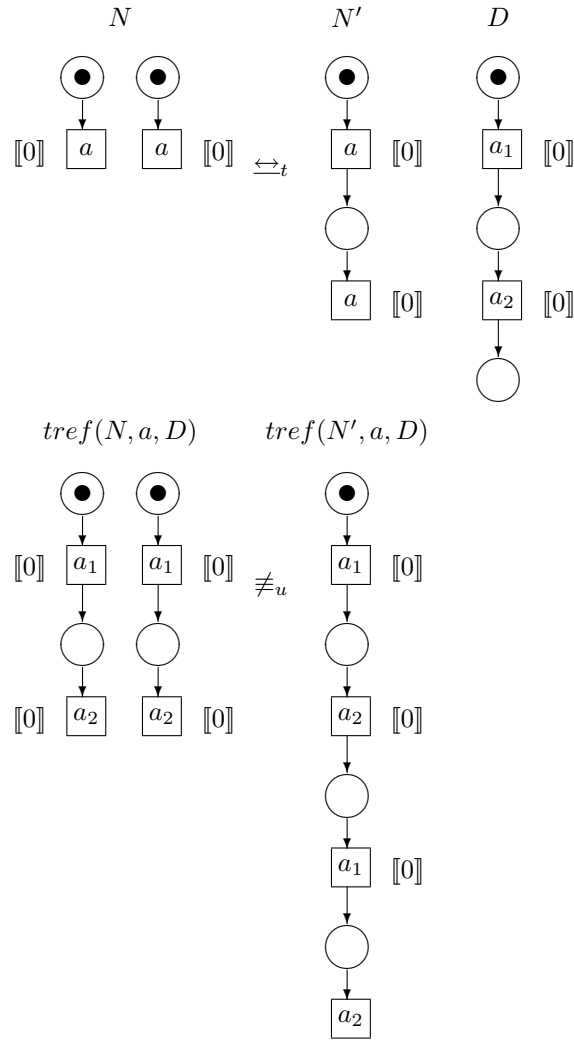


Рис. 2.7: Эквивалентности от  $\equiv_u$  до  $\xleftrightarrow{t}$  не сохраняются при временных SM-детализациях (автопараллелизм)

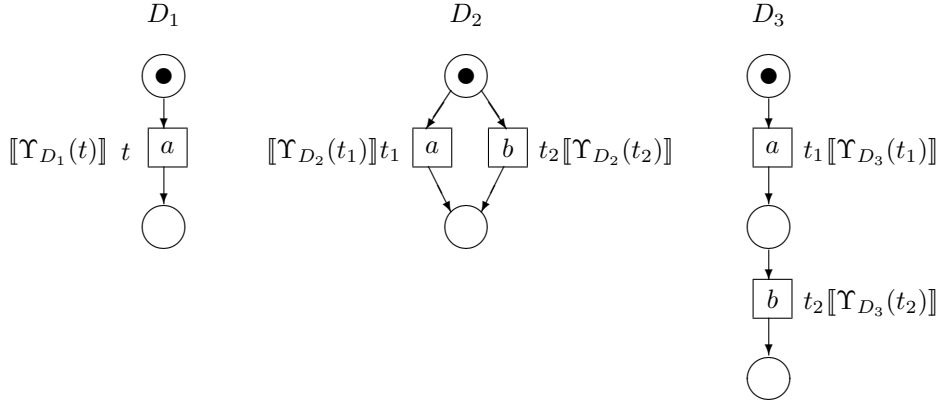


Рис. 2.8: Временные SM-сети переименования, простого выбора и простого расщепления

Заметим, что каждая SM-сеть может быть скомбинирована из элементарных SM-сетей (состоящих из одного перехода с единственными входным и выходным местами) с помощью операций альтернативы (выбора) и последовательной композиции описанных, например, в [161, 98, 43].

Следовательно, детализацию на произвольную временную SM-сеть можно заменить на последовательность *простых* временных SM-детализаций: *переименование*, *простой выбор* и *простое расщепление* которые заменяют переходы соответственно на временные SM-сети  $D_1$ ,  $D_2$  и  $D_3$ , изображенные на рисунке 2.8. При этом требование пункта 4 определения произвольной временной SM-сети преобразуется в следующие более простые условия.

- Для  $D_1$ :  $\Upsilon_{D_1}(t) = \Upsilon(D_1)$ .
- Для  $D_2$ :  $\Upsilon_{D_2}(t_1) = \Upsilon_{D_2}(t_2) = \Upsilon(D_2)$ .
- Для  $D_3$ :  $\Upsilon_{D_3}(t_1) + \Upsilon_{D_3}(t_2) = \Upsilon(D_3)$ .

Следующее предложение демонстрирует, что все рассмотренные следовые временные эквивалентности, а также все не-временные эквивалентности не сохраняются при временных SM-детализациях.

**Предложение 2.2.6** *Эквивалентности  $\equiv_t$ ,  $\equiv_u$  и  $\leftrightarrow_u$  не сохраняются при временных SM-детализациях.*

*Доказательство.*

- На рисунке 2.9  $N \equiv_t N'$ , но  $tref(N, b, D) \not\equiv_u tref(N', b, D)$ , так как только во временной сети  $tref(N', b, D)$  последовательность действий  $ab_1c$  не может выполняться. Следовательно, эквивалентности от  $\equiv_u$  до  $\equiv_t$  не сохраняются при временных SM-детализациях.
- На рисунке 2.10  $N \leftrightarrow_u N'$ , но  $tref(N, a, D) \not\equiv_u tref(N', a, D)$ , так как только во временной сети  $tref(N', a, D)$  действие  $b$  может выполняться. Следовательно, эквивалентности от  $\equiv_u$  до  $\leftrightarrow_u$  не сохраняются при временных SM-детализациях.

На рисунке 2.11 линиями обведены эквивалентности, не сохраняющиеся при временных SM-детализациях, что демонстрируется примерами на рисунках 2.9–2.10.  $\square$

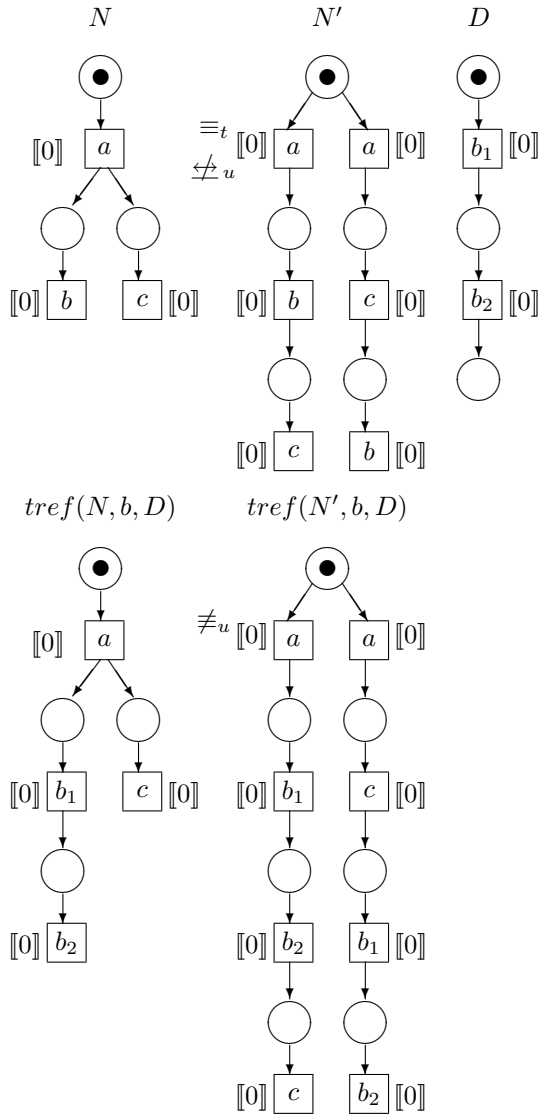


Рис. 2.9: Эквивалентности от  $\equiv_u$  до  $\equiv_t$  не сохраняются при временных SM-детализациях

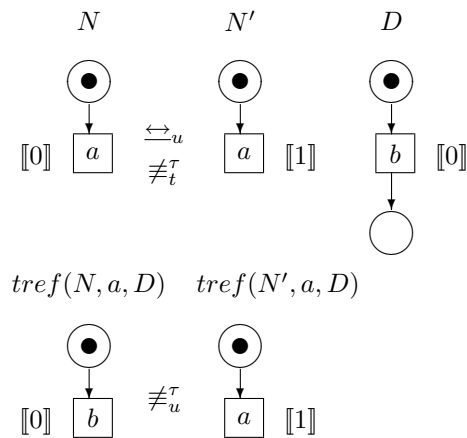


Рис. 2.10: Эквивалентности от  $\equiv_u$  до  $\leftrightarrow_u$  не сохраняются при временных SM-детализациях

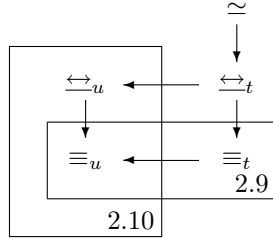


Рис. 2.11: Эквивалентности, не сохраняющиеся при временных SM-детализациях

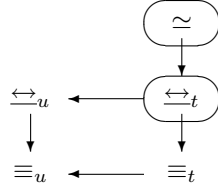


Рис. 2.12: Сохранение эквивалентностей при временных SM-детализациях

Следующая теорема демонстрирует, какие из рассмотренных эквивалентностей сохраняются при временных SM-детализациях.

**Теорема 2.2.2** Пусть  $\leftrightarrow \in \{\equiv, \leftrightarrow, \simeq\}$  и  $\star \in \{_, t, u\}$ . Для временных сетей  $N, N'$  таких, что  $a \in l_N(T_N) \cap l_{N'}(T_{N'})$  и временной SM-сети  $D$   $N \leftrightarrow_\star N' \Rightarrow \text{tref}(N, a, D) \leftrightarrow_\star \text{tref}(N', a, D)$  тогда и только тогда, когда эквивалентность  $\leftrightarrow_\star$  заключена в овал на рисунке 2.12.

*Доказательство.* Доказательство основано на факте, что в случае не-моделирования одной детализованной временной сети другой, одна из исходных сетей имеет конфликтные или автопараллельные действия, и мы входим в противоречие с определением временной SM-детализации.  $\square$

Заметим, что результат теоремы справедлив и для операции модификации временной SM-детализации, позволяющей заменять переходы с конфликтными и автопараллельными действиями на временные SM-сети определенного вида: *n-выбор*. Пример такой сети приведен на рисунке 2.13. Условие на начальные временные задержки переходов следующее:  $\forall i (1 \leq i \leq n) \Upsilon_D(t_i) = \Upsilon(D)$ .

## 2.2.6 Эквивалентности на не-временных сетях Петри

**Определение 2.2.10** Не-временная сеть — временная сеть  $\langle P_N, T_N, F_N, l_N, M_N, \Upsilon_N \rangle$  такая, что  $\forall t \in T_N \Upsilon_N(t) = 0$ .

Для не-временных сетей следующее предложение демонстрирует совпадение временных и не-временных эквивалентностей.

**Предложение 2.2.7** Пусть  $\leftrightarrow \in \{\equiv, \leftrightarrow\}$ . Для не-временных сетей  $N$  и  $N'$   $N \leftrightarrow_u N' \Leftrightarrow N \leftrightarrow_t N'$ .

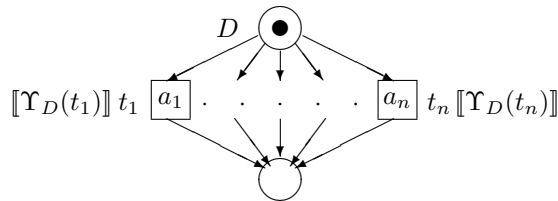


Рис. 2.13: Временная SM-сеть *n*-выбора



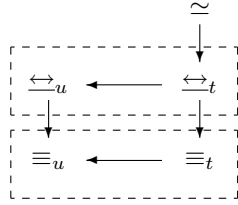


Рис. 2.14: Совпадение эквивалентностей на не-временных сетях

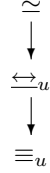


Рис. 2.15: Взаимосвязь эквивалентностей на не-временных сетях

*Доказательство.* Очевидно, так как в не-временных сетях нет временных задержек. □

На рисунке 2.14 линиями обведены эквивалентности, совпадающие на не-временных сетях.

**Теорема 2.2.3** Пусть  $\leftrightarrow, \leftrightarrow \in \{\equiv_u, \leftrightarrow_u, \simeq\}$ . Для не-временных сетей  $N$  и  $N'$   $N \leftrightarrow N' \Rightarrow N \leftrightarrow N'$  тогда и только тогда, когда в графе на рисунке 2.15 существует направленный путь от  $\leftrightarrow$  к  $\leftrightarrow$ .

*Доказательство.* ( $\Leftarrow$ ) По теореме 2.2.1.

( $\Rightarrow$ )

- На рисунке 2.3(b)  $N \equiv_u N'$  но  $N \not\leftrightarrow_u N'$ .
- На рисунке 2.3(c)  $N \leftrightarrow_u N'$  но  $N \not\equiv N'$ . □

## 2.3 Временные сети Петри с невидимыми переходами

Рассмотрим эквивалентности для временных сетей Петри с невидимыми переходами, моделирующих системы, некоторые действия которых считаются невидимыми для внешнего наблюдателя.

**Определение 2.3.1** Временная сеть с невидимыми переходами — *временная сеть*  $N = \langle P_N, T_N, F_N, l_N, M_N, \Upsilon_N \rangle$ , такая, что  $l_N : T_N \rightarrow Act_\tau$  и  $\forall t \in T_N (l_N(t) = \tau \Rightarrow \Upsilon_N(t) = 0)$  (то есть некоторые переходы временной сети помечены невидимым действием  $\tau$ , и такие переходы не имеют временной задержки).

Следующие базисные  $\tau$ -эквивалентности для временных формальных моделей с невидимыми действиями были рассмотрены в литературе.

- *Временные  $\tau$ -эквивалентности* (учитывают временные задержки в функционировании систем и абстрагируются от невидимых действий).
  - Временная  $\tau$ -бисимуляционная эквивалентность ( $\leftrightarrow_t^\tau$ ) введена в [92] на параллельных таймерных процессах.

Мы вводим следующие новые отношения: временная  $\tau$ -следовая эквивалентность ( $\equiv_t^\tau$ ), не-временная  $\tau$ -следовая эквивалентность ( $\equiv_u^\tau$ ) и не-временная  $\tau$ -бисимуляционная эквивалентность ( $\leftrightarrow_u^\tau$ ).

Мы исследуем также региональные  $\tau$ -эквивалентности. Региональная  $\tau$ -бисимуляционная эквивалентность ( $\leftrightarrow_r^\tau$ ) была рассмотрена [92] на параллельных таймерных процессах. Мы вводим новое понятие региональной  $\tau$ -следовой эквивалентности ( $\equiv_r^\tau$ ).

Проводится сравнение всех  $\tau$ -эквивалентностных понятий. Доказывается совпадение региональных и временных  $\tau$ -эквивалентностей.

$\tau$ -эквивалентности проверяются на сохранение при новой операции временной SM-детализации.

Проводится сравнение всех  $\tau$ -эквивалентностных отношений на не-временных сетях.

### 2.3.1 Временные $\tau$ -эквивалентности

Пусть  $N$  — временная сеть с невидимыми переходами и  $Q, \tilde{Q} \in States(N)$ . Введем следующие обозначения.

- $Q \Rightarrow \tilde{Q} \Leftrightarrow \exists Q_i (0 \leq i \leq n) Q = Q_0 \xrightarrow{\tau} Q_1 \xrightarrow{\tau} \dots \xrightarrow{\tau} Q_n \tilde{Q}$ ;
- $Q \xrightarrow{t} \tilde{Q} \Leftrightarrow \exists Q_1, Q_2 Q \Rightarrow Q_1 \xrightarrow{t} Q_2 \Rightarrow \tilde{Q}$ ;
- $Q \xrightarrow{a} \tilde{Q} \Leftrightarrow \exists t Q \xrightarrow{t} \tilde{Q}, l_N(t) = a$ ;
- $Q \xrightarrow{\delta} \tilde{Q} \Leftrightarrow \exists Q_1 Q \Rightarrow Q_1 \xrightarrow{\delta} \tilde{Q}$ .

Будем считать, что отношение  $\xrightarrow{\delta}$  обладает следующим свойством аддитивности:  $\exists \delta_1, \delta_2, Q_1 Q \xrightarrow{\delta_1} Q_1 \xrightarrow{\delta_2} \tilde{Q}$  и  $\delta_1 + \delta_2 = \delta$  влечет  $Q \xrightarrow{\delta} \tilde{Q}$ .

#### Временная $\tau$ -следовая эквивалентность

**Определение 2.3.2** Видимый временной след временной сети с невидимыми переходами  $N$  — последовательность  $x_1 \dots x_n \in (Act \cup \mathbf{R}^+)^*$  такая, что  $Q_N \xrightarrow{x_1} Q_1 \xrightarrow{x_2} \dots \xrightarrow{x_n} Q_n$ . Обозначим множество всех видимых временных следов временной сети с невидимыми переходами  $N$  через  $VisTimeTraces(N)$ . Временные сети с невидимыми переходами  $N$  и  $N'$  временно  $\tau$ -следово эквивалентны, запись  $N \equiv_t^\tau N'$ , если  $VisTimeTraces(N) = VisTimeTraces(N')$ .

#### Временная $\tau$ -бисимуляционная эквивалентность

**Определение 2.3.3** Пусть  $N$  и  $N'$  — временные сети с невидимыми переходами. Отношение  $\mathcal{R} \subseteq States(N) \times States(N')$  — временная  $\tau$ -бисимуляция между  $N$  и  $N'$ , запись  $\mathcal{R} : N \leftrightarrow_t^\tau N'$ , если:

1.  $(Q_N, Q_{N'}) \in \mathcal{R}$ .
2.  $(Q, Q') \in \mathcal{R}, Q \xrightarrow{x} \tilde{Q} (x \in Act \cup \mathbf{R}^+ \cup \{_\perp\}) \Rightarrow \exists \tilde{Q}' : Q' \xrightarrow{x} \tilde{Q}', (\tilde{Q}, \tilde{Q}') \in \mathcal{R}$ .
3. Как пункт 2, но роли  $N$  и  $N'$  меняются.

Временные сети с невидимыми переходами  $N$  и  $N'$  временно  $\tau$ -бисимуляционно эквивалентны, запись  $N \leftrightarrow_t^\tau N'$ , если  $\exists \mathcal{R} : N \leftrightarrow_t^\tau N'$ .

### 2.3.2 Не-временные $\tau$ -эквивалентности

Пусть  $N$  — временная сеть с невидимыми переходами и  $Q, \tilde{Q} \in States(N)$ . Введем следующие обозначения.

- $Q \Vdash \tilde{Q} \Leftrightarrow \exists \delta Q \xrightarrow{\delta} \tilde{Q}$ ;
- $Q \xrightarrow{t} \tilde{Q} \Leftrightarrow \exists Q_1, Q_2 Q \Vdash Q_1 \xrightarrow{t} Q_2 \Vdash \tilde{Q}$ ;
- $Q \xrightarrow{a} \tilde{Q} \Leftrightarrow \exists t Q \xrightarrow{t} \tilde{Q}, l_N(t) = a$ .

#### Не-временная $\tau$ -следовая эквивалентность

**Определение 2.3.4** Видимый не-временной след временной сети с невидимыми переходами  $N$  — последовательность  $a_1 \dots a_n \in Act^*$  такая, что  $Q_N \xrightarrow{a_1} Q_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} Q_n$ . Обозначим множество всех видимых не-временных следов временной сети с невидимыми переходами  $N$  через  $VisUntimeTraces(N)$ . Временные сети с невидимыми переходами  $N$  и  $N'$  не-временно  $\tau$ -следово эквивалентны, запись  $N \equiv_u^\tau N'$ , если  $VisUntimeTraces(N) = VisUntimeTraces(N')$ .

## Не-временная $\tau$ -бисимуляционная эквивалентность

**Определение 2.3.5** Пусть  $N$  и  $N'$  – временные сети с невидимыми переходами. Отношение  $\mathcal{R} \subseteq States(N) \times States(N')$  – не-временная  $\tau$ -бисимуляция между  $N$  и  $N'$ , запись  $\mathcal{R} : N \stackrel{\tau}{\sim}_u N'$ , если:

1.  $(Q_N, Q_{N'}) \in \mathcal{R}$ .
2.  $(Q, Q') \in \mathcal{R}$ ,  $Q \stackrel{a}{\mapsto} \tilde{Q}$  ( $a \in Act \cup \{\_ \}$ )  $\Rightarrow \exists \tilde{Q}' : Q' \stackrel{a}{\mapsto} \tilde{Q}'$ ,  $(\tilde{Q}, \tilde{Q}') \in \mathcal{R}$ .
3. Как пункт 2, но роли  $N$  и  $N'$  меняются.

Временные сети с невидимыми переходами  $N$  и  $N'$  не-временно  $\tau$ -бисимуляционно эквивалентны, запись  $N \stackrel{\tau}{\sim}_u N'$ , если  $\exists \mathcal{R} : N \stackrel{\tau}{\sim}_u N'$ .

### 2.3.3 Региональные $\tau$ -эквивалентности

**Предложение 2.3.1** Пусть  $N$  – временная сеть с невидимыми переходами,  $Q_1, Q_2 \in States(N)$  и  $\kappa$  – равномерное отображение такое, что  $\kappa(Q_1) = Q_2$ . Тогда:

1.  $Q_1 \stackrel{t}{\Rightarrow} \tilde{Q}_1$  влечет  $Q_2 \stackrel{t}{\Rightarrow} \kappa(\tilde{Q}_1)$  для некоторого  $t \in T_N$ ;
2.  $Q_1 \stackrel{\delta}{\Rightarrow} \tilde{Q}_1$  влечет  $Q_2 \stackrel{\kappa(\delta)}{\Rightarrow} \kappa_\delta(\tilde{Q}_1)$  для некоторого  $\delta \in \mathbf{R}^+$ .

*Доказательство.*

1. Случай  $Q_1 \stackrel{t}{\Rightarrow} \tilde{Q}_1$  рассматривается как в предложении 2.2.1, с использованием индукции по промежуточным состояниям.
2. Случай  $Q_1 \stackrel{\delta}{\Rightarrow} \tilde{Q}_1$  рассматривается как в предложении 2.2.1, с использованием индукции по промежуточным состояниям и следующего замечания:  $(\kappa_{\delta_1})_{\delta_2}(x) = \kappa_{\delta_1}(x + \delta_2) - \kappa_{\delta_1}(\delta_2)\kappa(x + (\delta_1 + \delta_2)) - \kappa(\delta_1) - (\kappa(\delta_1 + \delta_2) - \kappa(\delta_1))\kappa(x + (\delta_1 + \delta_2)) - \kappa(\delta_1 + \delta_2)\kappa_{(\delta_1 + \delta_2)}(x)$ .  $\square$

Пусть для временных сетей с невидимыми переходами  $N$  и  $N' Q \in States(N)$ ,  $Q' \in States(N')$ . Пишем  $Q \equiv_t^\tau Q'$ , если  $\exists x_i \in Act \cup \mathbf{R}^+$  ( $1 \leq i \leq n$ ) такие, что  $Q_N \xrightarrow{x_1} Q_1 \xrightarrow{x_2} \dots \xrightarrow{x_n} Q_n = Q$  и  $Q_{N'} \xrightarrow{x_1} Q'_1 \xrightarrow{x_2} \dots \xrightarrow{x_n} Q'_n = Q'$ .

**Предложение 2.3.2** Пусть  $N$  и  $N'$  – временные сети с невидимыми переходами,  $Q_1, Q_2 \in States(N)$ ,  $Q'_1, Q'_2 \in States(N')$ . Тогда  $Q_1 =_{reg} Q_2$  и  $Q'_1 =_{reg} Q'_2$  влечет:  $Q_1 \equiv_t^\tau Q'_1 \Leftrightarrow Q_2 \equiv_t^\tau Q'_2$ .

*Доказательство.* Аналогично предложению 2.2.2.  $\square$

Пусть для некоторых временных сетей с невидимыми переходами  $N$  и  $N' Q \in States(N)$ ,  $Q' \in States(N')$ . Пишем  $Q \stackrel{\tau}{\sim}_t Q'$ , если  $\exists \mathcal{R} : N \stackrel{\tau}{\sim}_t N'$  такое, что  $(Q, Q') \in \mathcal{R}$ .

**Предложение 2.3.3** Пусть  $N$  и  $N'$  – временная сетис невидимыми переходами,  $Q \in States(N)$ ,  $Q' \in States(N')$ . Тогда  $Q_1 =_{reg} Q_2$  и  $Q'_1 =_{reg} Q'_2$  влечет:  $Q_1 \stackrel{\tau}{\sim}_t Q'_1 \Leftrightarrow Q_2 \stackrel{\tau}{\sim}_t Q'_2$ .

*Доказательство.* Аналогично предложению 2.2.3.  $\square$

Пусть  $N$  – временная сеть с невидимыми переходами и  $S, \tilde{S} \in RegStates(N)$ . Введем следующие обозначения.

- $S \Rightarrow \tilde{S} \Leftrightarrow \exists Q \in S, \tilde{Q} \in \tilde{S} Q \Rightarrow \tilde{Q}$ ;
- $S \stackrel{t}{\Rightarrow} \tilde{S} \Leftrightarrow \exists Q \in S, \tilde{Q} \in \tilde{S} Q \stackrel{t}{\Rightarrow} \tilde{Q}$ ;
- $S \stackrel{a}{\Rightarrow} \tilde{S} \Leftrightarrow \exists t S \stackrel{t}{\Rightarrow} \tilde{S}, l_N(t) = a$ ;
- $S \stackrel{\forall}{\Rightarrow} \tilde{S} \Leftrightarrow \exists Q \in S, \tilde{Q} \in \tilde{S} Q \stackrel{\zeta(Q)}{\Rightarrow} \tilde{Q}$ .

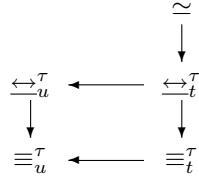


Рис. 2.16: Взаимосвязь  $\tau$ -эквивалентностей

### Региональная $\tau$ -следовая эквивалентность

**Определение 2.3.6** Видимый региональный след временной сети с невидимыми переходами  $N$  — последовательность  $x_1 \cdots x_n \in (Act \cup \{\sqrt{\cdot}\})^*$  такая, что  $S_N \xrightarrow{x_1} S_1 \xrightarrow{x_2} \dots \xrightarrow{x_n} S_n$ . Обозначим множество всех видимых региональных следов временной сети с невидимыми переходами  $N$  через  $VisRegTraces(N)$ . Временные сети  $N$  и  $N'$  регионально  $\tau$ -следово эквивалентны, запись  $N \equiv_r^\tau N'$ , если  $VisRegTraces(N) = VisRegTraces(N')$ .

### Региональная $\tau$ -бисимуляционная эквивалентность

**Определение 2.3.7** Пусть  $N$  и  $N'$  — временные сети с невидимыми переходами. Отношение  $\mathcal{R} \subseteq RegStates(N) \times RegStates(N')$  — региональная  $\tau$ -бисимуляция между  $N$  и  $N'$ , запись  $\mathcal{R} : N \leftrightarrow_r^\tau N'$ , если:

1.  $(S_N, S_{N'}) \in \mathcal{R}$ .
2.  $(S, S') \in \mathcal{R} \ S \xrightarrow{x} \tilde{S} \ (x \in Act \cup \{\_ \} \cup \{\sqrt{\cdot}\}) \Rightarrow \exists \tilde{S}' : S' \xrightarrow{x} \tilde{S}' \ , (\tilde{S}, \tilde{S}') \in \mathcal{R}$ .
3. Как пункт 2, но роли  $N$  и  $N'$  меняются.

Временные сети с невидимыми переходами  $N$  и  $N'$  регионально  $\tau$ -бисимуляционно эквивалентны, запись  $N \leftrightarrow_r^\tau N'$ , если  $\exists \mathcal{R} : N \leftrightarrow_r^\tau N'$ .

### 2.3.4 Сравнение $\tau$ -эквивалентностей

В этом разделе проводится сравнение всех рассмотренных  $\tau$ -эквивалентностей. В частности, доказывается совпадение региональных  $\tau$ -эквивалентностных понятий с временными, что дает возможность упрощения проверки последних за счет уменьшения количества рассматриваемых состояний.

**Предложение 2.3.4** Для временных сетей с невидимыми переходами  $N$  и  $N'$   $N \equiv_t^\tau N' \Leftrightarrow N \equiv_r^\tau N'$ .

*Доказательство.* Как для временной бисимуляционной  $\tau$ -эквивалентности в предложении 2.3.5, но проще.  $\square$

**Предложение 2.3.5** Для временных сетей  $N$  и  $N'$   $N \leftrightarrow_t^\tau N' \Leftrightarrow N \leftrightarrow_r^\tau N'$ .

*Доказательство.* Аналогично предложению 2.2.5.  $\square$

Следующая теорема устанавливает взаимосвязи между  $\tau$ -эквивалентностными отношениями

**Теорема 2.3.1** Пусть  $\leftrightarrow, \Leftrightarrow \in \{\equiv^\tau, \leftrightarrow^\tau, \simeq\}$  и  $\star, \star\star \in \{\_, t, u\}$ . Для временных сетей с невидимыми переходами  $N$  и  $N'$   $N \leftrightarrow_\star N' \Rightarrow N \Leftrightarrow_{\star\star} N'$  тогда и только тогда, когда в графе на рисунке 2.16 существует направленный путь от  $\leftrightarrow_\star$  к  $\Leftrightarrow_{\star\star}$ .

*Доказательство.* ( $\Leftarrow$ )

- Связи  $\leftrightarrow_t^\tau \rightarrow \leftrightarrow_u^\tau$ ,  $\leftrightarrow \in \{\equiv, \leftrightarrow\}$ , существуют, в результате абстракции от времени получают более слабые эквивалентности.
- Связи  $\leftrightarrow_\star^\tau \rightarrow \equiv_\star^\tau$ ,  $\star \in \{t, u\}$ , существуют, так как  $\tau$ -бисимуляционные эквивалентности влекут  $\tau$ -следовые.
- Связь  $\simeq \rightarrow \leftrightarrow_t^\tau$  очевидна.

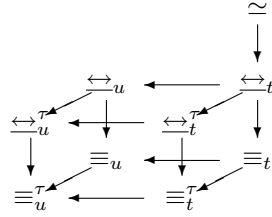


Рис. 2.17: Взаимосвязь  $\tau$ -эквивалентностей с эквивалентностями

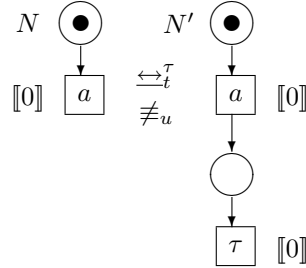


Рис. 2.18: Пример взаимосвязи эквивалентностей и  $\tau$ -эквивалентностей

( $\Rightarrow$ )

- На рисунке 2.3(a)  $N \leftrightarrow_u^\tau N'$  но  $N \not\equiv_t^\tau N'$ , так как только во временной сети  $N'$  один квант времени может пройти до срабатывания действия  $a$ .
- На рисунке 2.3(b)  $N \equiv_t^\tau N'$  но  $N \not\leftrightarrow_u^\tau N'$ , так как только во временной сети  $N'$  действие  $a$  выполнится так, что действие  $b$  не может произойти после него.
- На рисунке 2.3(c)  $N \leftrightarrow_t^\tau N'$  но  $N \not\equiv_u^\tau N'$ . □

### 2.3.5 Сравнение эквивалентностей и $\tau$ -эквивалентностей

**Теорема 2.3.2** Пусть  $\leftrightarrow, \leftrightarrow^* \in \{\equiv^\tau, \leftrightarrow^\tau, \simeq\}$  и  $\star, \star\star \in \{_, t, u\}$ . Для временных сетей с невидимыми переходами  $N$  и  $N'$   $N \leftrightarrow_\star N' \Rightarrow N \leftrightarrow_{\star\star}^\tau N'$  тогда и только тогда, когда в графе на рисунке 2.17 существует направленный путь от  $\leftrightarrow_\star$  к  $\leftrightarrow_{\star\star}$ .

*Доказательство.* ( $\Leftarrow$ ) По теоремам 2.2.1 и 2.3.1, с использованием следующих рассуждений.

- Связи  $\leftrightarrow \rightarrow \leftrightarrow^\tau$ ,  $\leftrightarrow \in \{\equiv_t^\tau, \equiv_u^\tau, \leftrightarrow_t^\tau, \leftrightarrow_u^\tau\}$ , существуют, так как в результате абстракции от невидимых действий получаются более слабые эквивалентности.

( $\Rightarrow$ )

- На рисунке 2.18  $N \leftrightarrow_t^\tau N'$  но  $N \not\equiv_u^\tau N'$ , так как только во временной сети с невидимыми переходами  $N'$  может выполниться действие  $\tau$ . □

### 2.3.6 Сохранение $\tau$ -эквивалентностей при временных детализациях

Следующее предложение демонстрирует, что все рассмотренные  $\tau$ -следовые временные эквивалентности, а также все не-временные  $\tau$ -эквивалентности не сохраняются при временных SM-детализациях.

**Предложение 2.3.6** Эквивалентности  $\equiv_t^\tau, \equiv_u^\tau, \leftrightarrow_u^\tau$  не сохраняются при временных SM-детализациях.

*Доказательство.*

- На рисунке 2.9  $N \equiv_t^\tau N'$ , но  $\text{tref}(N, b, D) \not\equiv_u^\tau \text{tref}(N', b, D)$ , так как только во временной сети с невидимыми переходами  $\text{tref}(N', b, D)$  последовательность действий  $ab_1c$  не может выполниться. Следовательно, эквивалентности от  $\equiv_u^\tau$  до  $\equiv_t^\tau$  не сохраняются при временных SM-детализациях.

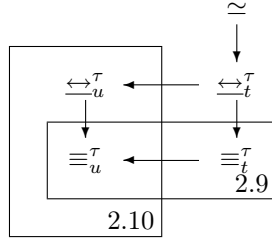


Рис. 2.19:  $\tau$ -эквивалентности, не сохраняющиеся при временных SM-детализациях

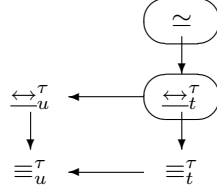


Рис. 2.20: Сохранение  $\tau$ -эквивалентностей при временных SM-детализациях

- На рисунке 2.10  $N \xleftrightarrow{\tau} N'$ , но  $\text{tref}(N, a, D) \not\equiv_u^{\tau} \text{tref}(N', a, D)$ , так как только во временной сети с невидимыми переходами  $\text{tref}(N', a, D)$  действие  $b$  может выполняться. Следовательно, эквивалентности от  $\equiv_u^{\tau}$  до  $\xleftrightarrow{\tau}$  не сохраняются при временных SM-детализациях.

На рисунке 2.19 линиями обведены  $\tau$ -эквивалентности, не сохраняющиеся при временных SM-детализациях, что демонстрируется примерами на рисунках 2.9–2.10. □

Следующая теорема демонстрирует, какие из рассмотренных  $\tau$ -эквивалентностей сохраняются при временных SM-детализациях.

**Теорема 2.3.3** Пусть  $\leftrightarrow \in \{\equiv_t^{\tau}, \xleftrightarrow{\tau}, \simeq\}$  и  $\star \in \{_, t, u\}$ . Для временных сетей с невидимыми переходами  $N, N'$  такое, что  $a \in l_N(T_N) \cap l_{N'}(T_{N'}) \cap \text{Act}$  и временной SM-сети с невидимыми переходами  $D$   $N \leftrightarrow_{\star} N' \Rightarrow \text{tref}(N, a, D) \leftrightarrow_{\star} \text{tref}(N', a, D)$  тогда и только тогда, когда эквивалентность  $\leftrightarrow_{\star}$  заключена в овал на рисунке 2.20.

*Доказательство.* Аналогично теореме 2.2.2 □

### 2.3.7 $\tau$ -эквивалентности на подклассах временных сетей Петри с невидимыми переходами

**$\tau$ -эквивалентности на временных сетях Петри с видимыми переходами**

Для временных сетей с видимыми переходами следующее предложение устанавливает совпадение  $\tau$ -эквивалентностей с эквивалентностями.

**Предложение 2.3.7** Пусть  $\leftrightarrow \in \{\equiv_t, \equiv_u, \xleftrightarrow{t}, \xleftrightarrow{u}\}$ . Для временных сетей с видимыми переходами  $N$  и  $N'$   $N \leftrightarrow^{\tau} N' \Leftrightarrow N \leftrightarrow N'$ .

*Доказательство.* Очевидно, так как во временных сетях с видимыми переходами нет переходов с пометкой  $\tau$ . □

На рисунке 2.21 пунктирными линиями обведены эквивалентности и  $\tau$ -эквивалентности, совпадающие на временных сетях с видимыми переходами.

**Теорема 2.3.4** Пусть  $\leftrightarrow, \Leftarrow \in \{\equiv_t^{\tau}, \equiv_u^{\tau}, \xleftrightarrow{t}^{\tau}, \xleftrightarrow{u}^{\tau}, \simeq\}$ . Для временных сетей с видимыми переходами  $N$  и  $N'$   $N \leftrightarrow N' \Rightarrow N \Leftarrow N'$  тогда и только тогда, когда в графе на рисунке 2.16 существует направленный путь от  $\leftrightarrow$  к  $\Leftarrow$ .

*Доказательство.* По теореме 2.3.1. □

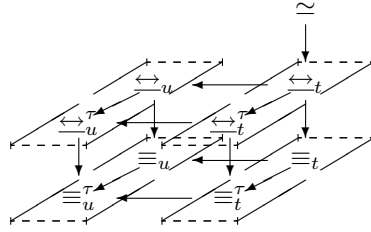


Рис. 2.21: Совпадение  $\tau$ -эквивалентностей и эквивалентностей на временных сетях с видимыми переходами

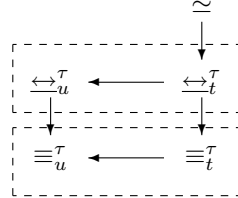


Рис. 2.22: Совпадение  $\tau$ -эквивалентностей на не-временных сетях с невидимыми переходами

### $\tau$ -эквивалентности на не-временных сетях Петри с невидимыми переходами

**Определение 2.3.8** Не-временная сеть с невидимыми переходами — временная сеть с невидимыми переходами  $N = \langle P_N, T_N, F_N, l_N, M_N, \Upsilon_N \rangle$  такая, что  $\forall t \in T_N \ \Upsilon_N(t) = 0$ .

Для не-временных сетей с невидимыми переходами следующее предложение устанавливает совпадение временных и не-временных  $\tau$ -эквивалентностей.

**Предложение 2.3.8** Пусть  $\leftrightarrow \in \{\leftrightarrow^\tau, \leftrightarrow^{\tau'}\}$ . Для не-временных сетей с невидимыми переходами  $N$  и  $N'$   $N \leftrightarrow_u N' \Leftrightarrow N \leftrightarrow_t N'$ .

*Доказательство.* Очевидно, так как в не-временных сетях с невидимыми переходами нет временных задержек.  $\square$

На рисунке 2.22 пунктирными линиями обведены эквивалентности и  $\tau$ -эквивалентности, совпадающие на не-временных сетях с невидимыми переходами.

**Теорема 2.3.5** Пусть  $\leftrightarrow, \Leftrightarrow \in \{\equiv_u^\tau, \leftrightarrow_u^\tau, \simeq\}$ . Для не-временных сетей с невидимыми переходами  $N$  и  $N'$   $N \leftrightarrow N' \Rightarrow N \Leftrightarrow N'$  тогда и только тогда, когда в графе в рисунке 2.23 существует направленный путь от  $\leftrightarrow$  к  $\Leftrightarrow$ .

*Доказательство.* ( $\Leftarrow$ ) По теореме 2.3.1.

( $\Rightarrow$ )

- На рисунке 2.3(b)  $N \equiv_u^\tau N'$  но  $N \not\leftrightarrow_u^\tau N'$ .
- На рисунке 2.3(c)  $N \leftrightarrow_u^\tau N'$  но  $N \not\equiv_u^\tau N'$ .  $\square$

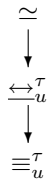


Рис. 2.23: Взаимосвязь  $\tau$ -эквивалентностей на не-временных сетях с невидимыми переходами

## Глава 3

# Алгебры процессов

Для описания параллельных систем и процессов и исследования их поведенческих свойств был предложен ряд моделей, различающихся представлением параллельности. Среди таких моделей алгебраические исчисления занимают особое место. В них процесс описывается алгебраической формулой и проверка свойств процесса выполняется посредством эквивалентностей, аксиом и правил вывода. Все исчисления, рассматриваемые ниже, имеют общее ядро. Они конструируют процесс из атомарных действий (некоторые из которых могут взаимодействовать) с использованием операций последовательного выполнения, параллелизма и недетерминизма. В зависимости от семантики операции параллелизма эти алгебраические исчисления делятся на три группы.

1. CCS [197, 198, 203, 204, 81], CSP [174, 81] и ее расширения [82], TCSP [65, 175, 220, 221], PA [32], BPA [70] и ее расширения [26, 71, 85] — наиболее известные алгебраические исчисления для описания параллельного процесса на основе интерливинговой семантики, в которой параллельность моделируется последовательным недетерминизмом.
2. SCCS [200, 201, 202], ACP [66, 67, 68] и ее расширения [25, 30, 31, 32, 76, 60, 61, 62, 63, 69, 147, 83, 84, 27, 28, 29, 135], CCSP [222], PBC [43, 42, 54, 55, 56, 57, 58, 59, 72, 73, 74, 75, 77, 78, 79, 123, 124, 125, 132, 183, 184, 51, 192, 210, 44, 179] были предложены для аксиоматизации группы моделей параллелизма, основанных на шаговой семантике, в которой параллельное выполнение двух процессов моделируется чередованием мультимножеств их атомарных действий. Множество операций было расширено (по сравнению с CCS и PA) введением нового оператора, описывающего одновременное выполнение двух действий.
3. Алгебраические исчисления (например, алгебры из [34, 35]) для группы моделей, описывающих *истинный параллелизм* и основанных на семантике ЧУММ. Среди таких моделей наиболее известны сети Петри [232], структуры событий [215], ЧУММ [244], О-сети (Occurrence Nets) [233] и А-сети [180]. Отношение причинной зависимости действий в модели задает на них отношение предшествования, порождающее частичный порядок. Соответственно, два действия параллельны, если они причинно независимы.

Интерливинговые исчисления являются более удобными в техническом отношении, в то время как алгебры, основанные на шаговой семантике и семантике ЧУММ, имеют более естественное представление параллелизма. В настоящее время все три подхода к определению семантики сосуществуют, чему дается обоснование в [48, 49]. Предлагаются новые семантики для введенных ранее исчислений. Например, были предложены семантики на основе ЧУММ для CCS [36, 130], CSP [246], SCCS [35], PBC [190], на основе сетей Петри — для CCS [126, 128, 155, 156, 38], PBC [72, 73, 74, 75] и на основе структур событий — для CCS [288, 129, 37, 38]. Также вводятся алгебраические операции на произвольных сетях Петри, что придает им модульность, характерную для процессных алгебр [20, 16, 17, 18].

Алгебры, предлагаемые для спецификации процесса, могут быть разделены на “описательные” и “аналитические”. В описательных алгебрах типа  $AFP_0, AFP_0^m$ , рассмотренных в [98, 105], спецификация процесса обеспечивает описание структурных свойств разрабатываемых параллельных систем. Аналитические алгебры содержат механизмы для проверки поведенческих свойств процессов.

Алгебры  $AFP_1, AFP_1^\lambda, AFP_2, AFP_2^\odot, SAFP_1, SAFP_2$  и  $SAFP_2$  с  $\lambda$ -действиями, рассмотренные в [96, 97, 98, 105, 99, 100, 107, 108, 109, 181, 182, 7], разработаны на основе алгебры регулярных сетей [106, 2, 189].



Они относятся к третьей группе алгебраических исчислений и объединяют механизмы как для описания параллельных недетерминированных процессов, так и для исследования их свойств.

Основные результаты данной главы следующие.

- Введена новая алгебра  $AFLP_2$  помеченных недетерминированных параллельных процессов (расширение известного исчисления  $AFP_2$  [98] посредством пометки формул символами действий), дающая возможность спецификации и анализа значительно более широкого класса процессов.
- Сетевые эквивалентности исследованы на таких подклассах как А-сети и слабо помеченные А-сети, которые соответствуют формулам алгебр  $AFP_2$  и  $AFLP_2$  что облегчило сравнения этих понятий с алгебраическими эквивалентностями и их переноса с сетей на формулы алгебр.
- Предложена операционная характеристика эквивалентности относительно денотационной семантики  $AFLP_2$ , позволяющая сравнивать семантическую эквивалентность с поведенческими отношениями.
- Дана полная и корректная аксиоматизация семантических эквивалентностей, облегчающая сравнение формульных представлений процессов.
- Алгебраические эквивалентности перенесены на сети и проведено сравнение сетевых и алгебраических отношений, что позволило лучше понять природу последних.
- Сетевые эквивалентности перенесены на формулы  $AFLP_2$ . Доказана согласованность алгебраических аналогов исходным сетевым отношениям. Это позволило переходить от сетевых к алгебраическим спецификациям и обратно с сохранением поведения и сочетать преимущества сетей и алгебр.
- Алгебраические аналоги сетевых эквивалентностей проверены на композициональность относительно структурных операций.
- Разработана система правил переписывания  $RWS_2$  для автоматизации проверки семантических эквивалентностей исчислений  $AFP_2$  и  $AFLP_2$ . Доказана ее сходимость при условии завершимости.
- На основе  $RWS_2$  разработана программа CANON на языке Си для машинной проверки эквивалентности формул упомянутых алгебр.

### 3.1 Алгебра конечных процессов $AFP_2$

Алгебра  $AFP_2$ , рассматриваемая в этом разделе, основана на семантике ЧУМ с не-действиями и тупиковыми действиями, с помощью которых сохраняется информация о недетерминизме. Синхронизация действий в  $AFP_2$  происходит по имени. То есть несколько одноименных действий, которые встречаются в разных частях формулы этой алгебры, синхронизируются. Считается, что этим действиям соответствует одно событие.

В [98] показано, что средствами  $AFP_2$  можно анализировать поведение А-сетей. В данном разделе рассмотренные ранее эквивалентности изучаются на этом подклассе сетей. Эквивалентности на формулах  $AFP_2$  переносятся на А-сети и исследуется их взаимосвязь с сетевыми эквивалентностями.

Также предлагается метод автоматизации проверки семантической эквивалентности формул на основе системы правил переписывания  $RWS_2$ . Доказана сходимость системы при условии завершимости. Метод реализован в виде программы CANON на языке Си.

#### 3.1.1 Синтаксис

Пусть  $\alpha = \{a, b, c, \dots\}$  — конечный алфавит символов *действий* (базис действий процесса). Действия комбинируются в процесс с помощью операций ; (*предшествование*), || (*параллельность*) и  $\nabla$  (*исключение или альтернатива*). В процессе  $(a; b)$  сначала выполняется действие  $a$  и только после этого —  $b$ . Формула  $(a||b)$  представляет процесс, в котором действия  $a$  и  $b$  выполняются параллельно. Процесс  $(a\nabla b)$  описывает два возможных поведения: если выбирается выполнение действия  $a$ , то  $b$  не случается и наоборот.

Предполагается, что каждое действие имеет свое уникальное имя. Например, формула  $(a; c)|| (b; c)$  описывает процесс, в котором действия  $a$  и  $b$  выполняются параллельно, и только затем выполняется  $c$ . Таким образом, выполнения действия  $c$  в подпроцессах  $(a; c)$  и  $(b; c)$  синхронизированы.

Алгебра  $AFP_2$  содержит механизмы для описания как самих процессов, так и их свойств. Для выражения и проверки различных свойств процессов к базису действий процесса добавляются другие множества.  $\bar{\alpha} = \{\bar{a}, \bar{b}, \bar{c}, \dots\}$  — двойной к  $\alpha$  алфавит символов *не-действий*, которые выражают факт, что соответствующие действия не выполняются во время работы процесса из-за выполнения некоторых альтернативных им действий.  $\Delta_\alpha = \{\delta_a, \delta_b, \delta_c, \dots\}$  — алфавит *тупиковых* действий, описывающий действия, которые не могут выполняться из-за некоторого противоречия или ошибки в описании процесса.

Вводятся также дополнительные операции:  $\parallel$  (“не случится”),  $\tilde{\parallel}$  (“не случится ошибочно”),  $\vee$  (*дизъюнкция* или *объединение*).

Операция  $\parallel$  — модифицированное отрицание.  $\parallel P$  означает, что процесс  $P$  не случится, то есть выполняются не-действия из  $P$ .

Операция  $\tilde{\parallel}$  — другой тип отрицания.  $\tilde{\parallel} P$  означает, что процесс  $P$  не случается в результате некоторой ошибки, то есть любое действие из  $P$  не случается во время функционирования процесса в результате некоторых противоречивых требований в описании процесса.

Формула  $(P \vee Q)$  определяет процесс, в котором выполняется либо подпроцесс  $P$ , либо  $Q$ . В этом случае множество возможных поведений процесса — объединение множеств поведений подпроцессов  $P$  и  $Q$ .

Обозначим  $\hat{\alpha} = \alpha \cup \bar{\alpha} \cup \Delta_\alpha$ .

**Определение 3.1.1** Формула  $AFP_2$  в базисе  $\hat{\alpha}$  определяется следующим образом.

$$P ::= a \mid \bar{a} \mid \delta_a \mid \parallel P \mid \tilde{\parallel} P \mid P; Q \mid P \parallel Q \mid P \nabla Q \mid P \vee Q$$

Здесь  $a \in \alpha$ ,  $\bar{a} \in \bar{\alpha}$ ,  $\delta_a \in \Delta_\alpha$  — элементарные формулы.

Обозначим через  $\mathbf{AFP}_2$  множество всех формул алгебры  $AFP_2$ .

### 3.1.2 Семантика

#### Денотационная семантика

Семантика процесса, который описывается формулой  $AFP_2$ , есть множество частичных порядков, то есть совокупность частично упорядоченных множеств (ЧУМ), рассмотренных в [98, 99] с порядком по предшествованию действий при выполнении данного процесса.

Рассмотрим ЧУМ  $\rho = (X, \prec)$ , где:

- $X$  — множество вершин, моделирующих действия, не-действия и тупиковые действия процесса, то есть  $X \subseteq \hat{\alpha}$ ;
- $\prec \subseteq X \times X$  — строгий частичный порядок над  $X$ , где  $a \prec b$  интерпретируется так: действие  $a$  обязательно предшествует  $b$  в процессе.

Обозначим через  $X^+ = X \cap \alpha$  подмножество действий  $X$ ,  $X^- = X \cap \bar{\alpha}$  подмножество не-действий  $X$ ,  $\Delta_X = X \cap \Delta_\alpha$  подмножество тупиковых действий  $X$ , то есть  $X = X^+ \cup X^- \cup \Delta_X$ . Мы будем рассматривать ЧУМ только двух типов: либо  $X = X^+ \cup X^-$ , либо  $X = X^+ \cup \Delta_X$ , то есть если в ЧУМ существует некоторое тупиковое действие, мы не будем различать не-действия и тупиковые действия и будем рассматривать все не-действия в “отрицательной” части процесса как тупиковые.

Над частично упорядоченными множествами вводятся операции  $\parallel, \tilde{\parallel}, ;, \parallel, \nabla$  и операция *модифицированного объединения*  $\tilde{\cup}$ , определяемые в [98, 99].

Пусть  $\mathcal{D}_{F2}[P]$  — совокупность частично упорядоченных множеств, связанных с процессом  $P$ .

**Определение 3.1.2** Денотационная семантика  $AFP_2$  определяется следующим образом.

1.  $\mathcal{D}_{F2}[a] = (\{a\}, \emptyset)$ ,  $\mathcal{D}_{F2}[\bar{a}] = (\{\bar{a}\}, \emptyset)$ ,  $\mathcal{D}_{F2}[\delta_a] = (\{\delta_a\}, \emptyset)$ ;
2.  $\mathcal{D}_{F2}[\neg P] = \neg \mathcal{D}_{F2}[P]$ ,  $\neg \in \{\parallel, \tilde{\parallel}\}$ ;
3.  $\mathcal{D}_{F2}[P \circ Q] = \mathcal{D}_{F2}[P] \circ \mathcal{D}_{F2}[Q]$ ,  $\circ \in \{;, \parallel, \nabla\}$ ;
4.  $\mathcal{D}_{F2}[P \vee Q] = \mathcal{D}_{F2}[P] \tilde{\cup} \mathcal{D}_{F2}[Q]$ .

**Пример 3.1.1** Рассмотрим процесс, описываемый формулой  $P = (a \nabla b) \parallel (b \nabla c)$ . Тогда  $\mathcal{D}_2[P] = \{(\{b, \bar{a}, \bar{c}\}, \emptyset), (\{a, c, \bar{b}\}, \emptyset)\}$  состоит из двух частично упорядоченных множеств.

Денотационная семантика задает эквивалентность на процессах.

**Определение 3.1.3** *Формулы  $P$  и  $P'$  эквивалентны относительно денотационной семантики  $\mathcal{D}_{F_2}$ , обозначение  $P =_{\mathcal{D}_{F_2}} P'$ , если  $\mathcal{D}_{F_2}[P] = \mathcal{D}_{F_2}[P']$ .*

Пусть  $\rho = (X, \prec)$  — ЧУМ. Тогда  $\rho^+ = (X^+, \prec)$  — это *действительная, “наблюдаемая” часть*  $\rho$  над множеством действий. Если процесс  $P$  характеризуется множеством ЧУМ  $\cup_{i=1}^n \rho_i$ , то есть  $\mathcal{D}_{F_2}[P] = \cup_{i=1}^n \rho_i$ , тогда  $\mathcal{D}_{F_2}^+[P] = \cup_{i=1}^n \rho_i^+$  обозначает его *“наблюдаемую” часть* над множеством действий.

**Определение 3.1.4** *Формулы  $P$  и  $P'$  наблюдаемо эквивалентны относительно денотационной семантики  $\mathcal{D}_{F_2}$ , запись  $P =_{\mathcal{D}_{F_2}^+} P'$ , если  $\mathcal{D}_{F_2}^+[P] = \mathcal{D}_{F_2}^+[P']$ .*

Таким образом, наблюдаемая эквивалентность — ограничение обычной эквивалентности. Эта эквивалентность учитывает только ЧУМ над множеством действий и не берет в расчет не-действия и тупиковые действия.

### Аксиоматизация

В соответствии с отношением эквивалентности вводится система аксиом  $\Theta_{F_2}$ .

В следующих равенствах  $P, Q, R$  обозначают формулы  $AFP_2$ , а  $a \in \alpha, \bar{a} \in \bar{\alpha}, \delta_a \in \Delta_\alpha$ .

#### 1. Ассоциативность

- 1.1  $P \parallel (Q \parallel R) = (P \parallel Q) \parallel R$
- 1.2  $P \nabla (Q \nabla R) = (P \nabla Q) \nabla R$
- 1.3  $P \vee (Q \vee R) = (P \vee Q) \vee R$
- 1.4  $P; (Q; R) = (P; Q); R$

#### 2. Коммутативность

- 2.1  $P \parallel Q = Q \parallel P$
- 2.2  $P \nabla Q = Q \nabla P$
- 2.3  $P \vee Q = Q \vee P$

#### 3. Дистрибутивность

- 3.1  $(P \parallel Q); R = (P; R) \parallel (Q; R)$
- 3.2  $P; (Q \parallel R) = (P; Q) \parallel (P; R)$
- 3.3  $(P \vee Q); R = (P; R) \vee (Q; R)$
- 3.4  $P; (Q \vee R) = (P; Q) \vee (P; R)$
- 3.5  $(P \vee Q) \parallel R = (P \parallel R) \vee (Q \parallel R)$
- 3.6  $P \nabla (Q \parallel R) = (P \nabla Q) \parallel (P \nabla R)$

#### 4. Аксиомы для $\nabla$ и $\parallel$

- 4.1  $P \nabla Q = (P \parallel (\parallel Q)) \vee ((\parallel P) \parallel Q)$
- 4.2  $\parallel (P \parallel Q) = (\parallel P) \parallel (\parallel Q)$
- 4.3  $\parallel (P \vee Q) = (\parallel P) \vee (\parallel Q)$
- 4.4  $\parallel (P; Q) = (\parallel P) \parallel (\parallel Q)$
- 4.5  $\parallel a = \bar{a}$
- 4.6  $\parallel \bar{a} = \bar{a}$
- 4.7  $\parallel \delta_a = \bar{a}$

#### 5. Структурные свойства

- 5.1  $\bar{a}; P = \bar{a} \| P$
- 5.2  $P; \bar{a} = P \| \bar{a}$
- 5.3  $P \| (P; Q) = (P; Q)$
- 5.4  $Q \| (P; Q) = (P; Q)$
- 5.5  $P; Q; R = (P; Q) \| (Q; R)$
- 5.6  $(P; Q) \| (Q; R) = (P; Q) \| (Q; R) \| (P; R)$
- 5.7  $P \| P = P$
- 5.8  $P \vee P = P$
- 5.9  $P \vee Q = P$ , если  $Q \triangleleft P$  (понятие строгого префикса  $\triangleleft$  на формулах будет определено дальше)

6. Аксиомы для тупиковых действий и  $\tilde{\parallel}$

- 6.1  $a \| \bar{a} = \delta_a$
- 6.2  $a; a = \delta_a$
- 6.3  $a \| \delta_a = \delta_a$
- 6.4  $\delta_a; P = \delta_a \| (\tilde{\parallel} P)$
- 6.5  $P; \delta_a = P \| \delta_a$
- 6.6  $\delta_a \| (\parallel P) = \delta_a \| (\tilde{\parallel} P)$
- 6.7  $\tilde{\parallel} a = \delta_a$
- 6.8  $\tilde{\parallel} \bar{a} = \delta_a$
- 6.9  $\tilde{\parallel} \delta_a = \delta_a$
- 6.10  $\tilde{\parallel} (P \| Q) = (\tilde{\parallel} P) \| (\tilde{\parallel} Q)$
- 6.11  $\tilde{\parallel} (P; Q) = (\tilde{\parallel} P) \| (\tilde{\parallel} Q)$
- 6.12  $\tilde{\parallel} (P \vee Q) = (\tilde{\parallel} P) \vee (\tilde{\parallel} Q)$

Для доказательства полноты системы аксиом  $\Theta_{FP_2}$  вводится понятие канонической формы формулы  $AFP_2$ .

**Каноническая форма формул**

Алфавит формулы  $P$ , обозначение  $\alpha(P)$ , — это подмножество алфавита действий  $\alpha$ , которое определяется следующим образом.

1.  $\alpha(a) = \alpha(\bar{a}) = \alpha(\delta_a) = a$ ;
2.  $\alpha(\neg P) = \alpha(P)$ ,  $\neg \in \{\parallel, \tilde{\parallel}\}$ ;
3.  $\alpha(P \circ Q) = \alpha(P) \cup \alpha(Q)$ ,  $\circ \in \{;, \|, \nabla, \vee\}$ .

Определим также  $\bar{\alpha}(P) = \{\bar{a} \mid a \in \alpha(P)\}$ ,  $\Delta_\alpha(P) = \{\delta_a \mid a \in \alpha(P)\}$  и  $\hat{\alpha}(P) = \alpha(P) \cup \bar{\alpha}(P) \cup \Delta_\alpha(P)$ .

Содержимое формулы  $P$ , обозначение  $cont(P)$ , — множество символов из  $\hat{\alpha}(P)$  следующего вида.

1.  $cont(a) = \{a\}$ ,  $cont(\bar{a}) = \{\bar{a}\}$ ,  $cont(\delta_a) = \{\delta_a\}$ ;
2.  $cont(\neg P) = cont(P)$ ,  $\neg \in \{\parallel, \tilde{\parallel}\}$ ;
3.  $cont(P \circ Q) = cont(P) \cup cont(Q)$ ,  $\circ \in \{;, \|, \nabla, \vee\}$ .

Как следует из определения,  $cont(P)$  — множество элементарных подформул формулы  $P$ . Определим также  $cont^+(P) = cont(P) \cap \alpha$ ,  $cont^-(P) = cont(P) \cap \bar{\alpha}$ ,  $\Delta_{cont}(P) = cont(P) \cap \Delta_\alpha$ .

Формулы  $P$  и  $P'$  *изоморфны*, обозначение  $P \simeq P'$ , тогда и только тогда, когда  $P$  может быть сведена к  $P'$  (и наоборот) с использованием аксиом ассоциативности для  $;$ ,  $\|$ ,  $\nabla$ ,  $\vee$  и коммутативности для операций  $\|$ ,  $\nabla$ ,  $\vee$ .

**Пример 3.1.2**  $(a \| b \| \bar{c}) \vee (c \| \bar{a} \| \bar{b}) \simeq (\bar{a} \| \bar{b} \| c) \vee (b \| a \| \bar{c})$ .

Введем понятия, связанные со структурой формулы.

*Предшествование* — формула вида  $P_1; \dots; P_n = \underset{i=1}{\overset{n}{\parallel}} P_i$ ,  $P_i \in \hat{\alpha}$  ( $1 \leq i \leq n$ );

*Конъюнкция* — формула вида  $P_1 \parallel \dots \parallel P_n = \underset{i=1}{\parallel} P_i$ , где  $P_i$  — предшествования ( $1 \leq i \leq n$ ).

*Дизъюнкция* — формула вида  $P = P_1 \vee \dots \vee P_n = \underset{i=1}{\vee} P_i$ , где  $P_i$  — конъюнкции ( $1 \leq i \leq n$ ).

*Нормальная конъюнкция* — конъюнкция  $P = \underset{i=1}{\parallel} P_i$ , для которой истинны следующие предложения.

1. Каждая формула  $P_i$  ( $1 \leq i \leq n$ ) имеет одну из следующих форм:

- (а) элементарная формула  $a$  ( $a \in \alpha$ ),  $\bar{a}$  ( $\bar{a} \in \bar{\alpha}$ ),  $\delta_a$  ( $\delta_a \in \Delta_\alpha$ );
- (б) элементарное предшествование  $(a; b)$ , где  $a, b \in \alpha$  и  $a \neq b$ ;

2. Если имеется формула  $P_i$  ( $1 \leq i \leq n$ ) в форме  $\delta_a$  ( $\delta_a \in \Delta_\alpha$ ), тогда нет другой формулы  $P_j$  ( $1 \leq j \leq n$ ) такой, что  $P_j = \bar{b}$  ( $\bar{b} \in \bar{\alpha}$ );

3. Для любых формул  $P_i$  и  $P_j$  ( $1 \leq i \neq j \leq n$ ) таких, что  $\alpha(P_i) \cap \alpha(P_j) \neq \emptyset$   $P_i$  и  $P_j$  должны иметь форму различных элементарных предшествований;

4. Для любой пары  $P_i = (a; b)$  и  $P_j = (b; c)$  ( $1 \leq i \neq j \leq n$ ) существует терм  $P_k = (a; c)$  ( $1 \leq k \leq n$ ), описывающий транзитивное замыкание отношения предшествования для действий  $a$ ,  $b$  и  $c$ .

Назовем 1 (соответственно 2, 3, 4)-*конъюнкцией* конъюнкцию, удовлетворяющую условию 1 (соответственно 2, 3, 4) из определения нормальной конъюнкции. Аналогично введем определение, например 1,2,3-конъюнкции. В соответствии с этими определениями нормальная конъюнкция есть 1,2,3,4-конъюнкция.

Пусть  $P$  и  $Q$  — нормальные конъюнкции. Формула  $P$  — *строгий префикс* формулы  $Q$ , обозначение  $P \triangleleft Q$ , если выполняются следующие условия.

1.  $\text{cont}^+(P) \subset \text{cont}^+(Q)$ ;

2. элементарное предшествование  $(a; b)$  является конъюнктивным членом  $Q$  и  $b \in \text{cont}^+(P)$  тогда и только тогда, когда  $(a; b)$  — конъюнктивный член  $P$ ;

Формула  $P$  — *префикс* формулы  $Q$ , обозначение  $P \triangleleft Q$ , если  $P \triangleleft Q$  или  $P \simeq Q$ .

**Пример 3.1.3** В формуле  $(a \parallel c \parallel \bar{b} \parallel \bar{d} \parallel \bar{e}) \vee (c \parallel \delta_a \parallel \delta_b \parallel \delta_d \parallel \delta_e) \vee (a \parallel \delta_b \parallel \delta_c \parallel \delta_d \parallel \delta_e) \vee ((b; d) \parallel (b; e) \parallel \bar{a} \parallel \bar{c})$  вторая и третья конъюнкции — строгие префиксы первой.

Формула  $P$  имеет *каноническую форму*, если она является дизъюнкцией  $P = \underset{i=1}{\vee} P_i$  и выполняются следующие условия.

- 1.  $P_i$  ( $1 \leq i \leq n$ ) — нормальная конъюнкция;
- 2. для любых формул  $P_i$  и  $P_j$  ( $1 \leq i \neq j \leq n$ )  $P_i \not\triangleleft P_j$ ;
- 3. для любых формул  $P_i$  и  $P_j$  ( $1 \leq i \neq j \leq n$ )  $\neg(P_i \triangleleft P_j \vee P_j \triangleleft P_i)$ .

Будем обозначать множество всех канонических форм формулы  $P$  через  $\text{canon}(P)$ . Заметим, что каждый дизъюнктивный член канонической формы формулы характеризует одно из возможных альтернативных поведений процесса и является представлением частичного порядка этого процесса.

**Пример 3.1.4** Формула  $(a \parallel c \parallel \bar{b} \parallel \bar{d} \parallel \bar{e}) \vee ((b; d) \parallel (b; e) \parallel \bar{a} \parallel \bar{c})$  имеет каноническую форму.

Аналогично определениям для конъюнкций вводим определения 1 (соответственно 2, 3)-*дизъюнкций* и другие. Таким образом, каноническая форма — 1,2,3-дизъюнкция.

Конъюнкция (дизъюнкция) *максимальна*, если она не является конъюнктивным (дизъюнктивным) членом никакой другой конъюнкции (дизъюнкции).

**Пример 3.1.5** В формуле  $((a \parallel b \parallel c) \vee d) \parallel e$  конъюнкции  $a \parallel b$  и  $b \parallel c$  не максимальные, так как они — конъюнктивные члены максимальной конъюнкции  $a \parallel b \parallel c$ . Вся формула — не конъюнкция, так как ее первый конъюнктивный член — дизъюнкция.

Запись  $P =_{\Theta_{F_2}} P'$  означает, что равенство формул  $P$  и  $P'$  может быть доказано с использованием системы аксиом  $\Theta_{F_2}$ .

Следующие теоремы устанавливают полноту системы  $\Theta_{F_2}$ .

**Теорема 3.1.1** [98, 99] Любая формула  $AFP_2$  может быть сведена к единственной до изоморфизма канонической форме.

**Теорема 3.1.2** [98, 99] Для любых формул  $P$  и  $P'$  алгебры  $AFP_2$  истинно  $P =_{\mathcal{D}_{F_2}} P' \Leftrightarrow P =_{\Theta_{F_2}} P'$ .

Таким образом, мы можем выяснить, эквивалентны ли любые две формулы  $P$  и  $P'$  алгебры  $AFP_2$ . Для этого достаточно привести эти формулы к их каноническим формам  $Q$  и  $Q'$  и проверить их на изоморфизм.

### 3.1.3 Эквивалентности на А-сетях

В [2, 189] введена алгебра регулярных сетей, на основе которой позднее была разработана алгебра  $AFP_0$ , формулам которой соответствуют конечные А-сети [180, 3]. В [2, 189] доказано, что любую конечную А-сеть можно представить формулой этой алгебры с помощью алгоритма регуляризации. В [98] построено отображение  $\Psi$  из множества формул  $AFP_0$  в соответствующее множество  $AFP_2$  такое, что совокупность ЧУМ сети, описываемой формулой  $P$  в  $AFP_0$ , совпадает с множеством ЧУМ недетерминированного процесса, описываемого формулой  $\Psi(P)$  в  $AFP_2$ . Это позволяет нам не различать формулы  $P$  и  $\Psi(P)$  в семантике ЧУММ. Таким образом, если у нас есть А-сеть, определяемая формулой  $P$  в  $AFP_0$ , мы можем анализировать ее свойства и поведение посредством той же самой формулы  $P$  в  $AFP_2$ . Приведем формальное определение А-сети и рассмотрим, как ведут себя эквивалентности на А-сетях.

**Определение 3.1.5** А-сеть [3, 180] — это ациклическая ординарная строго помеченная сеть  $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$  со следующими свойствами.

1.  $\forall p \in P_N (\bullet p \neq \emptyset) \vee (p \bullet \neq \emptyset)$ , то есть нет изолированных мест;
2.  $\forall p, q \in P_N (\bullet p = \bullet q) \wedge (p \bullet = q \bullet) \Rightarrow p = q$ , то есть нет “лишних” мест;
3.  $\forall t \in T_N (\bullet t \neq \emptyset) \wedge (t \bullet \neq \emptyset)$ , то есть все переходы имеют входные и выходные места;
4.  $\forall x \in P_N \cup T_N | \downarrow_N x | < \infty$ , то есть  $N$  — сеть с конечным множеством причин;
5.  $\forall p \in P_N \forall t, u \in T_N t, u \in \bullet p \Rightarrow t \text{ al } u$ , то есть переходы с общим выходным местом альтернативны;
6.  $M_N = \circ N$ , то есть начальная маркировка — множество входных мест сети.

Отношение *альтернативы*, обозначаемое **al**, определяется следующим образом. Пусть в сети  $N$   $t, u \in T_N$ , тогда  $t \text{ al } u$ , если выполняются следующие условия.

1.  $\neg(t \diamond_N u)$ ;
2.  $(\bullet t \cap \bullet u \neq \emptyset) \vee (\exists p \in \bullet t \forall t' \in \bullet p t' \text{ al } u) \vee (\exists q \in \bullet u \forall u' \in \bullet q t \text{ al } u') \vee (t = u)$ .

Заметим, что в исходном определении [3] А-сеть считается непомеченной, что в данном определении соответствует требованию о строгой помеченности. Так как мы рассматриваем только сети с конечными процессами, то пункт 4 определения А-сети можно опустить. Из свойств 5 и 6 этого определения следует, что А-сети безопасны.

Рассмотрим, как ведут себя эквивалентности на А-сетях. Сначала докажем следующей лемму, утверждающую, что для А-сети ее процессы с С-сетями над одним и тем же множеством действий изоморфны.

**Лемма 3.1.1** Пусть  $N$  — А-сеть,  $\pi = (C, \varphi)$ ,  $\bar{\pi} = (\bar{C}, \bar{\varphi}) \in \Pi(N)$  и  $l_C(T_C) = l_{\bar{C}}(T_{\bar{C}})$ . Тогда  $\pi \simeq \bar{\pi}$ .

*Доказательство.* Пусть  $l_C(T_C) = A \subseteq Act$ . Рассмотрим подсеть  $N_A$  сети  $N$ , определяемую следующим образом:

1.  $T_{N_A} = \varphi(T_C) = \{u \in T_N \mid l_N(u) \in A\}$ ,  $P_{N_A} = \varphi(P_C) = \bullet T_{N_A} \cup T_{N_A} \bullet \cup M_N$ ;
2.  $F_{N_A} = F_N|_{(P_{N_A} \times T_{N_A}) \cup (T_{N_A} \times P_{N_A})}$ ;
3.  $l_{N_A} = l_N|_{T_{N_A}}$ ;
4.  $M_{N_A} = M_N$ .

Таким образом,  $N_A$  — это замкнутая относительно предшествования подсеть  $N$ , содержащая все места из начальной маркировки. Кроме того,  $N_A$  определяется только на основе множества  $A$  и не зависит от функции встраивания  $\varphi$ .

Заметим, что, так как А-сети — ациклические, строго помеченные и безопасные, С-сети их процессов — тоже строго помеченные.

Пусть  $v, w \in T_C, v \neq w$  и  $\varphi(v) = \varphi(w)$ . Тогда по определению функции встраивания  $l_C(v) = l_C(w)$ , что противоречит строгой помеченности сети  $C$ . Поэтому  $\varphi|_{T_C} : \rho_C \simeq \rho_{N_A}$ .

Докажем, что  $\varphi|_{P_C} : P_C \rightarrow P_{N_A}$  — биекция. Предположим противное, то есть что  $\exists r, s \in P_C, r \neq s$  и  $\varphi(r) = \varphi(s)$ . Имеем следующие 2 случая.

1.  $r \diamond_C s$ . Достаточно рассмотреть вариант, когда  $r \prec_C s$ . Тогда в  $F_C$  существует цепочка  $(r, v) \cdots (w, s)$ . По определению  $\varphi$ , в  $F_N$  есть цепочка  $(\varphi(r), \varphi(v)) \cdots (\varphi(w), \varphi(s))$ . Так как  $\varphi(r) = \varphi(s)$ , в сети  $N$  существует цикл, что противоречит ациклическости А-сетей.
2.  $\neg(r \diamond_C s)$ , то есть в  $F_C$  нет цепочек между  $r$  и  $s$ . Пусть  $V = (\Downarrow_C r) \cap T_C$  и  $W = (\Downarrow_C s) \cap T_C$ . Тогда в результате срабатывания переходов из  $V \cup W$  получим маркировку  $L$  С-сети  $C$  такую, что  $r, s \in L$ . Следовательно, в  $N$  существует последовательность срабатываний переходов, приводящая к маркировке  $\varphi(L)$ , такой, что  $\varphi(r), \varphi(s) \in \varphi(L)$ . Так как  $\varphi(r) = \varphi(s)$ , в месте  $\varphi(r)$  сети  $N$  содержатся 2 фишки, и мы получаем противоречие с безопасностью А-сетей.

Итак,  $\varphi|_{P_C} : P_C \rightarrow P_{N_A}$  — биекция. По свойству 3 функции встраивания имеем  $\varphi : C \simeq N_A$ .

Пусть  $\bar{\pi} = (\bar{C}, \bar{\varphi}) \in \Pi(N)$  и  $l_{\bar{C}}(T_{\bar{C}}) = A$ . Тогда  $\bar{\varphi} : \bar{C} \simeq N_A$ . Следовательно,  $\bar{\varphi}^{-1} : N_A \simeq \bar{C}$  и Пусть  $\beta = \bar{\varphi}^{-1} \circ \varphi$ . Тогда  $\beta : C \simeq \bar{C}$  и  $\forall x \in P_C \cup T_C \varphi(x) = \bar{\varphi}(\beta(x))$ , то есть  $\beta : \pi \simeq \bar{\pi}$ .  $\square$

**Предложение 3.1.1** Для А-сетей  $N$  и  $N'$   $N \equiv_i N' \Leftrightarrow N \equiv_{mes} N'$ .

*Доказательство* ( $\Leftarrow$ ) Следствие теоремы 1.2.1.

( $\Rightarrow$ ) Пусть  $N \equiv_i N'$ . Положим  $\Sigma = IntTraces(N) = IntTraces(N')$  и  $A = \{a \in Act \mid \exists \sigma \in \Sigma a \text{ входит в } \sigma\}$ .  $\Sigma$  с точностью до изоморфизма определяет ПСС  $\xi = \langle X, \prec, \#, l \rangle$ , где:

- $X = \{x_a \mid a \in A\}$ ;
- $x_a \prec x_b \Leftrightarrow \forall \sigma \in \Sigma a$  и  $b$  входят в  $\sigma \Rightarrow a$  предшествует  $b$  в  $\sigma$ ;
- $x_a \# x_b \Leftrightarrow \forall \sigma \in \Sigma a$  и  $b$  не входят в  $\sigma$  вместе;
- $l(x_a) = a$ .

Заметим, что это определение корректно по следующим причинам.

1. По определению А-сетей, С-сети их процессов являются строго помеченными. Следовательно, мы можем идентифицировать переходы таких С-сетей с действиями, помечающими эти переходы.
2. По лемме 3.1.1, процессы А-сетей над одним и тем же множеством действий изоморфны. Следовательно, если  $a$  предшествует  $b$  в некотором процессе, то и во всех процессах, куда входят  $a$  и  $b$ ,  $a$  предшествует  $b$ . По лемме 1.2.2,  $a$  предшествует  $b$  в С-сети некоторого процесса тогда и только тогда, когда в любом полном выполнении этой С-сети  $a$  выполняется перед  $b$ , то есть в любом последовательном следе, соответствующем полному выполнению данной С-сети,  $a$  предшествует  $b$ .

Легко проверить, что  $\mathcal{E}(N) = \mathcal{E}(N')$  — класс изоморфизма  $\xi$ . Следовательно,  $N \equiv_{mes} N'$ .  $\square$

**Предложение 3.1.2** Для А-сетей  $N$  и  $N'$   $N \equiv_{pr} N' \Leftrightarrow N \leftrightarrow_{prh} N'$ .

*Доказательство* ( $\Leftarrow$ ) Следствие теоремы 1.2.1.

( $\Rightarrow$ ) Пусть  $N$  и  $N'$  — А-сети такие, что  $N \equiv_{pr} N'$ . Тогда по теореме 1.2.1  $N \equiv_i N'$ , следовательно  $N \leftrightarrow_{pomh} N'$  по предыдущему замечанию. Пусть  $\mathcal{R} : N \leftrightarrow_{pomh} N'$ . Докажем, что  $\mathcal{R} : N \leftrightarrow_{prh} N'$ .

1. Очевидно,  $(\pi_N, \pi_{N'}, \emptyset) \in \mathcal{R}$ .
2. По определению  $\mathcal{R}$ ,  $(\pi, \pi', \beta) \in \mathcal{R} \Rightarrow \beta : \rho_C = \rho_{C'}$ ;

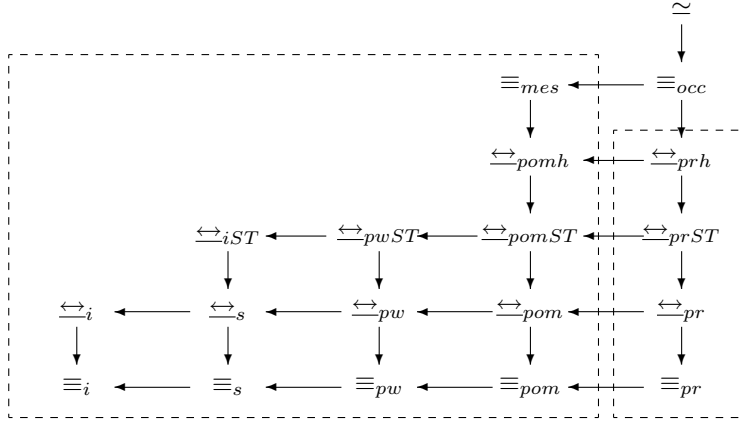


Рис. 3.1: Совпадение эквивалентностей на А-сетях

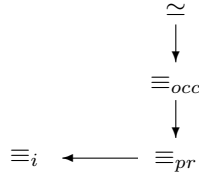


Рис. 3.2: Эквивалентности на А-сетях

3. Пусть  $(\pi, \pi', \beta) \in \mathcal{R}$ ,  $\pi \xrightarrow{v} \tilde{\pi}$ ,  $\tilde{\pi} = (\tilde{C}, \tilde{\varphi})$ . По определению сохраняющей историю бисимуляции ЧУММ существуют  $\tilde{\pi}'$ ,  $v'$ ,  $\tilde{\beta}$  такие, что  $\pi' \xrightarrow{v'} \tilde{\pi}'$ ,  $\tilde{\pi}' = (\tilde{C}', \tilde{\varphi}')$ ,  $\tilde{\beta}|_{T_C} = \beta$ ,  $\tilde{\beta} : \rho_{\tilde{C}} \simeq \rho_{\tilde{C}'}$ , и  $(\tilde{\pi}, \tilde{\pi}', \tilde{\beta}) \in \mathcal{R}$ . Пусть  $\tilde{C} \neq \tilde{C}'$ . Так как  $N \equiv_{pr} N'$ ,  $\exists \tilde{\pi}' = (\tilde{C}', \tilde{\varphi}') \in \Pi(N')$  такой, что  $\tilde{C} \simeq \tilde{C}'$ . Тогда  $\tilde{C}' \neq \tilde{C}'$ . Так как  $\rho_{\tilde{C}'} \simeq \rho_{\tilde{C}} \simeq \rho_{\tilde{C}'}$ , то  $l_{\tilde{C}'}(T_{\tilde{C}'}) \not\subseteq l_{\tilde{C}'}(T_{\tilde{C}'})$ . Имеем противоречие с леммой 3.1.1.
4. Как пункт 3, но роли  $N$  и  $N'$  меняются. □

На рисунке 3.1 пунктирными линиями обведены эквивалентности, совпадающие на А-сетях.

**Теорема 3.1.3** Пусть  $\leftrightarrow, \leftrightarrow^* \in \{\equiv, \simeq\}$ ,  $\star, \star\star \in \{_, i, pr, occ\}$ . Для А-сетей  $N$  и  $N'$   $N \leftrightarrow_{\star} N' \Rightarrow N \leftrightarrow_{\star\star} N'$  тогда и только тогда, когда в графе на рисунке 3.2 существует направленный путь от  $\leftrightarrow_{\star}$  к  $\leftrightarrow_{\star\star}$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие теоремы 1.2.1.

( $\Rightarrow$ ) Отсутствие дополнительных нетривиальных стрелок на рисунке 3.2 доказывается следующими примерами на А-сетях.

- На рисунке 3.3(a)  $N \equiv_i N'$ , но  $N \not\equiv_{pr} N'$ , так как в сети  $N'$  у процесса с действием  $a$  С-сеть не изоморфна ни одной С-сети процесса  $N$ .
- На рисунке 3.3(c)  $N \equiv_{pr} N'$ , но  $N \not\equiv_{occ} N'$ , так как только в развертке сети  $N'$  есть место, являющееся входным для трех переходов.
- На рисунке 3.3(b)  $N \equiv_{occ} N'$ , но  $N \not\equiv N'$ , так как только в сети  $N'$  есть переход с действием  $c$  (который никогда не сможет сработать). □

Заметим, что во всех доказательствах данного раздела мы пользовались только свойствами безопасности, строгой помеченности А-сетей и отсутствия в них циклов. Следовательно, все результаты, полученные для А-сетей, справедливы и для произвольных безопасных строго помеченных ациклических сетей.

### 3.1.4 Сравнение сетевых и алгебраических эквивалентностей

Понятия эквивалентностей в  $AFP_2$  можно расширить на сети, сопоставив любой конечной А-сети формулу  $AFP_2$ , как описано в начале предыдущего раздела.



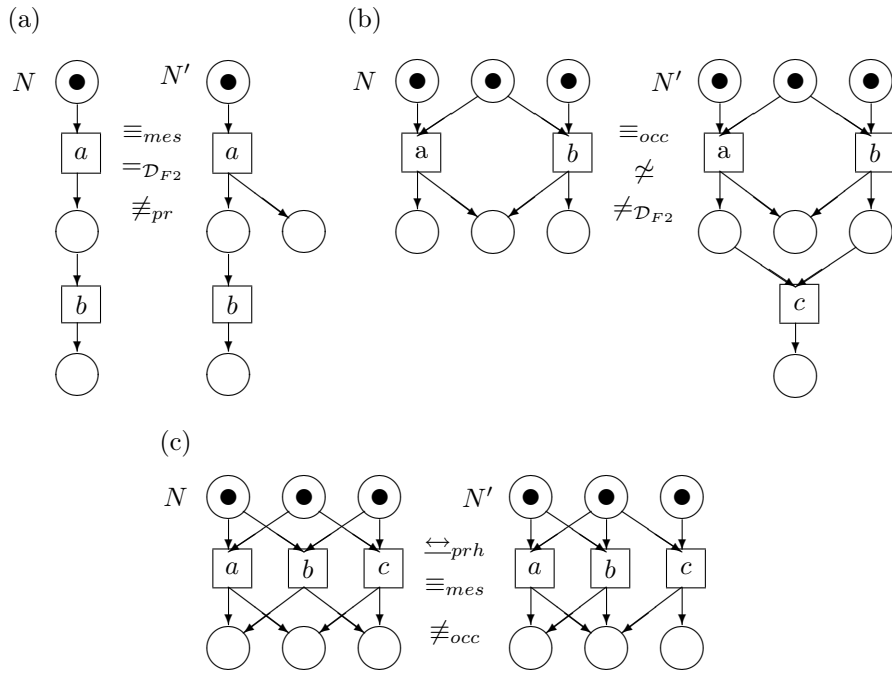


Рис. 3.3: Примеры эквивалентностей на А-сетях

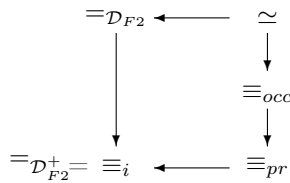


Рис. 3.4: Взаимосвязь сетевых и алгебраических эквивалентностей

**Определение 3.1.6** Пусть  $\leftrightarrow$  — некоторая формульная эквивалентность  $AFP_2$  и А-сетям  $N$  и  $N'$  соответствуют формулы  $E$  и  $E'$ . Сети  $N$  и  $N'$  эквивалентны (относительно  $\leftrightarrow$ ), запись  $N \leftrightarrow N'$ , тогда и только тогда, когда соответствующие им формулы эквивалентны, то есть  $E \leftrightarrow E'$ .

Следующее предложение устанавливает, что на А-сетях наблюдаемая эквивалентность относительно денотационной семантики совпадает со интерливинговой следовой.

**Предложение 3.1.3** Для А-сетей  $N$  и  $N'$   $N \equiv_i N' \Leftrightarrow N =_{\mathcal{D}_{F_2}^+} N'$ .

*Доказательство.* ( $\Leftarrow$ ) Если  $N =_{\mathcal{D}_{F_2}^+} N'$ , то множества максимальных ЧУММ  $N$  и  $N'$  совпадают. Следовательно, совпадают и множества всех их (не только максимальных) ЧУММ. Поэтому  $N \equiv_{pot} N'$ . По теореме 1.2.1  $N \equiv_i N'$ .

( $\Rightarrow$ ) Пусть  $N \equiv_i N'$ . Так как  $N$  и  $N'$  — А-сети, а на А-сетях сливаются все эквивалентности в семантиках от интерливинговой до ЧУММ, имеем  $N \equiv_{pot} N'$ . Следовательно, совокупности ЧУММ сетей совпадают. Из-за строгой помеченности А-сетей в таких сетях не существует пары разных максимальных ЧУММ таких, что одно из них — строгий префикс другого. Поэтому совокупности максимальных ЧУММ  $N$  и  $N'$  совпадают. Так как в  $=_{\mathcal{D}_{F_2}^+}$  рассматриваются только “наблюдаемые” действия, имеем  $N =_{\mathcal{D}_{F_2}^+} N'$ .  $\square$

Следующая теорема устанавливает взаимосвязь сетевых эквивалентностей и формульных эквивалентностей алгебры  $AFP_2$ .

**Теорема 3.1.4** Пусть  $\leftrightarrow, \Leftrightarrow \in \{\equiv, \simeq, =\}$ ,  $\star, \star\star \in \{\_, i, pr, occ, \mathcal{D}_{F_2}^+, \mathcal{D}_{F_2}\}$ . Для А-сетей  $N$  и  $N'$   $N \leftrightarrow_\star N' \Rightarrow N \Leftrightarrow_{\star\star} N'$  тогда и только тогда, когда в графе на рисунке 3.4 существует направленный путь от  $\leftrightarrow_\star$  к  $\Leftrightarrow_{\star\star}$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие теоремы 3.1.3 и следующих рассуждений.

- $=_{\mathcal{D}_{F_2}}$  — более строгая, чем  $=_{\mathcal{D}_{F_2}^+}$ , так как она учитывает “негативную” информацию, которая дается не-действиями и тупиковыми действиями.
- Очевидно, что  $=_{\mathcal{D}_{F_2}}$  — следствие  $\simeq$ , так как совокупности ЧУМ, соответствующие изоморфным сетям, совпадают.

( $\Rightarrow$ ) Отсутствие дополнительных нетривиальных стрелок на рисунке 3.4 доказывается следующими примерами на А-сетях.

- А-сетям  $N$  и  $N'$  на рисунке 3.3(a) соответствуют формулы  $P = a; b$  и  $P' = (a; b) \| a$ . Имеем  $\mathcal{D}_{F_2}[P] = \mathcal{D}_{F_2}[P'] = \{(\{a, b\}, <)\}$ , где  $a < b$ . Следовательно,  $N =_{\mathcal{D}_{F_2}} N'$ . Но  $N \not\equiv_{pr} N'$ , так как в сети  $N'$  переход с пометкой  $a$  имеет дополнительное выходное место.
- А-сетям  $N$  и  $N'$  на рисунке 3.3(b) соответствуют формулы  $P = (a \nabla b) \| a \| b$  и  $P' = (a \nabla b) \| (a; c) \| (b; c)$ . Ясно, что  $N \equiv_{occ} N'$ , так как действие  $c$  в сети  $N'$  никогда не сработает. Тем не менее,  $N \neq_{\mathcal{D}_{F_2}} N'$ . Это следует из того, что  $\mathcal{D}_{F_2}[P] = \{(\{a, \delta_b\}, \emptyset), (\{b, \delta_a\}, \emptyset)\}$ , а  $\mathcal{D}_{F_2}[P'] = \{(\{a, \delta_b, \delta_c\}, \emptyset), (\{b, \delta_a, \delta_c\}, \emptyset)\}$ , то есть тупиковое действие  $\delta_c$  оказывает влияние на эквивалентность  $=_{\mathcal{D}_{F_2}}$ .  $\square$

### 3.1.5 Автоматизация проверки алгебраической эквивалентности

Процесс приведения формулы  $AFP_2$  к каноническому виду с помощью аксиом системы  $\Theta_{F_2}$  довольно трудоемок, особенно если эта формула имеет большую длину и сложную структуру, что часто случается при формульном представлении реальных процессов. Кроме того, в процессе приведения аксиомы из  $\Theta_{F_2}$  приходится применять в ту и другую сторону.

Для того чтобы автоматически привести исходную формулу к одному из изоморфных между собой канонических видов, нужно создать систему правил переписывания, при применении которых формула приводится к нужному виду.

В соответствии с этими требованиями строится система правил переписывания  $RWS_2$ .

#### Система правил переписывания $RWS_2$

Перед описанием системы  $RWS_2$  введем необходимое определение.

*Замена* в формуле  $P$  подформулы  $P_i$  на  $Q$ , запись  $[P]_Q^{P_i}$ , есть формула  $P_1 \circ \dots \circ P_{i-1} \circ Q \circ P_{i+1} \circ \dots \circ P_n$ , где  $P = P_1 \circ \dots \circ P_{i-1} \circ P_i \circ P_{i+1} \circ \dots \circ P_n$ ,  $\circ \in \{;, \|, \nabla, \vee\}$ .

В следующих правилах  $RWS_2$   $P, Q, R$  обозначают формулы  $AFP_2$ ,  $a, b, c \in \alpha$ ,  $\bar{a}, \bar{b} \in \bar{\alpha}$ ,  $\delta_a, \delta_b \in \Delta_\alpha$ , а цифры в скобках — номера равенств системы  $\Theta_{F_2}$ , которые использовались при построении соответствующих правил.

- 1.1.  $\circ \in \{;, \|, \vee\} \Rightarrow$   
 $P \circ (Q \circ R) \rightarrow (P \circ Q) \circ R$   
(1.1, 1.3, 1.4);
- 2.1.  $(\bullet, \circ) \in \{(\|, ;), (\vee, ;), (\vee, \|)\} \Rightarrow$   
 $(P \circ Q) \bullet R \rightarrow (P \bullet R) \circ (Q \bullet R)$   
(3.1, 3.3, 3.5);
- 2.2.  $(\bullet, \circ) \in \{(\|, ;), (\vee, ;), (\vee, \|)\} \Rightarrow$   
 $P \bullet (Q \circ R) \rightarrow (P \bullet Q) \circ (P \bullet R)$   
(2.1, 3.2, 3.4, 3.5);
- 3.1  $P \nabla Q \rightarrow (P \| (\prod Q)) \vee ((\prod P) \| Q)$   
(4.1);
- 4.1.  $\circ \in \{\|, ;\}, \neg \in \{\prod, \tilde{\prod}\} \Rightarrow$   
 $\neg(P \circ Q) \rightarrow (\neg P) \| (\neg Q)$   
(4.2, 4.4, 6.10, 6.11);
- 4.2.  $\neg \in \{\prod, \tilde{\prod}\} \Rightarrow$   
 $\neg(P \vee Q) \rightarrow (\neg P) \vee (\neg Q)$   
(4.3, 6.12);

- 4.3.  $P = a$  или  $P = \bar{a}$  или  $P = \delta_a \Rightarrow$   
 $\parallel P \rightarrow \bar{a}$   
(4.5, 4.6, 4.7);
- 4.4.  $P = a$  или  $P = \bar{a}$  или  $P = \delta_a \Rightarrow$   
 $\tilde{\parallel} P \rightarrow \delta_a$   
(6.7, 6.8, 6.9);
- 5.1.  $P, Q, R \in \hat{\alpha} \Rightarrow$   
 $(P; Q); R \rightarrow ((P; Q) \parallel (Q; R)) \parallel (P; R)$   
(5.5, 5.6);
- 5.2.  $Q \in \hat{\alpha} \Rightarrow$   
 $\bar{a}; Q \rightarrow \bar{a} \parallel Q$   
(5.1);
- 5.3.  $P \in \hat{\alpha} \Rightarrow$   
 $P; \bar{a} \rightarrow P \parallel \bar{a}$   
(5.2);
- 5.4.  $a; a \rightarrow \delta_a$   
(6.2);
- 5.5.  $Q = b$  или  $Q = \bar{b}$  или  $Q = \delta_b \Rightarrow$   
 $\delta_a; Q \rightarrow \delta_a \parallel \delta_b$   
(6.4, 6.7, 6.8, 6.9);
- 5.6.  $P \in \hat{\alpha} \Rightarrow$   
 $P; \delta_a \rightarrow P \parallel \delta_a$   
(6.5);
- 6.1.  $P$  — 1-конъюнкция,  $P' = \delta_a$  — конъюнктивный член  $P \Rightarrow$   
 $P \parallel \bar{b} \rightarrow P \parallel \delta_b$   
(1.1, 2.1, 4.5, 6.6, 6.7);
- 6.2.  $P$  — 1-конъюнкция,  $P' = \bar{b}$  — конъюнктивный член  $P \Rightarrow$   
 $P \parallel \delta_a \rightarrow [P]_{\delta_b}^{P'} \parallel \delta_a$   
(1.1, 2.1, 4.5, 6.6, 6.7);
- 7.1.  $P$  — 1,2-конъюнкция,  $P'$  — конъюнктивный член  $P$ ,  $P' = a$  или  $P' = b \Rightarrow$   
 $P \parallel (a; b) \rightarrow [P]_{(a; b)}^{P'}$   
(1.1, 2.1, 5.3, 5.4);
- 7.2.  $P$  — 1,2-конъюнкция,  $P'$  — конъюнктивный член  $P$ ,  $P' = (a; b)$  или  $P' = (b; a) \Rightarrow$   
 $P \parallel a \rightarrow P$   
(1.1, 2.1, 5.3, 5.4);
- 7.3.  $P$  — 1,2-конъюнкция,  $P' = a$  — конъюнктивный член  $P$ ,  $\diamond \in \{-, \delta\} \Rightarrow$   
 $P \parallel \diamond a \rightarrow [P]_{\delta_a}^{P'}$   
(1.1, 2.1, 6.1, 6.3);
- 7.4.  $P$  — 1,2-конъюнкция,  $P'$  — конъюнктивный член  $P$ ,  $P' = \bar{a}$  или  $P' = \delta_a \Rightarrow$   
 $P \parallel a \rightarrow [P]_{\delta_a}^{P'}$   
(1.1, 2.1, 6.1, 6.3);
- 7.5.  $P$  — 1,2-конъюнкция,  $P' = (a; b)$  — конъюнктивный член  $P$ ,  $\diamond \in \{-, \delta\} \Rightarrow$   
 $P \parallel \diamond a \rightarrow [P]_{\delta_b}^{P'} \parallel \delta_a$   
(1.1, 1.4, 2.1, 5.1, 6.1, 6.3, 6.4, 6.7);
- 7.6.  $P$  — 1,2-конъюнкция,  $P' = (b; a)$  — конъюнктивный член  $P$ ,  $\diamond \in \{-, \delta\} \Rightarrow$   
 $P \parallel \diamond a \rightarrow [P]_b^{P'} \parallel \delta_a$   
(1.1, 2.1, 5.2, 6.1, 6.3, 6.5);

- 7.7.  $P$  — 1,2-конъюнкция,  $P'$  — конъюнктивный член  $P$ ,  $P' = \bar{a}$  или  $P' = \delta_a \Rightarrow$   
 $P \parallel (a; b) \rightarrow [P]_{\delta_a}^{P'} \parallel \delta_b$   
(1.1, 1.4, 2.1, 5.1, 6.1, 6.3, 6.4, 6.7);
- 7.8.  $P$  — 1,2-конъюнкция,  $P'$  — конъюнктивный член  $P$ ,  $P' = \bar{a}$  или  $P' = \delta_a \Rightarrow$   
 $P \parallel (b; a) \rightarrow [P]_{\delta_a}^{P'} \parallel b$   
(1.1, 2.1, 5.2, 6.1, 6.3, 6.5);
- 7.9.  $P$  — 1,2-конъюнкция,  $P' = Q$  — конъюнктивный член  $P \Rightarrow$   
 $P \parallel Q \rightarrow P$   
(1.1, 2.1, 5.7);
- 8.1.  $P$  — 1,2,3-конъюнкция,  $P' = (a; b)$  — конъюнктивный член  $P$ , в максимальной 1,2,3-конъюнкции, содержащей  $P$  в качестве конъюнктивного члена, нет члена  $P'' = (a; c) \Rightarrow$   
 $P \parallel (b; c) \rightarrow (P \parallel (b; c)) \parallel (a; c)$   
(1.1, 2.1, 5.6);
- 8.2.  $P$  — 1,2,3-конъюнкция,  $P' = (c; a)$  — конъюнктивный член  $P$ , в максимальной 1,2,3-конъюнкции, содержащей  $P$  в качестве конъюнктивного члена, нет члена  $P'' = (b; a) \Rightarrow$   
 $P \parallel (b; c) \rightarrow (P \parallel (b; c)) \parallel (b; a)$   
(1.1, 2.1, 5.6);
- 9.1.  $P$  — 1-дизъюнкция,  $P'$  — дизъюнктивный член  $P$ ,  $P' \simeq Q \Rightarrow$   
 $P \vee Q \rightarrow P$   
(1.1, 1.3, 2.1, 2.3, 5.8);
- 10.1.  $P$  — 1,2-дизъюнкция,  $Q$  — нормальная конъюнкция,  $P'$  — дизъюнктивный член  $P$ ,  $Q \triangleleft P' \Rightarrow$   
 $P \vee Q \rightarrow P$   
(1.3, 2.3, 5.9);
- 10.2.  $P$  — 1,2-дизъюнкция,  $Q$  — нормальная конъюнкция,  $P'$  — дизъюнктивный член  $P$ ,  $P' \triangleleft Q \Rightarrow$   
 $P \vee Q \rightarrow [P]_Q^{P'}$   
(1.3, 2.3, 5.9).

Сделаем краткий обзор системы правил переписывания.

Для избежания бесконечных цепочек вывода вида  $P \circ (Q \circ R) \rightarrow (P \circ Q) \circ R \rightarrow P \circ (Q \circ R) \rightarrow \dots$ ,  $\circ \in \{;, \parallel, \vee\}$  вводится правило левой ассоциативности 1.1.

В систему  $RWS_2$  нельзя включать правила коммутативности, применение которых может привести к бесконечным цепочкам вида  $P \circ Q \rightarrow Q \circ P \rightarrow P \circ Q \rightarrow \dots$ ,  $\circ \in \{\parallel, \vee\}$ . Поэтому дополнительно вводятся симметричные правила, необходимые при отсутствии правил коммутативности. На этой идее основаны правила дистрибутивности группы 2.

**Пример 3.1.6** В системе  $\Theta_{F_2}$  нет аксиомы, симметричной аксиоме 3.5  $(P \vee Q) \parallel R = (P \parallel R) \vee (Q \parallel R)$ . Поэтому, если мы не включим в нашу систему нового симметричного правила, основанного на аксиоме  $P \parallel (Q \vee R) = (P \parallel Q) \vee (P \parallel R)$ , мы не сможем преобразовать формулу  $a \parallel (b \vee c)$  к виду  $(a \parallel b) \vee (a \parallel c)$ .

Правило 3.1 позволяет избавиться от символа  $\nabla$ , а правила группы 4 — от символов  $\parallel$  и  $\tilde{\parallel}$ .

Правила группы 5 используются для удовлетворения свойству 1 из определения нормальной конъюнкции.

В связи с отсутствием правил коммутативности возникают и трудности другого рода, связанные с удалением друг от друга конъюнктивных (дизъюнктивных) членов формулы, к которым можно применить некоторую аксиому. Тогда в дополнение к подходу, основанному на введении симметричных правил, применяется другой метод. Рассмотрим формулу  $P \parallel Q$  (или  $P \vee Q$ ). В подформуле  $P$ , являющейся конъюнкцией (дизъюнкцией), ищется подформула  $P'$  такая, что к  $P' \parallel Q$  (к  $P' \vee Q$ ) применима некоторая аксиома из  $\Theta_{F_2}$ .

На этой идее основаны правила группы 6, предназначенные для удовлетворения свойству 2 нормальной конъюнкции, группы 7, необходимые для выполнения свойства 3, правила группы 8 для удовлетворения свойству 4, а также правила групп 9 и 10 для выполнения свойств 2 и 3 канонической формы соответственно.

**Пример 3.1.7** Аксиома 6.1  $a\|\bar{a} = \delta_a$  не применима при отсутствии правил коммутативности к формулам  $a\|b\|\bar{a}$  и  $\bar{a}\|b\|a$ . Поэтому вводятся симметричные правила 7.3 и 7.4. В первом случае, при применении правила 7.3 ( $P = a\|b$ ,  $P' = a$ ) получаем формулу  $\delta_a\|b$ . Во втором случае эту же формулу получаем по правилу 7.4 ( $P = \bar{a}\|b$ ,  $P' = \bar{a}$ ).

Для избежания бесконечных цепочек вывода вида  $(a; b)\|(b; c) \rightarrow ((a; b)\|(b; c))\|(a; c) \rightarrow (((a; b)\|(b; c))\|(a; c))\|(a; c) \rightarrow \dots$  в правилах группы 8 конъюнктивный член — транзитивное замыкание отношения предшествования — ищется в максимальной 1,2,3-конъюнкции, содержащей данную. Правило применяется только тогда, когда такого члена в максимальной 1,2,3-конъюнкции нет. Этот прием предохраняет от бесконечного увеличения длины формулы в процессе ее приведения.

Необходимо отметить, что правила 6.1, 6.2, 7.5–7.8 основаны на новых аксиомах, полученных из аксиом  $\Theta_{F_2}$ , доказательство которых приведено ниже. Цифры над знаками равенства — номера применяемых аксиом. Звездочка означает, что аксиома применяется справа налево. Цифры в скобках — номера новых, только что доказанных здесь аксиом.

1.  $\bar{a}\|(a; b) \stackrel{5.1*}{=} \bar{a}; (a; b) \stackrel{1.4}{=} (\bar{a}; a); b \stackrel{5.1}{=} (\bar{a}\|a); b \stackrel{2.1}{=} (a\|\bar{a}); b \stackrel{6.1}{=} \delta_a; b \stackrel{6.4}{=} \delta_a\|(\bar{\bar{b}}) \stackrel{6.7}{=} \delta_a\|\delta_b;$
2.  $\delta_a\|(a; b) \stackrel{6.1*}{=} (a\|\bar{a})\|(a; b) \stackrel{1.1*}{=} a\|(\bar{a}\|(a; b)) \stackrel{(1)}{=} a\|(\delta_a\|\delta_b) \stackrel{1.1}{=} (a\|\delta_a)\|\delta_b \stackrel{6.3}{=} \delta_a\|\delta_b;$
3.  $\bar{a}\|(b; a) \stackrel{2.1}{=} (b; a)\|\bar{a} \stackrel{5.2*}{=} (b; a); \bar{a} \stackrel{1.1*}{=} b; (a; \bar{a}) \stackrel{5.2}{=} b; (a\|\bar{a}) \stackrel{6.1}{=} b; \delta_a \stackrel{6.5}{=} b\|\delta_a \stackrel{2.1}{=} \delta_a\|b;$
4.  $\delta_a\|(b; a) \stackrel{6.1*}{=} (a\|\bar{a})\|(b; a) \stackrel{1.1*}{=} a\|(\bar{a}\|(b; a)) \stackrel{(3)}{=} a\|(\delta_a\|b) \stackrel{1.1}{=} (a\|\delta_a)\|b \stackrel{6.3}{=} \delta_a\|b;$
5.  $\delta_a\|\bar{b} \stackrel{4.5*}{=} \delta_a\|(\bar{\bar{b}}) \stackrel{6.6}{=} \delta_a\|(\bar{\bar{b}}) \stackrel{6.7}{=} \delta_a\|\delta_b.$

### Доказательство сходимости $RWS_2$

Докажем серию предложений о системе  $RWS_2$ , гарантирующих, что по окончании процесса приведения формулы мы получим одну из изоморфных между собой канонических форм.

**Предложение 3.1.4** К формуле  $AFP_2$  не применимо ни одно из правил групп 1–5 тогда и только тогда, когда она является дизъюнкцией 1-конъюнкций.

*Доказательство.* ( $\Rightarrow$ ) Заметим, что, если к формуле не применимо правило 1.1, то все скобки, объединяющие подформулы, соединенные одинаковыми знаками операций, смещены влево. В дальнейшем будем предполагать, что все формулы уже обладают этим свойством.

Если к формуле не применимы правила 1.1 и 3.1, то в этой формуле нет операции  $\nabla$ .

Очевидно также, что с помощью правил группы 4 мы избавляемся от  $\bar{\bar{\quad}}$  и  $\bar{\bar{\bar{\quad}}}$ . Если в формуле есть эти символы, стоящие перед сложными подформулами, то к ней обязательно применимы правила группы 4, смещающие знаки операций  $\bar{\bar{\quad}}$  и  $\bar{\bar{\bar{\quad}}}$  к элементарным подформулам видов  $a$ ,  $\bar{a}$ ,  $\delta_a$ . Затем, при применении правил 4.3 и 4.4, знаки  $\bar{\bar{\quad}}$  и  $\bar{\bar{\bar{\quad}}}$  исчезают. Итак, если к формуле не применимы правила групп 1, 3, 4, эта формула без символов  $\nabla$ ,  $\bar{\bar{\quad}}$  и  $\bar{\bar{\bar{\quad}}}$ .

Если, в дополнение к этому, к формуле не применимы правила группы 2, приводящие формулу к виду  $P = \bigvee_{i=1}^m \|\bigvee_{j=1}^{n_i} P_{ij}$ , то формула является дизъюнкцией конъюнкций с членами  $P_{ij}$ , являющимися предшествованиями или элементарными формулами.

Если к тому же к формуле не применимо правило 5.1, то можно заключить, что она — дизъюнкция конъюнкций с членами вида  $P_{ij} = (Q_{ij}; R_{ij})$  или  $P_{ij} = Q_{ij}$ , где  $Q_{ij}, R_{ij} \in \hat{\alpha}$ .

Правила 5.2–5.6 позволяют избавиться от символов из  $\bar{a} \cup \Delta_\alpha$  и одинаковых символов в двучленных предшествованиях. Таким образом, если к формуле не применимы правила групп 1–5, она является дизъюнкцией конъюнкций с членами вида  $a$ ,  $\bar{a}$ ,  $\delta_a$  или  $(a; b)$ ,  $a \neq b$ , то есть эта формула — дизъюнкция 1-конъюнкций.

( $\Leftarrow$ ) Так как формула — дизъюнкция 1-конъюнкций, в ней все скобки, объединяющие подформулы, соединенные одинаковыми знаками операций, смещены влево. Значит, к ней не применимо правило 1.1. Кроме того, формула имеет вид  $P = \bigvee_{i=1}^m \|\bigvee_{j=1}^{n_i} P_{ij}$ , где  $P_{ij}$  — элементарная формула или элементарное предшествование, то есть к формуле не применимы правила группы 2. Так как она не содержит символов  $\nabla$ ,  $\bar{\bar{\quad}}$ ,  $\bar{\bar{\bar{\quad}}}$ , то к ней не применимы правила групп 3, 4. Из элементарности конъюнктов-предшествований следует неприменимость правил группы 5.  $\square$

**Предложение 3.1.5** *К формуле  $AFP_2$  не применимо ни одно из правил групп 1–6 тогда и только тогда, когда она является дизъюнкцией 1,2-конъюнкций.*

*Доказательство.* ( $\Rightarrow$ ) По предложению 3.1.4, наша формула — дизъюнкция 1-конъюнкций. Пусть для формулы не выполняется свойство 2 нормальной конъюнкции. Значит, в формуле есть дизъюнкт, в котором имеются конъюнктивные члены видов  $\bar{b}$  и  $\delta_a$ . Так как формула — дизъюнкция 1-конъюнкций, то этот дизъюнкт — 1-конъюнкция. Таким образом, к ней применимо правило 6.1 или 6.2, что противоречит исходному предположению.

( $\Leftarrow$ ) По предложению 3.1.4, к формуле не применимы правила групп 1–5. Правила группы 6 не применимы, так как любой ее дизъюнктивный член не содержит в качестве подформул одновременно элементы из  $\bar{a}$  и  $\delta_a$ .  $\square$

**Предложение 3.1.6** *К формуле  $AFP_2$  не применимо ни одно из правил групп 1–7 тогда и только тогда, когда она является дизъюнкцией 1,2,3-конъюнкций.*

*Доказательство.* ( $\Rightarrow$ ) По предложению 3.1.5, формула — дизъюнкция 1,2-конъюнкций. Достаточно доказать, что из неприменимости правил группы 7 следует выполнение условия 3 нормальной конъюнкции для всех дизъюнктивных членов формулы. Рассмотрим ситуации, когда для дизъюнктивного члена формулы, 1,2-конъюнкции  $P_i = \prod_{j=1}^{n_i} P_{ij}$ , выполняется условие  $\alpha(P_{ik}) \cap \alpha(P_{il}) \neq \emptyset$  ( $1 \leq k \neq l \leq n$ ) и  $P_{ik}$ ,  $P_{il}$  не являются различными элементарными предшествованиями, то есть не выполняется условие 3. Мы покажем, что во всех этих ситуациях к формуле применимы правила, уменьшающие число таких пар дизъюнктов.

1. Конъюнкты видов  $\bar{a}$  и  $\delta_a a$ . Для нашей формулы этот случай не имеет места из-за того, что для нее выполняется свойство 2 нормальной конъюнкции.
2. Конъюнктивные члены имеют вид  $a$  и  $(a; b)$  или  $b$  и  $(a; b)$ . Тогда к конъюнкции, содержащей эти члены, применимо правило 7.1 или 7.2.
3. Конъюнктивные члены имеют вид  $a$  и  $\bar{a}$  или  $a$  и  $\delta_a$  или  $\bar{a}$  и  $\delta_a$ . Тогда к соответствующей конъюнкции применимо одно из правил 7.3, 7.4.
4. Конъюнктивные члены имеют вид  $\bar{a}$  и  $(a; b)$  или  $\bar{b}$  и  $(a; b)$ , или  $\delta_a$  и  $(a; b)$ , или  $\delta_b$  и  $(a; b)$ . Тогда к конъюнкции применимо одно из правил 7.5–7.8.
5. Конъюнктивные члены имеют вид  $a$  и  $a$  или  $\bar{a}$  и  $\bar{a}$ , или  $\delta_a$  и  $\delta_a$ , или  $(a; b)$  и  $(a; b)$ , то есть совпадают. Тогда к конъюнкции применимо правило 7.9, уничтожающее лишний конъюнктивный член.

Связи между конъюнктивными членами наглядно представляются схемой на рисунке 3.5, в которой линии с номерами пунктов доказательства этого предложения соединяют формулы, которые могут быть на месте  $P_{ik}$  и  $P_{il}$ . Эта схема показывает, что в предложении рассмотрены все случаи невыполнимости условия 3 нормальной конъюнкции.

Значит, если для формулы не выполняется условие 3, к ней обязательно применимо какое-либо из правил группы 7, что противоречит исходному утверждению. Итак, при неприменимости к формуле правил групп 1–7 можно утверждать, что она — дизъюнкция 1,2,3-конъюнкций.

( $\Leftarrow$ ) По предложению 3.1.5, к формуле не применимы правила групп 1–6. Правила группы 7 не применимы, так как в дизъюнктах 1,2,3-дизъюнкции нет конъюнктивных членов — не элементарных предшествований, содержащих одинаковые символы.  $\square$

**Предложение 3.1.7** *К формуле  $AFP_2$  не применимо ни одно из правил групп 1–8, тогда и только тогда, когда она является 1-дизъюнкцией.*

*Доказательство.* ( $\Rightarrow$ ) По предложению 3.1.6, формула — дизъюнкция 1,2,3-конъюнкций. Покажем, что из неприменимости правил группы 8 следует выполнение условия 4 нормальной конъюнкции для всех ее дизъюнктивных членов. Если правила 8.1 и 8.2 не применимы к формуле, то в каждом ее дизъюнктивном члене содержатся все транзитивные замыкания отношения предшествования, то есть для любого дизъюнкта выполняется условие 4 нормальной конъюнкции. В заключение заметим, что 1,2,3,4-конъюнкция — нормальная конъюнкция, а дизъюнкция нормальных конъюнкций — 1-дизъюнкция.

( $\Leftarrow$ ) По предложению 3.1.6, к формуле не применимы правила групп 1–7. Так как формула — 1-дизъюнкция, в любом ее дизъюнктивном члене содержатся все транзитивные замыкания отношения предшествования, то есть правила группы 8 не применимы.  $\square$

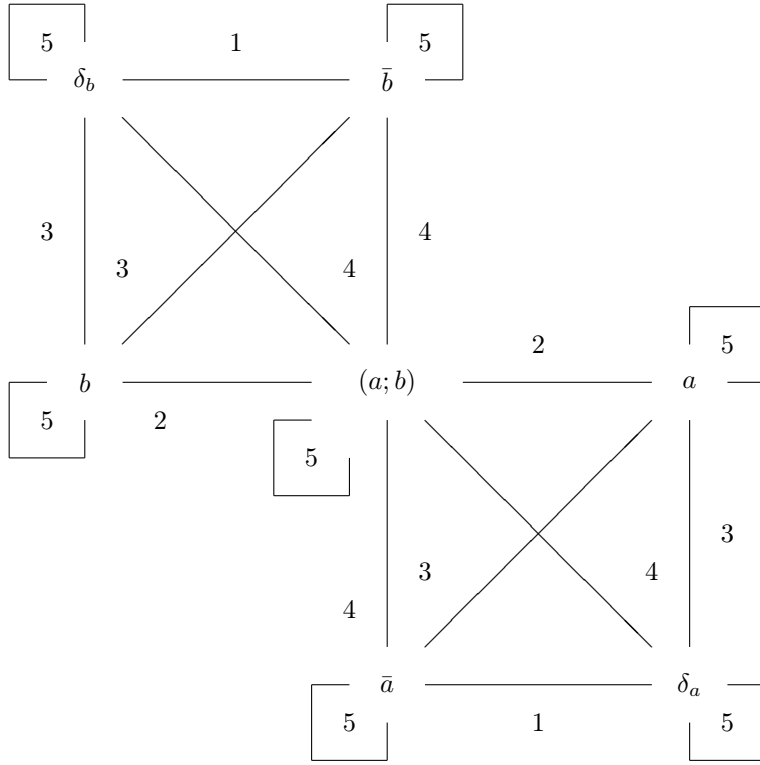


Рис. 3.5: Конъюнкты с пересекающимися алфавитами

**Предложение 3.1.8** *К формуле  $AFP_2$  не применимо ни одно из правил групп 1–9 тогда и только тогда, когда она — 1,2-дизъюнкция.*

*Доказательство.* ( $\Rightarrow$ ) По предложению 3.1.7, формула — 1-дизъюнкция. Если правило 9.1 не применимо, то все дизъюнктивные члены различны, то есть формула — 1,2-дизъюнкция.

( $\Leftarrow$ ) По предложению 3.1.7, к формуле не применимы правила групп 1–8. Правила группы 9 не применимы из-за того, что по свойству 2 канонической формы в ней все дизъюнктивные члены различны.  $\square$

**Предложение 3.1.9** *К формуле  $AFP_2$  не применимо ни одно из правил  $RWS_2$  тогда и только тогда, когда она находится в канонической форме.*

*Доказательство.* ( $\Rightarrow$ ) По предложению 3.1.8, формула — 1,2-дизъюнкция. Если правила группы 10 не применимы к формуле, то в ней нет дизъюнктов, один из которых — префикс другого, то есть выполняется свойство 3 канонической формы, а 1,2,3-дизъюнкция — это каноническая форма формулы.

( $\Leftarrow$ ) По предложению 3.1.8, к формуле не применимы правила групп 1–9. Правила группы 10 не применимы, так как в канонической форме нет ни одной пары префиксных один другому дизъюнктов.  $\square$

Таким образом, для проверки двух формул на эквивалентность достаточно, используя систему  $RWS_2$ , автоматически привести их к каноническим формам, а затем проверить эти канонические формы на изоморфизм.

Автором написана программа **CANON**, основанная на предложениях данного пункта, выполняющая автоматическое преобразование произвольной формулы  $AFP_2$  к каноническому виду. Описание и примеры работы этой программы можно найти в приложениях В и С.

## 3.2 Алгебра конечных помеченных процессов $AFLP_2$

Достоинство рассмотренной выше алгебры  $AFP_2$  — механизм синхронизации действий по имени, близкий к сетевому, позволяющий описывать процессы, которые нельзя представить формулами других алгебр

(например, CCS, TCSP, ACP, алгебры структур событий [34, 35]), в которых с каждым вхождением действия в формулу связывается уникальное событие [98]. В [128] дается семантика CCS на основе сетей Петри. Формулам CCS сопоставляются С/Е-системы (подкласс сетей Петри). Показано, что трансляция в обратном направлении (С/Е-сетей в формулы CCS) невозможна из-за бинарности операции синхронизации в CCS, тогда как в сетях Петри синхронизация может быть не только бинарной, так как у перехода могут быть более чем два предусловия. В этой же работе утверждается, что оператор синхронизации необходимо расширить, чтобы решить данную проблему.

В то же время в рамках алгебры  $AFP_2$  нельзя, например, задать процесс, в котором два одноименных действия выполняются параллельно.

Алгебра  $AFLP_2$  (Algebra of Finite Labelled Processes) строится на основе алгебры  $AFP_2$  с помощью введения глобальной пометки на символах событий, которые комбинируются в формулы. Таким образом, формулы алгебры  $AFLP_2$  описывают *помеченные* недетерминированные процессы, в которых два различных события могут быть помечены одним и тем же действием, в отличие от формул  $AFP_2$ , специфицирующих обычные (не помеченные) недетерминированные процессы. Следовательно, в рамках этой алгебры можно описать значительно более широкий класс процессов, чем в  $AFP_2$ .

Аналогично алгебре  $AFP_2$ , алгебра  $AFLP_2$  является как описательной, так и аналитической. Это значит, что средства  $AFLP_2$  позволяют как описывать процесс, так и анализировать его свойства.

От алгебры  $AFP_2$  алгебра  $AFLP_2$  унаследовала механизм не-действий и тупиковых действий, позволяющий сохранять информацию о недетерминизме. На базе помеченных частично упорядоченных множеств (ПЧУМ) в  $AFLP_2$  вводится денотационная семантика в виде отображения, сопоставляющего каждой формуле этой алгебры совокупность ПЧУМ с не-событиями и тупиковыми событиями. На основе денотационной семантики определяется эквивалентность формул  $AFLP_2$ . Строится система аксиом, соответствующая данной эквивалентности и доказывается полнота этой системы.

Отмечена возможность использования программы CANON для автоматизации проверки семантической эквивалентности  $AFLP_2$ .

Определяется операционная семантика  $AFLP_2$  на базе системы переходов, основанной на ПЧУМ. В этой системе преобразование одной формулы в другую по правилам перехода интерпретируется как выполнение некоторого ПЧУМ.

Устанавливается соответствие между денотационной и операционной семантиками алгебры  $AFLP_2$ .

Также показано, что средствами  $AFLP_2$  можно анализировать поведение слабо помеченных А-сетей (то есть А-сетей, помечающая функция которых не обязательно инъективна). Рассмотренные ранее эквивалентности изучены на этом подклассе сетей. Эквивалентности на формулах  $AFLP_2$  переносятся на слабо помеченные А-сети и исследуется их взаимосвязь с сетевыми эквивалентностями.

На формулах  $AFLP_2$  вводятся аналоги сетевых эквивалентностей и доказывается их согласованность с исходными эквивалентностями на сетях.

В заключение устанавливается, что эквивалентность относительно денотационной семантики  $AFLP_2$  — единственная из рассмотренных эквивалентностей на формулах этой алгебры, которая является конгруэнтностью.

### 3.2.1 Синтаксис

Пусть  $Ev = \{e, f, \dots\}$  — алфавит символов (*обычных*) событий,  $\bar{Ev} = \{\bar{e}, \bar{f}, \dots\}$  — алфавит символов *не-событий* и  $\Delta_{Ev} = \{\delta_e, \delta_f, \dots\}$  — алфавит символов *тупиковых событий*. Обозначим  $\widehat{Ev} = Ev \cup \bar{Ev} \cup \Delta_{Ev}$ . Символы  $\widehat{Ev}$  связываются с помощью операций ; (*предшествование*), || (*параллельность*),  $\nabla$  (*исключение* или *альтернатива*),  $\vee$  (*дизъюнкция* или *объединение*),  $\overline{\parallel}$  (“не случится”),  $\overline{\nabla}$  (“не случится ошибочно”). Введем  $Act = \{a, b, \dots\}$  — алфавит символов действий (*меток*). Глобальная функция пометки  $lab : Ev \rightarrow Act$  сопоставляет каждому событию некоторое действие. Эта функция расширяется на множество  $\bar{Ev} \cup \Delta_{Ev}$  следующим образом:  $lab(\bar{e}) = \overline{lab(e)}$  и  $lab(\delta_e) = \delta_{lab(e)}$ .

**Определение 3.2.1** Формула  $AFLP_2$  над алфавитом  $\widehat{Ev}$  определяется следующим образом.

$$E ::= e \mid \bar{e} \mid \delta_e \mid \overline{\parallel} E \mid \overline{\nabla} E \mid E; F \mid E \parallel F \mid E \nabla F \mid E \vee F$$

Здесь  $e \in Ev$ ,  $\bar{e} \in \bar{Ev}$ ,  $\delta_e \in \Delta_{Ev}$  — элементарные формулы.

Обозначим через  $\mathbf{AFLP}_2$  множество всех формул алгебры  $AFLP_2$ .

Пусть  $E$  — формула  $AFLP_2$ . Определим множество  $Ev(E)$  следующим образом.



1.  $Ev(e) = Ev(\bar{e}) = Ev(\delta_e) = e$ ;
2.  $Ev(\neg E) = Ev(E)$ ,  $\neg \in \{\llbracket, \tilde{\llbracket}\}$ ;
3.  $Ev(E \circ F) = Ev(E) \cup Ev(F)$ ,  $\circ \in \{;, \parallel, \nabla, \vee\}$ .

Определим также  $\overline{Ev}(E) = \{\bar{e} \mid e \in Ev(E)\}$ ,  $\Delta_{Ev}(E) = \{\delta_e \mid e \in Ev(E)\}$  и  $\widehat{Ev}(E) = Ev(E) \cup \overline{Ev}(E) \cup \Delta_{Ev}(E)$ .

С каждой формулой  $E$  алгебры  $AFLP_2$  можно связать *локальную функцию пометки*  $l_E = lab|_{Ev(E)}$ , помечающую символы событий, входящие в эту формулу.

Определим *содержимое* формулы  $E$ ,  $cont(E)$ , следующим образом.

1.  $cont(e) = e$ ,  $cont(\bar{e}) = \bar{e}$ ,  $cont(\delta_e) = \delta_e$ ;
2.  $cont(\neg E) = cont(E)$ ,  $\neg \in \{\llbracket, \tilde{\llbracket}\}$ ;
3.  $cont(E \circ F) = cont(E) \cup cont(F)$ ,  $\circ \in \{;, \parallel, \nabla, \vee\}$ .

Пусть также  $cont^+(E) = cont(E) \cap Ev$  — множество событий формулы  $E$ ,  $cont^-(E) = cont(E) \cap \overline{Ev}$  — множество не-событий формулы  $E$ ,  $\Delta_{cont}(E) = cont(E) \cap \Delta_{Ev}$  — множество тупиковых событий формулы  $E$ .

Пусть  $E$  и  $E'$  — формулы  $AFLP_2$ . Тогда  $E$  и  $E'$  *изоморфны*, обозначение  $E \simeq E'$ , если они совпадают с точностью до применения правил ассоциативности относительно  $;, \parallel, \nabla, \vee$  и коммутативности относительно  $\parallel, \nabla, \vee$ .

**Пример 3.2.1**  $(e \parallel f \parallel \bar{g}) \vee (g \parallel \bar{e} \parallel \bar{f}) \simeq (\bar{e} \parallel \bar{f} \parallel g) \vee (f \parallel e \parallel \bar{g})$ .

## 3.2.2 Семантика

### Денотационная семантика

Рассмотрим ПЧУМ  $\rho = \langle X, \prec, l \rangle$ , где:

- $X \subseteq \widehat{Ev}$ ;
- $\prec \subseteq X \times X$  — строгий частичный порядок над  $X$ , отношение предшествования;
- $l : Ev(X) \rightarrow Act$  — помечающая функция.

Заметим, что  $Ev(X) = \{e \mid (e \in X) \vee (\bar{e} \in X) \vee (\delta_e \in X)\}$ . Определим также  $\overline{Ev}(X) = \{\bar{e} \mid e \in Ev(X)\}$ ,  $\Delta_{Ev}(X) = \{\delta_e \mid e \in Ev(X)\}$  и  $\widehat{Ev}(X) = Ev(X) \cup \overline{Ev}(X) \cup \Delta_{Ev}(X)$ . Введем также следующие обозначения:  $X^+ = X \cap Ev$  — *подмножество событий*  $X$ ,  $X^- = X \cap \overline{Ev}$  — *подмножество не-событий*  $X$ ,  $\Delta_X = X \cap \Delta_{Ev}$  — *подмножество тупиковых событий*  $X$ .

В дальнейшем будем рассматривать ПЧУМ, удовлетворяющие следующим условиям.

1.  $e$ ,  $\bar{e}$  и  $\delta_e$  не встречаются в  $X$  вместе, то есть либо  $e$  входит в  $X$ , либо  $\bar{e}$ , либо  $\delta_e$ ;
2. отношение  $\prec$  иррефлексивно;
3.  $\forall x, y \in X^- \cup \Delta_X$  ( $x \not\prec y$ )  $\wedge$  ( $y \not\prec x$ ), то есть все элементы  $X^- \cup \Delta_X$  несравнимы;
4.  $\forall x \in X^+ \forall y \in X^- \cup \Delta_X$  ( $x \not\prec y$ )  $\wedge$  ( $y \not\prec x$ ), то есть все элементы  $X^+$  и  $X^- \cup \Delta_X$  несравнимы.

*Модифицированное объединение* ПЧУМ определяется следующим образом.

$$\rho \tilde{\cup} \rho' \begin{cases} \rho, & \rho' \trianglelefteq \rho; \\ \rho', & \rho \trianglelefteq \rho'; \\ \{\rho, \rho'\}, & \text{иначе.} \end{cases}$$

Модифицированное объединение удаляет вычисления, которые могут быть продолжены в другом поведении (детерминированном процессе) недетерминированного процесса, а также одинаковые вычисления.

Для определения денотационной семантики  $AFLP_2$  введем операции над ПЧУМ:  $;$  (*предшествование*),  $\parallel$  (*параллелизм*),  $\nabla$  (*альтернатива*),  $\llbracket$  (*не случится*),  $\tilde{\llbracket}$  (*не случится ошибочно*). Если ПЧУМ  $\rho$ , сконструированный с помощью этих операций, не удовлетворяет условиям 1–4

для ПЧУМ, приведенным выше, мы “корректируем” его с помощью новой операции *регуляризации*  $[\rho]$ . Эта операция отделяет максимальный префикс  $\rho$  “до” появления некоторых противоречий в спецификации процесса. Все события процесса, которые случаются “после” этих противоречий, считаются тупиковыми.

Пусть  $D_1 = \{\delta_e \mid (e \in X) \wedge (e \prec e)\} \cup \{\delta_e \mid (e \in X) \wedge (\bar{e} \in X)\} \cup \{\delta_e \mid (e \in X) \wedge (\delta_e \in X)\} \cup \{\delta_e \mid (\bar{e} \in X) \wedge (\delta_e \in X)\} \cup \Delta_X$ ,  $D_2 = \{\delta_e \mid (e \in X) \wedge (\delta_f \in D_1) \wedge (\delta_f \prec e)\}$  и  $D_3 = \{\delta_e \mid \bar{e} \in X\}$ . Определим множество  $D$  следующим образом.

$$D \begin{cases} \emptyset, & D_1 = \emptyset; \\ D_1 \cup D_2 \cup D_3, & \text{иначе.} \end{cases}$$

Тогда  $[\rho] = \langle D, \emptyset, l|_{Ev(D)} \rangle \cup \langle Y, \prec \cap (Y \times Y), l|_{Ev(Y)} \rangle$ , где  $Y = X \setminus \widehat{Ev}(D)$ . Легко убедиться, что если ПЧУМ  $\rho$  удовлетворяет условиям 1–4, то  $[\rho] = \rho$ .

Определим операции над ПЧУМ следующим образом. Пусть  $\rho = \langle X, \prec, l \rangle$ ,  $\rho' = \langle X, \prec', l' \rangle$ .

**Не случится**  $\parallel \rho = \langle \overline{Ev}(X), \emptyset, l \rangle$ .

**Не случится ошибочно**  $\tilde{\parallel} \rho = \langle \Delta_{Ev}(X), \emptyset, l \rangle$ .

**Предшествование**  $\rho; \rho' = [\langle X \cup X', \prec \cup \prec' \cup (X^+ \times (X')^+) \cup (\Delta_X \times (X')^+), l \cup l' \rangle]$ .

**Параллельность**  $\rho \parallel \rho' = [\langle X \cup X', (\prec \cup \prec')^*, l \cup l' \rangle]$ , где  $(\prec \cup \prec')^*$  — транзитивное замыкание отношения  $\prec \cup \prec'$ .

**Альтернатива**  $\rho \nabla \rho' = [\langle X \cup \overline{Ev}(X'), \prec, l \cup l' \rangle] \tilde{\cup} [\langle \overline{Ev}(X) \cup X', \prec', l \cup l' \rangle]$ . Заметим, что  $\rho \nabla \rho'$  — не ПЧУМ, а совокупность ПЧУМ, описывающих альтернативные поведения недетерминированного процесса. То есть если  $\rho$  случится, то  $\rho'$  — не случится и наоборот.

Операции над ПЧУМ расширяются на совокупности ПЧУМ следующим образом. Пусть  $\mathcal{P} = \cup_{i=1}^n \rho_i$  и  $\mathcal{P}' = \cup_{j=1}^m \rho'_j$  — совокупности ПЧУМ. Тогда  $\neg \mathcal{P} = \tilde{\cup}_{i=1}^n \neg \rho_i$ , где  $\neg \in \{\parallel, \tilde{\parallel}\}$  и  $\mathcal{P} \circ \mathcal{P}' = \tilde{\cup}_{i=1}^n (\cup_{j=1}^m \rho_i \circ \rho'_j)$ , где  $\circ \in \{;, \parallel, \nabla\}$ .

Недетерминированный параллельный процесс характеризуется совокупностью ПЧУМ, связанных со всеми его альтернативными поведениями.

**Определение 3.2.2** Денотационная семантика  $AFLP_2$  — отображение  $\mathcal{D}_{FL2}$  из множества формул  $AFLP_2$  в совокупность ПЧУМ, определяемое следующим образом.

1.  $\mathcal{D}_{FL2}[e] = \langle \{e\}, \emptyset, l_e \rangle$ ,  $\mathcal{D}_{FL2}[\bar{e}] = \langle \{\bar{e}\}, \emptyset, l_e \rangle$ ,  $\mathcal{D}_{FL2}[\delta_e] = \langle \{\delta_e\}, \emptyset, l_e \rangle$ , где  $l_e = (e, lab(e))$ ;
2.  $\mathcal{D}_{FL2}[\neg E] = \neg \mathcal{D}_{FL2}[E]$ ,  $\neg \in \{\parallel, \tilde{\parallel}\}$ ;
3.  $\mathcal{D}_{FL2}[E \circ F] = \mathcal{D}_{FL2}[E] \circ \mathcal{D}_{FL2}[F]$ ,  $\circ \in \{;, \parallel, \nabla\}$ ;
4.  $\mathcal{D}_{FL2}[E \vee F] = \mathcal{D}_{FL2}[E] \tilde{\cup} \mathcal{D}_{FL2}[F]$ .

**Определение 3.2.3** Формулы  $AFLP_2$   $E$  и  $E'$  эквивалентны относительно денотационной семантики  $\mathcal{D}_{FL2}$ , обозначение  $E =_{\mathcal{D}_{FL2}} E'$ , если  $\mathcal{D}_{FL2}[E] = \mathcal{D}_{FL2}[E']$ .

Если  $\rho = \langle X, \prec, l \rangle$  — ПЧУМ, то  $\rho^+ = \langle X^+, \prec, l|_{X^+} \rangle$  — ПЧУМ, соответствующий “наблюдаемой” части  $\rho$  над множеством  $Ev$ . Для каждой формулы  $E$  алгебры  $AFLP_2$   $\mathcal{D}_{FL2}[E] = \cup_{i=1}^n \rho_i$  — совокупность ПЧУМ, характеризующая помеченный процесс, определяемый этой формулой. “Наблюдаемую” часть этого множества определим так:  $\mathcal{D}_{FL2}^+[E] = \cup_{i=1}^n \rho_i^+$ .

**Определение 3.2.4** Формулы  $E$  и  $E'$  наблюдаемо эквивалентны относительно денотационной семантики  $\mathcal{D}_{FL2}$ , обозначение  $E =_{\mathcal{D}_{FL2}^+} E'$ , если  $\mathcal{D}_{FL2}^+[E] = \mathcal{D}_{FL2}^+[E']$ .

*Контекст*,  $\mathcal{C}$ , — выражение, представляющее собой формулу  $AFLP_2$ , в которой ноль или более подформул заменены “пустыми местами”, куда могут быть подставлены другие формулы  $AFLP_2$  [99].  $\mathcal{C}[E]$  будет обозначать помещение формулы  $E$  в каждое из таких пустых мест.

**Предложение 3.2.1** Для любых формул  $E$  и  $E'$  алгебры  $AFLP_2$   $E =_{\mathcal{D}_{FL2}} E' \Leftrightarrow \forall \mathcal{C} \mathcal{C}[E] =_{\mathcal{D}_{FL2}} \mathcal{C}[E']$ .

*Доказательство.* Как лемма 5.1 в [99]. □

Таким образом,  $=_{\mathcal{D}_{FL2}}$  — конгруэнтность относительно операций алгебры  $AFLP_2$ . Заметим, что  $=_{\mathcal{D}_{FL2}^+}$  не является конгруэнтностью, что демонстрируется следующим примером.

**Пример 3.2.2** Пусть  $E = e \nabla f$ ,  $E' = (e \nabla f) \| e \| f$  и  $lab(e) = a$ ,  $lab(f) = b$ ,  $lab(g) = c$ . Тогда  $\mathcal{D}_{FL2}^+[E] = \mathcal{D}_{FL2}^+[E'] = \{\langle \{e\}, \emptyset, l_1 \rangle, \langle \{f\}, \emptyset, l_2 \rangle\}$ , где  $l_1(e) = a$ ,  $l_2(f) = b$  и  $E =_{\mathcal{D}_{FL2}^+} E'$ . Но  $\mathcal{D}_{FL2}^+[E; g] = \{\langle \{e, g\}, \prec_1, l_1 \rangle, \langle \{f, g\}, \prec_2, l_2 \rangle\}$ , тогда как  $\mathcal{D}_{FL2}^+[E'; g] = \{\langle \{e\}, \emptyset, l_3 \rangle, \langle \{f\}, \emptyset, l_4 \rangle\}$ , где  $e \prec_1 g$ ,  $f \prec_2 g$ ,  $l_1(e) = l_3(e) = a$ ,  $l_2(f) = l_4(f) = b$ ,  $l_1(g) = l_2(g) = c$ , и  $E; g \neq_{\mathcal{D}_{FL2}^+} E'; g$ . Заметим, что в процессе, задаваемом формулой  $E'; g$ , в отличие от процесса с формулой  $E; g$ , действие с никогда не сможет сработать.

### Аксиоматизация

В соответствии с отношением эквивалентности  $=_{\mathcal{D}_{FL2}}$  вводится система аксиом  $\Theta_{FL2}$ .

В следующих равенствах  $E$ ,  $F$ ,  $G$  обозначают формулы  $AFLP_2$ , а  $e \in Ev$ ,  $\bar{e} \in \overline{Ev}$ ,  $\delta_e \in \Delta_{Ev}$ .

#### 1. Ассоциативность

- 1.1  $E \| (F \| G) = (E \| F) \| G$
- 1.2  $E \nabla (F \nabla G) = (E \nabla F) \nabla G$
- 1.3  $E \vee (F \vee G) = (E \vee F) \vee G$
- 1.4  $E; (F; G) = (E; F); G$

#### 2. Коммутативность

- 2.1  $E \| F = F \| E$
- 2.2  $E \nabla F = F \nabla E$
- 2.3  $E \vee F = F \vee E$

#### 3. Дистрибутивность

- 3.1  $(E \| F); G = (E; G) \| (F; G)$
- 3.2  $E; (F \| G) = (E; F) \| (E; G)$
- 3.3  $(E \vee F); G = (E; G) \vee (F; G)$
- 3.4  $E; (F \vee G) = (E; F) \vee (E; G)$
- 3.5  $(E \vee F) \| G = (E \| G) \vee (F \| G)$
- 3.6  $E \nabla (F \| G) = (E \nabla F) \| (E \nabla G)$

#### 4. Аксиомы для $\nabla$ и $\|$

- 4.1  $E \nabla F = (E \| (\| F)) \vee ((\| E) \| F)$
- 4.2  $\| (E \| F) = (\| E) \| (\| F)$
- 4.3  $\| (E \vee F) = (\| E) \vee (\| F)$
- 4.4  $\| (E; F) = (\| E) \| (\| F)$
- 4.5  $\| e = \bar{e}$
- 4.6  $\| \bar{e} = \bar{e}$
- 4.7  $\| \delta_e = \bar{e}$

#### 5. Структурные свойства

- 5.1  $\bar{e}; E = \bar{e} \| E$
- 5.2  $E; \bar{e} = E \| \bar{e}$
- 5.3  $E \| (E; F) = (E; F)$
- 5.4  $F \| (E; F) = (E; F)$
- 5.5  $E; F; G = (E; F) \| (F; G)$
- 5.6  $(E; F) \| (F; G) = (E; F) \| (F; G) \| (E; G)$
- 5.7  $E \| E = E$

$$5.8 \ E \vee E = E$$

$$5.9 \ E \vee F = E, \text{ если } F \triangleleft E \text{ (понятие строгого префикса } \triangleleft \text{ на формулах будет определено дальше)}$$

6. Аксиомы для тупиковых событий и  $\tilde{\parallel}$

$$6.1 \ e \parallel \bar{e} = \delta_e$$

$$6.2 \ e; e = \delta_e$$

$$6.3 \ e \parallel \delta_e = \delta_e$$

$$6.4 \ \delta_e; E = \delta_e \parallel (\tilde{\parallel} E)$$

$$6.5 \ E; \delta_e = E \parallel \delta_e$$

$$6.6 \ \delta_e \parallel (\parallel E) = \delta_e \parallel (\tilde{\parallel} E)$$

$$6.7 \ \tilde{\parallel} e = \delta_e$$

$$6.8 \ \tilde{\parallel} \bar{e} = \delta_e$$

$$6.9 \ \tilde{\parallel} \delta_e = \delta_e$$

$$6.10 \ \tilde{\parallel} (E \parallel F) = (\tilde{\parallel} E) \parallel (\tilde{\parallel} F)$$

$$6.11 \ \tilde{\parallel} (E; F) = (\tilde{\parallel} E) \parallel (\tilde{\parallel} F)$$

$$6.12 \ \tilde{\parallel} (E \vee F) = (\tilde{\parallel} E) \vee (\tilde{\parallel} F)$$

Система аксиом  $\Theta_{FL2}$  соответствует отношению эквивалентности  $=_{\mathcal{D}_{FL2}}$ , то есть если  $E = E'$  — аксиома  $\Theta_{FL2}$ , то  $E =_{\mathcal{D}_{FL2}} E'$ . Доказательство этого факта состоит в определении денотационных семантик  $E$  и  $E'$  и в проверке на равенство совокупностей ПЧУМ, соответствующих этим формулам. Это можно сделать непосредственно из определения операций над ПЧУМ.

Для доказательства *полноты* системы аксиом  $\Theta_{FL2}$  вводится понятие канонической формы формулы  $AFLP_2$ .

### Каноническая форма формул

Введем понятия, связанные со структурой формулы.

*Предшествование* — формула вида  $E_1; \dots; E_n = \parallel_{i=1}^n E_i$ ,  $E_i \in \widehat{Ev}$  ( $1 \leq i \leq n$ );

*Конъюнкция* — формула вида  $E_1 \parallel \dots \parallel E_n = \parallel_{i=1}^n E_i$ , где  $E_i$  — предшествования ( $1 \leq i \leq n$ ).

*Дизъюнкция* — формула вида  $E = E_1 \vee \dots \vee E_n = \vee_{i=1}^n E_i$ , где  $E_i$  — конъюнкции ( $1 \leq i \leq n$ ).

*Нормальная конъюнкция* — конъюнкция  $E = \parallel_{i=1}^n E_i$ , для которой истинны следующие утверждения.

1. Каждая формула  $E_i$  ( $1 \leq i \leq n$ ) имеет одну из следующих форм:

(а) элементарная формула  $e$  ( $e \in Ev$ ),  $\bar{e}$  ( $\bar{e} \in \overline{Ev}$ ),  $\delta_e$  ( $\delta_e \in \Delta_{Ev}$ );

(б) элементарное предшествование  $(e; f)$ , где  $e, f \in Ev$  и  $e \neq f$ ;

2. Если имеется формула  $E_i$  ( $1 \leq i \leq n$ ) в форме  $\delta_e$  ( $\delta_e \in \Delta_{Ev}$ ), тогда нет другой формулы  $E_j$  ( $1 \leq j \leq n$ ) такой, что  $E_j = \bar{f}$  ( $\bar{f} \in \overline{Ev}$ );

3. Для любых формул  $E_i$  и  $E_j$  ( $1 \leq i \neq j \leq n$ ) таких, что  $Ev(E_i) \cap Ev(E_j) \neq \emptyset$ ,  $E_i$  и  $E_j$  должны иметь форму различных элементарных предшествований;

4. Для любой пары  $E_i = (e; f)$  и  $E_j = (f; g)$  ( $1 \leq i \neq j \leq n$ ) существует формула  $E_k = (e; g)$  ( $1 \leq k \leq n$ ), описывающая транзитивное замыкание отношения предшествования для событий  $e, f$  и  $g$ .

Пусть  $E$  и  $F$  — нормальные конъюнкции. Формула  $E$  — *строгий префикс* формулы  $F$ , обозначение  $E \triangleleft F$ , если выполняются следующие условия.

1.  $cont^+(E) \subset cont^+(F)$ ;

2. элементарное предшествование  $(e; f)$  является конъюнктивным членом  $F$  и  $f \in cont^+(E)$  тогда и только тогда, когда  $(e; f)$  — конъюнктивный член  $E$ ;

Формула  $E$  — *префикс* формулы  $F$ , обозначение  $E \triangleleft F$ , если  $E \triangleleft F$  или  $E \simeq F$ .

**Пример 3.2.3** В формуле  $(e\|g\|\bar{f}\|\bar{h}\|\bar{k}) \vee (g\|\delta_e\|\delta_f\|\delta_h\|\delta_k) \vee (e\|\delta_f\|\delta_g\|\delta_h\|\delta_k) \vee ((f; h)\|(f; k)\|\bar{e}\|\bar{g})$  вторая и третья конъюнкции — строгие префиксы первой.

Формула  $E$  имеет каноническую форму, если она является дизъюнкцией  $E = \bigvee_{i=1}^n E_i$  и выполняются следующие условия.

1.  $E_i$  ( $1 \leq i \leq n$ ) — нормальная конъюнкция;
2. для любых  $E_i$  и  $E_j$  ( $1 \leq i \neq j \leq n$ )  $E_i \not\prec E_j$ ;
3. для любых  $E_i$  и  $E_j$  ( $1 \leq i \neq j \leq n$ )  $\neg(E_i \triangleleft E_j \vee E_j \triangleleft E_i)$ .

Будем обозначать множество всех канонических форм формулы  $E$  через  $\text{canon}(E)$ .

Заметим, что каждый дизъюнктивный член канонической формы формулы характеризует одно из возможных альтернативных поведений недетерминированного процесса, который описывается этой формулой, и является представлением ПЧУМ, соответствующего этому поведению.

**Пример 3.2.4** Формула  $(e\|g\|\bar{f}\|\bar{h}\|\bar{k}) \vee ((f; h)\|(f; k)\|\bar{e}\|\bar{g})$  имеет каноническую форму, которая является представлением двух ПЧУМ, соответствующих детерминированным процессам недетерминированного процесса, описываемого этой формулой.

Запись  $E =_{\Theta_{FL2}} E'$  означает, что равенство формул  $E$  и  $E'$  может быть доказано с использованием системы аксиом  $\Theta_{FL2}$ .

**Теорема 3.2.1** Любая формула  $AFLP_2$  может быть сведена с помощью системы аксиом  $\Theta_{FL2}$  к единственной до изоморфизма канонической форме.

*Доказательство.* Как теорема 6.1 в [99]. □

**Теорема 3.2.2** Для любых формул  $E$  и  $E'$  алгебры  $AFLP_2$  истинно  $E =_{\mathcal{D}_{FL2}} E' \Leftrightarrow E =_{\Theta_{FL2}} E'$ .

*Доказательство.* Как теорема 6.2 в [99]. □

Таким образом, мы можем выяснить, эквивалентны ли любые две формулы  $E$  и  $E'$  алгебры  $AFLP_2$  относительно денотационной семантики. Для этого достаточно привести эти формулы к каноническим формам  $F$  и  $F'$  и проверить их на изоморфизм.

Для автоматизации этой проверки можно использовать программу CANON, учитывая пометки формул.

## Операционная семантика

**Определение 3.2.5** Система переходов — четверка  $TS = \langle S, L, \rightarrow, s_{TS} \rangle$ , где:

- $S$  — множество состояний;
- $L$  — множество меток;
- $\rightarrow \subseteq S \times L \times S$  — множество переходов;
- $s_{TS}$  — начальное состояние.

Переход  $(s, a, \tilde{s})$  будем изображать как  $s \xrightarrow{a} \tilde{s}$ . Будем также рассматривать только конечные системы переходов, то есть системы с конечным множеством состояний.

Рассмотрим систему переходов над формулами алгебры  $AFLP_2$ . Пусть  $F$  — формула  $AFLP_2$ , имеющая каноническую форму. Тогда  $TS(F) = \langle \mathbf{AFLP}_2 \cup \{\nu\}, \mathbf{AFLP}_2, \rightarrow_{TS}, F \rangle$ , где:

- Множество состояний,  $\mathbf{AFLP}_2 \cup \{\nu\}$ , — множество формул  $AFLP_2$ , дополненное специальным символом “пустой” формулы  $\nu$ , обозначающей исчерпание, то есть то, что произошли все события процесса, задаваемого формулой. Предполагается, что для любой формулы  $E$  алгебры  $AFLP_2$   $E\|\nu = \nu\|E = E$  и  $\text{cont}(\nu) = \emptyset$ .
- Множество меток — совокупность конъюнкций алгебры  $AFLP_2$  над алфавитом  $Ev$ . Каждая из таких конъюнкций  $G$  является представлением ПЧУМ  $\rho_G = \langle \text{cont}(G), \prec_G^*, l_G \rangle$ , где  $e \prec_G f \Leftrightarrow (e; f)$  — конъюнктивный член  $G$ , а  $\prec_G^*$  — транзитивное замыкание отношения  $\prec_G$ .

- Переход  $E \xrightarrow{G} \tilde{E} \in \rightarrow_{TS}$  описывает преобразование формулы  $E$  в  $\tilde{E}$  как результат выполнения ПЧУМ  $\rho_G$ .
- Начальное состояние системы переходов — формула  $F$ .

Определим множество переходов  $TS(F)$  на основе следующих правил вывода.

1. Элементарное событие

$$1.1 \quad e \xrightarrow{e} \nu$$

2. Элементарное предшествование

$$2.1 \quad e; f \xrightarrow{e} f$$

$$2.2 \quad e; f \xrightarrow{e;f} \nu$$

3. Параллельность

$$3.1 \quad \frac{E \xrightarrow{G} \tilde{E}}{E \parallel F \xrightarrow{G} \tilde{E} \parallel F} \quad cont(G) \cap cont(F) = \emptyset$$

$$3.2 \quad \frac{F \xrightarrow{G} \tilde{F}}{E \parallel F \xrightarrow{G} E \parallel \tilde{F}} \quad cont(G) \cap cont(E) = \emptyset$$

$$3.3 \quad \frac{E \xrightarrow{G} \tilde{E}, F \xrightarrow{H} \tilde{F}}{E \parallel F \xrightarrow{G \parallel H} \tilde{E} \parallel \tilde{F}} \quad cont(G) \cap cont(\tilde{F}) = \emptyset, \quad cont(H) \cap cont(\tilde{E}) = \emptyset$$

4. Дизъюнкция

$$4.1 \quad \frac{E \xrightarrow{G} \tilde{E}}{E \vee F \xrightarrow{G} \tilde{E}} \quad cont(G) \not\subseteq cont(F)$$

$$4.2 \quad \frac{F \xrightarrow{G} \tilde{F}}{E \vee F \xrightarrow{G} \tilde{F}} \quad cont(G) \not\subseteq cont(E)$$

$$4.3 \quad \frac{E \xrightarrow{G} \tilde{E}, F \xrightarrow{H} \tilde{F}}{E \vee F \xrightarrow{G} \tilde{E} \vee \tilde{F}} \quad canon(G) \simeq canon(H)$$

Пусть  $\mathbf{TS}(F) = \{G \mid \exists \tilde{F} \quad F \xrightarrow{G} \tilde{F}\}$  — множество формул системы  $TS(F)$ . Если  $F \xrightarrow{G} \tilde{F}$  — некоторый переход системы  $TS(F)$ , и к формуле  $\tilde{F}$  не применимо ни одно из правил вывода этой системы, то  $\tilde{F}$  — конечная формула, а  $G$  — максимальная формула системы  $TS(F)$ . Легко заметить, что  $\tilde{F}$  — либо пустая формула  $\nu$ , либо конъюнкция символов из  $\overline{Ev}$  или  $\Delta_{Ev}$ , и что  $cont(G) \subseteq Ev$ . Конечная формула системы  $TS(F)$  содержит информацию о том, какие события не смогли произойти в данном поведении недетерминированного процесса, который описывает формула  $F$ . Обозначим через  $\mathbf{TS}_{\max}(F)$  множество максимальных формул системы  $TS(F)$ .

**Определение 3.2.6** Операционная семантика алгебры  $AFLP_2$  — отображение  $\mathcal{O}_{FL2}$  из  $\mathbf{AFLP}_2$  в совокупность ПЧУМ, определяемое следующим образом:  $\mathcal{O}_{FL2}[E] = \{\rho_{G \parallel \tilde{F}} \mid G \in \mathbf{TS}_{\max}(F) \wedge F \xrightarrow{G} \tilde{F}\}$ .

Для любой формулы  $E$  алгебры  $AFLP_2$   $\mathcal{O}_{FL2}[E] = \cup_{i=1}^n \rho_i$  — совокупность ПЧУМ, характеризующая помеченный процесс, определяемый этой формулой. “Наблюдаемую” часть этого множества определим следующим образом.  $\mathcal{O}_{FL2}^+[E] = \cup_{i=1}^n \rho_i^+$ .

Пусть  $E$  — нормальная конъюнкция  $AFLP_2$ . Обозначим через  $E^+$  формулу, полученную в результате удаления из  $E$  символов множества  $\overline{Ev} \cup \Delta_{Ev}$ .  $E^+$  определяется следующим образом.

1.  $e^+ = e$ ,  $\bar{e}^+ = \delta_e^+ = \nu$ ,
2.  $(e; f)^+ = e; f$ ,
3.  $(E \circ F)^+ = E^+ \circ F^+$ ,  $\circ \in \{\parallel, \vee\}$ .

Следующее предложение устанавливает взаимосвязь между максимальными формулами  $TS(F)$  и дизъюнктивными членами  $F$ .

**Предложение 3.2.2** Пусть  $F = \bigvee_{i=1}^n F_i$  — каноническая форма. Тогда:

1. Для любой  $G \in \mathbf{TS}_{\max}(F)$  и терминальной формулы  $\tilde{F}$ , для которых  $F \xrightarrow{G} \tilde{F}$  существует дизъюнктивный член  $F_j$  ( $1 \leq j \leq n$ ) формулы  $F$  такой, что  $G \parallel \tilde{F} \simeq F_j$ .
2. Для любого дизъюнктивного члена  $F_j$  ( $1 \leq j \leq n$ ) формулы  $F$  существуют  $G \in \mathbf{TS}_{\max}(F)$  и терминальная формула  $\tilde{F}$ , для которых  $F \xrightarrow{G} \tilde{F}$ , такие, что  $G \parallel \tilde{F} \simeq F$ .

*Доказательство.*

1. Имеем  $\bigvee_{i=1}^n F_i \xrightarrow{G} \tilde{F}$ . Так как  $\tilde{F}$  не содержит операций дизъюнкции, правила 4.1 и 4.2 были применены несколько раз к  $\bigvee_{i=1}^n F_i$ . Следовательно,  $\exists j$  ( $1 \leq j \leq n$ )  $F_j \xrightarrow{G} \tilde{F}$ . Так как  $\text{cont}(G) \subseteq Ev$  и  $\text{cont}(\tilde{F}) \subseteq \overline{Ev} \cup \Delta_{Ev}$ ,  $F_j \simeq F_j^+ \parallel \tilde{F}$  и  $F_j^+ \parallel \tilde{F} \xrightarrow{G} \tilde{F}$ . По правилу 3.1 имеем  $F_j^+ \xrightarrow{G} \nu$ . Индукцией по структуре формул легко доказать, что для некоторой нормальной конъюнкции  $E$   $E \xrightarrow{G} \nu$  влечет  $E = G$ . В нашем случае имеем  $F_j^+ = G$ . Таким образом,  $F_j \simeq F_j^+ \parallel \tilde{F} = G \parallel \tilde{F}$ .
2. Очевидно,  $F_j^+ \xrightarrow{F_j^+} \nu$ . Для конъюнкции  $\tilde{F}$  символов из  $\text{cont}^-(F_j)$  или  $\Delta_{\text{cont}(F_j)}$  (так как символы из  $\overline{Ev}$  и  $\Delta_{Ev}$  не могут встречаться в  $F$  вместе) имеем  $F_j \simeq F_j^+ \parallel \tilde{F}$ . По правилу 3.1  $F_j^+ \parallel \tilde{F} \xrightarrow{F_j^+} \tilde{F}$ . Следовательно,  $F_j \xrightarrow{F_j^+} \tilde{F}$ . По правилам 4.1, 4.2  $\bigvee_{i=1}^n F_i \xrightarrow{F_j^+} \tilde{F}$ , так как для дизъюнктивных членов канонической формы истинно следующее:  $\text{cont}(F_k) \not\subseteq \text{cont}(F_l)$  ( $1 \leq k \neq l \leq n$ ). Поэтому,  $F_j^+ = G \in \mathbf{TS}_{\max}(F)$ .  $\square$

Следующее предложение устанавливает взаимосвязь между наблюдаемой частью  $\mathcal{O}_{FL2}[E]$  и множеством максимальных формул  $TS(F)$ .

**Предложение 3.2.3** Пусть  $E$  — формула  $AFLP_2$  и  $F \in \text{canon}(E)$ . Тогда  $\mathcal{O}_{FL2}^+[E] = \{\rho_G \mid G \in \mathbf{TS}_{\max}(F)\}$ .

*Доказательство.* Пусть  $\rho_{G \parallel \tilde{F}} \in \mathcal{O}_{FL2}[E]$ . Так как  $\text{cont}(G) \subseteq Ev$  и  $\text{cont}(\tilde{F}) \subseteq \overline{Ev} \cup \Delta_{Ev}$ , имеем  $(G \parallel \tilde{F})^+ = G$ . Следовательно,  $\rho_{G \parallel \tilde{F}}^+ = \rho_{(G \parallel \tilde{F})^+} = \rho_G$ .  $\square$

Теперь мы можем сформулировать главный результат, касающийся взаимосвязи денотационной и операционной семантик  $AFLP_2$ .

**Теорема 3.2.3** Пусть  $E$  — формула  $AFLP_2$ . Тогда  $\mathcal{O}_{FL2}[E] = \mathcal{D}_{FL2}[E]$ .

*Доказательство.* Пусть  $F = \bigvee_{i=1}^n F_i \in \text{canon}(E)$ . По определению канонической формы имеет место следующее равенство:  $\mathcal{D}_{FL2}[E] = \bigcup_{i=1}^n \rho_{F_i}$ . Пусть  $\rho_{G \parallel \tilde{F}} \in \mathcal{O}_{FL2}[E]$ . По предыдущим предложениям, существует дизъюнктивный член  $F_j$  ( $1 \leq j \leq n$ ) формулы  $F$  такой, что  $G \parallel \tilde{F} \simeq F_j$ . Следовательно,  $\rho_{G \parallel \tilde{F}} = \rho_{F_j}$ , и мы имеем  $\mathcal{O}_{FL2}[E] \subseteq \mathcal{D}_{FL2}[E]$ . Включение в обратную сторону доказывается аналогично.  $\square$

### 3.2.3 Эквивалентности на слабо помеченных А-сетях

Подобно тому, как была введена пометка на формулах  $AFP_2$  и получена алгебра  $AFLP_2$ , описывающая помеченные недетерминированные процессы, можно ввести пометку на формулах  $AFP_0$  и получить алгебру  $AFLP_0$ . Так как формулы алгебры  $AFP_0$  описывают конечные А-сети, формулы алгебры  $AFLP_0$  будут описывать конечные *слабо помеченные А-сети* (то есть А-сети, помечающая функция которых не обязательно инъективна).

Определим отображение  $\Psi_L : \mathbf{AFLP}_0 \rightarrow \mathbf{AFLP}_2$  из множества формул  $AFLP_0$  в множество формул  $AFLP_2$  следующим образом.

1.  $\Psi_L(e) = e$ ,
2.  $\Psi_L(E;_{L_0} F) = E;_{L_2} F$ ,
3.  $\Psi_L(E \parallel_{L_0} F) = E \parallel_{L_2} F$ ,

$$4. \Psi_L(E \nabla_{L0} F) = E \nabla_{L2} F.$$

Заметим, что символом “ $L0$ ” помечены операции  $AFLP_0$ , а символом “ $L2$ ” — операции  $AFLP_2$ . Денотационная семантика алгебры  $AFLP_0$  — отображение  $\mathcal{D}_{L0}$ , сопоставляющее каждой формуле  $E$  этой алгебры множество максимальных  $C$ -подсетей (максимальных  $O$ -подсетей, в терминах [98]) конечной  $A$ -сети  $N$ , которая описывается этой формулой.

**Теорема 3.2.4** Пусть  $E_0$  — формула  $AFLP_0$  и  $E_2$  — формула  $AFLP_2$  такая, что  $E_2 = \Psi_L(E_0)$ . Тогда  $\{\rho_C \mid C \in \mathcal{D}_{L0}[E_0]\} = \mathcal{D}_{FL2}^+[E_2]$ .

*Доказательство.* Как теорема 4.3 в [98], с учетом дополнительной информации о пометке событий, входящих в формулы  $E_0$  и  $E_2$ .  $\square$

Таким образом, каждой формуле  $E_0$  алгебры  $AFLP_0$ , описывающей конечную слабо помеченную  $A$ -сеть  $N$ , соответствует формула  $E_2$  алгебры  $AFLP_2$  такая, что совокупность ПЧУМ максимальных  $C$ -подсетей  $N$  совпадает с совокупностью ПЧУМ максимальных детерминированных процессов недетерминированного процесса, задаваемого формулой  $E_2$ . Заметим, что результат теоремы справедлив также для произвольных (не только максимальных) начальных  $C$ -подсетей  $N$  и для произвольных детерминированных процессов, описываемых формулой  $E_2$ . В этом случае начальным  $C$ -подсетям будут соответствовать начальные детерминированные процессы.

Заметим также, что отображение  $\Psi_L$  всего лишь заменяет операции  $AFLP_0$  на операции  $AFLP_2$  в исходной формуле. Следовательно, если у нас есть конечная слабо помеченная  $A$ -сеть  $N$ , определяемая формулой  $E$  в  $AFLP_0$ , мы можем анализировать ее свойства и поведение посредством той же самой формулы  $E$  в  $AFLP_2$ .

**Пример 3.2.5** Рассмотрим, какие формулы  $E$  и  $E'$  алгебры  $AFLP_2$  соответствуют сетям  $N$  и  $N'$  в примерах на рисунках 3.6 и 3.7. Положим  $lab(e) = lab(e_i) = a$ ,  $lab(f) = lab(f_i) = b$ ,  $lab(g) = lab(g_i) = c$ ,  $lab(h) = lab(h_i) = d$  ( $1 \leq i \leq 3$ ).

- На рисунке 3.6(a)  $E = e \parallel f$ ,  $E' = (e_1; f_1) \nabla (f_2; e_2)$ .
- На рисунке 3.6(b)  $E = e \parallel f$ ,  $E' = (e; f_1) \parallel (f_1 \nabla f_2)$ .
- На рисунке 3.6(c)  $E = (e; f) \parallel (g; h)$ ,  $E' = (e; (f_1 \nabla f_2)) \parallel (e; (f_2 \nabla h_1)) \parallel (g; (f_2 \nabla h_1)) \parallel (g; (h_1 \nabla h_2)) \parallel (f_1 \nabla h_2)$ .
- На рисунке 3.6(d)  $E = e; f$ ,  $E' = (e_1; f) \nabla e_2$ .
- На рисунке 3.7(a)  $E = ((e_1 \nabla e_2); f_1) \parallel (f_1 \nabla f_2) \parallel e_1 \parallel e_2 \parallel f_2$ ,  $E' = ((e_1; f_1) \nabla (e_2; f_3)) \parallel (f_1 \nabla f_2) \parallel (e_2 \nabla f_2) \parallel e_1 \parallel f_3$ .
- На рисунке 3.7(b)  $E = (e; f; h) \parallel (e; g_2) \parallel (g_1 \nabla g_2) \parallel f \parallel g_1$ ,  $E' = (e; (f_1 \nabla f_2); h) \parallel (e; g_2) \parallel (f_2 \nabla g_1) \parallel (g_1 \nabla g_2) \parallel f_1$ .
- На рисунке 3.7(c)  $E = e$ ,  $E' = e_1 \nabla e_2$ .

Рассмотрим, как ведут себя эквивалентности на слабо помеченных  $A$ -сетях. В отличие от  $A$ -сетей, на слабо помеченных  $A$ -сетях взаимосвязь эквивалентностей выглядит так же, как и на сетях общего вида и изображается тем же рисунком 1.11.

**Теорема 3.2.5** Пусть  $\leftrightarrow, \leftrightarrow \in \{\equiv, \leftrightarrow, \simeq\}$  и  $\star, \star \in \{\_, i, s, pw, pom, pr, iST, pwST, pomST, prST, pomh, prh, tes, oss\}$ . Для слабо помеченных  $A$ -сетей  $N$  и  $N'$   $N \leftrightarrow_\star N' \Rightarrow N \leftrightarrow_{\star\star} N'$  тогда и только тогда, когда в графе на рисунке 1.11 существует направленный путь от  $\leftrightarrow_\star$  к  $\leftrightarrow_{\star\star}$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие теоремы 1.2.1.

( $\Rightarrow$ ) Отсутствие дополнительных нетривиальных стрелок на рисунке 1.11 доказывается следующими примерами на слабо помеченных  $A$ -сетях.

- На рисунке 3.6(a)  $N \leftrightarrow_i N'$ , но  $N \not\equiv_s N'$ , так как только в сети  $N'$  действия  $a$  и  $b$  не могут выполняться параллельно.
- На рисунке 3.6(c)  $N \leftrightarrow_{iST} N'$ , но  $N \not\equiv_{pw} N'$ , так как сети  $N$  соответствует ЧУММ такое, что даже менее последовательное ЧУММ не может быть выполнено в сети  $N'$ .



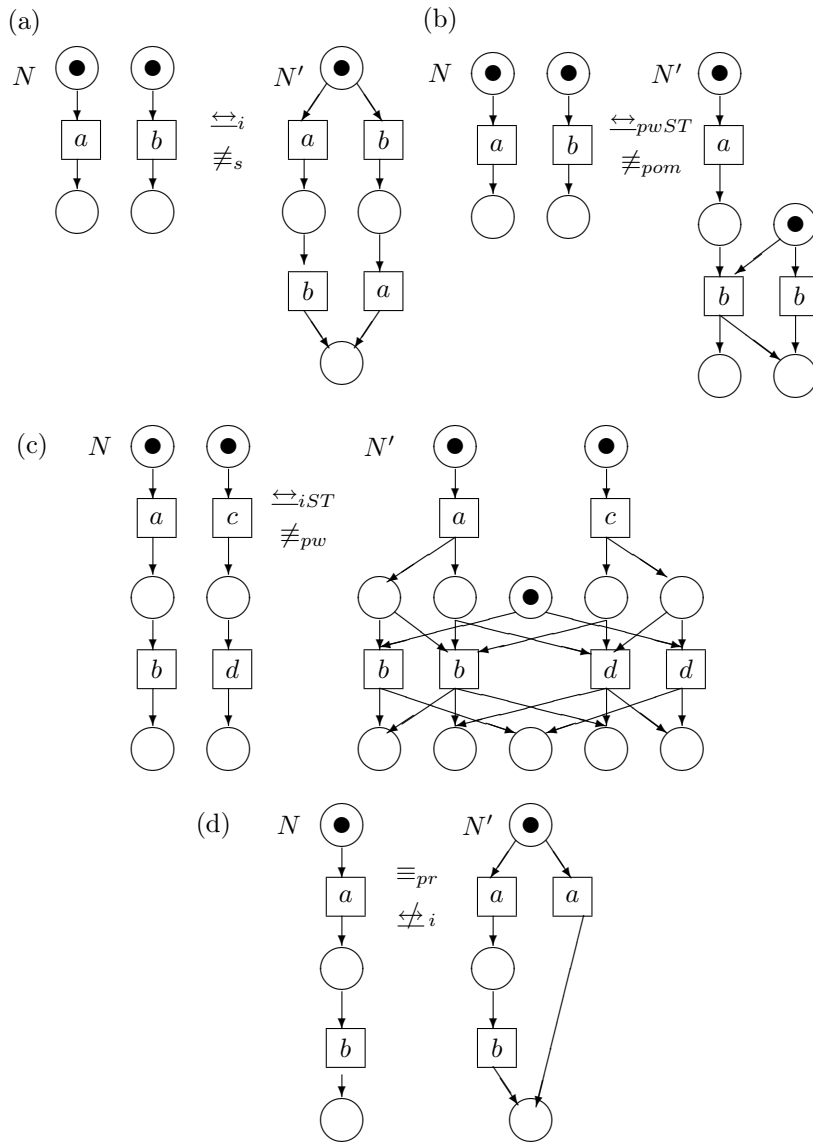


Рис. 3.6: Примеры эквивалентностей на слабо помеченных A-сетях

- На рисунке 3.6(b)  $N \leftrightarrow_{pwST} N'$ , но  $N \not\equiv_{pom} N'$ , так как только в сети  $N'$  действие  $b$  может зависеть от  $a$ .
- На рисунке 3.3(a)  $N \equiv_{mes} N'$ , но  $N \not\equiv_{pr} N'$ .
- На рисунке 3.6(d)  $N \equiv_{pr} N'$ , но  $N \not\leftrightarrow_i N'$ , так как только в сети  $N'$  можно выполнить действие  $a$  так, что после него нельзя выполнить  $b$ .
- На рисунке 3.7(a)  $N \leftrightarrow_{pr} N'$ , но  $N \not\leftrightarrow_{iST} N'$ , так как в только в сети  $N'$  действие  $a$  может так начать работать, что никакое  $b$  уже не может стартовать, пока  $a$  не окончит работу.
- На рисунке 3.7(b)  $N \leftrightarrow_{prST} N'$ , но  $N \not\leftrightarrow_{pomh} N'$ , так как только в сети  $N'$  можно выполнить действия  $a$  и  $b$  так, чтобы следующее действие,  $c$ , обязательно зависело от  $a$ .
- На рисунке 3.7(c)  $N \leftrightarrow_{prh} N'$ , но  $N \not\equiv_{mes} N'$ , так как сети  $N'$  соответствует ПСС с двумя конфликтными между собой действиями  $a$ .
- На рисунке 3.3(b)  $N \equiv_{occ} N'$ , но  $N \not\equiv N'$ . □

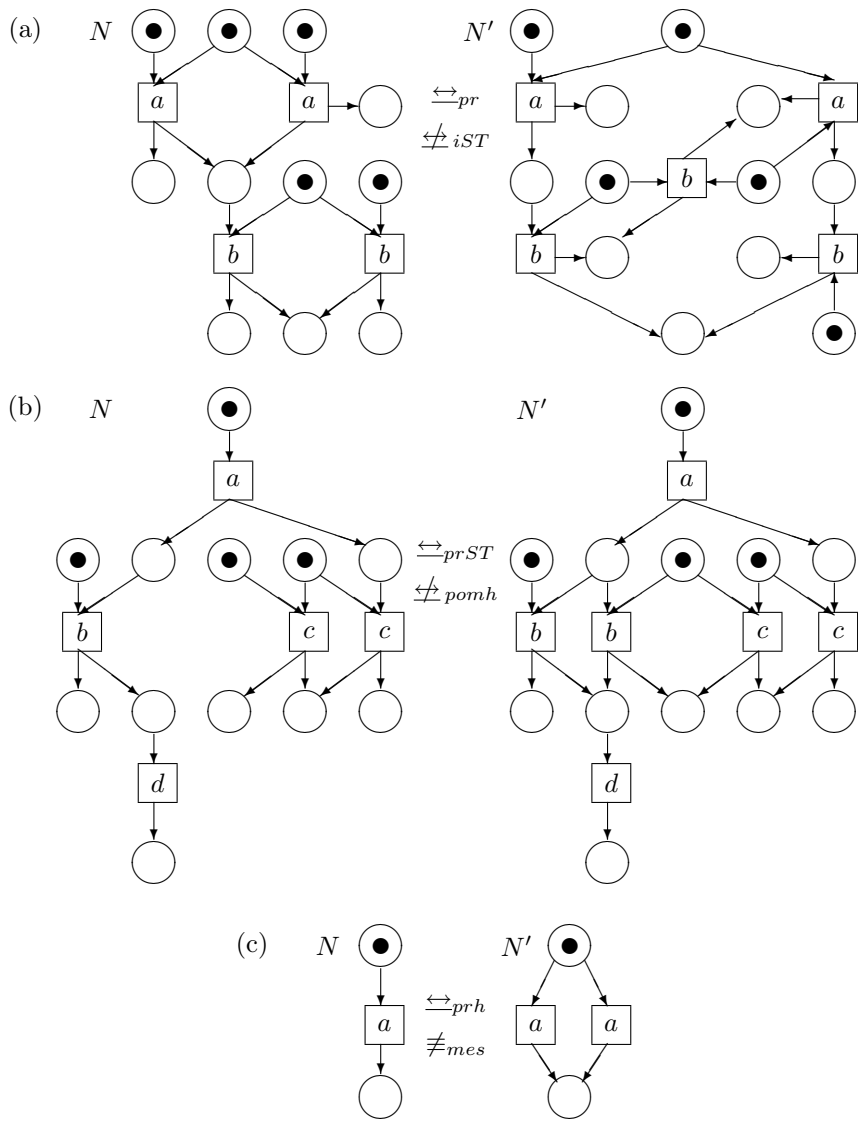


Рис. 3.7: Примеры эквивалентностей на слабо помеченных A-сетях (продолжение)

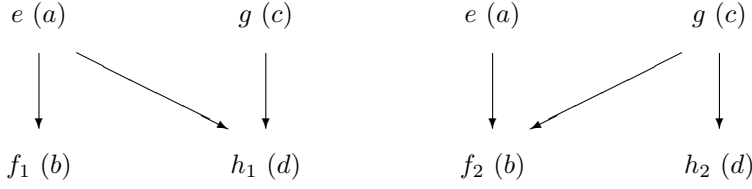


Рис. 3.8: Совокупность ПЧУМ помеченного недетерминированного процесса

**Пример 3.2.6** Рассмотрим сеть  $N'$  на рисунке 3.6(c). Ей соответствует формула  $AFLP_2$

$E' = (e; (f_1 \nabla f_2)) \parallel (e; (f_2 \nabla h_1)) \parallel (g; (f_2 \nabla h_1)) \parallel (g; (h_1 \nabla h_2)) \parallel (f_1 \nabla h_2)$ ,  $lab(e) = a$ ,  $lab(f_1) = lab(f_2) = b$ ,  $lab(g) = c$ ,  $lab(h_1) = lab(h_2) = d$ . Ее каноническая форма

$F' = ((e; f_1) \parallel (e; h_1) \parallel (g; h_1) \parallel \bar{f}_2 \parallel \bar{h}_2) \vee ((e; f_2) \parallel (g; f_2) \parallel (g; h_2) \parallel \bar{f}_1 \parallel \bar{h}_1)$ . Помеченный недетерминированный процесс, задаваемый формулой  $E'$ , имеет два ПЧУМ, представленных на рисунке 3.8. На этом рисунке возле имен событий в скобках стоят действия, помечающие эти события, а частичный порядок изображается стрелками.

Покажем, что в системе  $TS(F')$  из начальной формулы,  $F'$ , может сработать часть первого ПЧУМ, не включающая событие  $f_1$ . В следующих применениях правил перехода системы  $TS(F')$  к подформулам  $F'$  под стрелкой будет стоять номер применяемого правила, а в скобках справа – проверка условий этого правила.

1.  $e; f_1 \xrightarrow{e}_{2.1} f_1$
2.  $e; h_1 \xrightarrow{e; h_1}_{2.2} \nu$
3.  $(e; f_1) \parallel (e; h_1) \xrightarrow{e \parallel (e; h_1)}_{3.3} f_1 \parallel \nu$  ( $\{e\} \cap \emptyset = \emptyset$ ,  $\{e, h_1\} \cap \{f_1\} = \emptyset$ )
4.  $g; h_1 \xrightarrow{g; h_1}_{2.2} \nu$
5.  $(e; f_1) \parallel (e; h_1) \parallel (g; h_1) \xrightarrow{e \parallel (e; h_1) \parallel (g; h_1)}_{3.3} f_1 \parallel \nu \parallel \nu$  ( $\{e, h_1\} \cap \emptyset = \emptyset$ ,  $\{g, h_1\} \cap \{f_1\} = \emptyset$ )
6.  $(e; f_1) \parallel (e; h_1) \parallel (g; h_1) \parallel \bar{f}_2 \xrightarrow{e \parallel (e; h_1) \parallel (g; h_1)}_{3.1} f_1 \parallel \nu \parallel \nu \parallel \bar{f}_2$  ( $\{e, g, h_1\} \cap \{\bar{f}_2\} = \emptyset$ )
7.  $(e; f_1) \parallel (e; h_1) \parallel (g; h_1) \parallel \bar{f}_2 \parallel \bar{h}_2 \xrightarrow{e \parallel (e; h_1) \parallel (g; h_1)}_{3.1} f_1 \parallel \nu \parallel \nu \parallel \bar{f}_2 \parallel \bar{h}_2$  ( $\{e, g, h_1\} \cap \{\bar{h}_2\} = \emptyset$ )
8.  $((e; f_1) \parallel (e; h_1) \parallel (g; h_1) \parallel \bar{f}_2 \parallel \bar{h}_2) \vee ((e; f_2) \parallel (g; f_2) \parallel (g; h_2) \parallel \bar{f}_1 \parallel \bar{h}_1) \xrightarrow{e \parallel (e; h_1) \parallel (g; h_1)}_{4.1} f_1 \parallel \nu \parallel \nu \parallel \bar{f}_2 \parallel \bar{h}_2$  ( $\{e, g, h_1\} \not\subseteq \{e, g, f_2, h_2, \bar{f}_1, \bar{h}_1\}$ )

Итак,  $F' \xrightarrow{G} \tilde{F}'$  – переход системы  $TS(F')$ , где  $G = e \parallel (e; h_1) \parallel (g; h_1)$ ,  $\tilde{F}' = f_1 \parallel \bar{f}_2 \parallel \bar{h}_2$ . Таким образом, в  $TS(F')$  из начального состояния может сработать ПЧУМ  $\rho_G = \langle \{e, g, h_1\}, \prec, l \rangle$ , где  $e \prec h_1$ ,  $g \prec h_1$ ,  $l(e) = a$ ,  $l(g) = c$ ,  $l(h_1) = d$ . В результате срабатывания этого ПЧУМ получается формула  $\tilde{F}' = f_1 \parallel \bar{f}_2 \parallel \bar{h}_2$ , содержащая информацию о том, что в данном поведении помеченного недетерминированного процесса, задаваемого формулой  $E'$ , не смогли произойти события  $f_2$  и  $h_2$  из-за того, что случились альтернативные им события. Также видно, что в текущем состоянии, которое задается формулой  $\tilde{F}'$ , может случиться событие  $f_1$ . В результате мы приходим в состояние, описываемое конечной формулой  $\bar{f}_2 \parallel \bar{h}_2$  системы  $TS(F')$ .

Определим денотационную семантику формулы  $E'$ .  $\mathcal{D}_{FL2}[E'] = \{ \langle \{e, f_1, g, h_1, \bar{f}_2, \bar{h}_2\}, \prec_1, l \rangle, \langle \{e, f_2, g, h_2, \bar{f}_1, \bar{h}_1\}, \prec_2, l \rangle \}$ ,  $\mathcal{D}_{FL2}^+[E'] = \{ \langle \{e, f_1, g, h_1\}, \prec_1, l_1 \rangle, \langle \{e, f_2, g, h_2\}, \prec_2, l_2 \rangle \}$ , где  $e \prec_1 f_1$ ,  $e \prec_1 h_1$ ,  $g \prec_1 h_1$ ,  $e \prec_2 f_2$ ,  $g \prec_2 f_2$ ,  $g \prec_2 h_2$ ,  $l(e) = l_1(e) = l_2(e) = a$ ,  $l(f_1) = l_1(f_1) = l_2(f_1) = b$ ,  $l(g) = l_1(g) = l_2(g) = c$ ,  $l(h_1) = l_1(h_1) = l_2(h_1) = d$ .

### 3.2.4 Сравнение сетевых и алгебраических эквивалентностей

В [2, 189] доказано, что любую конечную A-сеть можно представить формулой  $AFP_0$  с помощью алгоритма регуляризации. Следовательно, любую конечную слабо помеченную A-сеть с помощью аналогичного алгоритма можно представить формулой  $AFLP_0$ . В предыдущем разделе построено отображение  $\Psi_L$ ,

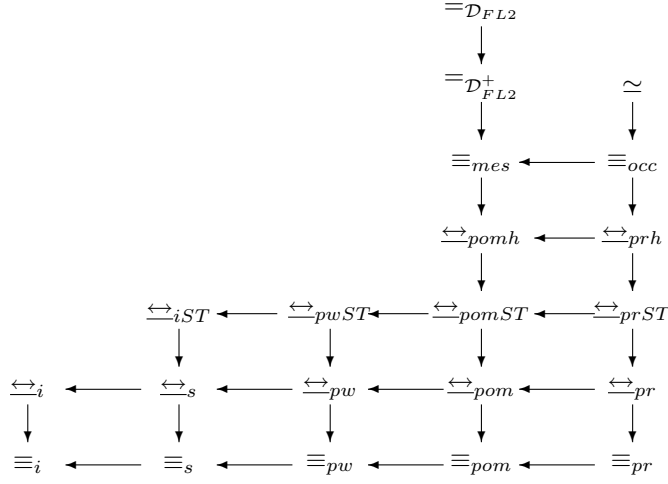


Рис. 3.9: Взаимосвязь сетевых и алгебраических эквивалентностей

сопоставляющее каждой формуле  $AFLP_0$  формулу  $AFLP_2$  и сохраняющее совокупности ПЧУМ. Таким образом, любой конечной слабо помеченной А-сети  $N$  можно сопоставить формулу  $E$  алгебры  $AFLP_2$  такую, что совокупность ПЧУМ начальных С-подсетей  $N$  совпадает с совокупностью ПЧУМ детерминированных процессов, задаваемых формулой  $E$ .

В таком случае очевидно, что понятия формульных эквивалентностей алгебры  $AFLP_2$  можно расширить на сети.

**Определение 3.2.7** Пусть  $\leftrightarrow$  — некоторая формульная эквивалентность  $AFLP_2$  и слабо помеченным А-сетям  $N$  и  $N'$  соответствуют формулы  $E$  и  $E'$ . Сети  $N$  и  $N'$  эквивалентны (относительно  $\leftrightarrow$ ), запись  $N \leftrightarrow N'$ , тогда и только тогда, когда соответствующие им формулы эквивалентны, то есть  $E \leftrightarrow E'$ .

Следующая теорема устанавливает взаимосвязь сетевых эквивалентностей и формульных эквивалентностей алгебры  $AFLP_2$ .

**Теорема 3.2.6** Пусть  $\leftrightarrow, \Leftrightarrow \in \{\equiv, \leftrightarrow, \cong, =\}$  и  $\star, \star\star \in \{\_, i, s, pw, pom, pr, iST, pwST, pomST, prST, pomh, prh, mes, occ, \mathcal{D}_{FL2}, \mathcal{D}_{FL2}^+\}$ . Для слабо помеченных А-сетей  $N$  и  $N'$   $N \leftrightarrow_{\star} N' \Rightarrow N \Leftrightarrow_{\star\star} N'$  тогда и только тогда, когда в графе на рисунке 3.9 существует направленный путь от  $\leftrightarrow_{\star}$  к  $\Leftrightarrow_{\star\star}$ .

*Доказательство.* ( $\Leftarrow$ ) Следствие теоремы 3.2.5 и следующих рассуждений.

- $\equiv_{mes}$  — следствие  $=_{\mathcal{D}_{FL2}^+}$ . Этот факт доказывается следующими рассуждениями. Пусть  $N =_{\mathcal{D}_{FL2}^+} N'$ , а  $P$  и  $P'$  — формулы  $AFLP_2$ , соответствующие А-сетям  $N$  и  $N'$  соответственно. Положим  $\mathcal{D}_{FL2}^+[P] = \mathcal{D}_{FL2}^+[P'] = \bigcup_{i=1}^n \rho_i$ ,  $\rho_i = \langle X_i, \prec_i, l_i \rangle$  ( $1 \leq i \leq n$ ). Тогда этой совокупности ПЧУМ однозначно соответствует ПСС  $\xi = \langle \bigcup_{i=1}^n X_i, \bigcup_{i=1}^n \prec_i, \#, \bigcup_{i=1}^n l_i \rangle$ , где  $x \# y \Leftrightarrow \forall i (1 \leq i \leq n) (x \notin X_i) \vee (y \notin X_i)$ . Легко проверить, что  $\mathcal{E}(N) = \mathcal{E}(N')$  — класс изоморфизма  $\xi$ . Следовательно,  $N \equiv_{mes} N'$ .
- $=_{\mathcal{D}_{FL2}^+}$  — следствие  $=_{\mathcal{D}_{FL2}}$ , так как  $=_{\mathcal{D}_{FL2}^+}$  не учитывает не-события и тупиковые события.

( $\Rightarrow$ ) Отсутствие дополнительных нетривиальных стрелок на рисунке 3.9 доказывается следующими примерами на слабо помеченных А-сетях.

- А-сетям  $N$  и  $N'$  на рисунке 3.3(а) соответствуют формулы  $AFLP_2$   $E = e; f$  и  $E' = (e; f) \parallel e$ ,  $lab(e) = a$ ,  $lab(f) = b$ . Имеем  $\mathcal{D}_{FL2}[E] = \mathcal{D}_{FL2}[E'] \{ \{e, f\}, \prec, l \}$ , где  $e \prec f$ ,  $l(e) = a$ ,  $l(f) = b$ . Следовательно,  $N =_{\mathcal{D}_{FL2}} N'$ . Но  $N \not\equiv_{pr} N'$ .
- Рассмотрим некоторые слабо помеченные А-сети  $N$  и  $N'$ , различающиеся лишь именами переходов. Тогда  $N \cong N'$ , но  $N \neq_{\mathcal{D}_{FL2}^+} N'$ , так как эквивалентность  $=_{\mathcal{D}_{FL2}^+}$  учитывает конкретные имена переходов (события).

- А-сетям  $N$  и  $N'$  на рисунке 3.3(b) соответствуют формулы  $E = (e \nabla f) \| e \| f$  и  $E' = (e \nabla f) \| (e; g) \| (f; g)$ ,  $lab(e) = a$ ,  $lab(f) = b$ ,  $lab(g) = c$ . Так как  $\mathcal{D}_{FL2}^+[E] = \mathcal{D}_{FL2}^+[E'] = \{\{\{e\}, \emptyset, l_1\}, \{\{f\}, \emptyset, l_2\}\}$ ,  $l_1(e) = a$ ,  $l_2(f) = b$ , имеем  $N =_{\mathcal{D}_{FL2}^+} N'$ . Тем не менее,  $N \neq_{\mathcal{D}_{FL2}} N'$ . Это следует из того, что  $\mathcal{D}_{FL2}[E] = \{\{\{e, \delta_f\}, \emptyset, l\}, \{\{f, \delta_e\}, \emptyset, l\}\}$ ,  $l(e) = a$ ,  $l(f) = b$ , а  $\mathcal{D}_{FL2}[E'] = \{\{\{e, \delta_f, \delta_g\}, \emptyset, l'\}, \{\{f, \delta_e, \delta_g\}, \emptyset, l'\}\}$ ,  $l'(e) = a$ ,  $l'(f) = b$ ,  $l'(g) = c$ , то есть тупиковое событие  $\delta_g$  оказывает влияние на эквивалентность  $=_{\mathcal{D}_{FL2}}$ .  $\square$

### 3.2.5 Введение алгебраических аналогов сетевых эквивалентностей

#### Процессные подформулы

**Определение 3.2.8** Пусть  $E$  — формула AFLP<sub>2</sub> и  $F \in \text{canon}(E)$ . Процессная подформула формулы  $E$  — формула  $G$  такая, что  $G \in \text{canon}(H)$  для некоторой формулы  $H \in \mathbf{TS}(F)$  или  $G = \nu$ .

Обозначим множество всех процессных подформул формулы  $E$  через  $PSF(E)$ . Из определения следует, что процессная подформула — либо пустая формула  $\nu$ , либо нормальная конъюнкция, которая является дизъюнктивным членом формулы  $F$  или префиксом некоторого дизъюнктивного члена  $F$ . Будем рассматривать процессные подформулы с точностью до изоморфизма. Так как процессные подформулы — нормальные конъюнкции, изоморфизм на них означает совпадение с точностью до перестановки конъюнктивных членов. Будем считать, что пустой формуле  $\nu$  соответствует ПЧУМ  $\rho_\nu = \langle \emptyset, \emptyset, \emptyset \rangle$ .

Будем писать  $G \xrightarrow{\hat{G}} \tilde{G}$ , если  $F \xrightarrow{H} F'$ ,  $F' \xrightarrow{\hat{H}} F''$ ,  $F \xrightarrow{\tilde{H}} F''$  — переходы  $TS(F)$  и  $G = \text{canon}(H)$ ,  $\hat{G} = \text{canon}(\hat{H})$ ,  $\tilde{G} = \text{canon}(\tilde{H})$ . В этом случае процессная подформула  $\tilde{G}$  — расширение процессной подформулы  $G$  на  $\hat{G}$ , а  $\hat{G}$  — расширяющая процессная подформула. Положим для всех  $G \in PSF(E)$   $\nu \xrightarrow{G} G$  и  $G \xrightarrow{\nu} G$ . Пишем  $G \rightarrow \tilde{G}$ , если  $G \xrightarrow{\hat{G}} \tilde{G}$  для некоторой процессной подформулы  $\hat{G}$ .

$\tilde{G}$  — расширение  $G$  на одно действие, если  $G \xrightarrow{\hat{G}} \tilde{G}$  и  $G = e$ ,  $e \in Ev$ . В этом случае пишем  $G \xrightarrow{e} \tilde{G}$  или  $G \xrightarrow{a} \tilde{G}$ , если  $lab(e) = a$ .

$\tilde{G}$  — расширение  $G$  на мультимножество действий или шаг, если  $G \xrightarrow{\hat{G}} \tilde{G}$  и  $G = \parallel_{i=1}^n e_i$ . В этом случае пишем  $G \xrightarrow{U} \tilde{G}$  или  $G \xrightarrow{A} \tilde{G}$ , если  $U = \{e_1, \dots, e_n\}$ ,  $A = \{lab(e_1), \dots, lab(e_n)\} \in \mathcal{M}(Act)$ .

Пусть  $G \in PSF(E)$ . Тогда  $G$  — максимальная процессная подформула формулы  $E$ , если ее нельзя расширить ни на одно действие. Обозначим множество всех максимальных процессных подформул формулы  $E$  через  $PSF_{\max}(E)$ .

**Пример 3.2.7** Для формулы  $E'$ , описывающей сеть  $N'$  на рисунке 3.6(c),  $PSF_{\max}(E') = \{(e; f_1) \| (e; h_1) \| (g; h_1), (e; f_2) \| (g; f_2) \| (g; h_2)\}$ . Заметим, что каждая из 2 процессных подформул, входящих в  $PSF_{\max}(E)$ , представляет класс изоморфизма в формул, которые получаются перестановками конъюнктивных членов. Так как мы рассматриваем процессные подформулы с точностью до изоморфизма, мы пишем только 2 формулы вместо 12.

#### Следовые эквивалентности

**Определение 3.2.9** Интерливинговый след формулы  $E$  — это последовательность  $a_1 \cdots a_n \in Act^*$  такая, что  $\nu \xrightarrow{a_1} G_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} G_n$ , где  $G_i \in PSF(E)$  ( $1 \leq i \leq n$ ). Обозначим множество всех интерливинговых следов  $E$  через  $IntTraces(E)$ . Формулы  $E$  и  $E'$  интерливингово следово эквивалентны, запись  $E \equiv_i E'$ , если  $IntTraces(E) = IntTraces(E')$ .

**Определение 3.2.10** Шаговый след формулы  $E$  — это последовательность  $A_1 \cdots A_n \in (\mathcal{M}(Act))^*$  такая, что  $\nu \xrightarrow{A_1} G_1 \xrightarrow{A_2} \dots \xrightarrow{A_n} G_n$ , где  $G_i \in PSF(E)$  ( $1 \leq i \leq n$ ). Обозначим множество всех шаговых следов  $E$  через  $StepTraces(E)$ . Формулы  $E$  и  $E'$  шагово следово эквивалентны, запись  $E \equiv_s E'$ , если  $StepTraces(E) = StepTraces(E')$ .

**Определение 3.2.11** ЧУММ след формулы  $E$  — это ЧУММ  $\rho$  — класс изоморфизма  $\rho_G$  для  $G \in PSF(E)$ . Пишем  $\rho \sqsubseteq \rho'$ , если  $\rho_G \sqsubseteq \rho_{G'}$  для  $\rho_G \in \rho$  и  $\rho_{G'} \in \rho'$ . В этом случае говорим, что ЧУММ  $\rho$  менее последователен или более параллелен, чем  $\rho'$ . Обозначим через  $Pomsets(E)$  множество всех ЧУММ следов  $E$ . Формулы  $E$  и  $E'$  ЧС следово эквивалентны, запись  $E \equiv_{pw} E'$ , если  $Pomsets(E) \sqsubseteq Pomsets(E')$  и  $Pomsets(E') \sqsubseteq Pomsets(E)$ , то есть для любого  $\rho' \in Pomsets(E')$  существует  $\rho \in Pomsets(E)$  такой, что  $\rho \sqsubseteq \rho'$  и наоборот.

**Определение 3.2.12** Формулы  $E$  и  $E'$  ЧУММ следово эквивалентны, запись  $E \equiv_{\text{пот}} E'$ , если  $\text{Pomsets}(E) = \text{Pomsets}(E')$ .

### Бисимуляционные эквивалентности

#### Обычные бисимуляционные эквивалентности

**Определение 3.2.13** Пусть  $E$  и  $E'$  — некоторые формулы  $AFLP_2$ . Отношение  $\mathcal{R} \subseteq \text{PSF}(E) \times \text{PSF}(E')$  —  $\star$ -бисимуляция между  $E$  и  $E'$ ,  $\star \in \{\text{интерливингивая, шаговая, ЧС, ЧУММ}\}$ , запись  $\mathcal{R} : E \leftrightarrow_{\star} E'$ ,  $\star \in \{i, s, pw, \text{пот}\}$ , если:

1.  $(\nu, \nu) \in \mathcal{R}$ ;

2.  $(G, G') \in \mathcal{R}$ ,  $G \xrightarrow{\hat{G}} \tilde{G}$ ,

(a)  $|\text{cont}(\hat{G})| = 1$ , если  $\star = i$ ;

(b)  $\prec_{\hat{G}} = \emptyset$ , если  $\star = s$ ;

тогда  $\exists \tilde{G}' : G' \xrightarrow{\hat{G}'} \tilde{G}'$ ,  $(\tilde{G}, \tilde{G}') \in \mathcal{R}$  и

(a)  $\rho_{\tilde{G}'} \sqsubseteq \rho_{\tilde{G}}$ , если  $\star = pw$ ;

(b)  $\rho_{\tilde{G}} \simeq \rho_{\tilde{G}'}$ , если  $\star \in \{i, s, \text{пот}\}$ .

3. Как пункт 2, но роли  $E$  и  $E'$  меняются.

Формулы  $E$  и  $E'$   $\star$ -бисимуляционно эквивалентны,  $\star \in \{\text{интерливингивая, шаговая, ЧС, ЧУММ}\}$ , запись  $E \leftrightarrow_{\star} E'$ , если  $\exists \mathcal{R} : E \leftrightarrow_{\star} E'$ ,  $\star \in \{i, s, pw, \text{пот}\}$ .

#### ST-бисимуляционные эквивалентности

**Определение 3.2.14** ST-процессная подформула формулы  $E$  — пара  $(G, H)$  такая, что  $G, H \in \text{PSF}(E)$ ,  $H \xrightarrow{K} G$  и  $\forall e, f \in \rho_G$   $e \prec_G f \Rightarrow e \in \text{cont}(H)$ .

В этом случае  $G$  — процессная подформула, которая начала работать, то есть все события  $G$  начали происходить. Процессная подформула  $H$  соответствует той части  $G$ , которая уже завершилась, а  $K$  — той части, которая еще работает. Очевидно, что  $\prec_K = \emptyset$ .  $ST - \text{PSF}(E)$  обозначает множество всех ST-процессных подформул формулы  $E$ .

$(\nu, \nu)$  будет начальной ST-процессной подформулой. Пусть  $(G, H), (\tilde{G}, \tilde{H}) \in ST - \text{PSF}(E)$ . Пишем  $(G, H) \rightarrow (\tilde{G}, \tilde{H})$ , если  $G \rightarrow \tilde{G}$  и  $H \rightarrow \tilde{H}$ .

**Определение 3.2.15** Пусть  $E$  и  $E'$  — некоторые формулы  $AFLP_2$ . Отношение  $\mathcal{R} \subseteq ST - \text{PSF}(E) \times ST - \text{PSF}(E') \times \mathcal{B}$ , где  $\mathcal{B} = \{\beta \mid \beta : \text{cont}(G) \rightarrow \text{cont}(G'), G \in \text{PSF}(E), G' \in \text{PSF}(E')\}$  —  $\star$ -ST-бисимуляция между  $E$  и  $E'$ ,  $\star \in \{\text{интерливингивая, ЧС, ЧУММ}\}$ , запись  $\mathcal{R} : E \leftrightarrow_{\star ST} E'$ ,  $\star \in \{i, pw, \text{пот}\}$ , если:

1.  $((\nu, \nu), (\nu, \nu), \emptyset) \in \mathcal{R}$ .

2.  $((G, H), (G', H'), \beta) \in \mathcal{R} \Rightarrow \beta : \rho_G = \rho_{G'}$  и  $\beta(\text{cont}(H)) = \text{cont}(H')$ .

3.  $((G, H), (G', H'), \beta) \in \mathcal{R}$ ,  $(G, H) \rightarrow (\tilde{G}, \tilde{H}) \Rightarrow \exists \tilde{\beta}, (\tilde{G}', \tilde{H}') : (G', H') \rightarrow (\tilde{G}', \tilde{H}')$ ,  $\tilde{\beta}|_{\text{cont}(G)} = \beta$ ,  $((\tilde{G}, \tilde{H}), (\tilde{G}', \tilde{H}'), \tilde{\beta}) \in \mathcal{R}$ , и если  $H \xrightarrow{K} \tilde{G}$ ,  $H' \xrightarrow{K'} \tilde{G}'$ ,  $\gamma = \tilde{\beta}|_{\text{cont}(K)}$ , то:

(a)  $\gamma^{-1} : \rho_{K'} \sqsubseteq \rho_K$ , если  $\star = pw$ ;

(b)  $\gamma : \rho_K \simeq \rho_{K'}$ , если  $\star = \text{пот}$ .

4. Как пункт 3, но роли  $E$  и  $E'$  меняются.

Формулы  $E$  и  $E'$   $\star$ -ST-бисимуляционно эквивалентны,  $\star \in \{\text{интерливингивая, ЧС, ЧУММ}\}$ , запись  $E \leftrightarrow_{\star ST} E'$ , если  $\exists \mathcal{R} : E \leftrightarrow_{\star ST} E'$ ,  $\star \in \{i, pw, \text{пот}\}$ .

### Сохраняющие историю бисимуляционные эквивалентности

**Определение 3.2.16** Пусть  $E$  и  $E'$  — некоторые формулы  $AFLP_2$ . Отношение  $\mathcal{R} \subseteq PSF(E) \times PSF(E') \times \mathcal{B}$ , где  $\mathcal{B} = \{\beta \mid \beta : cont(G) \rightarrow cont(G'), G \in PSF(E), G' \in PSF(E')\}$  — ЧУММ сохраняющая историю бисимуляция между  $E$  и  $E'$ , запись  $\mathcal{R} : E \xrightarrow{pot_h} E'$ , если:

1.  $(\nu, \nu, \emptyset) \in \mathcal{R}$ .
2.  $(G, G', \beta) \in \mathcal{R} \Rightarrow \beta : \rho_G \simeq \rho_{G'}$ .
3.  $(G, G', \beta) \in \mathcal{R}, G \rightarrow \tilde{G} \Rightarrow \exists \tilde{\beta}, \tilde{G}' : G' \rightarrow \tilde{G}', \tilde{\beta}|_{cont(G)} = \beta, (\tilde{G}, \tilde{G}', \tilde{\beta}) \in \mathcal{R}$ ;
4. Как пункт 3, но роли  $E$  и  $E'$  меняются.

Формулы  $E$  и  $E'$  ЧУММ сохраняющие историю бисимуляционно эквивалентны, запись  $E \xleftrightarrow{pot_h} E'$ , если  $\exists \mathcal{R} : E \xleftrightarrow{pot_h} E'$ .

### Сохраняющие конфликт эквивалентности

Пусть  $E$  — формула  $AFLP_2$  и  $F = \bigvee_{i=1}^n F_i \in canon(E)$ . Формуле  $F$  можно сопоставить ПСС  $\xi_F = \langle cont^+(F), \prec_F, \#_F, l_F|_{cont^+(F)} \rangle$ , где:

- $e \prec_F f \Leftrightarrow \exists i (1 \leq i \leq n) (e; f)$  входит в  $F_i$ ;
- $e \#_F f \Leftrightarrow \forall i (1 \leq i \leq n) e$  и  $f$  не входят вместе в  $F_i$ .

Обозначим через  $\mathcal{E}(E)$  МСС — класс изоморфизма ПСС  $\xi_F$  для  $F \in canon(E)$ .

**Определение 3.2.17** Две формулы  $E$  и  $E'$  МСС-сохраняющие конфликт эквивалентны, обозначение  $E \equiv_{mes} E'$ , если  $\mathcal{E}(E) = \mathcal{E}(E')$ .

### Сравнение сетевых эквивалентностей с их алгебраическими аналогами

Пусть  $E$  — формула  $AFLP_2$ , соответствующая конечной слабо помеченной А-сети  $N$ . В разделе о взаимосвязи сетевых эквивалентностей с эквивалентностями алгебры  $AFLP_2$  было установлено, что совокупность ПЧУМ начальных С-подсетей  $N$  совпадает с совокупностью ПЧУМ всех детерминированных процессов, описываемых формулой  $E$ . Следующее предложение устанавливает взаимосвязь ПЧУМ процессов  $N$  и ПЧУМ процессных подформул  $E$ .

**Предложение 3.2.4** Пусть  $E$  — формула  $AFLP_2$ , соответствующая конечной слабо помеченной А-сети  $N$ . Тогда  $\{\rho_C \mid \pi = (C, id) \in \Pi(N)\} = \{\rho_G \mid G \in PSF(E)\}$ .

*Доказательство.*

1. Как было замечено в [98], множество максимальных С-подсетей конечной А-сети формирует множество ее максимальных процессов. Очевидно, что множество начальных С-подсетей формирует множество всех (не только максимальных) процессов конечной А-сети. Аналогичное утверждение справедливо для конечных слабо помеченных А-сетей. Таким образом, с точностью до изоморфизма можно считать, что  $\Pi(N)$  состоит из процессов вида  $\pi = (C, id)$ , где  $id$  — идентичное отображение над множеством  $P_C \cup T_C$ . Каждому такому процессу соответствует ПЧУМ  $\rho_C = \langle T_C, \prec_C \cap (T_C \times T_C), l_C \rangle$ .
2. С другой стороны, любому дизъюнктивному члену  $F_j$  ( $1 \leq j \leq n$ ) формулы  $F = \bigvee_{i=1}^n F_i \in canon(E)$  соответствует ПЧУМ одного из максимальных детерминированных процессов, описываемых этой формулой,  $\rho_{F_j}^+ = \langle cont^+(F_j), \prec_{F_j}^*, l_{F_j} \rangle$ . Тогда дизъюнктивным членам формулы  $F$  и их префиксам будут соответствовать ПЧУМ всех (не только максимальных) детерминированных процессов, описываемых формулой  $E$ . Заметим, что для любого дизъюнктивного члена (или его префикса)  $F_j$  формулы  $F$  существует процессная подформула  $G \in PSF(E)$  такая, что  $G \simeq F_j^+$  и  $\rho_{F_j}^+ = \rho_{F_j^+} = \rho_G$ .

Таким образом,  $\{\rho_C \mid \pi = (C, id) \in \Pi(N)\} = \{\rho_G \mid G \in PSF(E)\}$ . □

Следующее предложение устанавливает биекцию между множеством процессов сети  $N$  и множеством процессных подформул  $E$ , которая сохраняет ПЧУМ.

**Предложение 3.2.5** Пусть  $E$  — формула  $AFLP_2$ , соответствующая конечной слабо помеченной  $A$ -сети  $N$ . Тогда существует биекция  $\chi : \Pi(N) \rightarrow PSF(E)$  такая, что если  $\pi \in \Pi(N)$ ,  $G \in PSF(E)$  и  $\chi(\pi) = G$ ,  $\pi = (C, id)$ , то  $\rho_C = \rho_G$ .

*Доказательство.* Покажем, что ПЧУМы определяют с точностью до изоморфизма как процессы сети  $N$ , так и процессные подформулы формулы  $E$ .

1. Определим отображение  $\chi_1$  из  $\Pi(N)$  в совокупность ПЧУМ следующим образом. Пусть  $\pi = (C, id) \in \Pi(N)$ . Тогда положим  $\chi_1(\pi) = \rho_C$ . Очевидно, что каждому процессу соответствует только одно ПЧУМ. Следовательно,  $\chi_1$  — функция. По определению она сюръективна. Кроме того, каждый процесс  $\pi = (C, id) \in \Pi(N)$  однозначно определяется своей  $C$ -сетью  $C$ . А сеть  $C$  является начальной  $C$ -подсетью  $N$ , и, следовательно, однозначно определяется множеством своих переходов  $T_C$ . Значит, никаким двум разным процессам из  $\Pi(N)$  не соответствует одно ПЧУМ. Тогда  $\chi_1$  — биекция.
2. Определим отображение  $\chi_2$  из  $PSF(E)$  в совокупность ПЧУМ следующим образом. Пусть  $G \in PSF(E)$ . Тогда положим  $\chi_2(G) = \rho_G$ . Очевидно, что каждой процессной подформуле соответствует только одно ПЧУМ. Следовательно,  $\chi_2$  — функция. По определению она сюръективна. Кроме того, никаким двум разным (не изоморфным) процессным подформулам не соответствует одно ПЧУМ, так как эти подформулы — нормальные конъюнкции, которые являются, по существу, представлением ПЧУМ (если учесть информацию о пометке событий, входящих в эти подформулы). Тогда  $\chi_2$  — биекция.

Пусть  $\chi = \chi_2^{-1} \circ \chi_1$ . Тогда  $\chi : \Pi(N) \rightarrow PSF(E)$  — биекция, сохраняющая ПЧУМы, то есть если  $\chi(\pi) = G$ ,  $\pi = (C, id)$ , то  $\rho_C = \rho_G$ .  $\square$

Установим результат, касающийся правил расширений для процессов и процессных подформул.

**Предложение 3.2.6** Пусть  $E$  — формула  $AFLP_2$ , соответствующая конечной слабо помеченной  $A$ -сети  $N$ . Тогда  $\forall \pi, \pi' \in \Pi(N)$   $\pi \xrightarrow{\hat{\pi}} \tilde{\pi} \Leftrightarrow \chi(\pi) \xrightarrow{\chi(\tilde{\pi})} \chi(\tilde{\pi})$ .

*Доказательство.* Достаточно заметить, что определения расширений процессов и процессных подформул основаны на следующем правиле расширений для ПЧУМ. Пусть  $\rho = \langle X, \prec, l \rangle$ ,  $\tilde{\rho} = \langle \tilde{X}, \tilde{\prec}, \tilde{l} \rangle$ ,  $\hat{\rho}$  — ПЧУМ. Тогда  $\tilde{\rho}$  — расширение  $\rho$  на  $\hat{\rho}$ , запись  $\rho \xrightarrow{\hat{\rho}} \tilde{\rho}$ , если  $\rho \prec \tilde{\rho}$  и  $\hat{\rho} = \tilde{\rho}|_{\tilde{X} \setminus X}$ .  $\square$

Теперь мы можем доказать главный результат, касающийся взаимосвязи сетевых эквивалентностей и их аналогов на формулах  $AFLP_2$ .

**Теорема 3.2.7** Пусть  $\leftrightarrow \in \{\equiv, \leftrightarrow\}$ ,  $\star \in \{i, s, pw, pot, iST, pwST, potST, pot h, mes\}$ . и  $E (E')$  — формула  $AFLP_2$ , соответствующая конечной слабо помеченной  $A$ -сети  $N (N')$ . Тогда  $N \leftrightarrow_{\star} N' \Leftrightarrow E \leftrightarrow_{\star} E'$ .

*Доказательство.*  $(\Rightarrow)$  Любой след  $N$  является следом  $E$ . Для доказательства этого факта достаточно заметить, что определение следа  $E$  получается из определения следа  $N$  заменой каждого процесса  $\pi \in \Pi(N)$  на процессную подформулу  $G \in PSF(E)$  такую, что  $\chi(\pi) = G$ . Аналогично доказывается, что любой след  $E$  является следом  $N$ . Таким образом, множества следов  $N$  и  $E$  совпадают. Доказательство того, что множества следов  $N'$  и  $E'$  совпадают, аналогично приведенному выше. Следовательно,  $N \equiv_{\star} N' \Leftrightarrow E \equiv_{\star} E'$ ,  $\star \in \{i, s, pw, pot\}$ .

По предложению 3.2.6,  $\mathcal{R} : N \leftrightarrow_{\star} N' \Leftrightarrow \mathcal{S} : E \leftrightarrow_{\star} E'$ ,  $\star \in \{i, s, pw, pot, iST, pwST, potST, pot h\}$ , где отношение  $\mathcal{S}$  определяется следующим образом.

**Обычные бисимуляции**  $\mathcal{R} = \{(\pi, \pi') \mid (\chi(\pi), \chi'(\pi')) \in \mathcal{S}\}$ ;

**ST-бисимуляции**  $\mathcal{R} = \{((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \mid ((\chi(\pi_E), \chi(\pi_P)), (\chi'(\pi'_E), \chi'(\pi'_P)), \chi' \circ \beta \circ \chi^{-1}) \in \mathcal{S}\}$ .

**Сохраняющие историю бисимуляции**  $\mathcal{R} = \{(\pi, \pi', \beta) \mid (\chi(\pi), \chi'(\pi'), \chi' \circ \beta \circ \chi^{-1}) \in \mathcal{S}\}$ .

Заметим, что формула  $E$  специфицирует недетерминированный процесс, который является максимальным  $O$ -процессом сети  $N$ . Следовательно,  $\mathcal{E}(N) = \mathcal{E}(E)$ . Аналогично доказывается, что  $\mathcal{E}(N') = \mathcal{E}(E')$ . Тогда мы имеем  $N \equiv_{mes} N' \Leftrightarrow E \equiv_{mes} E'$ .

$(\Leftarrow)$  Как предыдущий пункт, но с использованием  $\chi^{-1}$  и  $(\chi')^{-1}$  вместо  $\chi$  и  $\chi'$  соответственно.  $\square$

Очевидно, взаимосвязь формульных эквивалентностей и аналогов сетевых эквивалентностей в алгебре  $AFLP_2$  изображается графом на рисунке 3.9, из которого удалены процессные эквивалентности.

После определения аналогов сетевых эквивалентностей на формулах  $AFLP_2$  возникает вопрос, является ли какой-либо из таких аналогов конгруэнтностью относительно операций этой алгебры. Рассмотрим следующий пример.



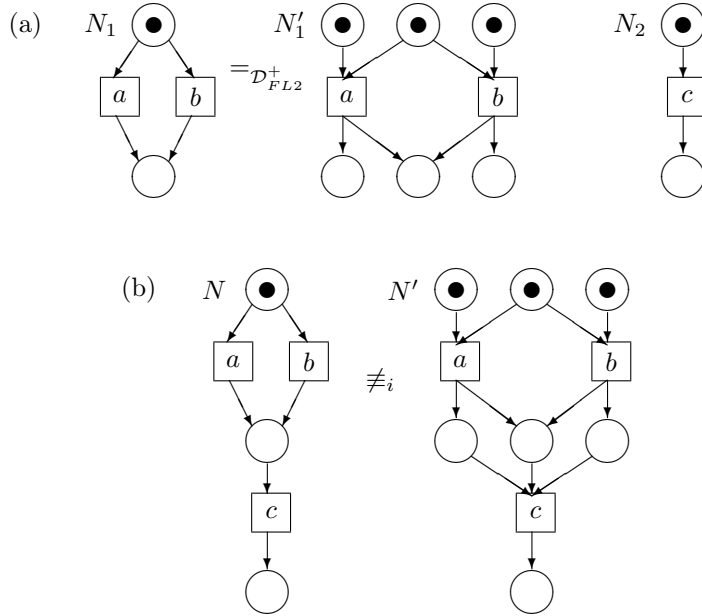


Рис. 3.10: А-сети из примера на конгруэнтность

**Пример 3.2.8** Пусть для формул  $E_1 = e \nabla f$ ,  $E'_1 = (e \nabla f) \| e \| f$  и  $E_2 = g$ ,  $lab(e) = a$ ,  $lab(f) = b$ ,  $lab(g) = c$ . Имеем  $E_1 =_{\mathcal{D}_{FL2}^+} E'_1$ , но  $E_1; E_2 \not\equiv_i E'_1; E_2$ , так как  $PSF(E_1; E_2) = \{\nu, e, f, (e; g), (f; g)\}$ , а  $PSF(E'_1; E_2) = \{\nu, e, f\}$ . Поэтому  $IntTraces(E_1; E_2) = \{a, b, ac, bc\}$ , тогда как  $IntTraces(E'_1; E_2) = \{a, b\}$ .

Заметим, что формулы  $E_1$ ,  $E'_1$  и  $E_2$  соответствуют А-сетям  $N_1$ ,  $N'_1$  и  $N_2$  на рисунке 3.10(a), а формулы  $E = E_1; E_2$  и  $E' = E'_1; E_2$  — А-сетям  $N$  и  $N'$  на рисунке 3.10(b).

Мы доказали согласованность эквивалентностей на сетях с их аналогами на формулах  $AFLP_2$ . Следовательно, тот факт, что  $E \not\equiv_i E'$ , можно установить, рассматривая сети  $N$  и  $N'$ , для которых  $N \not\equiv_i N'$ , так как в сети  $N'$ , в отличие от  $N$ , действие  $c$  никогда не сможет сработать.

Таким образом, никакая из рассмотренных эквивалентностей на формулах  $AFLP_2$ , за исключением  $=_{\mathcal{D}_{FL2}}$ , не является конгруэнтностью, то есть  $=_{\mathcal{D}_{FL2}}$  — самая слабая эквивалентность, которая обладает свойством композициональности.

# Заключение

Перечислим основные **результаты**, полученные в данной работе.

1. В рамках сетей Петри с видимыми и невидимыми переходами введен и исследован ряд поведенческих эквивалентностей. Установлены взаимосвязи указанных эквивалентностей для различных подклассов сетей. Дана логическая характеристика ряда эквивалентностных отношений. Установлена возможность использования некоторых эквивалентностей для эффективной, сохраняющей поведение редукции сетей. Исследован композиционный подход к сравнению поведения параллельных систем.
2. Для временных сетей Петри с видимыми и невидимыми переходами определен ряд временных, невременных и региональных эквивалентностей. Установлено их соотношение для некоторых подклассов временных сетей. Дана региональная характеристика временных эквивалентностей, упрощающая их проверку. Исследована стабильность эквивалентностных отношений при композиционной разработке временных параллельных систем.
3. Предложено расширение известного алгебраического исчисления  $AFP_2$  функцией пометки — алгебра помеченных недетерминированных параллельных процессов  $AFLP_2$ . Дана операционная характеристика, полная и корректная аксиоматизация семантических эквивалентностей указанных алгебр. Проведено сравнение алгебраических и сетевых эквивалентностей и их перенос из одной модели в другую и обратно. Разработан и программно реализован способ автоматизации проверки формул алгебр на семантическую эквивалентность.

Таким образом, основные **выводы** состоят в следующем. В работе введен и систематически исследован широкий набор поведенческих эквивалентностных отношений для трех важнейших моделей параллелизма: обычных и временных сетей Петри, а также процессных алгебр. Полученные результаты позволяют оценить и выбрать наиболее подходящий формализм и критерий равенства в процессе разработки параллельных и распределенных систем, в том числе и систем реального времени.

# Приложение А

## Доказательства

В данную главу перенесены наиболее длинные доказательства.

### Доказательство предложения 1.2.4.

( $\Leftarrow$ ) Пусть  $\mathcal{R} : N \xleftrightarrow{\star h} N'$ . Определим отношение  $\mathcal{S}$  следующим образом:  $\mathcal{S} = \{((\pi, \sigma), (\pi', \sigma')) \mid (\pi, \sigma) \in \text{Runs}(N), (\pi', \sigma') \in \text{Runs}(N'), |\sigma| = |\sigma'|, \forall i (0 \leq i \leq |\sigma|) (\pi(i), \pi'(i), \beta_{\sigma(i)}^{\sigma'(i)}) \in \mathcal{R}\}$ . Докажем, что  $\mathcal{S} : N \xleftrightarrow{\star b \star f} N'$ .

1.  $((\pi_N, \varepsilon), (\pi_{N'}, \varepsilon)) \in \mathcal{R}$ , так как  $\beta_{\varepsilon}^{\varepsilon} = \emptyset$  и  $(\pi_N, \pi_{N'}, \emptyset) \in \mathcal{R}$ .

2. Пусть  $((\pi, \sigma), (\pi', \sigma')) \in \mathcal{R}$ .

- (обратно)

Пусть  $(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\hat{\pi}} (\pi, \sigma)$ . По определению  $\mathcal{S}$ ,  $\exists (\tilde{\pi}', \tilde{\sigma}')$  такое, что  $|\tilde{\sigma}| = |\tilde{\sigma}'|$ ,  $(\tilde{\pi}', \tilde{\sigma}') \xrightarrow{\hat{\pi}'} (\pi', \sigma')$  и  $((\tilde{\pi}, \tilde{\sigma}), (\tilde{\pi}', \tilde{\sigma}')) \in \mathcal{S}$ .

Так как по определению  $\mathcal{S}$   $(\pi, \pi', \beta_{\sigma}^{\sigma'}) \in \mathcal{R}$  и  $\mathcal{R} : N \xleftrightarrow{\star h} N'$ , имеем:

- $\beta_{\sigma}^{\sigma'} : \rho_C \simeq \rho_{C'}$ , если  $\star \in \{\text{pot}, \text{pr}\}$ ;
- $C \simeq C'$ , если  $\star = \text{pr}$ .

Следовательно,

- $\rho_{\hat{C}} \simeq \rho_{\hat{C}'}$ , если  $\star = \text{pot}$ ;
- $\hat{C} \simeq \hat{C}'$ , если  $\star = \text{pr}$ .

- (прямо)

Пусть  $(\pi, \sigma) \xrightarrow{\hat{\pi}} (\tilde{\pi}, \sigma v_1 \cdots v_n)$ . Тогда по определению  $\mathcal{S}$   $(\pi, \pi', \beta_{\sigma}^{\sigma'}) \in \mathcal{R}$  и  $\exists \pi_i (1 \leq i \leq n) : \pi \xrightarrow{v_1} \pi_1 \xrightarrow{v_2} \cdots \xrightarrow{v_n} \pi_n = \tilde{\pi}$ .

Так как  $\mathcal{R} : N \xleftrightarrow{\star h} N'$ ,  $\exists v'_i, \pi'_i : \pi' \xrightarrow{v'_1} \pi'_1 \xrightarrow{v'_2} \cdots \xrightarrow{v'_n} \pi'_n = \tilde{\pi}'$  такие, что  $(\pi_i, \pi'_i, \beta_{\sigma v_1 \cdots v_i}^{\sigma' v'_1 \cdots v'_i}) \in \mathcal{R}$  ( $1 \leq i \leq n$ ). Следовательно, для некоторого  $\hat{\pi}' (\pi', \sigma') \xrightarrow{\hat{\pi}'} (\tilde{\pi}', \sigma' v'_1 \cdots v'_n)$  и  $((\tilde{\pi}, \sigma v_1 \cdots v_n), (\tilde{\pi}', \sigma' v'_1 \cdots v'_n)) \in \mathcal{S}$ .

Так как по определению  $\mathcal{S}$   $(\tilde{\pi}, \tilde{\pi}', \beta_{\sigma v_1 \cdots v_n}^{\sigma' v'_1 \cdots v'_n}) \in \mathcal{R}$  и  $\mathcal{R} : N \xleftrightarrow{\star h} N'$ , имеем:

- $\beta_{\sigma v_1 \cdots v_n}^{\sigma' v'_1 \cdots v'_n} : \rho_{\tilde{C}} \simeq \rho_{\tilde{C}'}$ , если  $\star \in \{\text{pot}, \text{pr}\}$ ;
- $\tilde{C} \simeq \tilde{C}'$ , если  $\star = \text{pr}$ .

Следовательно,

- $\rho_{\hat{C}} \simeq \rho_{\hat{C}'}$ , если  $\star = \text{pot}$ ;
- $\hat{C} \simeq \hat{C}'$ , если  $\star = \text{pr}$ .

3. Как пункт 2, но роли  $N$  и  $N'$  меняются.

( $\Rightarrow$ ) Пусть  $\mathcal{R} : N \xleftrightarrow{\star b \star f} N'$ . Определим отношение  $\mathcal{S}$  следующим образом:  $\mathcal{S} = \{(\pi, \pi', \beta_{\sigma}^{\sigma'}) \mid ((\pi, \sigma), (\pi', \sigma')) \in \mathcal{R} \text{ и}$

- $\beta_{\sigma}^{\sigma'} : \rho_C \simeq \rho_{C'}$ , если  $\star \in \{pot, pr\}$ ;
- $C \simeq C'$ , если  $\star = pr$ .

Докажем, что  $\mathcal{S} : N \xleftrightarrow{\star h} N'$ .

1.  $(\pi_N, \pi_{N'}, \emptyset) \in \mathcal{S}$  так как  $\beta_{\varepsilon}^{\varepsilon} = \emptyset$  и  $((\pi_N, \varepsilon), (\pi_{N'}, \varepsilon)) \in \mathcal{R}$ .

2. Пусть  $(\pi, \pi', \beta_{\sigma}^{\sigma'}) \in \mathcal{S}$ . Тогда по определению  $\mathcal{S}$

- $\beta_{\sigma}^{\sigma'} : \rho_C \simeq \rho_{C'}$ , если  $\star \in \{pot, pr\}$ ;
- $C \simeq C'$ , если  $\star = pr$ .

3. Пусть  $(\pi, \pi', \beta_{\sigma}^{\sigma'}) \in \mathcal{S}$  и  $\pi \xrightarrow{v} \tilde{\pi}$ . Тогда по определению  $\mathcal{S}$   $((\pi, \sigma), (\pi', \sigma')) \in \mathcal{R}$  и  $(\pi, \sigma) \rightarrow (\tilde{\pi}, \sigma v)$ . Так как  $\mathcal{R} : N \xleftrightarrow{\star b \star f} N'$ ,  $\exists v', \tilde{\pi}' : (\pi', \sigma') \rightarrow (\tilde{\pi}', \sigma' v')$  и  $((\tilde{\pi}, \sigma v), (\tilde{\pi}', \sigma' v')) \in \mathcal{R}$ . Имеем  $\pi' \xrightarrow{v'} \tilde{\pi}'$ .

Докажем, что

- $\beta_{\sigma v}^{\sigma' v'} : \rho_{\tilde{C}} \simeq \rho_{\tilde{C}'}$ , если  $\star \in \{pot, pr\}$ ;
- $\tilde{C} \simeq \tilde{C}'$ , если  $\star = pr$ .

Пусть  $\beta_{\sigma v}^{\sigma' v'}$  — не изоморфизм. Если  $\sigma = v_1 \cdots v_n$ ,  $\sigma' = v'_1 \cdots v'_n$ , определим  $j = \max\{i \mid ((v_i \prec_{\tilde{C}} v) \wedge (v'_i \not\prec_{\tilde{C}'} v')) \vee ((v_i \not\prec_{\tilde{C}} v) \wedge (v'_i \prec_{\tilde{C}'} v'))\}$ . Если  $\pi(i-1) \xrightarrow{\tilde{\pi}_i} \tilde{\pi}$  и  $\pi'(i-1) \xrightarrow{\tilde{\pi}'_i} \tilde{\pi}'$  ( $1 \leq i \leq n$ ), то  $\rho_{\tilde{C}_j} \not\approx \rho_{\tilde{C}'_j}$  и  $\rho_{\tilde{C}_{j+1}} \simeq \rho_{\tilde{C}'_{j+1}}$ . Следовательно, обратное расширение  $(\pi(j-1), \sigma(j-1)) \xrightarrow{\tilde{\pi}_j} (\tilde{\pi}, \sigma v)$  не может имитироваться обратным расширением  $(\pi'(j-1), \sigma'(j-1)) \xrightarrow{\tilde{\pi}'_j} (\tilde{\pi}', \sigma' v')$  так, чтобы  $\rho_{\tilde{C}_j} \simeq \rho_{\tilde{C}'_j}$ .

Имеем противоречие с тем, что  $\mathcal{R} : N \xleftrightarrow{\star b \star f} N'$ . Поэтому  $\beta_{\sigma v}^{\sigma' v'} : \rho_{\tilde{C}} \simeq \rho_{\tilde{C}'}$  и  $(\tilde{\pi}, \tilde{\pi}', \beta_{\sigma v}^{\sigma' v'}) \in \mathcal{S}$ . Итак, мы все доказали для случая  $\star = pot$ .

Докажем для случая  $\star = pr$ . Так как  $((\tilde{\pi}, \sigma v), (\tilde{\pi}', \sigma' v')) \in \mathcal{R}$  и  $\mathcal{R} : N \xleftrightarrow{\star b \star f} N'$ , обратное расширение  $(\pi_N, \varepsilon) \xrightarrow{\tilde{\pi}} (\tilde{\pi}, \sigma v)$  имитируется обратным расширением  $(\pi_{N'}, \varepsilon) \xrightarrow{\tilde{\pi}'} (\tilde{\pi}', \sigma' v')$ , где  $\tilde{C} \simeq \tilde{C}'$ .

4. Как пункт 3, но роли  $N$  и  $N'$  меняются. □

### Доказательство предложения 1.2.11.

Пусть  $\bar{N} = ref(N, a, D)$ ,  $\bar{N}' = ref(N', a, D)$ . Заметим, что С-сети процессов SM-сетей — простые цепочки, то есть сети, каждая вершина которых имеет ровно одного предшественника (за исключением единственного входного места) и ровно одного последователя (за исключением единственного выходного места).

*Конструкция (\*)*

1. Пусть  $\bar{\pi} = (\bar{C}, \bar{\varphi}) \in \Pi(\bar{N})$ . Тогда каждая вершина  $\bar{C}$ , которая не встраивается в множество  $P_N \cup T_N$ , обладает следующими свойствами:

- имеет вид  $\langle e, f \rangle$  ( $e \in P_{C_D} \cup T_{C_D}$ ,  $\pi_D = (C_D, \varphi_D) \in \Pi(D)$  и  $f \in T_C$ ,  $\pi = (C, \varphi) \in \Pi(N)$ ) и встраивается в элемент вида  $\langle x, u \rangle$ ,  $x \in T_D \cup (P_D \setminus \{p_{in}, p_{out}\})$ ,  $u \in l_N^{-1}(a)$ ;
- имеет единственного предшественника  $\langle e_{min}, f \rangle$ , отображаемого функцией встраивания в  $\langle t_{min}, u \rangle$ ,  $t_{min} \in p_{in}^{\bullet}$ ;
- принадлежит единственной максимальной цепочке  $\vartheta$  (соответствующей сети  $C_D$ ), начинающейся в  $\langle e_{min}, f \rangle$ , все вершины которой встраиваются в элементы вида  $\langle y, u \rangle$ ,  $y \in T_D \cup (P_D \setminus \{p_{in}, p_{out}\})$  и единственные связи которой с окружением процесса следующие:
  - через входные места  $\langle e_{min}, f \rangle$  (всегда);
  - (а) через выходные места максимальной вершины цепочки  $\langle e_{max}, f \rangle$ , которая является переходом, отображаемым функцией встраивания в  $\langle t_{max}, u \rangle$ ,  $t_{max} \in p_{out}^{\bullet}$ ;
  - (б) или цепочка обрывается на максимальном месте раньше.

Следовательно, каждую такую цепочку  $\vartheta$ , входящую в сеть  $\bar{C}$ , можно заменить:

- (a) на переход  $f$ , отображаемый функцией встраивания в  $u$ , так как они имеют одинаковые входы и выходы;
- (b) на переход  $f$ , отображаемый функцией встраивания в  $u$ , с новыми выходными местами, соответствующими  $u$ , так как они имеют одинаковые входы, и после  $f$  ничего нет (в этом случае  $f$  — максимальный переход).

В результате получим процесс  $\pi = (C, \varphi) \in \Pi(N)$ .

2. Так как  $N \equiv_{\star} N'$ ,  $\star \in \{pw, rot, pr\}$ , всегда можно выбрать  $\pi' = (C', \varphi') \in \Pi(N')$  и  $\beta$  такие, что:

- $\beta^{-1} : \rho_{C'} \sqsubseteq \rho_C$ , если  $\star = pw$ ;
- $\beta : \rho_C \simeq \rho_{C'}$ , если  $\star = rot$ ;
- $\beta : C \simeq C'$ , если  $\star = pr$ .

3. Для любой цепочки  $\vartheta$ , сконструированной таким образом, заменим в  $C'$  переход  $\beta(f)$ , встраивающийся в  $u'$ , на копию  $\vartheta'$  цепочки  $\vartheta$ , где все вершины вида  $\langle e, f \rangle$  заменяются на  $\langle e, \beta(f) \rangle$ . Возможны два случая:

- (a) если цепочка полная,  $\beta(f)$  и  $\vartheta'$  имеют одинаковые выходы (из  $u'$ );
- (b) если цепочка неполная, то есть заканчивается на месте, отбрасываем все выходные места  $\beta(f)$ .

В обоих случаях  $\beta(f)$  и  $\vartheta'$  имеют одинаковые входы (в  $u'$ ).

Ясно, что сконструированный таким образом объект — процесс  $\bar{\pi}' = (\bar{C}', \bar{\varphi}') \in \Pi(\bar{N}')$ .

4. Пусть  $g \in P_{\bar{C}} \cup T_{\bar{C}}$ . Определим отображение  $\bar{\beta}$  следующим образом.

$$\bar{\beta}(g) = \begin{cases} \beta(g), & g \text{ не входит ни в какую цепочку;} \\ \langle e, \beta(f) \rangle, & g = \langle e, f \rangle \text{ входит в некоторую цепочку } \vartheta. \end{cases}$$

□ (Конструкция (\*))

Осталось доказать следующие утверждения.

- $\bar{\beta}^{-1} : \rho_{\bar{C}'} \sqsubseteq \rho_{\bar{C}}$ , если  $\star = pw$ ;
- $\bar{\beta} : \rho_{\bar{C}} \simeq \rho_{\bar{C}'}$ , если  $\star = rot$ ;
- $\bar{\beta} : \bar{C} \simeq \bar{C}'$ , если  $\star = pr$ .

Рассмотрим случай, когда  $\star = pw$ , так как случаи  $\star = rot$  и  $\star = pr$  — более простые. Пусть  $g, h \in T_{\bar{C}}$ . Возможны пять случаев:

1.  $g$  и  $h$  не входят ни в какую цепочку;
2.  $g$  входит в некоторую цепочку  $\vartheta$ ,  $h$  не входит ни в какую цепочку;
3.  $g$  не входит ни в какую цепочку,  $h$  входит в некоторую цепочку  $\vartheta$ ;
4.  $g$  и  $h$  входят в одну и ту же цепочку  $\vartheta$ ;
5.  $g$  входит в цепочку  $\vartheta_1$ ,  $h$  входит в цепочку  $\vartheta_2$  и  $\vartheta_1 \neq \vartheta_2$ .

Рассмотрим случай 5, так как случаи 1–4 — более простые. Тогда  $g = \langle e_1, f_1 \rangle$ ,  $h = \langle e_2, f_2 \rangle$ , где  $e_1 \in T_{C_{D_1}}$ ,  $e_2 \in T_{C_{D_2}}$  для  $\pi_{D_1} = (C_{D_1}, \varphi_{D_1}), \pi_{D_2} = (C_{D_2}, \varphi_{D_2}) \in \Pi(D)$ ,  $f_1, f_2 \in T_C$  для  $\pi = (C, \varphi) \in \Pi(N)$ ,  $f_1$  и  $f_2$  детализуются в  $\bar{C}$  в разные цепочки  $\vartheta_1$  и  $\vartheta_2$  соответственно. Имеем:  $\bar{\beta}(g) \prec_{\bar{C}'} \bar{\beta}(h) \Rightarrow \bar{\beta}(\langle e_1, f_1 \rangle) \prec_{\bar{C}'} \bar{\beta}(\langle e_2, f_2 \rangle) \Rightarrow$  (по определению  $\bar{\beta}$ )  $\langle e_1, \beta(f_1) \rangle \prec_{\bar{C}'} \langle e_2, \beta(f_2) \rangle \Rightarrow$  (так как единственные связи цепочек с окружением процесса — через их минимальные и максимальные переходы)  $\langle e_{max1}, \beta(f_1) \rangle \prec_{\bar{C}'} \langle e_{min2}, \beta(f_2) \rangle \Rightarrow$  (по конструкции (\*))  $\beta(f_1) \prec_{C'} \beta(f_2) \Rightarrow$  (так как  $\beta^{-1} : \rho_{C'} \sqsubseteq \rho_C$ )  $f_1 \prec_C f_2 \Rightarrow$  (по конструкции (\*))  $\langle e_{max1}, f_1 \rangle \prec_{\bar{C}} \langle e_{min2}, f_2 \rangle \Rightarrow \langle e_1, f_1 \rangle \prec_{\bar{C}} \langle e_2, f_2 \rangle \Rightarrow g \prec_{\bar{C}} h$ . □

*Доказательство предложения 1.2.12.*

Пусть  $\bar{N} = ref(N, a, D)$ ,  $\bar{N}' = ref(N', a, D)$  и  $\mathcal{R} : N \leftrightarrow_{\star ST} N'$ ,  $\star \in \{i, pw, rot, pr\}$ .  
Конструкция (\*\*)

1. Пусть  $(\bar{\pi}_E, \bar{\pi}_P) \in ST - \Pi(\bar{N})$  и  $\pi_E, \pi_P \in \Pi(N)$  получены из  $\bar{\pi}_E$  и  $\bar{\pi}_P$  соответственно с помощью части 1 конструкции (\*) из доказательства предложения 1.2.11.

*Лемма 1*  $(\pi_E, \pi_P) \in ST - \Pi(N)$ .

*Доказательство (леммы 1)*. Пусть  $g, h \in T_{C_E}$  и  $g \prec_{C_E} h$ . Возможны четыре случая:

- (a)  $l_{C_E}(g) \neq a \neq l_{C_E}(h)$ ;
- (b)  $l_{C_E}(g) = a \neq l_{C_E}(h)$ ;
- (c)  $l_{C_E}(g) \neq a = l_{C_E}(h)$ ;
- (d)  $l_{C_E}(g) = a = l_{C_E}(h)$ .

Рассмотрим случай 4, так как случаи 1–3 — более простые. Тогда  $g$  и  $h$  детализуются в  $\bar{C}_E$  в разные цепочки  $\vartheta_1$  и  $\vartheta_2$  с вершинами вида  $\langle e_1, g \rangle$  и  $\langle e_2, h \rangle$  соответственно, где  $e_1 \in T_{C_{D1}}$ ,  $e_2 \in T_{C_{D2}}$  для  $\pi_{D1} = (C_{D1}, \varphi_{D1}), \pi_{D2} = (C_{D2}, \varphi_{D2}) \in \Pi(D)$ . Имеем:  $g \prec_{\bar{C}_E} h \Rightarrow$  (по конструкции (\*))  $\langle e_{max1}, g \rangle \prec_{\bar{C}_E} \langle e_{min2}, h \rangle \Rightarrow$  (так как  $(\bar{\pi}_E, \bar{\pi}_P) \in ST - \Pi(\bar{N})$  и  $\langle e_{min2}, h \rangle \in T_{\bar{C}_E}$ )  $\langle e_{max1}, g \rangle \in T_{\bar{C}_P} \Rightarrow$  (по конструкции (\*))  $g \in T_{C_P}$ .  $\square$  (Лемма 1)

2. Выберем  $(\pi'_E, \pi'_P) \in ST - \Pi(N')$  и  $\beta$  такие, что  $((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{R}$ .

3. Получим  $\bar{\pi}'_E, \bar{\pi}'_P \in \Pi(\bar{N}')$  из  $\pi'_E$  и  $\pi'_P$  соответственно с помощью части 3 конструкции (\*) из доказательства предложения 1.2.11.

*Лемма 2*  $(\bar{\pi}'_E, \bar{\pi}'_P) \in ST - \Pi(\bar{N}')$ .

*Доказательство (леммы 2)*. Пусть  $g', h' \in T_{\bar{C}'_E}$  и  $g' \prec_{\bar{C}'_E} h'$ . Возможны пять случаев:

- (a)  $g'$  и  $h'$  не входят ни в какую цепочку;
- (b)  $g'$  входит в некоторую цепочку  $\vartheta'$ ,  $h'$  не входит ни в какую цепочку;
- (c)  $g'$  не входит ни в какую цепочку,  $h'$  входит в некоторую цепочку  $\vartheta'$ ;
- (d)  $g'$  и  $h'$  входят в одну и ту же цепочку  $\vartheta'$ ;
- (e)  $g'$  входит в цепочку  $\vartheta'_1$ ,  $h'$  входит в цепочку  $\vartheta'_2$  и  $\vartheta'_1 \neq \vartheta'_2$ .

Рассмотрим случай 5, так как случаи 1–4 — более простые. Тогда  $g' = \langle e_1, f'_1 \rangle$ ,  $h' = \langle e_2, f'_2 \rangle$ , где  $e_1 \in T_{C_{D1}}$ ,  $e_2 \in T_{C_{D2}}$  для  $\pi_{D1} = (C_{D1}, \varphi_{D1}), \pi_{D2} = (C_{D2}, \varphi_{D2}) \in \Pi(D)$ ,  $f'_1, f'_2 \in T_{C'}$  для  $\pi' = (C', \varphi') \in \Pi(N')$ ,  $f'_1$  и  $f'_2$  детализуются в  $\bar{C}'_E$  в разные цепочки  $\vartheta'_1$  и  $\vartheta'_2$  соответственно. Имеем:  $g' \prec_{\bar{C}'_E} h' \Rightarrow \langle e_1, f'_1 \rangle \prec_{\bar{C}'_E} \langle e_2, f'_2 \rangle \Rightarrow$  (так как единственные связи цепочек с окружением процесса — через их минимальные и максимальные переходы)  $\langle e_{max1}, f'_1 \rangle \prec_{\bar{C}'_E} \langle e_{min2}, f'_2 \rangle \Rightarrow$  (по конструкции (\*))  $f'_1 \prec_{\bar{C}'_E} f'_2 \Rightarrow$  (так как  $(\pi'_E, \pi'_P) \in ST - \Pi(N')$ )  $f'_1 \in T_{C'_P} \Rightarrow$  (по конструкции (\*))  $g' = \langle e_1, f'_1 \rangle \in T_{\bar{C}'_P}$ .  $\square$  (Лемма 2)

Пусть  $g \in T_{\bar{C}_E}$ . Определим отображение  $\bar{\beta}$  следующим образом:

$$\bar{\beta}(g) = \begin{cases} \beta(g), & g \text{ не входит ни в какую цепочку;} \\ \langle e, \beta(f) \rangle, & g = \langle e, f \rangle \text{ входит в некоторую цепочку } \vartheta. \end{cases}$$

$\square$  (Конструкция (\*\*))

Пусть  $\mathcal{S}$  состоит из элементов вида  $((\bar{\pi}_E, \bar{\pi}_P), (\bar{\pi}'_E, \bar{\pi}'_P), \bar{\beta})$ , полученных с помощью конструкции (\*\*) из элементов  $((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{R}$ . Докажем, что  $\mathcal{S} : \bar{N} \xleftrightarrow{ST} \bar{N}'$ .

1.  $((\pi_{\bar{N}}, \pi_{\bar{N}}), (\pi_{\bar{N}'}, \pi_{\bar{N}'}), \emptyset) \in \mathcal{S}$ , так как по части 1 конструкции (\*)  $\pi_N$  получается из  $\pi_{\bar{N}}$ , берем процесс  $\pi_{N'}$ , на основе которого по части 3 конструкции (\*) всегда можно выбрать  $\pi_{\bar{N}'}$ .
2. Пусть  $((\bar{\pi}_E, \bar{\pi}_P), (\bar{\pi}'_E, \bar{\pi}'_P), \bar{\beta}) \in \mathcal{S}$ . Очевидно, что по конструкции (\*\*)  $\bar{\beta} : \rho_{\bar{C}_E} \approx \rho_{\bar{C}'_E}$  и  $\bar{\beta}(T_{\bar{C}_P}) = T_{\bar{C}'_P}$ , так как  $\beta(T_{C_P}) = T_{C'_P}$ .
3. Пусть  $((\bar{\pi}_E, \bar{\pi}_P), (\bar{\pi}'_E, \bar{\pi}'_P), \bar{\beta}) \in \mathcal{S}$  и  $(\bar{\pi}_E, \bar{\pi}_P) \rightarrow (\tilde{\pi}_E, \tilde{\pi}_P)$ .

Элемент  $((\bar{\pi}_E, \bar{\pi}_P), (\bar{\pi}'_E, \bar{\pi}'_P), \bar{\beta})$  получен по конструкции (\*\*) из элемента

$((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{R}$ . По части 1 конструкции (\*\*) получим  $(\tilde{\pi}_E, \tilde{\pi}_P) \in ST - \Pi(N)$  из  $(\tilde{\pi}_E, \tilde{\pi}_P)$ .

Очевидно, что  $(\pi_E, \pi_P) \rightarrow (\tilde{\pi}_E, \tilde{\pi}_P)$ . Так как  $\mathcal{R} : N \leftrightarrow_{\star ST} N'$ ,  $\star \in \{i, pw, rot, pr\}$ , имеем:  $\exists \tilde{\beta}, (\tilde{\pi}'_E, \tilde{\pi}'_P)$  такие, что:  $\tilde{\beta}|_{T_{C_E}} = \beta$ ,  $(\pi'_E, \pi'_P) \rightarrow (\tilde{\pi}'_E, \tilde{\pi}'_P)$  и  $((\tilde{\pi}_E, \tilde{\pi}_P), (\tilde{\pi}'_E, \tilde{\pi}'_P), \tilde{\beta}) \in \mathcal{R}$ . В соответствии с частями 3 и 4 конструкции (\*\*\*) получим  $(\tilde{\pi}'_E, \tilde{\pi}'_P) \in ST - \Pi(\bar{N})$  и  $\tilde{\beta}$  из  $(\tilde{\pi}'_E, \tilde{\pi}'_P)$  и  $\tilde{\beta}$  соответственно.

*Лемма 3*  $((\tilde{\pi}_E, \tilde{\pi}_P), (\tilde{\pi}'_E, \tilde{\pi}'_P), \tilde{\beta}) \in \mathcal{S}$ .

*Доказательство (леммы 3)*. Очевидно из конструкции (\*\*\*) □ (Лемма 3)

*Лемма 4*  $\tilde{\beta}|_{T_{\bar{C}_E}} = \tilde{\beta}$ .

*Доказательство (леммы 4)*. Пусть  $g \in T_{\bar{C}_E}$ . Возможны два случая:

- (a)  $g$  не входит ни в какую цепочку;
- (b)  $g$  входит в некоторую цепочку  $\vartheta$ .

Рассмотрим случай (b), так как случай (a) тривиален. Тогда  $g = \langle e, f \rangle$ , где  $e \in T_{C_D}$  для  $\pi_D = (C_D, \varphi_D) \in \Pi(D)$ ,  $f \in T_C$  для  $\pi = (C, \varphi) \in \Pi(N)$ ,  $f$  детализуется в  $\bar{C}_E$  в цепочку  $\vartheta$ . Имеем:  $\tilde{\beta}(\langle e, f \rangle) = \langle e, \tilde{\beta}(f) \rangle =$  (так как  $f \in T_{C_E}$  и  $\tilde{\beta}|_{T_{C_E}} = \beta$ )  $\langle e, \beta(f) \rangle =$  (по определению  $\tilde{\beta}$ )  $\tilde{\beta}(\langle e, f \rangle)$ . □  
(Лемма 4)

*Лемма 5*  $(\pi'_E, \pi'_P) \rightarrow (\tilde{\pi}'_E, \tilde{\pi}'_P)$ .

*Доказательство (леммы 5)*. Докажем, что  $\pi'_E \rightarrow \tilde{\pi}'_E$  (доказательство того, что  $\pi'_P \rightarrow \tilde{\pi}'_P$ , проводится аналогично). Надо показать, что  $\bar{C}'_E \tilde{C}'_E|_{P_{\bar{C}'_E} \cup T_{\bar{C}'_E}}$  и  $\bar{\varphi}'_E \tilde{\varphi}'_E|_{P_{\bar{C}'_E} \cup T_{\bar{C}'_E}}$ .

Достаточно доказать, что  $T_{\bar{C}'_E} \subseteq T_{\tilde{C}'_E}$ . Пусть  $g' \in T_{\bar{C}'_E}$ . Возможны два случая:

- (a)  $g'$  не входит ни в какую цепочку;
- (b)  $g'$  входит в некоторую цепочку  $\vartheta'$ .

Рассмотрим случай (b), так как случай (a) тривиален. Тогда  $g' = \langle e, f' \rangle$ , где  $e \in T_{C_D}$  для  $\pi_D = (C_D, \varphi_D) \in \Pi(D)$ ,  $f' \in T_{C'_E}$  для  $\pi' = (C', \varphi') \in \Pi(N')$ ,  $f'$  детализуется в  $\bar{C}'_E$  в цепочку  $\vartheta'$ . Так как  $f' \in T_{C'_E}$  и  $(\pi'_E, \pi'_P) \rightarrow (\tilde{\pi}'_E, \tilde{\pi}'_P)$ , получаем  $f' \in T_{\tilde{C}'_E} \Rightarrow$  (по конструкции (\*\*\*))  $g' = \langle e, f' \rangle \in T_{\tilde{C}'_E}$ . □  
(Лемма 5)

*Замечание 1* Так как по лемме 4  $\tilde{\beta}|_{\bar{C}_E} = \tilde{\beta}$  и из  $(\pi_E, \pi_P) \in ST - \Pi(\bar{N})$  следует  $\tilde{\beta}(T_{\bar{C}_P}) = T_{\bar{C}'_P}$ , имеем  $\tilde{\beta}(T_{\tilde{C}_E} \setminus T_{\bar{C}_P})T_{\tilde{C}'_E} \setminus T_{\bar{C}'_P}$ . Следовательно,  $\tilde{\beta}(T_{\tilde{C}}) = T_{\tilde{C}'}$ . □ (Замечание 1)

*Замечание 2* Так как из того, что  $f \in T_{C_P}$ , следует  $\langle e, f \rangle \in T_{\bar{C}_P}$ , то  $\langle e, f \rangle \notin T_{\bar{C}_P}$  влечет  $f \notin T_{C_P}$ . Тогда  $\langle e, f \rangle \in T_{\tilde{C}_E} \setminus T_{\bar{C}_P} = T_{\tilde{C}}$  влечет  $f \in T_{\tilde{C}_E} \setminus T_{C_P} = T_{\tilde{C}}$ . □ (Замечание 2)

Осталось доказать следующие утверждения.

- $\tilde{\beta}^{-1} : \rho_{\bar{C}'_E} \sqsubseteq \rho_{\bar{C}_E}$ , если  $\star = pw$ ;
- $\tilde{\beta} : \rho_{\bar{C}_E} \simeq \rho_{\bar{C}'_E}$ , если  $\star \in \{i, rot, pr\}$ ;
- $\bar{C}_E \simeq \bar{C}'_E$ , если  $\star = pr$ .

Рассмотрим случай  $\star = pr$ , так как случай  $\star = pw$  рассматривается, как в предложении 1.2.11, а случай  $\star = rot$  — более простой. Сначала докажем, что  $\tilde{\beta} : \rho_{\bar{C}_E} \simeq \rho_{\bar{C}'_E}$ . Пусть  $g, h \in T_{\tilde{C}}$ . Возможны пять случаев:

- (a)  $g$  и  $h$  не входят ни в какую цепочку;
- (b)  $g$  входит в некоторую цепочку  $\vartheta$ ,  $h$  не входит ни в какую цепочку;
- (c)  $g$  не входит ни в какую цепочку,  $h$  входит в некоторую цепочку  $\vartheta$ ;
- (d)  $g$  и  $h$  входят в одну и ту же цепочку  $\vartheta$ ;
- (e)  $g$  входит в цепочку  $\vartheta_1$ ,  $h$  входит в цепочку  $\vartheta_2$  и  $\vartheta_1 \neq \vartheta_2$ .

Рассмотрим случай (е), так как случаи (а)–(д) — более простые. Тогда  $g = \langle e_1, f_1 \rangle$ ,  $h = \langle e_2, f_2 \rangle$ , где  $e_1 \in T_{C_{D1}}$ ,  $e_2 \in T_{C_{D2}}$  для  $\pi_{D1} = (C_{D1}, \varphi_{D1}), \pi_{D2} = (C_{D2}, \varphi_{D2}) \in \Pi(D)$ ,  $f_1, f_2 \in T_C$  для  $\pi = (C, \varphi) \in \Pi(N)$ ,  $f_1$  и  $f_2$  детализуются в  $\bar{C}_E$  в разные цепочки  $\vartheta_1$  и  $\vartheta_2$  соответственно. Имеем:  $g \prec_{\bar{C}} h \Leftrightarrow \langle e_1, f_1 \rangle \prec_{\bar{C}} \langle e_2, f_2 \rangle \Leftrightarrow$  (так как единственные связи цепочек с окружением процесса — через их минимальные и максимальные переходы)  $\langle e_{max1}, f_1 \rangle \prec_{\bar{C}} \langle e_{min2}, f_2 \rangle \Leftrightarrow$  (по конструкции (\*\*))  $f_1 \prec_C f_2 \Leftrightarrow$  (так как  $\tilde{\beta} : \rho_C \simeq \rho_{C'}$  и по замечанию 2  $f_1, f_2 \in T_C$ )  $\tilde{\beta}(f_1) \prec_{C'} \tilde{\beta}(f_2) \Leftrightarrow$  (по конструкции (\*\*))  $\langle e_{max1}, \tilde{\beta}(f_1) \rangle \prec_{\bar{C}'} \langle e_{min2}, \tilde{\beta}(f_2) \rangle \Leftrightarrow \langle e_1, \tilde{\beta}(f_1) \rangle \prec_{\bar{C}'} \langle e_2, \tilde{\beta}(f_2) \rangle \Leftrightarrow$  (по определению  $\tilde{\beta}$ )  $\tilde{\beta}(\langle e_1, f_1 \rangle) \prec_{\bar{C}'} \tilde{\beta}(\langle e_2, f_2 \rangle) \Leftrightarrow \tilde{\beta}(g) \prec_{\bar{C}'} \tilde{\beta}(h)$ .

Теперь докажем, что  $\bar{C} \simeq \bar{C}'$ . Так как  $\mathcal{R} : N \leftrightarrow_{prST} N'$ , имеем  $\exists \alpha : C \simeq C'$ . Тогда отображение  $\bar{\alpha} : \bar{C} \simeq \bar{C}'$  получается из  $\alpha$  следующим образом. Пусть  $g \in P_{\bar{C}} \cup T_{\bar{C}}$ . Тогда

$$\bar{\alpha}(g) = \begin{cases} \alpha(g), & g \text{ не входит ни в какую цепочку;} \\ \langle e, \alpha(f) \rangle, & g = \langle e, f \rangle \text{ входит в некоторую цепочку } \vartheta. \end{cases}$$

4. Как пункт 3, но роли  $\bar{N}$  и  $\bar{N}'$  меняются. □

### Доказательство предложения 1.2.15.

Пусть  $\bar{N} = ref(N, a, D)$ ,  $\bar{N}' = ref(N', a, D)$ . Заметим, что О-сети О-процессов SM-сетей — деревья, то есть сети, каждая вершина которых имеет ровно одного предшественника (за исключением единственного входного места).

*Конструкция (\*\*\*)*

1. Пусть  $\bar{\omega} = (\bar{O}, \bar{\psi}) \in \wp_{max}(\bar{N})$  каждая вершина  $\bar{O}$ , которая не встраивается в множество  $P_N \cup T_N$ , обладает следующими свойствами:

- имеет вид  $\langle e, f \rangle$  ( $e \in P_{O_D} \cup T_{O_D}$ ,  $\varpi_D = (O_D, \psi_D) \in \wp_{max}(D)$  и  $f \in T_O$ ,  $\varpi = (O, \psi) \in \wp_{max}(N)$ ) и встраивается в элемент вида  $\langle x, u \rangle$ ,  $x \in T_D \cup (P_D \setminus \{p_{in}, p_{out}\})$ ,  $u \in l_N^{-1}(a)$ ;
- имеет единственного предшественника  $\langle e_{min}^i, f \rangle$  ( $1 \leq i \leq n$ ), отображаемого функцией встраивания в  $\langle t_{min}^i, u \rangle$ ,  $t_{min}^i \in p_{in}^\bullet$ ;
- принадлежит единственному максимальному дереву  $\vartheta^i$  (входящему в совокупность деревьев  $\vartheta = \bigcup_{i=1}^n \vartheta^i$ , соответствующую сети  $O_D$ ), начинающемуся в  $\langle e_{min}^i, f \rangle$ , все вершины которого встраиваются в элементы вида  $\langle y, u \rangle$ ,  $y \in T_D \cup (P_D \setminus \{p_{in}, p_{out}\})$  и единственные связи которого с окружением процесса следующие:
  - через входные места  $\langle e_{min}^i, f \rangle$  (всегда);
  - через выходные места максимальных вершин дерева  $\langle e_{max}^{ij}, f \rangle$  ( $1 \leq j \leq m$ ), которые являются переходами, отображаемыми функцией встраивания в  $\langle t_{max}^{ij}, u \rangle$ ,  $t_{max}^{ij} \in \bullet p_{out}$ .

Заметим, что все  $e_{min}^i$  ( $1 \leq i \leq n$ ) имеют одинаковые связи с окружением процесса (как и все  $e_{max}^{ij}$  ( $1 \leq i \leq n, 1 \leq j \leq m$ )).

Следовательно, каждую такую совокупность деревьев  $\vartheta$ , входящую в  $\bar{O}$ , можно заменить на переход  $f$ , отображаемый функцией встраивания в  $u$ , так как они имеют одинаковые входы и выходы.

В результате получим О-процесс  $\varpi = (O, \psi) \in \wp_{max}(N)$ .

2. Так как  $N \equiv_\star N'$ ,  $\star \in \{mes, occ\}$ , всегда можно выбрать  $\varpi' = (O', \psi') \in \wp_{max}(N')$  и  $\beta$  такие, что:

- $\beta : \xi_O \simeq \xi_{O'}$ , если  $\star = mes$ ;
- $\beta : O \simeq O'$ , если  $\star = occ$ .

3. Для  $\vartheta$ , сконструированной таким образом, заменим в  $O'$  переход  $\beta(f)$ , встраивающийся в  $u'$ , на копию  $\vartheta'$  совокупности деревьев  $\vartheta$ , где все вершины вида  $\langle e, f \rangle$  заменяются на  $\langle e, \beta(f) \rangle$ . Тогда  $\beta(f)$  и  $\vartheta'$  имеют одинаковые выходы (из  $u'$ ) и одинаковые входы (в  $u'$ ).

Ясно, что сконструированный таким образом объект — О-процесс  $\bar{\omega}' = (\bar{O}', \bar{\psi}') \in \wp_{max}(\bar{N}')$ .



4. Пусть  $g \in T_{\bar{O}}$ . Определим отображение  $\bar{\beta}$  следующим образом:

$$\bar{\beta}(g) = \begin{cases} \beta(g), & g \text{ не входит ни в какую совокупность деревьев;} \\ \langle e, \beta(f) \rangle, & g = \langle e, f \rangle \text{ входит в некоторую совокупность деревьев } \vartheta. \end{cases}$$

□ (Конструкция (\*\*\*))

Осталось доказать следующие утверждения.

- $\bar{\beta} : \xi_{\bar{O}} \simeq \xi_{\bar{O}'}$ , если  $\star = mes$ ;
- $\bar{\beta} : \bar{O} \simeq \bar{O}'$ , если  $\star = occ$ .

Рассмотрим случай  $\star = mes$ , так как случай  $\star = occ$  — более простой. Пусть  $g, h \in T_{\bar{O}}$ . Возможны пять случаев:

1.  $g$  и  $h$  не входят ни в какую совокупность деревьев;
2.  $g$  входит в некоторую совокупность деревьев  $\vartheta$ ,  $h$  не входит ни в какую совокупность деревьев;
3.  $g$  не входит ни в какую совокупность деревьев,  $h$  входит в некоторую совокупность деревьев  $\vartheta$ ;
4.  $g$  и  $h$  входят в одну и ту же совокупность деревьев  $\vartheta$ ;
5.  $g$  входит в совокупность деревьев  $\vartheta_1$ ,  $h$  входит в совокупность деревьев  $\vartheta_2$  и  $\vartheta_1 \neq \vartheta_2$ .

Рассмотрим случай 5, так как случаи 1–4 — более простые. Тогда  $g = \langle e_1, f_1 \rangle$ ,  $h = \langle e_2, f_2 \rangle$ , где  $e_1 \in T_{O_{D_1}}$ ,  $e_2 \in T_{O_{D_2}}$  для  $\varpi_{D_1} = (O_{D_1}, \psi_{D_1})$ ,  $\varpi_{D_2} = (O_{D_2}, \psi_{D_2}) \in \wp_{max}(D)$ ,  $f_1, f_2 \in T_O$  для  $\varpi = (O, \psi) \in \wp_{max}(N)$ ,  $f_1$  и  $f_2$  детализуются в  $\bar{O}$  в разные совокупности деревьев  $\vartheta_1$  и  $\vartheta_2$  соответственно. Докажем сохранение отношений предшествования и конфликта.

- $g \prec_{\bar{O}} h \Leftrightarrow \langle e_1, f_1 \rangle \prec_{\bar{O}} \langle e_2, f_2 \rangle \Leftrightarrow$  (так как единственные связи  $\vartheta_1$  и  $\vartheta_2$  с окружением процесса — через их минимальные и максимальные переходы и все минимальные (максимальные) переходы имеют с этим окружением одинаковые связи)  $\forall i, j, k \langle e_{max1}^{ij}, f_1 \rangle \prec_{\bar{O}} \langle e_{min2}^k, f_2 \rangle \Leftrightarrow$  (по конструкции (\*\*\*))  $f_1 \prec_O f_2 \Leftrightarrow$  (так как  $\beta : \xi_O \simeq \xi_{O'}$ )  $\beta(f_1) \prec_{O'} \beta(f_2) \Leftrightarrow$  (по конструкции (\*\*\*))  $\forall i, j, k \langle e_{max1}^{ij}, \beta(f_1) \rangle \prec_{O'} \langle e_{min2}^k, \beta(f_2) \rangle \Leftrightarrow \langle e_1, \beta(f_1) \rangle \prec_{\bar{O}'} \langle e_2, \beta(f_2) \rangle \Leftrightarrow$  (по определению  $\bar{\beta}$ )  $\bar{\beta}(\langle e_1, f_1 \rangle) \prec_{\bar{O}'} \bar{\beta}(\langle e_2, f_2 \rangle) \Leftrightarrow \bar{\beta}(g) \prec_{\bar{O}'} \bar{\beta}(h)$ .
- $g \#_{\bar{O}} h \Leftrightarrow \langle e_1, f_1 \rangle \#_{\bar{O}} \langle e_2, f_2 \rangle \Leftrightarrow$  (так как единственные связи  $\vartheta_1$  и  $\vartheta_2$  с окружением процесса — через их минимальные и максимальные переходы и все минимальные (максимальные) переходы имеют с этим окружением одинаковые связи)  $\forall i, k \langle e_{min1}^i, f_1 \rangle \#_{\bar{O}} \langle e_{min2}^k, f_2 \rangle \Leftrightarrow$  (по конструкции (\*\*\*))  $f_1 \#_O f_2 \Leftrightarrow$  (так как  $\beta : \xi_O \simeq \xi_{O'}$ )  $\beta(f_1) \#_{O'} \beta(f_2) \Leftrightarrow$  (по конструкции (\*\*\*))  $\forall i, k \langle e_{min1}^i, \beta(f_1) \rangle \#_{\bar{O}'} \langle e_{min2}^k, \beta(f_2) \rangle \Leftrightarrow \langle e_1, \beta(f_1) \rangle \#_{\bar{O}'} \langle e_2, \beta(f_2) \rangle \Leftrightarrow$  (по определению  $\bar{\beta}$ )  $\bar{\beta}(\langle e_1, f_1 \rangle) \#_{\bar{O}'} \bar{\beta}(\langle e_2, f_2 \rangle) \Leftrightarrow \bar{\beta}(g) \#_{\bar{O}'} \bar{\beta}(h)$ . □

### Доказательство предложения 1.3.4.

Для  $\pi \in \Pi(N)$  обозначим  $[\pi] = \{\bar{\pi} \in \Pi(N) \mid \pi \xleftrightarrow{\tau}_{pomhbr} \bar{\pi}\}$ . Если  $(\pi, \sigma) \in Runs(N)$  и  $\sigma = v_1 \cdots v_n$ , определим след пары  $(\pi, \sigma)$  как  $trace(\pi, \sigma) = [\pi_N]l_C(v_1)[\pi_1] \cdots [\pi_{n-1}]l_C(v_n)[\pi_n]$ . След по модулю  $\tau$  пары  $(\pi, \sigma)$ , обозначение  $stutt(\pi, \sigma)$ , получается из  $trace(\pi, \sigma)$  заменой всех троек вида  $R\tau R$  на  $R$ .

( $\Leftarrow$ ) Пусть  $N \xleftrightarrow{\tau}_{pomhbr} N'$ ,  $(\pi, \sigma) \in Runs(N)$ ,  $(\pi', \sigma') \in Runs(N')$  и  $stutt(\pi, \sigma) = R_1 a_1 R_2 \cdots R_{n-1} a_n R_n$ ,  $stutt(\pi', \sigma') = R'_1 a'_1 R'_2 \cdots R'_{m-1} a'_m R'_m$ .  $stutt(\pi, \sigma)$  и  $stutt(\pi', \sigma')$  изоморфны, запись  $stutt(\pi, \sigma) \simeq stutt(\pi', \sigma')$ , если:

1.  $n = m$ ;
2.  $\forall i (1 \leq i \leq n) a_i = a'_i$ ;
3.  $\forall i (1 \leq i \leq n)$  и  $\pi_i \in R_i$ ,  $\pi'_i \in R'_i : \pi_i \xleftrightarrow{\tau}_{pomhbr} \pi'_i$ .

Определим отношение  $\mathcal{S}$  следующим образом:  $\mathcal{S} = \{((\pi, \sigma), (\pi', \sigma')) \mid (\pi, \sigma) \in Runs(N), (\pi', \sigma') \in Runs(N'), stutt(\pi, \sigma) \simeq stutt(\pi', \sigma')\}$ . Докажем, что  $\mathcal{S} : N \xleftrightarrow{\tau}_{pomhbr} N'$ .

1.  $((\pi_N, \varepsilon), (\pi_{N'}, \varepsilon)) \in \mathcal{S}$ , так как  $\pi_N \xleftrightarrow{\tau}_{\text{pomhbr}} \pi_{N'}$ .

2. Пусть  $((\pi, \sigma), (\pi', \sigma')) \in \mathcal{S}$ .

- (обратно)

Имеем  $\exists \beta : \text{vis}(\rho_C) \simeq \text{vis}(\rho_{C'})$ . Пусть  $(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\hat{\pi}} (\pi, \sigma)$ . Тогда  $\exists i$  ( $1 \leq i \leq n$ )  $(\tilde{\pi}, \tilde{\sigma}) \in R_i$  из  $\text{trace}(\pi, \sigma)$ . Так как  $\text{stutt}(\pi, \sigma) \simeq \text{stutt}(\pi', \sigma')$ , имеем  $\exists k$  ( $1 \leq k \leq n$ ) такое, что  $R_i$  соответствует  $R'_k$  из  $\text{trace}(\pi', \sigma')$ . Тогда  $\tilde{\pi} \xleftrightarrow{\tau}_{\text{pomhbr}} \pi'(k)$  и  $\exists \beta : \text{vis}(\rho_{\tilde{C}}) \simeq \text{vis}(\rho_{C'(k)})$ ,  $((\tilde{\pi}, \tilde{\sigma}), (\pi(k)', \sigma(k)')) \in \mathcal{S}$ .

Рассмотрим обратное расширение  $(\pi'(k), \sigma'(k)) \xrightarrow{\hat{\pi}'} (\pi', \sigma')$ . Так как  $\beta$  и  $\tilde{\beta}$  — изоморфизмы, имеем  $\text{vis}(\rho_{\tilde{C}}) \simeq \text{vis}(\rho_{\tilde{C}'})$ .

- (прямо) Очевидно.

3. Как пункт 2, но роли  $N$  и  $N'$  меняются.

( $\Rightarrow$ ) Пусть  $N \xleftrightarrow{\tau}_{\text{pombrpomf}} N'$ . Определим отношение  $\mathcal{S}$  следующим образом:  
 $\mathcal{S} = \{(\pi, \pi', \beta_{\sigma}^{\sigma'}) \mid (\pi, \sigma) \xleftrightarrow{\tau}_{\text{pombrpomf}} (\pi', \sigma')\}$ . Докажем, что  $\mathcal{S} : N \xleftrightarrow{\tau}_{\text{pomhbr}} N'$ .

1.  $(\pi_N, \pi_{N'}, \emptyset) \in \mathcal{S}$  так как  $\beta_{\varepsilon}^{\varepsilon} = \emptyset$  и  $(\pi_N, \varepsilon) \xleftrightarrow{\tau}_{\text{pombrpomf}} (\pi_{N'}, \varepsilon)$ .

2. Пусть  $(\pi, \pi', \beta_{\sigma}^{\sigma'}) \in \mathcal{S}$ . Тогда по определению  $\mathcal{S}$ ,  $(\pi, \sigma) \xleftrightarrow{\tau}_{\text{pombrpomf}} (\pi', \sigma')$  и обратное расширение  $(\pi_N, \varepsilon) \xrightarrow{\pi} (\pi, \sigma)$  имитируется  $(\tilde{\pi}', \varepsilon) \xrightarrow{\tilde{\pi}'} (\pi', \sigma')$  для некоторого  $\tilde{\pi}'$  такого, что  $\pi_{N'} \Rightarrow \tilde{\pi}'$ . Если  $\pi = (C, \varphi)$  и  $\tilde{\pi}' = (\tilde{C}, \tilde{\varphi})$ , имеем  $\beta_{\sigma}^{\sigma'} : \text{vis}(\rho_C) \simeq \text{vis}(\rho_{\tilde{C}'})$ . Так как  $\text{vis}(T_C) = \text{vis}(T_{\tilde{C}'})$ , где  $\pi' = (C', \varphi')$ , имеем  $\beta_{\sigma}^{\sigma'} : \text{vis}(\rho_C) \simeq \text{vis}(\rho_{C'})$ .

3. Пусть  $(\pi, \pi', \beta_{\sigma}^{\sigma'}) \in \mathcal{S}$  и  $\pi \xrightarrow{v} \tilde{\pi}$ . Тогда по определению  $\mathcal{S}$ ,  $(\pi, \sigma) \xleftrightarrow{\tau}_{\text{pombrpomf}} (\pi', \sigma')$  и  $(\pi, \sigma) \rightarrow (\tilde{\pi}, \sigma v)$ . Возможны следующие два случая.

(a)  $l_{\tilde{C}}(v) \neq \tau$ .

Так как  $N \xleftrightarrow{\tau}_{\text{pombrpomf}} N'$ , имеем  $\exists v'_i, w'_j$  ( $1 \leq i \leq n$ ,  $1 \leq j \leq m$ ),  $v', \pi'_1, \pi'_2$  такие, что  $(\pi', \sigma') \xrightarrow{v'_1} \dots \xrightarrow{v'_n} (\pi'_1, \sigma' v'_1 \dots v'_n) \xrightarrow{v'} (\pi'_2, \sigma' v'_1 \dots v'_n v')$   $\xrightarrow{w'_1} \dots \xrightarrow{w'_m} (\tilde{\pi}', \sigma' v'_1 \dots v'_n v' w'_1 \dots w'_m)$ ,  $(\tilde{\pi}, \sigma v) \xleftrightarrow{\tau}_{\text{pombrpomf}} (\tilde{\pi}', \sigma' v'_1 \dots v'_n v' w'_1 \dots w'_m)$  и  $l_{\tilde{C}}(v) = l_{\tilde{C}'}(v')$ ,  $\forall i, j$  ( $1 \leq i \leq n$ ,  $1 \leq j \leq m$ )  $l_{\tilde{C}'}(v'_i) = l_{\tilde{C}'}(w'_j) = \tau$ .

Следовательно,  $\pi' \xrightarrow{v'_1} \dots \xrightarrow{v'_n} \pi'_1 \xrightarrow{v'} \pi'_2 \xrightarrow{w'_1} \dots \xrightarrow{w'_m} \tilde{\pi}'$ .

Обратное расширение  $(\pi'_2, \sigma' v'_1 \dots v'_n v') \rightarrow (\tilde{\pi}', \sigma' v'_1 \dots v'_n v' w'_1 \dots w'_m)$  имитируется пустым обратным расширением  $(\tilde{\pi}, \sigma v)$ . Следовательно,  $(\tilde{\pi}, \sigma v) \xleftrightarrow{\tau}_{\text{pombrpomf}} (\pi'_2, \sigma' v'_1 \dots v'_n v')$ . Поэтому

$(\tilde{\pi}, \pi'_2, \beta_{\sigma v}^{\sigma' v'_1 \dots v'_n v'}) \in \mathcal{S}$ .

Рассмотрим обратное расширение  $(\pi'_1, \sigma' v'_1 \dots v'_n) \rightarrow (\pi'_2, \sigma' v'_1 \dots v'_n v')$ . Оно имитируется некоторым обратным расширением  $(\tilde{\pi}, \tilde{\sigma}) \Rightarrow (\pi, \sigma) \rightarrow (\tilde{\pi}, \sigma v)$  таким, что  $(\tilde{\pi}, \tilde{\sigma}) \xleftrightarrow{\tau}_{\text{pombrpomf}} (\pi'_1, \sigma' v'_1 \dots v'_n)$ . Так как  $(\pi', \sigma') \Rightarrow (\pi'_1, \sigma' v'_1 \dots v'_n)$  и  $(\pi, \sigma) \xleftrightarrow{\tau}_{\text{pombrpomf}} (\pi', \sigma')$ , по X-лемме 2 имеем

$(\pi, \sigma) \xleftrightarrow{\tau}_{\text{pombrpomf}} (\pi'_1, \sigma' v'_1 \dots v'_n)$ . Таким образом, получаем  $(\pi, \pi'_1, \beta_{\sigma}^{\sigma' v'_1 \dots v'_n}) \in \mathcal{S}$ .

Следовательно, имеем моделирование, так как  $\pi' \Rightarrow \pi'_1 \xrightarrow{a} \pi'_2$  и  $(\pi, \pi'_1, \beta_{\sigma}^{\sigma' v'_1 \dots v'_n}) \in \mathcal{S}$ ,

$(\tilde{\pi}, \pi'_2, \beta_{\sigma v}^{\sigma' v'_1 \dots v'_n v'}) \in \mathcal{S}$ .

(b)  $l_{\tilde{C}}(v) = \tau$ .

Так как  $N \xleftrightarrow{\tau}_{\text{pombrpomf}} N'$ , имеем  $\exists \pi'_i$  ( $1 \leq i \leq n$ ) такие, что  $(\pi', \sigma') \Rightarrow (\pi'_1, \sigma' v_1) \Rightarrow \dots \Rightarrow (\pi'_n, \sigma' v'_1 \dots v'_n) \Rightarrow (\tilde{\pi}', \sigma' v'_1 \dots v'_n)$  и  $(\tilde{\pi}, \sigma v) \xleftrightarrow{\tau}_{\text{pombrpomf}} (\tilde{\pi}', \sigma' v'_1 \dots v'_n)$ .

i. Если  $n = 0$ , все доказано.

ii. Если  $n \geq 1$ , и обратное расширение  $(\pi'_{n-1}, \sigma' v'_1 \dots v'_{n-1}) \Rightarrow (\pi'_n, \sigma' v'_1 \dots v'_n)$  имитируется пустым обратным расширением  $(\tilde{\pi}, \sigma v)$ , все доказано для  $n = 1$ , а для  $n \geq 2$  продолжим это рассуждение. Возможны два случая.

В первом случае  $(\tilde{\pi}, \sigma v) \xleftrightarrow{\tau}_{\text{pombrpomf}} (\pi', \sigma')$  и  $(\tilde{\pi}, \pi', \beta_{\sigma v}^{\sigma'}) \in \mathcal{S}$ .

Во втором случае  $\exists m$  ( $1 \leq m \leq n - 1$ ) такое, что  $(\tilde{\pi}, \sigma v) \xleftrightarrow{\tau}_{\text{pombrpomf}} (\pi'_m, \sigma' v'_1 \dots v'_m)$  и  $(\tilde{\pi}, \pi'_m, \beta_{\sigma v}^{\sigma' v'_1 \dots v'_m}) \in \mathcal{S}$ .

Обратное расширение  $(\pi'_{m-1}, \sigma' v'_1 \cdots v'_{m-1}) \Rightarrow (\pi'_m, \sigma' v'_1 \cdots v'_m)$  имитируется некоторым обратным расширением  $(\bar{\pi}, \bar{\sigma}) \Rightarrow (\pi, \sigma)$  таким, что  $(\bar{\pi}, \bar{\sigma}) \xleftrightarrow{\tau} \text{rombromf} (\pi'_{m-1}, \sigma' v'_1 \cdots v'_{m-1})$ . По X-лемме 2,  $(\pi, \sigma) \xleftrightarrow{\tau} \text{rombromf} (\pi'_{m-1}, \sigma' v'_1 \cdots v'_{m-1})$ . Таким образом, получим  $(\pi, \pi'_{m-1}, \beta_{\sigma}^{\sigma' v'_1 \cdots v'_{m-1}}) \in \mathcal{S}$ .

Следовательно, имеем моделирование, так как  $\pi' \Rightarrow \pi'_{m-1} \xrightarrow{\tau} \pi'_m$  и  $(\pi, \pi'_{m-1}, \beta_{\sigma}^{\sigma' v'_1 \cdots v'_{m-1}}) \in \mathcal{S}$ ,  $(\tilde{\pi}, \pi'_m, \beta_{\sigma v}^{\sigma' v'_1 \cdots v'_m}) \in \mathcal{S}$ .

4. Как пункт 3, но роли  $N$  и  $N'$  меняются. □

## Приложение В

# Описание программы CANON

Программа CANON объемом около 3000 строк на языке Си предназначена для преобразования произвольных формул алгебры  $AFP_2$  к каноническому виду. Эта программа основана на предложениях главы 3.

Тело функции main имеет следующий вид.

вывод информации о предназначении программы и формате вводимой формулы;

представление формулы в виде двусвязного списка;

выдача сообщения "формула считана";

преобразование списка в двусвязное дерево;

уничтожение списка;

печать формулы;

```
step=1; /*номер шага*/
```

```
do
```

```
{
```

```
  вывод step;
```

```
  nar=0; /*число применений правил на шаге с номером step*/
```

```
  применение правил;
```

```
  вывод nar;
```

```
  step++; /*следующий шаг*/
```

```
}
```

```
while(nar!=0);
```

вывод канонической формы формулы;

Формула должна вводиться по определенному формату. Символы  $\bar{\quad}$ ,  $\delta$ ,  $\bar{\bar{\quad}}$ ,  $\bar{\bar{\bar{\quad}}}$ ,  $\bar{\bar{\bar{\bar{\quad}}}}$ ,  $\bar{\bar{\bar{\bar{\bar{\quad}}}}$ ,  $\bar{\bar{\bar{\bar{\bar{\bar{\quad}}}}$ ,  $\nabla$ ,  $\vee$  заменяются на имеющиеся на клавиатуре в соответствии с таблицей В.1.

Кроме того, формула может иметь один из следующих видов.

1. a;
2. -a , \*a ;

Исходное обозначение символа	-	$\delta$	$\parallel$	$\tilde{\parallel}$	;	$\parallel$	$\nabla$	$\vee$
Имя символьной константы	NOT	DLT	NOC	MNO	PRC	CNC	ALT	DSJ
ASCII-представление символа	-	*	'	~	;		#	+

Таблица В.1: Представление символов в программе CANON

3. 'a , ~a ;
4. '(P) , ~(P) ;
5. a#b , a+b , a|b , a;b ;
6. a#(P) , a+(P) , a|(P) , a;(P) ;
7. (P)#a , (P)+a , (P)|a , (P);a ;
8. (P)#(Q) , (P)+(Q) , (P)|(Q) , (P);(Q) .

Здесь  $P$  и  $Q$  — формулы видов 2–8,  $a$  и  $b$  — символы элементарных действий. Как видно из описания формата формулы, она не должна иметь внешних скобок.

В процессе ввода формула представляется в виде двусвязного списка с элементами, которые содержат указатели на левого и правого соседей, а также один символ.

Затем по этому списку строится представление формулы в виде двусвязного дерева с вершинами, содержащими указатели на отца, 1 и 2 сыновей, а также два символа, которые могут представлять элементарную формулу, то есть формулу видов 1 и 2. Таким образом, формула преобразуется в дерево с промежуточными вершинами, которые соответствуют функциональным символам и листьями, представляющими элементарные формулы. В этом представлении подформулам соответствуют поддеревья.

После этого, для экономии памяти, список уничтожается и формула, представленная в виде дерева, печатается (проверка правильности преобразования в дерево).

**Пример В.0.9** На рисунке В.1(а) показано представление в виде двусвязного списка, а на рисунке В.1(б) — в виде дерева формулы  $a \vee \delta b$ . Крестиками помечены поля структур, имеющие значение NULL. В дальнейшем будем использовать упрощенное изображение дерева В.1(с), вершины которого изображаются окружностями с вписанными в них исходными обозначениями символов (а не их ASCII-символами, используемыми в программе). Двойные связи будут изображаться одинарными линиями, а неизвестные поддеревья — треугольниками.

Потом начинается процесс применения правил к формуле. На каждом шаге с номером step просматриваются все правила, и, если они применимы, применяются, а число применений правил на данном шаге, шаг, увеличивается. В конце шага проверяется, равно ли число примененных на нем правил нулю. Если это так, то ни одно из правил не применимо к формуле, а это, по предложению 3.1.9 означает, что она имеет канонический вид. В этом случае циклическое применение правил заканчивается, формула выводится на экран и программа заканчивает работу. В противном случае номер шага увеличивается на единицу, и к формуле опять применяются правила.

Назовем правило непосредственно применимым к дереву, если оно применимо ко всему этому дереву (то есть формула, соответствующая дереву, интерпретируется как левая часть некоторого правила переписывания). Правило косвенно применимо к дереву, если оно непосредственно применимо к некоторому его поддереву, за исключением самого этого дерева. Правило применимо к дереву, если оно непосредственно или косвенно применимо к нему.

**Пример В.0.10** Правило 1.1 косвенно применимо к дереву на рисунке В.2, соответствующему формуле  $a \parallel (b; (c; d))$ , а именно, к правому его поддереву, корневая вершина которого содержит первый символ предшествования (на нее установлен указатель root).

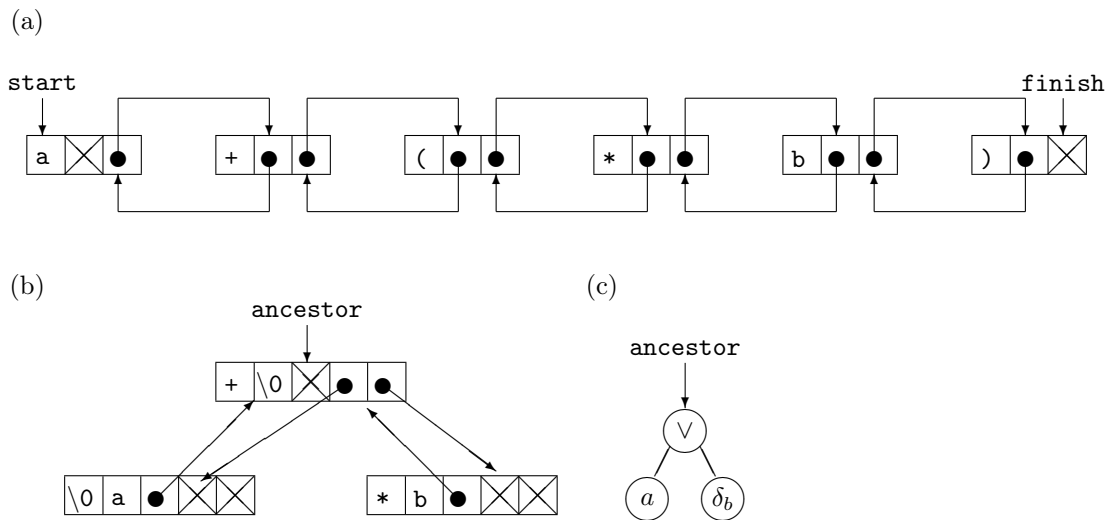


Рис. В.1: Представление формулы двусвязным списком и деревом

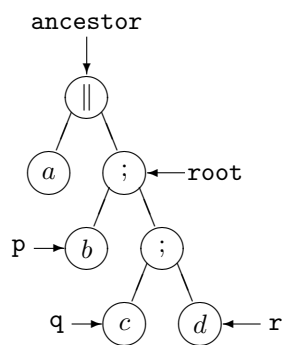


Рис. В.2: Дерево, к которому применимо правило 1.1

Тела правил имеют следующий вид.

```
if(root!=NULL)
{
  if(правило непосредственно применимо к дереву с указателем на корень root)
  {
    установка указателей на поддеревья, соответствующие подформулам в правиле вывода;

    вывод информации о применяемом правиле и печать поддеревьев;

    преобразование дерева в соответствии с применяемым правилом;

    (*addrnar)++; /*увеличение на единицу счетчика числа примененных на данном шаге правил*/

    вывод новой формулы;
  }
  else
  {
    применение правила к левому поддереву;

    применение правила к правому поддереву;
  }
}
```

Таким образом, при применении правила к дереву просматриваются все его поддеревья, пока указатель на корень текущего поддерева, *root*, не станет равным *NULL*. То есть, если правило применимо к дереву, оно применяется.

На рисунке В.3 показано, как преобразуются соответствующие формулам дерева при применении правил 1.1 и 2.1 системы  $RWS_2$ . Символом ' $\approx r$ ' помечено поддерево, структура которого совпадает со структурой поддерева с указателем на корень *r*.

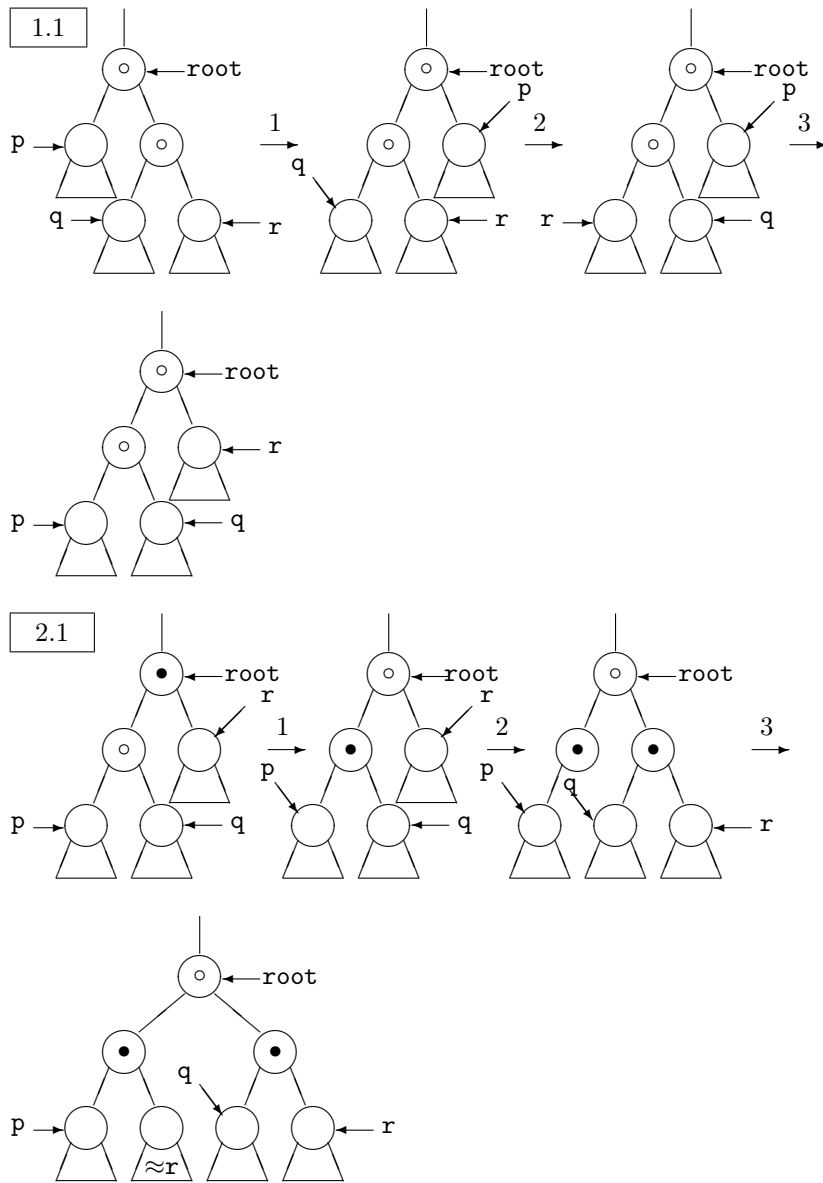


Рис. В.3: Преобразование деревьев при применении правил 1.1 и 2.1



## Приложение С

# Примеры работы программы CANON

В этой главе демонстрируется работа программы CANON по приведению формул алгебры  $AFP_2$  к каноническому виду.

1. Формула  $(a; (b \nabla c)) \parallel (a \nabla b)$ .

The author of this program is I.V. Tarasyuk  
Program CANON transforms formulas of algebras AFP\_2, AFLP\_2  
into canonical form

Input formula should be in one of the following forms:

1. a
2. -a \*a
3. 'a ~a
4. '(P) ~(P)
5. a;b a|b a#b a+b
6. a;(P) a|(P) a#(P) a+(P)
7. (P);a (P)|a (P)#a (P)+a
8. (P);(Q) (P)|(Q) (P)#(Q) (P)+(Q)

where a and b are symbols of elementary actions,  
P and Q are formulas types 2-8

Input formula  
Sign of end is EOF

Formula has been read

Your formula is:  
 $(a; (b \# c)) \parallel (a \# b)$

Step 1

Rule 3.1 is applied  
P=b  
Q=c  
New formula is:  
 $(a; ((b \parallel ('c)) + (('b \parallel c)))) \parallel (a \# b)$

Rule 3.1 is applied  
P=a  
Q=b

New formula is:  
 $(a; ((b|('c))+ (('b)|c))) | ((a|('b))+ (('a)|b))$

Rule 4.3 is applied

P=c

New formula is:  
 $(a; ((b|(-c))+ (('b)|c))) | ((a|('b))+ (('a)|b))$

Rule 4.3 is applied

P=b

New formula is:  
 $(a; ((b|(-c))+ ((-b)|c))) | ((a|('b))+ (('a)|b))$

Rule 4.3 is applied

P=b

New formula is:  
 $(a; ((b|(-c))+ ((-b)|c))) | ((a|(-b))+ (('a)|b))$

Rule 4.3 is applied

P=a

New formula is:  
 $(a; ((b|(-c))+ ((-b)|c))) | ((a|(-b))+ ((-a)|b))$

Number of applied rules in step 1 is 6

Step 2

Rule 2.2 is applied

P=(a; ((b|(-c))+ ((-b)|c)))

Q=(a|(-b))

R=((-a)|b)

New formula is:  
 $((a; ((b|(-c))+ ((-b)|c))) | (a|(-b))) + ((a; ((b|(-c))+ ((-b)|c))) | ((-a)|b))$

Number of applied rules in step 2 is 1

Step 3

Rule 1.1 is applied

P=(a; ((b|(-c))+ ((-b)|c)))

Q=a

R=(-b)

New formula is:  
 $((a; ((b|(-c))+ ((-b)|c))) | a) | (-b) + ((a; ((b|(-c))+ ((-b)|c))) | ((-a)|b))$

Rule 1.1 is applied

P=(a; ((b|(-c))+ ((-b)|c)))

Q=(-a)

R=b

New formula is:  
 $((a; ((b|(-c))+ ((-b)|c))) | a) | (-b) + ((a; ((b|(-c))+ ((-b)|c))) | (-a)) | b$

Rule 2.2 is applied

P=a

Q=(b|(-c))

R=((-b)|c)

New formula is:

$((((a;(b|(-c)))+(a;((-b)|c)))|a)|(-b))+(((a;(b|(-c)))+((-b)|c)))|(-a)|b)$

Rule 2.2 is applied

P=a

Q=(b|(-c))

R=((-b)|c)

New formula is:

$((((a;(b|(-c)))+(a;((-b)|c)))|a)|(-b))+(((a;(b|(-c)))+(a;((-b)|c)))|(-a)|b)$

Number of applied rules in step 3 is 4

Step 4

Rule 2.1 is applied

P=(a;(b|(-c)))

Q=(a;((-b)|c))

R=a

New formula is:

$((((a;(b|(-c)))|a)+(a;((-b)|c))|a))|(-b))+$   
 $((((a;(b|(-c)))+(a;((-b)|c)))|(-a))|b)$

Rule 2.1 is applied

P=(a;(b|(-c)))

Q=(a;((-b)|c))

R=(-a)

New formula is:

$((((a;(b|(-c)))|a)+(a;((-b)|c))|a))|(-b))+$   
 $((((a;(b|(-c)))|(-a))+(a;((-b)|c))|(-a)))|b)$

Rule 2.2 is applied

P=a

Q=b

R=(-c)

New formula is:

$(((((a;b)|(a;(-c)))|a)+(a;((-b)|c))|a))|(-b))+$   
 $((((a;(b|(-c)))|(-a))+(a;((-b)|c))|(-a)))|b)$

Rule 2.2 is applied

P=a

Q=(-b)

R=c

New formula is:

$(((((a;b)|(a;(-c)))|a)+(a;((-b))|(a;c))|a))|(-b))+$   
 $((((a;(b|(-c)))|(-a))+(a;((-b)|c))|(-a)))|b)$

Rule 2.2 is applied

P=a

Q=b

R=(-c)

New formula is:

$(((((a;b)|(a;(-c)))|a)+(a;((-b))|(a;c))|a))|(-b))+$   
 $(((((a;b)|(a;(-c)))|(-a))+(a;((-b)|c))|(-a)))|b)$

Rule 2.2 is applied

P=a

$Q=(-b)$   
 $R=c$   
 New formula is:  
 $(((((a;b)|(a;(-c)))|a)+(((a;(-b))|(a;c))|a))|(-b))+$   
 $(((((a;b)|(a;(-c)))|(-a))+(((a;(-b))|(a;c))|(-a)))|b)$

Rule 5.3 is applied

$P=a$   
 $Q=(-c)$   
 New formula is:  
 $(((((a;b)|(a|(-c)))|a)+(((a;(-b))|(a;c))|a))|(-b))+$   
 $(((((a;b)|(a;(-c)))|(-a))+(((a;(-b))|(a;c))|(-a)))|b)$

Rule 5.3 is applied

$P=a$   
 $Q=(-b)$   
 New formula is:  
 $(((((a;b)|(a|(-c)))|a)+(((a|(-b))|(a;c))|a))|(-b))+$   
 $(((((a;b)|(a;(-c)))|(-a))+(((a;(-b))|(a;c))|(-a)))|b)$

Rule 5.3 is applied

$P=a$   
 $Q=(-c)$   
 New formula is:  
 $(((((a;b)|(a|(-c)))|a)+(((a|(-b))|(a;c))|a))|(-b))+$   
 $(((((a;b)|(a|(-c)))|(-a))+(((a;(-b))|(a;c))|(-a)))|b)$

Rule 5.3 is applied

$P=a$   
 $Q=(-b)$   
 New formula is:  
 $(((((a;b)|(a|(-c)))|a)+(((a|(-b))|(a;c))|a))|(-b))+$   
 $(((((a;b)|(a|(-c)))|(-a))+(((a|(-b))|(a;c))|(-a)))|b)$

Number of applied rules in step 4 is 10

Step 5

Rule 1.1 is applied

$P=(a;b)$   
 $Q=a$   
 $R=(-c)$   
 New formula is:  
 $(((((a;b)|a)|(-c))|a)+(((a|(-b))|(a;c))|a))|(-b))+$   
 $(((((a;b)|a)|(-c))|(-a))+(((a|(-b))|(a;c))|(-a)))|b)$

Rule 1.1 is applied

$P=(a;b)$   
 $Q=a$   
 $R=(-c)$   
 New formula is:  
 $(((((a;b)|a)|(-c))|a)+(((a|(-b))|(a;c))|a))|(-b))+$   
 $(((((a;b)|a)|(-c))|(-a))+(((a|(-b))|(a;c))|(-a)))|b)$

Rule 2.1 is applied

$P = (((a;b) | a) | (-c)) | a$

$Q = (((a | (-b)) | (a;c)) | a)$

$R = (-b)$

New formula is:

$(((((a;b) | a) | (-c)) | a) | (-b)) + (((a | (-b)) | (a;c)) | a) | (-b)) +$   
 $(((((a;b) | a) | (-c)) | (-a)) + ((a | (-b)) | (a;c)) | (-a)) | b)$

Rule 2.1 is applied

$P = (((a;b) | a) | (-c)) | (-a)$

$Q = ((a | (-b)) | (a;c)) | (-a)$

$R = b$

New formula is:

$(((((a;b) | a) | (-c)) | a) | (-b)) + (((a | (-b)) | (a;c)) | a) | (-b)) +$   
 $(((((a;b) | a) | (-c)) | (-a)) | b) + (((a | (-b)) | (a;c)) | (-a)) | b)$

Number of applied rules in step 5 is 4

Step 6

Rule 1.1 is applied

$P = ((((((a;b) | a) | (-c)) | a) | (-b)) + (((a | (-b)) | (a;c)) | a) | (-b))$

$Q = (((a;b) | a) | (-c)) | (-a) | b$

$R = (((a | (-b)) | (a;c)) | (-a)) | b$

New formula is:

$(((((a;b) | a) | (-c)) | a) | (-b)) + (((a | (-b)) | (a;c)) | a) | (-b)) +$   
 $((((a;b) | a) | (-c)) | (-a)) | b) + (((a | (-b)) | (a;c)) | (-a)) | b$

Number of applied rules in step 6 is 1

Step 7

Rule 7.1 is applied

$P = (a | (-b))$

$P' = a$

$Q = (a;c)$

New formula is:

$(((((a;b) | a) | (-c)) | a) | (-b)) + (((a;c) | (-b)) | a) | (-b)) +$   
 $((((a;b) | a) | (-c)) | (-a)) | b) + (((a | (-b)) | (a;c)) | (-a)) | b$

Rule 7.1 is applied

$P = (a | (-b))$

$P' = a$

$Q = (a;c)$

New formula is:

$(((((a;b) | a) | (-c)) | a) | (-b)) + (((a;c) | (-b)) | a) | (-b)) +$   
 $((((a;b) | a) | (-c)) | (-a)) | b) + (((a;c) | (-b)) | (-a)) | b$

Rule 7.2 is applied

$P = ((a;b) | a) | (-c)$

$P' = (a;b)$

$Q = a$

New formula is:

$(((((a;b) | a) | (-c)) | (-b)) + (((a;c) | (-b)) | a) | (-b)) +$   
 $((((a;b) | a) | (-c)) | (-a)) | b) + (((a;c) | (-b)) | (-a)) | b$

Rule 7.2 is applied

$P = (a;c) | (-b)$

$P'=(a;c)$   
 $Q=a$   
 New formula is:  
 $(((((a;b)|a)|(-c))|(-b))+((a;c)|(-b))|(-b))+$   
 $(((((a;b)|a)|(-c))|(-a))|b))+((a;c)|(-b))|(-a)|b)$

Rule 7.2 is applied  
 $P=(((a;b)|a)|(-c))|(-a)$   
 $P'=(a;b)$   
 $Q=b$

New formula is:  
 $(((((a;b)|a)|(-c))|(-b))+((a;c)|(-b))|(-b))+$   
 $((a;b)|a)|(-c))|(-a))+((a;c)|(-b))|(-a)|b)$

Rule 7.3 is applied

$P=(((a;b)|a)|(-c))$   
 $P'=a$   
 $Q=(-a)$

New formula is:  
 $(((((a;b)|a)|(-c))|(-b))+((a;c)|(-b))|(-b))+$   
 $((a;b)|(*a))|(-c))+((a;c)|(-b))|(-a)|b)$

Rule 7.4 is applied

$P=(((a;c)|(-b))|(-a))$   
 $P'=(-b)$   
 $Q=b$

New formula is:  
 $(((((a;b)|a)|(-c))|(-b))+((a;c)|(-b))|(-b))+$   
 $((a;b)|(*a))|(-c))+((a;c)|(*b))|(-a)$

Rule 7.5 is applied

$P=(a;b)$   
 $P'=(a;b)$   
 $Q>(*a)$

New formula is:  
 $(((((a;b)|a)|(-c))|(-b))+((a;c)|(-b))|(-b))+$   
 $(((*b)|(*a))|(-c))+((a;c)|(*b))|(-a)$

Rule 7.5 is applied

$P=((a;c)|(*b))$   
 $P'=(a;c)$   
 $Q=(-a)$

New formula is:  
 $(((((a;b)|a)|(-c))|(-b))+((a;c)|(-b))|(-b))+$   
 $(((*b)|(*a))|(-c))+(((c)|(*b))|(*a))$

Rule 7.6 is applied

$P=(((a;b)|a)|(-c))$   
 $P'=(a;b)$   
 $Q=(-b)$

New formula is:  
 $(((((a|a)|(-c))|(*b))+((a;c)|(-b))|(-b))+$   
 $(((*b)|(*a))|(-c))+(((c)|(*b))|(*a))$

Rule 7.9 is applied

P=a

P'=a

Q=a

New formula is:

$((a|-c)|(*b)) + ((a;c)|(-b)) + ((*b)|(*a)) + ((*c)|(*b)) + ((*a))$

Rule 7.9 is applied

P= $((a;c)|(-b))$

P'=(-b)

Q=(-b)

New formula is:

$((a|-c)|(*b)) + ((a;c)|(-b)) + ((*b)|(*a)) + ((*c)|(*b)) + ((*a))$

Number of applied rules in step 7 is 12

Step 8

Rule 6.1 is applied

P= $((*b)|(*a))$

P'=(\*a)

Q=(-c)

New formula is:

$((a|-c)|(*b)) + ((a;c)|(-b)) + ((*b)|(*a)) + ((*c)) + ((*c)|(*b)) + ((*a))$

Rule 6.2 is applied

P= $(a|-c)$

P'=(-c)

Q>(\*b)

New formula is:

$((a|(*c))|(*b)) + ((a;c)|(-b)) + ((*b)|(*a)) + ((*c)) + ((*c)|(*b)) + ((*a))$

Number of applied rules in step 8 is 2

Step 9

Rule 9.1 is applied

P= $((a|(*c))|(*b)) + ((a;c)|(-b)) + ((*b)|(*a)) + ((*c))$

P'= $((*b)|(*a)) + ((*c))$

Q= $((*c)|(*b)) + ((*a))$

New formula is:

$((a|(*c))|(*b)) + ((a;c)|(-b)) + ((*b)|(*a)) + ((*c))$

Number of applied rules in step 9 is 1

Step 10

Rule 10.1 is applied

P= $((a|(*c))|(*b)) + ((a;c)|(-b))$

P'= $((a;c)|(-b))$

Q= $((*b)|(*a)) + ((*c))$

New formula is:

$(a|(*c))|(*b)) + ((a;c)|(-b))$

Rule 10.2 is applied

P= $(a|(*c))|(*b)$

P'= $(a|(*c))|(*b)$

$Q = ((a;c) | (-b))$   
New formula is:  
 $(a;c) | (-b)$

Number of applied rules in step 10 is 2

Step 11

Number of applied rules in step 11 is 0

Canonical form is:  
 $(a;c) | (-b)$

Каноническая форма формулы  $(a;c) || \bar{b}$ .

2. Формула  $(a \nabla (b;e)) || (d \nabla (c;e))$ .

The author of this program is I.V. Tarasyuk  
Program CANON transforms formulas of algebras AFP\_2, AFLP\_2  
into canonical form

Input formula should be in one of the following forms:

1. a
2. -a \*a
3. 'a ~a
4. '(P) ~(P)
5. a;b a|b a#b a+b
6. a;(P) a|(P) a#(P) a+(P)
7. (P);a (P)|a (P)#a (P)+a
8. (P);(Q) (P)|(Q) (P)#(Q) (P)+(Q)

where a and b are symbols of elementary actions,  
P and Q are formulas types 2-8

Input formula  
Sign of end is EOF

Formula has been read

Your formula is:  
 $(a\#(b;e)) | (d\#(c;e))$

Step 1

Rule 3.1 is applied  
 $P = a$   
 $Q = (b;e)$   
New formula is:  
 $((a | ('(b;e))) + (('a | (b;e))) | (d\#(c;e)))$

Rule 3.1 is applied  
 $P = d$   
 $Q = (c;e)$   
New formula is:  
 $((a | ('(b;e))) + (('a | (b;e))) | ((d | ('(c;e))) + (('d | (c;e))))$



Rule 4.1 is applied

P=b

Q=e

New formula is:

$((a|((b|e)))+(a|(b;e))|(d|((c;e)))+(d|(c;e)))$

Rule 4.1 is applied

P=c

Q=e

New formula is:

$((a|((b|e)))+(a|(b;e))|(d|((c|e)))+(d|(c;e)))$

Rule 4.3 is applied

P=b

New formula is:

$((a|((-b|e)))+(a|(b;e))|(d|((c|e)))+(d|(c;e)))$

Rule 4.3 is applied

P=e

New formula is:

$((a|((-b|e)))+(a|(b;e))|(d|((c|e)))+(d|(c;e)))$

Rule 4.3 is applied

P=a

New formula is:

$((a|((-b|e)))+(a|(b;e))|(d|((c|e)))+(d|(c;e)))$

Rule 4.3 is applied

P=c

New formula is:

$((a|((-b|e)))+(a|(b;e))|(d|((-c|e)))+(d|(c;e)))$

Rule 4.3 is applied

P=e

New formula is:

$((a|((-b|e)))+(a|(b;e))|(d|((-c|e)))+(d|(c;e)))$

Rule 4.3 is applied

P=d

New formula is:

$((a|((-b|e)))+(a|(b;e))|(d|((-c|e)))+(d|(c;e)))$

Number of applied rules in step 1 is 10

Step 2

Rule 1.1 is applied

P=a

Q=(-b)

R=(-e)

New formula is:

$((a|((-b))|(-e)))+(a|(b;e))|(d|((-c|(-e)))+(d|(c;e)))$

Rule 1.1 is applied

P=d

Q=(-c)

R=(-e)  
 New formula is:  
 $((a|(-b))|(-e))+((-a)|(b;e))|(((d|(-c))|(-e))+((-d)|(c;e)))$

Rule 2.1 is applied  
 P= $((a|(-b))|(-e))$   
 Q= $((-a)|(b;e))$   
 R= $((d|(-c))|(-e))+((-d)|(c;e))$   
 New formula is:  
 $((a|(-b))|(-e))|(((d|(-c))|(-e))+((-d)|(c;e)))+$   
 $((-a)|(b;e))|(((d|(-c))|(-e))+((-d)|(c;e)))$

Rule 2.2 is applied  
 P= $((a|(-b))|(-e))$   
 Q= $((d|(-c))|(-e))$   
 R= $((-d)|(c;e))$   
 New formula is:  
 $((a|(-b))|(-e))|(((d|(-c))|(-e))+((-a)|(-b))|((-d)|(c;e)))+$   
 $((-a)|(b;e))|(((d|(-c))|(-e))+((-d)|(c;e)))$

Rule 2.2 is applied  
 P= $((-a)|(b;e))$   
 Q= $((d|(-c))|(-e))$   
 R= $((-d)|(c;e))$   
 New formula is:  
 $((a|(-b))|(-e))|(((d|(-c))|(-e))+((-a)|(-b))|((-d)|(c;e)))+$   
 $((-a)|(b;e))|(((d|(-c))|(-e))+((-d)|(c;e)))$

Number of applied rules in step 2 is 5

Step 3

Rule 1.1 is applied  
 P= $((a|(-b))|(-e))|(((d|(-c))|(-e))+((-a)|(-b))|((-d)|(c;e)))$   
 Q= $((-a)|(b;e))|(((d|(-c))|(-e)))$   
 R= $((-a)|(b;e))|((-d)|(c;e))$   
 New formula is:  
 $((a|(-b))|(-e))|(((d|(-c))|(-e))+((-a)|(-b))|((-d)|(c;e)))+$   
 $((-a)|(b;e))|(((d|(-c))|(-e))+((-a)|(-b))|((-d)|(c;e)))$

Number of applied rules in step 3 is 1

Step 4

Rule 1.1 is applied  
 P= $((a|(-b))|(-e))$   
 Q= $(d|(-c))$   
 R= $(-e)$   
 New formula is:  
 $((a|(-b))|(-e))|((d|(-c))|(-e))+((-a)|(-b))|((-d)|(c;e)))+$   
 $((-a)|(b;e))|(((d|(-c))|(-e))+((-a)|(-b))|((-d)|(c;e)))$

Rule 1.1 is applied  
 P= $((a|(-b))|(-e))$   
 Q= $(-d)$

R=(c;e)

New formula is:

((((a|(-b))|(-e))|d|(-c))|(-e))+(((a|(-b))|(-e))|(-d))|(c;e))+  
(((a|(-b;e))|(d|(-c))|(-e)))+(((a|(-b;e))|(-d))|(c;e))

Rule 1.1 is applied

P=((-a)|(b;e))

Q=(d|(-c))

R=(-e)

New formula is:

((((a|(-b))|(-e))|d|(-c))|(-e))+(((a|(-b))|(-e))|(-d))|(c;e))+  
(((a|(-b;e))|(d|(-c))|(-e)))+(((a|(-b;e))|(-d))|(c;e))

Rule 1.1 is applied

P=((-a)|(b;e))

Q=(-d)

R=(c;e)

New formula is:

((((a|(-b))|(-e))|d|(-c))|(-e))+(((a|(-b))|(-e))|(-d))|(c;e))+  
(((a|(-b;e))|(d|(-c))|(-e)))+(((a|(-b;e))|(-d))|(c;e))

Number of applied rules in step 4 is 4

Step 5

Rule 1.1 is applied

P=((a|(-b))|(-e))

Q=d

R=(-c)

New formula is:

((((a|(-b))|(-e))|d|(-c))|(-e))+(((a|(-b))|(-e))|(-d))|(c;e))+  
(((a|(-b;e))|(d|(-c))|(-e)))+(((a|(-b;e))|(-d))|(c;e))

Rule 1.1 is applied

P=((-a)|(b;e))

Q=d

R=(-c)

New formula is:

((((a|(-b))|(-e))|d|(-c))|(-e))+(((a|(-b))|(-e))|(-d))|(c;e))+  
(((a|(-b;e))|d|(-c))|(-e)))+(((a|(-b;e))|(-d))|(c;e))

Number of applied rules in step 5 is 2

Step 6

Rule 7.6 is applied

P=(((a|(-b;e))|d)|(-c))

P'=(b;e)

Q=(-e)

New formula is:

((((a|(-b))|(-e))|d)|(-c))|(-e))+(((a|(-b))|(-e))|(-d))|(c;e))+  
(((a|(-b;e))|d)|(-c))|(\*e))+(((a|(-b;e))|(-d))|(c;e))

Rule 7.8 is applied

P=(((a|(-b))|(-e))|(-d))

$P' = (-e)$   
 $Q = (c; e)$   
 New formula is:  
 $(((((a|(-b))|(-e))|d)|(-c))|(-e)) + (((a|(-b))|(*e))|(-d))|c) +$   
 $(((((a|(-b))|(-e))|d)|(-c))|(*e)) + ((((-a)|b;e))|(-d))|c;e)$

Rule 7.9 is applied  
 $P = (((a|(-b))|(-e))|d)|(-c)$   
 $P' = (-e)$   
 $Q = (-e)$

New formula is:  
 $(((((a|(-b))|(-e))|d)|(-c)) + (((a|(-b))|(*e))|(-d))|c) +$   
 $(((((a|(-b))|(-e))|d)|(-c))|(*e)) + ((((-a)|b;e))|(-d))|c;e)$

Number of applied rules in step 6 is 3

Step 7

Rule 6.1 is applied

$P = ((a|(-b))|(*e))$

$P' = (*e)$

$Q = (-d)$

New formula is:

$(((((a|(-b))|(-e))|d)|(-c)) + (((a|(-b))|(*e))|(*d))|c) +$   
 $(((((a|(-b))|(-e))|d)|(-c))|(*e)) + ((((-a)|b;e))|(-d))|c;e)$

Rule 6.2 is applied

$P = ((a|(-b))|(*e))$

$P' = (-b)$

$Q = (*d)$

New formula is:

$(((((a|(-b))|(-e))|d)|(-c)) + (((a|(*b))|(*e))|(*d))|c) +$   
 $(((((a|(-b))|(-e))|d)|(-c))|(*e)) + ((((-a)|b;e))|(-d))|c;e)$

Rule 6.2 is applied

$P = ((((-a)|b)|d)|(-c))$

$P' = (-c)$

$Q = (*e)$

New formula is:

$(((((a|(-b))|(-e))|d)|(-c)) + (((a|(*b))|(*e))|(*d))|c) +$   
 $(((((a|(-b))|(-e))|d)|(-c))|(*e)) + ((((-a)|b;e))|(-d))|c;e)$

Number of applied rules in step 7 is 3

Step 8

Rule 6.2 is applied

$P = ((((-a)|b)|d)|(*c))$

$P' = (-a)$

$Q = (*e)$

New formula is:

$(((((a|(-b))|(-e))|d)|(-c)) + (((a|(*b))|(*e))|(*d))|c) +$   
 $(((((a|(-b))|(-e))|d)|(-c))|(*e)) + ((((-a)|b;e))|(-d))|c;e)$

Number of applied rules in step 8 is 1

Step 9

Number of applied rules in step 9 is 0

Canonical form is:

$(((((a|(-b))|(-e))|d)|(-c))+(((a|(*b))|(*e))|(*d))|c))+$   
 $(((((a|b)|d)|(*c))|(*e)))+(((a|b|e))|(-d))|c|e)$

Каноническая форма формулы  $(a||d||\bar{b}||\bar{c}||\bar{e}) \vee (a||c||\delta_b||\delta_d||\delta_e) \vee (b||d||\delta_a||\delta_c||\delta_e) \vee ((b;e)||c;e)||\bar{a}||\bar{d}$ .

# Литература

- [1] КОТОВ В.Е., ЧЕРКАСОВА Л.А. *Исчисления процессов. Системная Информатика* **2**, с. 6–38, Новосибирск, Наука, 1993.
- [2] КОТОВ В.Е. *Алгебра регулярных сетей. Препринт ВЦ СО РАН* **98**, Новосибирск, 1978.
- [3] КОТОВ В.Е. *Сети Петри*. Москва. Наука, 160 с., 1984.
- [4] ТАРАСЮК И.В. *Приведение формул алгебры  $AFP_2$  к каноническому виду. Проблемы Теоретического и Экспериментального Программирования*, с. 47–59, ИСИ СО РАН, Новосибирск, 1993.
- [5] ТАРАСЮК И.В. *Исследование сетевых эквивалентностей. Материалы 4-ой Международной Конференции по Прикладной Логике - 95 (AL'95)*, с. 74–75, ИГУ, Иркутск, 15–17 июня 1995.
- [6] ТАРАСЮК И.В. *Алгебра  $AFLP_2$ : исчисление помеченных недетерминированных параллельных процессов. Проблемы Спецификации и Верификации Параллельных Систем*, с. 22-49, ИСИ СО РАН, Новосибирск, 1995.
- [7] УСТИМЕНКО А.П., ЧЕРКАСОВА Л.А. *Алгебра параллельных недетерминированных процессов: денотационная и операционная семантики. Препринт ВЦ СО РАН* **903**, 38 с., Новосибирск, 1990.
- [8] AUTANT C., BELMESK Z., SCHNOEBELEN PH. *Strong bisimilarity on nets revisited. Research Report 847-I*, LIFIA-IMAG, Grenoble, France, 28 p., March 1991.
- [9] AUTANT C., BELMESK Z., SCHNOEBELEN PH. *Strong bisimilarity on nets revisited. Extended abstract. LNCS 506*, p. 295–312, June 1991.
- [10] ALUR R., COURCOUBETIS C., DILL D. *Model-checking for real-time systems. Proceedings of LICS'90*, p. 414–425, 1990.
- [11] ACETO L. *History preserving, causal and mixed-ordering equivalence for stable event structures. Technical Report HPL-PSC-91-28*, Hewlett-Packard Laboratories, Pisa Science Center, Pisa, 1991.
- [12] ALUR R., COURCOUBETIS C., HENZINGER T.A. *The observational power of clocks. LNCS 836*, p. 162–177, 1994.
- [13] ACETO L., HENNESSY M.C.B. *Towards action refinement in process algebras. Proceedings of 4<sup>th</sup> LICS*, IEEE Computer Society Press, Washington, p. 138–145, 1989.
- [14] ACETO L., HENNESSY M.C.B. *Adding action refinement to a finite process algebra. LNCS 510*, p. 506–519, 1991.
- [15] ACETO L., HENNESSY M.C.B. *Adding action refinement to a finite process algebra. Information and Computation* **115**, p. 179–247, 1994.
- [16] ANISIMOV N.A., KOVALENKO A.A. *Towards Petri net calculi based on synchronization via places. Proceedings of the IEEE Symposium on Parallel Algorithms and Architecture Synthesis - 95*, p. 264–270, Aizu, Japan, March 15–17, 1995.
- [17] ANISIMOV N.A., KOUTNY M. *On compositionality and Petri nets in protocol engineering. Protocol Specification, Testing and Verification XV*, Dembiński P., Średniawa eds. (Chapman & Hall), p. 71–86, 1995.

- [18] ANISIMOV N.A., KOVALENKO A.A. *Asynchronous composition of Petri nets via places*. Proceedings of the 2<sup>nd</sup> A.P. Ershov International Memorial Conference on Perspectives of System Informatics - 96 (PSI'96), p. 214–219, Institute of Informatics Systems, Novosibirsk, 1996.
- [19] ACETO L., DE NICOLA R., FANTECHI A. *Testing equivalences for event structures*. LNCS **280**, p. 1–20, 1987.
- [20] ANISIMOV N.A. *An algebra of regular macronets for formal specification of communication protocols*. Computers and Artificial Intelligence **10**, p. 541–560, 1991.
- [21] AUTANT C., PFISTER W., SCHNOEBELEN PH. *Place bisimulations for the reduction of labelled Petri nets with silent moves*. Proceedings of International Conference on Computing and Information, 1994.
- [22] AUTANT C., PFISTER W., SCHNOEBELEN PH. *Place bisimulations for the reduction of labelled Petri nets with silent moves*. Draft paper, 18 p., 1995.
- [23] AUTANT C., SCHNOEBELEN PH. *Place bisimulations in Petri nets*. LNCS **616**, p. 45–61, June 1992.
- [24] AUTANT C. *Petri nets for the semantics and the implementation of parallel processes*. Ph.D. thesis, Institut National Polytechnique de Grenoble, May 1993 (in French).
- [25] BAETEN J.C.M., BERGSTRA J.A. *Global renaming operators in concrete process algebra*. Report CS-R **8521**, Centrum voor Wiskunde en Informatica, 30 p., September 1985.
- [26] BAETEN J.C.M., BERGSTRA J.A. *Recursive process definitions with the state operator*. Report CS-R **8920**, Centrum voor Wiskunde en Informatica, 15 p., May 1989.
- [27] BAETEN J.C.M., BERGSTRA J.A. *Process algebra with zero object*. LNCS **458**, p. 83–98, 1990.
- [28] BAETEN J.C.M., BERGSTRA J.A. *Non interleaving process algebra*. LNCS **715**, p. 308–323, 1993.
- [29] BAETEN J.C.M., BERGSTRA J.A. *Process algebra with partial choice*. LNCS **836**, p. 465–480, 1994.
- [30] BAETEN J.C.M., BERGSTRA J.A., KLOP J.W. *Syntax and defining equations for an interrupt mechanism in process algebra*. Report CS-R **8503**, Centrum voor Wiskunde en Informatica, 45 p., February 1985.
- [31] BAETEN J.C.M., BERGSTRA J.A., KLOP J.W. *Ready trace semantics for concrete process algebra with priority operator*. Report CS-R **8517**, Centrum voor Wiskunde en Informatica, 21 p., August 1985.
- [32] BAETEN J.C.M., BERGSTRA J.A., KLOP J.W. *An operational semantics for process algebra*. Report CS-R **8522**, Centrum voor Wiskunde en Informatica, 30 p., September 1985.
- [33] BERNARDINELLO L., DE CINDIO F. *A survey of basic net models and modular net classes*. Design methods based on nets, *Esprit Basic Research Action* **3148**, June 1989–June 1990.
- [34] BOUDOL G., CASTELLANI I. *On the semantics of concurrency: partial orders and transition systems*. LNCS **249**, p. 123–137, 1987.
- [35] BOUDOL G., CASTELLANI I. *Concurrency and atomicity*. TCS **59**, p. 25–84, 1988.
- [36] BOUDOL G., CASTELLANI I. *Permutation of transitions: an event structure semantics for CCS and SCCS*. LNCS **354**, p. 411–427, 1989.
- [37] BOUDOL G., CASTELLANI I. *Three equivalent semantics for CCS*. LNCS **469**, p. 96–141, 1990.
- [38] BOUDOL G., CASTELLANI I. *Flow models of distributed computations: three equivalent semantics for CCS*. *Information and Computation* **114**, p. 247–314, 1994.
- [39] BOUDOL G., CASTELLANI I., HENNESSY M.C.B., KIEHN A. *A theory of processes with localities*. Report **13/91**, University of Sussex, 1991.
- [40] BOUDOL G., CASTELLANI I., HENNESSY M.C.B., KIEHN A. *Observing localities*. TCS **114**, p. 31–61, 1993.

- [41] BEST E., DEVILLERS R. *Sequential and concurrent behaviour in Petri net theory*. TCS **55**, p. 87–136, 1987.
- [42] BEST E., DEVILLERS R., ESPARZA J. *General refinement and recursion operations in the box calculus*. LNCS **665**, p. 130–140, 1993.
- [43] BEST E., DEVILLERS R., HALL J.G. *The box calculus: a new causal algebra with multi-label communication*. LNCS **609**, p. 21–69, 1992.
- [44] BEST E., DEVILLERS R., KOUTNY M. *Petri net algebra*. Draft paper, 181 p., April 1997.
- [45] BEST E., DEVILLERS R., KIEHN A., POMELLO L. *Fully concurrent bisimulation*. Technical Report LIT-**202**, Univ. Bruxelles, 1989.
- [46] BEST E., DEVILLERS R., KIEHN A., POMELLO L. *Concurrent bisimulations in Petri nets*. Acta Informatica **28**, p. 231–264, 1991.
- [47] BEDNARCZYK M.A. *Heredity history preserving bisimulations*. Draft paper, 1991.
- [48] BEST E. *Concurrent behaviour: sequences, processes and axioms*. LNCS **197**, p. 221–245, 1985.
- [49] BEST E. *Partial order semantics of concurrent programs*. (Abstract of invited lecture at CONCUR'90.) LNCS **458**, p. 1, 1990.
- [50] BEST E. *Semantics of sequential and parallel programs*. 347 p., Prentice Hall International, 1996.
- [51] BEST E. *Petri nets, process algebras and programming languages*. Draft tutorial, 140 p., 1996.
- [52] VAN BENTHEM J., VAN EIJCK J., STEBLETSOVA V. *Modal logic, transition systems and processes*. Report CS-R **9321**, Centrum voor Wiskunde en Informatica, 1993.
- [53] BEST E., FERNANDEZ C. *Nonsequential processes*. EATCS Monographs **13**, 112 p., 1988.
- [54] BEST E., FLEISCHHACK H., FRĄCZAK W., HOPKINS R.P., KLAUDEL H., PELZ E. *A high level Petri net semantics of  $B(PN)^2$* . Draft paper, March 28, 1994.
- [55] BEST E., FLEISCHHACK H., FRĄCZAK W., HOPKINS R.P., KLAUDEL H., PELZ E. *An M-net semantics of  $B(PN)^2$* . Proceedings of STRICT'95, 1995.
- [56] BEST E., FLEISCHHACK H., FRĄCZAK W., HOPKINS R.P., KLAUDEL H., PELZ E. *A class of composable high level Petri nets*. LNCS **935**, p. 103–120, 1995.
- [57] BEST E., FLEISCHHACK H., FRĄCZAK W., HOPKINS R.P., KLAUDEL H., PELZ E. *An algebra of high level Petri nets, with an application to the semantics of concurrent program languages*. Draft paper, 34 p., November 1995.
- [58] BEST E., FLEISCHHACK H. *PEP: programming environment based on Petri nets*. Hildesheimer Informatik-Bericht **14/95**, 100 p., May 1995.
- [59] BEST E., GRAHLMANN B. *Programming environment based on Petri nets. Documentation and user guide version 1.4*. Universität Hildesheim, Institut für Informatik, Hildesheim, Germany, 78 p., November 1995.
- [60] BAETEN J.C.M, VAN GLABBEEK R.J. *Another look at abstraction in process algebra*. LNCS **267**, p. 84–94, 1987.
- [61] BAETEN J.C.M, VAN GLABBEEK R.J. *Another look at abstraction in process algebra*. Report CS-R **8701**, Centrum voor Wiskunde en Informatica, 32 p., January 1987.
- [62] BAETEN J.C.M, VAN GLABBEEK R.J. *Merge and termination in process algebra*. Report CS-R **8716**, Centrum voor Wiskunde en Informatica, 26 p., April 1987.
- [63] BAETEN J.C.M, VAN GLABBEEK R.J. *Abstraction and empty process in process algebra*. Report CS-R **8721**, Centrum voor Wiskunde en Informatica, 24 p., April 1987.



- [64] BRAUER W., GOLD R., VOGLER W. *A survey of behaviour and equivalence preserving refinements of Petri nets*. LNCS **483**, p. 1–46, 1990.
- [65] BROOKES S.D., HOARE C.A.R., ROSCOE A.W. *A theory of communicating sequential processes*. *Journal of the ACM* **31**(3), p. 560–599, July 1984.
- [66] BERGSTRA J.A., KLOP J.W. *Process algebra for synchronous communication*. *Information and Control* **60**, p. 109–137, 1984.
- [67] BERGSTRA J.A., KLOP J.W. *The algebra of recursively defined processes and the algebra of regular processes*. LNCS **172**, p. 82–94, 1984.
- [68] BERGSTRA J.A., KLOP J.W. *Algebra of communicating processes with abstraction*. *TCS* **37**, p. 77–121, 1985.
- [69] BERGSTRA J.A., KLOP J.W. *ACP<sub>τ</sub>: a universal axiom system for process specification*. Report CS-R **8725**, Centrum voor Wiskunde en Informatica, 17 p., May 1987.
- [70] BERGSTRA J.A., KLOP J.W. *Process theory based on bisimulation semantics*. LNCS **354**, p. 50–122, 1989.
- [71] BERGSTRA J.A., KLOP J.W. *BMACP*. J.W. de Bakker, 25 Jaar Semantiek, Liber Amicorum, p. 91–111, 1989.
- [72] BEST E., KOUTNY M. *A refined view of the box algebra. Solving recursive net equations*. *Hildesheimer Informatik-Bericht* **9/95**, 16+15 p., April 1995.
- [73] BEST E., KOUTNY M. *Operational and denotational semantics for the box algebra*. *Hildesheimer Informatik-Bericht* **33/95**, 53 p., October 1995.
- [74] BEST E., KOUTNY M. *A refined view of the box algebra*. LNCS **935**, p. 1–20, 1995.
- [75] BEST E., KOUTNY M. *Solving recursive net equations*. LNCS **944**, p. 605–623, 1995.
- [76] BERGSTRA J.A., KLOP J.W., TUCKER J.V. *Process algebra with synchronism communication mechanisms*. LNCS **197**, p. 180–196, 1985.
- [77] BEST E., LAVROV A. *Operational semantics of place synchronization in the box calculus*. Draft paper, 5 p., October 1995.
- [78] BEST E., LAVROV A. *Syntactical representation of place synchronization in the box calculus*. Draft paper, 10 p., October 1995.
- [79] BEST E., LINDE-GOERS H.G. *Compositional process semantics of Petri boxes*. Proceedings of Mathematical Foundations of Programming Semantics, New Orleans, LNCS, 1993.
- [80] BROOKES S.D., ROSCOE A.W. *An improved failures model for communicating processes*. LNCS **197**, p. 281–305, 1985.
- [81] BROOKES S.D. *On the relationship of CCS and CSP*. LNCS **154**, p. 83–96, 1983.
- [82] BROOKES S.D. *A semantically based proof system for partial correctness and deadlock in CSP*. Proceedings of 1<sup>st</sup> LICS, p. 58–65, 1986.
- [83] BAETEN J.C.M., VAANDRAGER F.W. *An algebra for process creation*. Report CS-R **8907**, Centrum voor Wiskunde en Informatica, 24 p., March 1989.
- [84] BAETEN J.C.M., VAANDRAGER F.W. *An algebra for process creation*. J.W. de Bakker, 25 Jaar Semantiek, Liber Amicorum, p. 57–81, 1989.
- [85] BAETEN J.C.M., VAANDRAGER F.W. *An algebra for process creation*. *Acta Informatica* **29**(4), p. 303–334, 1992.
- [86] DE BAKKER J.W., WARMERDAM J.H.A. *Metric pomset semantics for a concurrent language with restriction*. LNCS **469**, p. 21–49, 1990.

- [87] DE BAKKER J.W., ZUCKER J.I. *Processes and denotational semantics of concurrency*. *Information and Control* **54**, p. 70–120, 1982.
- [88] CASTELLANI I. *Bisimulations for concurrency*. Ph.D. thesis, University of Edinburgh, 1988.
- [89] CASTELLANI I. *Observing distribution in processes*. *LNCS* **711**, p. 321–331, 1993.
- [90] CASTELLANI I. *Observing distribution in processes: static and dynamic localities*. *INRIA Research Report* **2276**, 45 p., May 1994.
- [91] CLARKE E.M., EMERSON E.A., SISTLA A.P. *Automatic verification of finite-state concurrent systems using temporal logic specifications*. *ACM Transactions on Programming Languages and Systems* **8(2)**, p. 244–263, 1986.
- [92] ČERĀNS K. *Decidability of bisimulation equivalences for parallel timer processes*. *LNCS* **663**, p. 302–315, 1993.
- [93] ČERĀNS K. *CRT: a calculus of timed refinement*. *LNCS* **962**, p. 516–530, 1995.
- [94] CZAJA I., VAN GLABBEK R.J., GOLTZ U. *Interleaving semantics and action refinement with atomic choice*. *LNCS* **609**, p. 89–107, 1992.
- [95] CHERIEF F., LAROUSSINIE F., PINCHINAT S. *Modal logics with past for true concurrency*. *Proceedings of CONCUR'92*, February 1992.
- [96] CHERKASOVA L.A. *On models and algebras for concurrent processes*. *LNCS* **324**, p. 27–43, 1988.
- [97] CHERKASOVA L.A. *Completeness of axiom set and inference rules for algebra of generalized concurrent processes*. *Papers presented at the International Conference in Computer Logic, COLOG-88*, Institute of Cybernetics of the Academy of Sciences of the Estonian SSR, Part II, p. 209–223, December 1988.
- [98] CHERKASOVA L.A. *Posets with non-actions: a model for concurrent nondeterministic processes*. *Arbeitspapiere der GMD* **403**, 68 p., July 1989.
- [99] CHERKASOVA L.A. *Algebra  $AFP_2$  for concurrent nondeterministic processes: fully abstract model and complete axiomatization*. *Reihe Informatik* **72**, 28 p., July 1990.
- [100] CHERKASOVA L.A. *A fully abstract model for concurrent nondeterministic processes based on posets with non-actions*. *Report CS-R* **9031**, Centrum voor Wiskunde en Informatica, Amsterdam, 42 p., July 1990.
- [101] CASTELLANI I., HENNESSY M.C.B. *Distributed bisimulations*. *Journal of the ACM* **36(4)**, p. 887–911, 1989.
- [102] CHERIEF F. *Back and forth bisimulations on prime event structures*. *LNCS* **605**, p. 843–858, June 1992.
- [103] CHERIEF F. *Contributions à la sémantique du parallélisme: bisimulations pour le raffinement et le vrai parallélisme*. Ph.D. thesis, Institut National Polytechnique de Grenoble, France, October 1992 (in French).
- [104] CHERIEF F. *Investigations of back and forth bisimulations on prime event structures*. *Computers and Artificial Intelligence* **11(5)**, p. 481–496, 1992.
- [105] CHERKASOVA L.A., FILURIN A.S. *Concurrent processes with synchronization: net and algebraic approach*. *LNCS* **363**, p. 63–80, 1989.
- [106] CHERKASOVA L.A., KOTOV V.E. *Structured nets*. *LNCS* **118**, p. 242–251, 1981.
- [107] CHERKASOVA L.A., KOTOV V.E. *Concurrent nondeterministic processes: adequacy of structure and behaviour*. *LNCS* **379**, p. 67–87, 1989.
- [108] CHERKASOVA L.A., KOTOV V.E. *Descriptive and analytical process algebras*. *LNCS* **424**, p. 77–104, 1989.
- [109] CHERKASOVA L.A., KOTOV V.E. *An algebra for concurrent nondeterministic processes*. *TCS* **90**, p. 151–170, 1991.

- [110] CHRISTENSEN S., HIRSHFELD Y., MOLLER F. *Bisimulation equivalence is decidable for basic parallel processes*. *LNCS* **715**, p. 143–157, 1993.
- [111] CHRISTENSEN S., HIRSHFELD Y., MOLLER F. *Decomposability, decidability and axiomatisability for bisimulation equivalence on basic parallel processes*. *Proceedings of 8<sup>th</sup> LICS*, p. 386–396, 1993.
- [112] CHERIEF F., SCHNOEBELEN PH.  *$\tau$ -bisimulations and full abstraction for refinement of actions*. *Information Processing Letters* **40**, p. 219–222, November 1991.
- [113] CASTELLANO L., DE MICHELIS G., POMELLO L. *Concurrency vs interleaving: an instructive example*. *Bulletin of the EATCS* **31**, p. 12–15, 1987.
- [114] CORRADINI F., DE NICOLA R. *Locality based semantics for process algebras*. *Acta Informatica* **34**(4), p. 294–324, 1997.
- [115] CLEAVELAND F., SMOLKA S.A. *Strategic directions in computer research: concurrency working group research*. *Bulletin of the EATCS* **60**, p. 97–122, October 1996.
- [116] DARONDEAU PH., DEGANO P. *Causal trees*. *LNCS* **372**, p. 234–248, 1989.
- [117] DARONDEAU PH., DEGANO P. *Event structures, causal trees and refinements*. *LNCS* **452**, p. 239–245, 1990.
- [118] DARONDEAU PH., DEGANO P. *Causal trees, interleaving + causality*. *LNCS* **469**, p. 239–255, 1990.
- [119] DARONDEAU PH., DEGANO P. *Refinement of actions in event structures and causal trees*. *TCS* **118**, p. 21–48, 1993.
- [120] DEVILLERS R. *Maximality preserving bisimulation*. *Technical Report LIT-214*, Lab. Informatique Theorique, Universite Libre de Bruxelles, March 1990.
- [121] DEVILLERS R. *Maximality preserving bisimulation*. *TCS* **102**, p. 165–184, 1992.
- [122] DEVILLERS R. *Maximality preservation and the ST-idea for action refinements*. *LNCS* **609**, p. 108–151, 1992.
- [123] DEVILLERS R. *Construction of S-invariants and S-components for refined Petri boxes*. *LNCS* **691**, p. 242–261, 1993.
- [124] DEVILLERS R. *S-invariant analysis of general recursive Petri boxes*. *Acta Informatica* **32**(4), p. 313–345, 1995.
- [125] DEVILLERS R. *Petri boxes and finite processes*. *LNCS* **1119**, p. 465–480, 1996.
- [126] DEGANO P., DE NICOLA R., MONTANARI U. *CCS is an (augmented) contact free C/E system*. *LNCS* **280**, p. 144–165, 1987.
- [127] DEGANO P., DE NICOLA R., MONTANARI U. *Observational equivalences for concurrency models*. *Proceedings 3<sup>rd</sup> IFIP WG 2.2 Working Conference*, p. 105–132, North-Holland, Amsterdam, 1987.
- [128] DEGANO P., DE NICOLA R., MONTANARI U. *A distributed operational semantics for CCS based on condition / event systems*. *Acta Informatica* **26**, p. 59–91, 1988.
- [129] DEGANO P., DE NICOLA R., MONTANARI U. *Partial orderings descriptions and observations of nondeterministic concurrent processes*. *LNCS* **354**, p. 438–466, 1989.
- [130] DEGANO P., DE NICOLA R., MONTANARI U. *A partial ordering semantics for CCS*. *TCS* **75**, p. 223–255, 1990.
- [131] DEGANO P., DE NICOLA R., MONTANARI U. *Universal axioms for bisimulations*. *TCS* **114**, p. 63–91, 1993.
- [132] ESPARZA J., BRUNS G. *Trapping mutual execution in the box calculus*. *TCS* **153**, p. 95–128, 1996.
- [133] ENGELFRIET J. *Determinacy  $\rightarrow$  (observation equivalence = trace equivalence)*. *TCS* **36**, p. 21–25, 1985.

- [134] ENGELFRIET J. *Branching processes of Petri nets*. *Acta Informatica* **28**(6), p. 575–591, 1991.
- [135] FOKKINK W., KLUSENER S. *An effective axiomatization for real time ACP*. *Information and Computation* **122**, p. 286–299, 1995.
- [136] FELDER M., MORZENTI A. *A temporal logic approach to implementation and refinement in timed Petri nets*. *LNCS ?*, p. 365–381, 1994.
- [137] VAN GLABBEEK R.J., GOLTZ U. *Equivalence notions for concurrent systems and refinement of actions*. *Arbeitspapiere der GMD* **366**, 1989.
- [138] VAN GLABBEEK R.J., GOLTZ U. *Equivalence notions for concurrent systems and refinement of actions*. *LNCS* **379**, p. 237–248, 1989.
- [139] VAN GLABBEEK R.J., GOLTZ U. *Partial order semantics for the refinement of actions: neither necessary nor always sufficient but appropriate when used with care*. Note CS-N **8901**, 10 p., Centrum voor Wiskunde en Informatica, Amsterdam, July 1989.
- [140] VAN GLABBEEK R.J., GOLTZ U. *Refinement of actions in causality based models*. *Arbeitspapiere der GMD* **428**, 1990.
- [141] VAN GLABBEEK R.J., GOLTZ U. *Refinement of actions in causality based models*. *LNCS* **430**, p. 267–300, 1990.
- [142] VAN GLABBEEK R.J., GOLTZ U. *Equivalence notions and refinement of actions for flow event structures*. *DEMON deliverables, Esprit Basic Research Action* **3148**, 1990.
- [143] VAN GLABBEEK R.J., GOLTZ U. *Equivalences and refinement*. *LNCS* **469**, p. 309–333, 1990.
- [144] GOLTZ U., GORRIERI R., RENSINK A. *On syntactic and semantic action refinement*. *Hildesheimer Informatik-Bericht* **17/92**, Institut für Informatik, Universität Hildesheim, Hildesheim, Germany, 1992.
- [145] GOLTZ U., GORRIERI R., RENSINK A. *On syntactic and semantic action refinement*. *LNCS* **789**, p. 385–404, 1994.
- [146] GOLTZ U., KUIPER R., PENCZEK W. *Propositional temporal logic and equivalences*. *LNCS* **630**, p. 222–236, August 1992.
- [147] VAN GLABBEEK R.J. *Bounded nondeterminism and the approximation induction principle in process algebra*. *LNCS* **247**, p. 336–347, 1987.
- [148] VAN GLABBEEK R.J. *The refinement theorem for ST-bisimulation semantics*. Report CS-R **9002**, Centrum voor Wiskunde en Informatica, Amsterdam, 1990.
- [149] VAN GLABBEEK R.J. *The refinement theorem for ST-bisimulation semantics*. M.Broy and C.B.Jones, editors. *Proceedings IFIP Working Conference of Programming Concepts and Methods*, Sea of Galilee, Israel, 1990.
- [150] VAN GLABBEEK R.J. *The linear time – branching time spectrum. Extended abstract*. *LNCS* **458**, p. 278–297, 1990.
- [151] VAN GLABBEEK R.J. *Comparative concurrency semantics and refinement of actions*. Ph.D. Thesis, Free University, Amsterdam, 1990.
- [152] VAN GLABBEEK R.J. *The linear time – branching time spectrum II: the semantics of sequential systems with silent moves. Extended abstract*. *LNCS* **715**, p. 66–81, 1993.
- [153] VAN GLABBEEK R.J. *What is branching time semantics and why to use it ?* *Bulletin of the EATCS* **53**, p. 191–198, June 1994.
- [154] GOLTZ U., MYCROFT A. *On the relationship of CCS and Petri nets*. *LNCS* **172**, p. 196–208, 1984.
- [155] GOLTZ U. *On representing CCS programs by finite Petri nets*. *LNCS* **324**, p. 339–350, 1988.
- [156] GOLTZ U. *CCS and Petri nets*. *LNCS* **469**, p. 334–357, 1990.

- [157] GOLTZ U., REISIG W. *Processes of place-transition nets*. LNCS **154**, p. 164–177, 1983.
- [158] GOLTZ U., REISIG W. *The non-sequential behaviour of Petri nets*. *Information and Control* **57**, p. 125–147, 1983.
- [159] GOLTZ U., REISIG W. *CSP-programs as nets with individual tokens*. LNCS **188**, p. 169–196, 1985.
- [160] GRABOWSKI J. *On partial languages*. *Fundamenta Informaticae* **IV(2)**, p. 428–498, 1981.
- [161] VAN GLABBEEK R.J., VAANDRAGER F.W. *Petri net models for algebraic theories of concurrency*. LNCS **259**, p. 224–242, 1987.
- [162] VAN GLABBEEK R.J., WEIJLAND W.P. *Branching time and abstraction in bisimulation semantics (extended abstract)*. Report CS-R **8911**, 14 p., Centrum voor Wiskunde en Informatica, Amsterdam, April 1989.
- [163] VAN GLABBEEK R.J., WEIJLAND W.P. *Branching time and abstraction in bisimulation semantics (extended abstract)*. G.X. Ritter, editor: *Information Processing 89, Proceedings of 11<sup>th</sup> World Computer Congress*, San Francisco, p. 613–618, North-Holland, August 1989.
- [164] VAN GLABBEEK R.J., WEIJLAND W.P. *Refinement in branching time semantics*. *J.W. de Bakker, 25 Jaar Semantiek, Liber Amicorum*, p. 247–252, 1989.
- [165] VAN GLABBEEK R.J., WEIJLAND W.P. *Branching time and abstraction in bisimulation semantics*. Report CS-R **9120**, Centrum voor Wiskunde en Informatica, Amsterdam, 1991.
- [166] VAN GLABBEEK R.J., WEIJLAND W.P. *Branching time and abstraction in bisimulation semantics*. *Journal of the ACM* **43(3)**, p. 555–600, 1996.
- [167] GOLTZ U., WEHRHEIM H. *Causal testing*. LNCS **1113**, p. 394–406, 1996.
- [168] HENNESSY M.C.B. *Algebraic theory of processes*. Cambridge, Massachusetts, 1988.
- [169] HENNESSY M.C.B. *Observing processes*. LNCS **354**, p. 173–200, 1989.
- [170] HOOGERS P.W., KLEIJN H.C.M., THIAGARAJAN P.S. *An event structure semantics for general Petri nets*. TCS **153**, p. 129–170, 1996.
- [171] HENNESSY M., MILNER R.A.J. *Algebraic laws for nondeterminism and concurrency*. *Journal of the ACM* **32(1)**, p. 137–161, January 1985.
- [172] HOARE C.A.R. *Communicating sequential processes*. *Communications ACM* **21(8)**, p. 666–677, 1978.
- [173] HOARE C.A.R. *Communicating sequential processes, on the construction of programs*. (McKeag R.M., Macnaghten A.M. eds.) Cambridge University Press, p. 229–254, 1980.
- [174] HOARE C.A.R. *A model for communicating sequential processes*. Technical Monograph, Prg **22**, Computing Laboratory, University of Oxford, 1981.
- [175] HOARE C.A.R. *Communicating sequential processes*. Prentice-Hall, London, 1985.
- [176] HOOGERS P.W. *Behavioural aspects of Petri nets*. Ph.D. thesis, 1994.
- [177] JOYAL A., NIELSEN M., WINSKEL G. *Bisimulation and open maps*. *Proceedings of 8<sup>th</sup> LICS*, p. 418–427, 1993.
- [178] JOYAL A., NIELSEN M., WINSKEL G. *Bisimulation and open maps*. BRICS RS-94-7, Computer Science Department, Aarhus University, 42 p., May 1994.
- [179] KOUTNY M., BEST E. *Operational and denotational semantics for the box algebra*. Draft paper, 70 p., March 1997.
- [180] KOTOV V.E., CHERKASOVA L.A. *On structural properties of generalized processes*. LNCS **188**, p. 288–306, 1985.

- [181] KOTOV V.E., CHERKASOVA L.A. *From nets to logic and back in the specification of process. Concurrency and Nets*, p. 253–268, Springer-Verlag, 1987.
- [182] KOTOV V.E., CHERKASOVA L.A. *On generalized process logic. LNCS 278*, p. 246–250, 1987.
- [183] KOUTNY M., ESPARZA J., BEST E. *Operational semantics for the Petri box calculus. Hildesheimer Informatik-Bericht 13/93*, October 1993.
- [184] KOUTNY M., ESPARZA J., BEST E. *Operational semantics for the Petri box calculus. LNCS 836*, p. 210–225, 1994.
- [185] KELLER R.M. *Formal verification of parallel programs. Communications of the ACM 19(7)*, p. 371–384, 1976.
- [186] KIEHN A. *On the interrelation between synchronized and non-synchronized behaviour of Petri nets. Journal Inform. Process. Cybernet. EIK 24(1/2)*, p. 3–18, 1988.
- [187] KIEHN A. *Proof systems for cause based equivalences. LNCS 711*, p. 547–556, 1993.
- [188] KIEHN A. *Comparing locality and causality based equivalences. Acta Informatica 31(8)*, p. 697–718, 1994.
- [189] KOTOV V.E. *An algebra for parallelism based on Petri nets. LNCS 64*, p. 39–55, 1978.
- [190] KOUTNY M. *Partial order semantics of box expressions. LNCS 815*, p. 318–337, 1994.
- [191] LAROUSSINIE F. *Temporal logic for specification and verification of reactive systems. Ph.D. thesis, Institut National Polytechnique de Grenoble, France, 151 p., 1994 (in French).*
- [192] LILIUS J. *OB(PN)<sup>2</sup>: an object oriented language with a truly concurrent semantics. Version 1.2. Draft paper, 22 p., January 19, 1996.*
- [193] LAROUSSINIE F., SCHNOEBELEN PH. *A hierarchy of temporal logics with past. TCS 148*, p. 303–324, 1995.
- [194] MAZURKIEWICZ A. *Trace theory. LNCS 255*, p. 279–324, 1987.
- [195] MAZURKIEWICZ A. *Basic notions of trace theory. LNCS 354*, p. 285–363, 1988.
- [196] MERLIN P., FARBER D.J. *Recoverability of communication protocols. IEEE Transactions on Communication Protocols COM-24(9)*, 1976.
- [197] MILNER R.A.J. *Algebras for communicating systems. Technical Report CSR-25-78*, Computer Science Department, University of Edinburgh, Edinburgh, England, 18 p., 1978.
- [198] MILNER R.A.J. *A calculus of communicating systems. LNCS 92*, p. 172–180, 1980.
- [199] MILNER R.A.J. *A modal characterization of observable machine behaviour. LNCS 112*, p. 25–34, 1981.
- [200] MILNER R.A.J. *A complete inference system for a class of regular behaviours. Technical Report CSR-111-82*, Computer Science Department, University of Edinburgh, Edinburgh, England, 1982.
- [201] MILNER R.A.J. *A finite delay operator in synchronous CCS. Technical Report CSR-116-82*, Computer Science Department, University of Edinburgh, Edinburgh, England, 1982.
- [202] MILNER R.A.J. *Calculi for synchrony and asynchrony. TCS 25*, p. 267–310, 1983.
- [203] MILNER R.A.J. *Lectures on a calculus of communicating systems. LNCS 197*, p. 197–221, 1984.
- [204] MILNER R.A.J. *Communication and concurrency*. Prentice-Hall International, New York, 1989.
- [205] MESEGUER J., MONTANARI U., SASSONE V. *On the semantics of place / transition Petri nets. Pisa University Technical Report TR-27/92*, 1992.
- [206] MESEGUER J., MONTANARI U., SASSONE V. *On the semantics of Petri nets. LNCS 630*, p. 286–301, 1992.

- [207] MESEGUER J., MONTANARI U., SASSONE V. *On the model of computation of place / transition Petri nets*. LNCS **815**, p. 16–38, 1994.
- [208] MESEGUER J., MONTANARI U., SASSONE V. *Process versus unfolding semantics for place / transition Petri nets*. TCS **153**, p. 171–210, 1996.
- [209] MUKUND M., NIELSEN M. *CCS, locations and asynchronous transition systems*. LNCS **652**, p. 328–341, 1993.
- [210] MELZER S., RÖMER S., ESPARZA J. *Verification using PEP*. LNCS **1101**, p. 591–594, 1996.
- [211] NIELSEN M., CLAUSEN C. *Bisimulation for models in concurrency*. LNCS **836**, p. 385–400, 1994.
- [212] NIELSEN M., ENGBERG U., LARSEN K.S. *Fully abstract models for a process language with refinement*. LNCS **354**, p. 523–548, 1989.
- [213] NAKAGAWA M., KUMAGAI S., MIYAMOTO T., LEE D.-I. S. *Equivalent net reduction for firing sequence*. IEICE Transactions of Fundamentals of Electronics, Communications and Computer Science **E 78-A(11)**, p. 1447–1457, November 1995.
- [214] DE NICOLA R., MONTANARI U., VAANDRAGER F.W. *Back and forth bisimulations*. LNCS **458**, p. 152–165, 1990.
- [215] NIELSEN M., PLOTKIN G., WINSKEL G. *Petri nets, event structures and domains*. TCS **13**, p. 85–108, 1981.
- [216] NIELSEN M., ROZENBERG G., THIAGARAJAN P.S. *Behavioural notions for elementary net systems*. Distributed Computing **4**, p. 45–57, 1990.
- [217] NIELSEN M., ROZENBERG G., THIAGARAJAN P.S. *Elementary transition system and refinement*. Acta Informatica **29(6/7)**, p. 555–578, 1992.
- [218] NIELSEN M., THIAGARAJAN P.S. *Degrees of non-determinism and concurrency: A Petri net view*. LNCS **181**, p. 89–117, December 1984.
- [219] NIELSEN M., WINSKEL G. *Petri nets and bisimulations*. BRICS RS-95-4, Computer Science Department, Aarhus University, 36 p., January 1995.
- [220] OLDEROG E.-R. *Process theory: semantics, specification and verification*. LNCS **224**, p. 442–509, 1986.
- [221] OLDEROG E.-R. *TCSP: theory of communicating sequential processes*. LNCS **255**, p. 441–465, 1987.
- [222] OLDEROG E.-R. *Operational Petri net semantics for CCSP*. LNCS **266**, p. 196–223, 1987.
- [223] OLDEROG E.-R. *Nets, terms and formulas, three views of concurrent processes and their relationship*. Habilitationsschrift, Christian-Albrechts Universität, Kiel, July 1989.
- [224] OLDEROG E.-R. *Strong bisimilarity on nets: a new concept for comparing net semantics*. LNCS **354**, p. 549–573, 1989.
- [225] OLDEROG E.-R. *Nets, terms and formulas*. Cambridge Tracts in Theoretical Computer Science **23**, Cambridge University Press, 1991.
- [226] PETERSON J.L. *Petri net theory and modelling of systems*. Prentice-Hall, 1981. (Имеется перевод на русский язык: ПИТЕРСОН ДЖ. *Теория сетей Петри и моделирование систем*. Москва, Мир, 1984.)
- [227] PARK D.M.R. *Concurrency and automata on infinite sequences*. Computer Science Department, University of Warwick, Great Britain, 1981.
- [228] PARK D.M.R. *Concurrency and automata on infinite sequences*. LNCS **104**, p. 167–183, March 1981.
- [229] PENSZEK W. *A temporal logic for event structures*. Fundamenta Informaticae **11(3)**, p. 297–326, 1988.
- [230] PENSZEK W. *A concurrent branching time temporal logic*. LNCS **11(3)**, p. 297–326, 1988.

- [231] PENSZEK W. *Branching time and partial order in temporal logic*. Technical Report **UMCS-91-??**, Department of Computer Science, University of Manchester, 1991.
- [232] PETRI C.A. *Kommunikation mit Automaten*. Ph.D. thesis, Universität Bonn, Schriften des Instituts für Instrumentelle Mathematik, 1962 (in German).
- [233] PETRI C.A. *Non-sequential processes*. GMD-InformationSystemForschung, Interner Bericht **77-05**, 1977.
- [234] PETRI C.A. *Introduction to general net theory*. LNCS **84**, p. 1–20, 1980.
- [235] PETRI C.A. *Nets, time and space*. TCS **153**, p. 3–48, 1996.
- [236] PFISTER W. *Simplification sémantique des réseaux de Petri par la bisimulation de places*. Technical Report of DEA, University of Grenoble, Grenoble, France, June 1992 (in French).
- [237] PINCHINAT S. *Bisimulations for the semantics of reactive systems*. Ph.D. thesis, Institut National Polytechnique de Grenoble, January 1993 (in French).
- [238] PLOTKIN G.D. *A structural approach to operational semantics*. Technical Report **DAIMI FN-19**, Computer Science Department, Aarhus University, 1981.
- [239] POMELO L. *Some equivalence notions for concurrent systems: an overview*. GMD Report **103**, 1984.
- [240] POMELO L. *Some equivalence notions for concurrent systems. An overview*. LNCS **222**, p. 381–400, 1986.
- [241] POMELO L. *Refinement of concurrent systems based on local state transformations*. LNCS **430**, p. 641–668, 1990.
- [242] PRATT V.R. *On the composition of processes*. Proceedings of 9<sup>th</sup> POPL, p. 213–223, 1982.
- [243] PRATT V.R. *The pomset model of parallel processes: unifying the temporal and the spatial*. LNCS **197**, p. 180–196, 1986.
- [244] PRATT V.R. *Modelling concurrency with partial orders*. International Journal of Parallel Programming **15**(1), p. 33–71, 1986.
- [245] POMELO L., ROZENBERG G., SIMONE C. *A survey of equivalence notions for net based systems*. LNCS **609**, p. 410–472, 1992.
- [246] REISIG W. *Partial order semantics versus interleaving semantics for CSP-like languages and its impact on fairness*. LNCS **172**, p. 403–413, 1984.
- [247] REISIG W. *Petri nets. An introduction*. EATCS Monographs on Theoretical Computer Science **4**, Springer-Verlag, 1985.
- [248] REM M. *Trace theory and systolic computations*. LNCS **258**, p. 14–33, 1987.
- [249] RENSINK A. *Models and methods for action refinement*. Ph.D. thesis, University of Twente, August 1993.
- [250] ROZENBERG G., THIAGARAJAN P.S. *Petri nets: basic notions, structure, behaviour*. LNCS **224**, p. 585–668, 1986.
- [251] RABINOVITCH A., TRAKHTENBROT B.A. *Behaviour structures and nets*. Fundamenta Informaticae **XI**, p. 357–404, 1988.
- [252] SASSONE V. *On the semantics of Petri nets: processes, unfoldings and infinite computations*. Ph.D. thesis, 351 p., 1994.
- [253] SLOAN R.H., BUY U. *Reduction rules for time Petri nets*. Acta Informatica **33**(7), p. 687–706, 1996.
- [254] SASSONE V., NIELSEN M., WINSKEL G. *Deterministic behavioural models for concurrency*. LNCS **711**, p. 682–692, 1993.
- [255] SASSONE V., NIELSEN M., WINSKEL G. *A classification of models for concurrency*. LNCS **715**, p. 82–96, 1993.



- [256] TARASYUK I.V. *Equivalences on Petri nets. Specification, Verification and Net Models of Concurrent Systems*, p. 35–57, Institute of Informatics Systems, Novosibirsk, 1994.
- [257] TARASYUK I.V. *An investigation of equivalence notions on some subclasses of Petri nets. Bulletin of the Novosibirsk Computing Center, Series Computer Science* **3**, p. 89–100, Computing Center, Novosibirsk, 1995.
- [258] TARASYUK I.V. *Equivalence notions for design of concurrent systems using Petri nets. Hildesheimer Informatik-Bericht* **4/96**, part 1, 19 p., Institut für Informatik, Universität Hildesheim, Hildesheim, Germany, January 1996.
- [259] TARASYUK I.V. *Algebra AFLP<sub>2</sub>: a calculus of labelled nondeterministic processes. Hildesheimer Informatik-Bericht* **4/96**, part 2, 18 p., Institut für Informatik, Universität Hildesheim, Hildesheim, Germany, January 1996.
- [260] TARASYUK I.V. *Petri net equivalences for design of concurrent systems. Proceedings of 5<sup>th</sup> Workshop on Concurrency, Specification and Programming - 96 (CSP'96), September 25–27, 1996, Informatik-Bericht* **69**, p. 190–204, Institut für Informatik, Humboldt-Universität zu Berlin, Berlin, Germany, 1996.
- [261] TARASYUK I.V. *An algebra of labelled nondeterministic processes. Joint Novosibirsk Computing Center and Institute of Informatics Systems Bulletin, Series Computer Science* **5**, p. 83–100, Novosibirsk, 1996.
- [262] TARASYUK I.V. *Back-forth equivalences for design of concurrent systems. S. Adian, A. Nerode, eds., Proceedings of 4<sup>th</sup> International Symposium on Logical Foundations of Computer Science - 97 (LFCS'97), LNCS* **1234**, p. 374–384, Yaroslavl, 1997.
- A full version of this paper is: TARASYUK I.V. *An investigation of back-forth and place bisimulation equivalences. Hildesheimer Informatik-Bericht* **8/97**, 30 p., Institut für Informatik, Universität Hildesheim, Hildesheim, Germany, April 1997.
- [263] TARASYUK I.V. *An investigation of  $\tau$ -equivalences. Hildesheimer Informatik-Bericht* **9/97**, 28 p., Institut für Informatik, Universität Hildesheim, Hildesheim, Germany, April 1997.
- [264] TARASYUK I.V. *Equivalences for behavioural analysis of multilevel systems. Joint Novosibirsk Computing Center and Institute of Informatics Systems Bulletin, Series Computer Science* **7**, Novosibirsk, 1997.
- [265] TRAKHTENBROT B.A., RABINOVITCH A., HIRSHFELD J. *Discerning causality in the behaviour of automata. Technical report* **104/88**, Tel Aviv University, 1988.
- [266] TAUBNER D.A., VOGLER W. *The step failure semantics. LNCS* **247**, p. 348–359, 1987.
- [267] VAANDRAGER F.W. *Determinism  $\rightarrow$  (event structure isomorphism step sequence equivalence). Report CS-R* **8839**, Centrum voor Wiskunde en Informatica, Amsterdam, 1988.
- [268] VAANDRAGER F.W. *An explicit representation of equivalence classes of history preserving bisimulation. Unpublished manuscript, Centrum voor Wiskunde en Informatica, Amsterdam, 1989.*
- [269] VAANDRAGER F.W. *Algebraic techniques for concurrency and their applications. Ph.D. thesis, Centrum voor Wiskunde en Informatica, Amsterdam, 244 p., 1989.*
- [270] VAANDRAGER F.W. *Determinism  $\rightarrow$  (event structure isomorphism step sequence equivalence). TCS* **79**, p. 275–294, February 1991.
- [271] VOGLER W. *Behaviour preserving refinements of Petri nets. LNCS* **246**, p. 82–93, 1987.
- [272] VOGLER W. *Failures semantics based on interval semiwords is a congruence for refinement. Technical Report TUM-I* **8905**, Institut für Informatik, Technische Universität München, 1989.
- [273] VOGLER W. *Failures semantics based on interval semiwords is a congruence for refinement. LNCS* **415**, p. 285–297, 1990.
- [274] VOGLER W. *Failures semantics of Petri nets and the refinement of places and transitions. Technical Report TUM-I* **9003**, Institut für Informatik, Technische Universität München, 1990.

- [275] VOGLER W. *Bisimulation and action refinement*. SFB Bericht **342/10/90A**, Technische Universität München, 1990.
- [276] VOGLER W. *Bisimulation and action refinement*. LNCS **480**, p. 309–321, 1991.
- [277] VOGLER W. *Deciding history preserving bisimilarity*. LNCS **510**, p. 495–505, 1991.
- [278] VOGLER W. *Failures semantics based on interval semiwords is a congruence for refinement*. *Distributed Computing* **4**, p. 139–162, 1991.
- [279] VOGLER W. *Partial words versus processes: a short comparison*. LNCS **609**, p. 292–303, 1992.
- [280] VOGLER W. *Modular construction and partial order semantics of Petri nets*. LNCS **625**, 252 p., 1992.
- [281] VOGLER W. *The limit of split<sub>n</sub>-language equivalence*. Report **288**, Institut für Mathematik, Universität Augsburg, 1993.
- [282] VOGLER W. *Bisimulation and action refinement*. TCS **114**, p. 173–200, 1993.
- [283] VOGLER W. *The limit of split<sub>n</sub>-language equivalence*. LNCS **944**, p. 636–647, 1995.
- [284] VIRBITSKAITE I.B., TARASYUK I.V. *Investigating equivalence notions for time Petri nets*. Proceedings of 4<sup>th</sup> Workshop on Logic, Languages, Information and Computation (WoLLIC'97), Fortaleza (Ceará), Brazil, August 19–22, 1997, *Logic Journal of the IGPL* **5**(6), Oxford University Press, November 1997.
- [285] WALKER D.J. *Bisimulation and divergence*. *Information and Computation* **85**(2), p. 201–241, 1990.
- [286] WEIJLAND W.P. *Synchrony and asynchrony in process algebra*. Ph.D. thesis, University of Amsterdam, 1989.
- [287] WINSKEL G. *Events in computation*. Ph.D. thesis, University of Edinburgh, 1980.
- [288] WINSKEL G. *Event structure semantics for CCS and related languages*. LNCS **140**, p. 561–567, 1982.
- [289] WINSKEL G. *Synchronization trees*. TCS **34**, p. 33–82, 1985.
- [290] WINSKEL G. *Event structures*. LNCS **255**, p. 325–392, 1987.
- [291] WINSKEL G. *An introduction to event structures*. LNCS **354**, p. 364–397, 1989.
- [292] WOLFSTAHL J., YOELI M. *An equivalence theorem for labeled marked graphs*. *IEEE Transactions on Parallel and Distributed Systems* **5**(8), p. 886–890, 1994.
- [293] YONEDA T., SHIBAYAMA A., SCHLINGLOFF B.-H., CLARKE E.M. *Efficient verification of parallel real-time systems*. LNCS **697**, p. 321–332, 1993.