Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский Томский государственный университет»

На правах рукописи

Твардовский Александр Сергеевич

КОНЕЧНО АВТОМАТНЫЕ МЕТОДЫ АНАЛИЗА И СИНТЕЗА ДИСКРЕТНЫХ СИСТЕМ С ОДНОЙ ВРЕМЕННОЙ ПЕРЕМЕННОЙ

05.13.01 – Системный анализ, управление и обработка информации
(в отраслях информатики, вычислительной техники и автоматизации)

Диссертация
на соискание ученой степени
кандидата физико-математических наук

Научный руководитель
доктор технических наук, профессор
Евтушенко Нина Владимировна

Томск – 2019

Оглавление

Введение ... 5
1 Основные определения и обозначения ... 22
1.1 Конечные автоматы ... 22
1.2 Временные автоматы ... 28
1.3 Отношения конформности между (временными) автоматами ... 34
1.4 Модели неисправности и проверяющие тесты ... 35
1.4.1 Модель неисправности ... 35
1.4.2 Проверяющие тесты ... 37
1.4.3 Методы синтеза тестов для конечных автоматов ... 37
1.5 Конечно автоматные абстракции ... 39
2 Минимизация временных автоматов ... 43
2.1 Свойства конечно автоматных абстракций ... 43
2.2 Минимизация временных автоматов по состояниям ... 46
2.3 Единственность минимальной формы для временного автомата ... 50
2.4 Частные случаи автоматов с таймаутами и временными ограничениями ... 55
2.5 Экспериментальные результаты по оценке времени минимизации в зависимости от размеров автомата ... 58
2.6 Выводы по главе 2 ... 59
3 Композиции временных автоматов ... 61
3.1 Параллельная композиция классических и временных детерминированных автоматов ... 61
3.1.1 Бинарная параллельная композиция конечных автоматов ... 62
3.1.2 Бинарная параллельная композиция временных автоматов ... 64
3.1.3 Операция параллельной композиции для автоматов с временными ограничениями ... 67
3.2 Замкнутость класса временных детерминированных автоматов относительно операции параллельной композиции ... 73
3.3 Выводы по главе 3 ... 80

4 Построение проверяющих тестов для детерминированных временных автоматов 81
4.1 Методы синтеза тестов для детерминированных автоматов с таймаутами и временными ограничениями 81
4.1.1 Свойства конечно автоматных абстракций 82
4.1.2 Модель неисправности для синтеза тестов на основе временных автоматов 85
4.1.3 Синтез тестов с гарантированной полнотой для временных автоматов 87
4.2 Экспериментальные результаты 90
4.2.1 Исследование длины проверяющих тестов для временных автоматов . 90
4.2.2 Синтез тестов для сервисных систем 92
4.3 Выводы по главе 4 95
5 Синтез и оптимизация проверяющих тестов для недетерминированных временных автоматов 96
5.1 Синтез проверяющих тестов для недетерминированных автоматов с таймаутами и временными ограничениями 96
5.1.1 Классические методы синтеза тестов для недетерминированных конечных автоматов 97
5.1.2 Конечно автоматная абстракция для недетерминированных временных автоматов 99
5.1.3 Метод синтеза проверяющих тестов для недетерминированных автоматов с таймаутами и временными ограничениями 102
5.2 Методы оптимизации проверяющих тестов 104
5.2.1 Адаптивные различающие тестовые последовательности для классических автоматов 105
5.2.2 Экспериментальные результаты по построению адаптивных различающих последовательностей для классических автоматов 109
5.2.3 Оптимизация различающих тестовых примеров для классических автоматов 111
5.2.4 Сокращение спецификации для классических автоматов 115

5.3 Выводы по главе 5 119
Заключение 120
Список использованной литературы 123

Введение

Актуальность работы. В настоящее время практически любая сфера деятельности человека зависит от качества работы той или иной информационной системы. Стремление обеспечить стабильную и надежную работу программного и аппаратного обеспечения таких систем стимулировало развитие соответствующих формальных методов анализа и синтеза [2, 10, 11, 23, 25, 26, 29, 32, 43, 44, 48, 50, 52, 56, 57, 58, 59, 67, 68, 74, 82, 84, 85]. Конечно автоматные методы анализа и синтеза [1, 3, 4, 6, 7, 31, 33, 34, 41, 49, 50, 54, 55] нашли широкое применение при проверке функциональных и нефункциональных требований к дискретным и гибридным управляющим системам, их оптимизации, синтезе сложных систем как композиции более простых в некотором смысле и др. Модель конечного автомата обладает естественной реактивностью и позволяет описывать системы с конечным числом состояний, входных воздействий и выходных реакций [4]. Однако, системы, поведение которых зависит от времени, не могут быть достаточно адекватно описаны в рамках модели классического конечного автомата, что привело к расширению конечно автоматных моделей временными переменными и введению понятия временного автомата [27, 35-37, 39-41, 43, 45, 46, 51, 60]. Модель автомата с одной временной переменной [27] является частным случаем модели полуавтомата с конечным множеством временных переменных [25, 32, 72, 75], для которой многие задачи алгоритмически неразрешимы. В ряде случаев рассмотрение конечного автомата с одной временной переменной (далее временного автомата) представляет собой достаточно хороший компромисс между сложностью и выразительностью модели. Несмотря на большое количество исследований в области временных автоматов, многие задачи анализа и синтеза для временных моделей остаются нерешёнными.

Одной из задач оптимизации в теории автоматов является построение эквивалентного автомата с минимальным числом состояний [4, 24, 43, 56, 66, 75], и эта задача остается нерешенной для временных автоматов. Методы синтеза автомата, описывающего работу композиций различной структуры, достаточно

хорошо изучены для классических автоматов [30, 53, 62, 63, 83], однако, методы композиции для временных автоматов [9, 51] разработаны лишь для некоторых частных случаев. Более того, для временных автоматов не определены условия замкнутости различных классов автоматов относительно вводимых операций композиции.

В рамках задач анализа для дискретных управляющих систем исследуются свойства конечного автомата, в частности, вопросы идентификации состояний конечного автомата и возможность построения конечного множества входных последовательностей (тестов с гарантированной полнотой), позволяющих идентифицировать заданный автомат с определенной точностью в известном множестве других автоматов, например, в множестве автоматов-мутантов, которые описывают поведение тестируемой системы [3, 6, 26, 31, 33, 54, 55, 61, 64, 65, 71, 86, 87]. Методы синтеза проверяющих тестов для детерминированных и недетерминированных конечных автоматов изучаются достаточно давно [3, 31, 33, 38, 49, 54, 55, 61, 64, 65, 71, 86], однако, при попытке «переложить» эти методы на временные автоматы [27, 35, 36, 37, 39, 60, 91] полученные тесты имеют слишком большую длину даже для детерминированных временных автоматов и требуют дополнительной оптимизации. Более того, в ряде случаев вопрос о полноте таких тестов для временных автоматов остаётся нерешённым. Достаточно часто авторы работ по синтезу тестов на основе автоматов с одной временной переменной рассматривают только отношения конформности, оставляя открытыми вопросы построения тестов [27, 49, 50].

Решаемые в кандидатской диссертации задачи синтеза направлены на оптимизацию конечных автоматов с одной временной переменной, в то время как задачи анализа представлены алгоритмами генерации и оптимизации проверяющих тестов с гарантированной полнотой для таких автоматов.

Целью работы является разработка методов оптимизации временных автоматов и их композиций и синтеза тестов с гарантированной полнотой для контроля выполнения функциональных и нефункциональных требований к системам, поведение которых описано такой моделью.

Для выполнения поставленной цели, необходимо решить следующие **задачи.**

1. Разработать метод минимизации временного автомата и исследовать, существует ли для временных автоматов единственная (с точностью до изоморфизма) приведённая форма.

2. Адаптировать операцию параллельной композиции классических конечных автоматов для временных автоматов и исследовать замкнутость класса детерминированных временных автоматов относительно такой операции.

3. Адаптировать для временных автоматов классические конечно автоматные методы синтеза тестов с гарантированной полнотой.

4. Предложить метод оптимизации длины проверяющих тестов с гарантированной полнотой для полностью определённых недетерминированных, возможно, временных автоматов.

Методы исследования.

Для решения задач анализа и синтеза автоматов с одной временной переменной используется аппарат дискретной математики, в частности, аппарат классической теории автоматов и математической логики. Для анализа эффективности предложенных алгоритмов используются компьютерные эксперименты.

Научная новизна.

1. Предложен метод построения минимальной формы полностью определенного детерминированного неинициального временного автомата; показано, что минимальная форма такого автомата единственна с точностью до изоморфизма.

2. Показано, что параллельная композиция взаимодействующих временных детерминированных автоматов не всегда является детерминированным временным автоматом. Выделены классы детерминированных временных автоматов, замкнутые относительно операции параллельной композиции.

3. Предложен метод синтеза проверяющих тестов с гарантированной полнотой для детерминированных временных автоматов, обобщающий известные

методы построения таких тестов для автоматов только с таймаутами или только с временными ограничениями.

4. Предложен алгоритм синтеза проверяющих тестов с гарантированной полнотой для недетерминированных временных автоматов; предложен подход к оптимизации построенных проверяющих тестов.

Основные положения, выносимые на защиту.

1. Метод минимизации детерминированных полностью определенных автоматов с таймаутами и временными ограничениями; единственность минимальной формы для неинициальных временных автоматов.

2. Алгоритм построения композиции детерминированных временных автоматов; классы временных автоматов, замкнутые относительно операции параллельной композиции.

3. Конечно автоматный метод синтеза проверяющих тестов с гарантированной полнотой для детерминированных временных автоматов, обобщающий известные методы синтеза проверяющих тестов для автоматов только с таймаутами или только с временными ограничениями.

4. Алгоритмы синтеза и оптимизации проверяющих тестов с гарантированной полнотой для недетерминированных, возможно, временных автоматов.

Достоверность результатов.

Все научные положения и выводы, полученные в работе, основаны на утверждениях и теоремах, доказанных с использованием аппарата дискретной математики.

Теоретическая и практическая ценность.

В работе предложены методы минимизации и операция композиции для автоматов с одной временной переменной, которые расширяют классическую теорию автоматов. Кроме того, классическая теория автоматов расширяется новыми моделями неисправности для временных автоматов и методами синтеза проверяющих тестов с гарантированной полнотой относительно таких моделей. Разработанные методы и алгоритмы могут быть использованы при построении

проверяющих тестов с гарантированной полнотой для проверки функциональных и нефункциональных требований к программным и аппаратным реализациям дискретных управляющих систем, для которых важны временные аспекты. Таким образом, предложенные в работе методы анализа и синтеза позволяют расширить области применения конечно автоматных подходов методами и алгоритмами для временных автоматов.

Реализация полученных результатов. Исследования, результаты которых изложены в диссертации, проводились в рамках следующих проектов:

1. Грант РНФ 16-49-03012 “Надежность, безопасность и доверие в системах, используемых в качестве сервисов: масштабируемые решения для эффективного анализа и менеджмента”, 2016-2018 гг.

2. Грант РФФИ № 15-58-46013 “Разработка новых эвристик и автоматных представлений для эффективной проверки функциональных требований для цифровых схем”, 2015-2016 гг.

3. НИР «Исследование и разработка вероятностных, статистических и логических методов и средств оценки качества компонентов телекоммуникационных систем» в рамках проектной части госзадания РФ № 739, 2014-2015 гг.

Апробация работы.

Основные положения и результаты были представлены на следующих конференциях и семинарах:

1. 10-я Российская конференция с международным участием “Новые информационные технологии в исследовании сложных структур», пос. Катунь Алтайского края, Россия, 2014.

2. Семинар для аспирантов и докторантов в рамках 27-й IFIP международной конференции по тестированию систем и программного обеспечения (ICTSS’2015), Дубай, ОАЭ, 2015.

3. 11-я Международная конференция “Новые информационные технологии в исследовании сложных структур”, Екатеринбург, Россия, 2016.

4. 17-я международная конференция молодых специалистов по микро/нанотехнологиям и электронным приборам (EDM), Алтайский край, Россия, 2016.

5. Международный Французско-Российский семинар "Software Verification, Testing, and Quality Estimation", Париж, Франция, 2014.

6. 11-й коллоквиум Spring/Summer Young Researchers Colloquium of Software Engineering (SYRCoSE), Иннополис, Россия, 2017.

7. 29-я IFIP международная конференция по тестированию систем и программного обеспечения (ICTSS), Санкт-Петербург, Россия, 2017.

8. 12-й коллоквиум “Spring/Summer Young Researchers Colloquium of Software Engineering” (SYRCoSE), Великий Новгород, Россия, 2018.

9. 12-я Международная конференция “Новые информационные технологии в исследовании сложных структур”, Алтайский край, Россия, 2018.

10. 30-я IFIP международная конференция по тестированию систем и программного обеспечения (ICTSS), Кадис, Испания, 2018.

11. 13-й коллоквиум “Spring/Summer Young Researchers Colloquium of Software Engineering” (SYRCoSE), Саратов, Россия, 2019.

12. 10-й международный семинар “Program Semantics, Specification and Verification: Theory and Applications” (PSSV), Новосибирский Академгородок, Россия, 2019.

По результатам проведенных исследований опубликовано 18 работ в научных журналах, докладах и тезисах докладов на конференциях различного уровня [13-22, 28, 47, 76-81], в том числе 6 статей в рецензируемых журналах из перечня ВАК [14, 15, 18, 19, 21, 22] и 5 публикаций [28, 47, 76-78], входящих в международные базы цитирования Scopus и Web of Science.

Структура и объем работы. Диссертация состоит из введения, 5 глав, заключения и списка использованной литературы, включающего 91 наименование. Диссертация содержит 35 рисунков и одну таблицу. Объем диссертации составляет 132 страницы.

Краткое содержание работы.

В первой главе вводятся основные определения и обозначения, используемые в работе, и кроме того, эта глава содержит краткий обзор литературы по теме диссертации. Конечный автомат [4] представляет собой систему с конечным числом состояний, переходы между которыми выполняются под действием входных символов, и при этом выдается предписанный выходной символ. В первом разделе представлены необходимые понятия из классической теории автоматов. Автомат называется полностью определенным, если в каждом состоянии автомата определен, по крайней мере, один переход под действием каждого входного символа. Конечный автомат может быть доопределён до полностью определённого различными способами [48], ввиду чего задачи анализа и синтеза достаточно часто формулируются и исследуются для полностью определённых автоматов. Автомат называется детерминированным, если в каждом состоянии под действием каждого входного символа определено не более одного перехода, в противном случае автомат называется недетерминированным. Причин появления недетерминизма в описаниях поведения систем достаточно много, одной из таковых может быть опциональность спецификации [61, 64]. В настоящей работе мы рассматриваем только полностью определённые автоматы, в то время как причины (не-)детерминизма в используемой модели будут уточняться в зависимости от решаемой задачи.

Временной автомат представляет собой конечный автомат, расширенный одной временной переменной [27, 37] и связанными с ней временными аспектами; автомат может иметь временные ограничения [27, 35], таймауты [9, 27, 43] и выходные задержки, часто называемые выходными таймаутами [45]. Такой автомат является частным случаем модели временного полуавтомата [25], содержащего конечное число временных переменных. Известно, что ряд задач анализа и синтеза алгоритмически неразрешим для временных полуавтоматов [25, 32, 75], что делает автомат с одной временной переменной достаточно подходящим «компромиссом» между сложностью и выразительностью модели. Во временном автомате (автомате с одной временной переменной) временная переменная

постоянно увеличивает своё значение и сбрасывается при поступлении входного символа и выдаче выходного символа. Время между подачей входного символа и выдачей выходного символа называется выходной задержкой или выходным таймаутом. В ряде работ считается, что переходы выполняются мгновенно, т.е. выходные таймауты отсутствуют [27], однако, наличие таких таймаутов предоставляет больше возможностей для описания поведения реальных систем, в которых практически все переходы выполняются с некоторой задержкой [50, 60]. В некоторых работах вводится понятие выходных временных интервалов [36], определяющих допустимые интервалы времени для выполнения перехода, однако такое расширение не рассматривается в данной работе. Функция таймаута [91] определяет максимальное время ожидания входного воздействия для каждого состояния временного автомата, и если входной символ не поступает до истечения таймаута, то автомат может спонтанно перейти в другое состояние. Предполагается, что переход по таймауту происходит мгновенно и «сбрасывает» значение временной переменной в ноль. При поступлении входного символа мы полагаем, что функция таймаута «отключается» до перехода автомата в следующее состояние, возможно, с некоторой выходной задержкой. Временные ограничения представляют собой интервалы на переходах, содержащие значения временной переменной, при которых соответствующий переход по входному символу может быть выполнен. Исследуемые в работе типы временных автоматов представлены во втором разделе первой главы. В большинстве случаев мы рассматриваем временные автоматы, содержащие как таймауты, так и временные ограничения (автомат с таймаутами и временными ограничениями), однако, частные случаи, такие как автоматы только с таймаутами или только с временными ограничениями, достаточно часто встречаются в практических приложениях и отражены в данной работе. Отметим, что два частных случая временного автомата не являются взаимозаменяемыми [5, 27], т.е. существует автомат с таймаутами, который не может быть описан автоматом с временными ограничениями, и наоборот. Частные случаи позволяют упростить модель, но адекватно описывают поведение более узкого класса систем.

В третьем разделе первой главы вводятся отношения конформности для конечных и временных автоматов. Отношения конформности используются как для решения задач анализа, определяя критерий соответствия между эталонным автоматом и исследуемой системой, так и для определения требований к синтезируемой системе. В настоящей работе мы рассматриваем отношения эквивалентности, редукции и изоморфизма как для неинициальных, так и для инициальных временных автоматов, т.е. автоматов, в которых выделено начальное состояние. Два состояния (временных) полностью определенных автоматов эквивалентны, если реакции автоматов в этих состояниях совпадают на любую последовательность входных воздействий. Два инициальных (временных) автомата эквивалентны, если эквивалентны их начальные состояния. Неинициальные (временные) автоматы эквивалентны, если каждому состоянию одного автомата соответствует эквивалентное состояние другого автомата и наоборот. Изоморфизм (временных) автоматов предполагает, что между их состояниями и переходами можно установить взаимно однозначное соответствие, т.е. структура автоматов совпадает точностью до переименования состояний. При рассмотрении недетерминированных автоматов часто используется отношение редукции; один автомат является редукцией другого, если поведение первого автомата содержится в поведении второго.

Четвёртый раздел первой главы содержит определения модели неисправности и (полного) проверяющего теста относительно модели неисправности [31, 33, 35, 50, 63, 64, 71]. В тестировании на основе конечно автоматных моделей достаточно часто рассматриваются дискретные системы, обладающие надежным сигналом сброса, что позволяет описывать поведение спецификации и тестируемой системы инициальными автоматами, т.е. автоматами с единственным начальным состоянием. Такой подход значительно сокращает длину проверяющих тестов с гарантированной полнотой [23] и является одним из эффективных решений при синтезе тестопригодных дискретных систем. Модель неисправности содержит эталонный автомат (спецификацию), отношение конформности, позволяющее определить, соответствует ли спецификации

проверяемая система (реализация), и область неисправности, определяющую множество автоматов-реализаций, которые должны быть идентифицированы построенным тестом. В свою очередь, проверяющий тест представляет собой конечное множество входных последовательностей, по реакции на которые можно определить, конформна ли тестируемая реализация спецификации. При синтезе тестов для инициальных детерминированных автоматов в качестве отношения конформности обычно используется отношение эквивалентности [4, 27, 87], при котором реакция конформной реализации на любую последовательность входных воздействий должна совпадать с таковой для спецификации. Однако, как отмечалось выше, спецификации реальных систем часто оказываются недетерминированными; в этом случае, для анализа конформности соответствующих реализаций рассматривается отношение редукции и, в то же время, полагается, что реализация является детерминированной [36, 38, 46, 49, 61, 64, 65, 89]. Иными словами, для конформной реализации выходная реакция на каждую входную последовательность должна содержаться в множестве таковых для автомата-спецификации. Проверяющие тесты конечной длины обычно строятся для конечной области неисправности (конечного множества неисправностей, подлежащих обнаружению). Поэтому одним из способов формирования области неисправности является явное перечисление возможных реализаций; в этом случае, для синтеза полного проверяющего теста используется модель «белого ящика», поскольку внутренняя структура реализации известна. Однако, такой подход может быть успешно использован лишь при небольшом количестве возможных реализаций, и в настоящей диссертации мы рассматриваем методы синтеза полных проверяющих тестов относительно модели «чёрного ящика». В большинстве случаев область неисправности для такой модели представляет собой множество автоматов с ограниченным числом состояний и с таким же входным алфавитом, как у автомата-спецификации. Проверяющий тест называется полным относительно заданной модели неисправности, если каждая неконформная реализация из области неисправности обнаруживается тестом, в то время как каждая конформная реализация из области неисправности «проходит»

тест. Для классических конечных детерминированных автоматов проверяющий тест может быть построен методом Василевского, известным также как W-метод [3, 31]. Большинство используемых в настоящий момент методов синтеза тестов на основе автоматных моделей являются модификациями данного метода. Тест при этом формируется на основе специальных входных последовательностей, позволяющих перевести автомат в требуемое состояние и отличить каждое состояние автомата от других (идентифицировать состояние до и после подачи входного символа). Такой тест позволяет обнаружить не только ошибки переходов и выходов в реализации, но и большое количество других ошибок.

В пятом разделе мы иллюстрируем, что в ряде случаев поведение временного автомата может быть адекватно описано соответствующим конечным автоматом, т.е. конечно автоматной абстракцией. Алгоритмы построения конечно автоматной абстракции для автоматов с таймаутами и временными ограничениями и их частных случаев были предложены в [27]. Метод построения конечно автоматной абстракции для временного автомата общего вида основан на сохранении информации о значении временной переменной в состояниях соответствующий абстракции. Аналогичный подход применяется и для автоматов с таймаутами. Таким образом, наличие таймаутов в модели временного автомата влечёт увеличение числа состояний при переходе к конечному автомату. В то же время, для автоматов только с временными ограничениями значение временной переменной можно представлять частью входного символа, что увеличивает входной алфавит при построении абстракции. В таком случае, между состояниями временного автомата и его конечно автоматной абстракции может быть установлено взаимно однозначное соответствие, что, в частности, облегчает задачу синтеза полных проверяющих тестов классическими конечно автоматными методами. Отметим также, что конечно автоматные абстракции из [27] не учитывают выходных задержек, и в настоящей работе мы расширяем понятие конечно автоматной абстракции для этого случая.

Во второй главе рассматривается задача минимизации полностью определенных детерминированных автоматов с таймаутами и временными

ограничениями. Под минимизацией [4] классического детерминированного неинициального автомата понимается построение приведённого автомата, эквивалентного исходному, т.е. синтез автомата с заданным поведением, в котором любые два различные состояния не являются эквивалентными. Неинициальные автоматы, т.е. автоматы, у которых каждое состояние может быть начальным, называются эквивалентными, если для каждого состояния одного автомата существует эквивалентное состояние другого, и наоборот. В настоящей работе мы обобщаем различные подходы к минимизации автоматов с одной временной переменной и предлагаем метод минимизации таких временных автоматов на основе соответствующей конечно автоматной абстракции. В первом разделе устанавливаются свойства конечно автоматной абстракции временного автомата, позволяющие использовать такую абстракцию для проверки эквивалентности между состояниями временного автомата. Второй раздел посвящён минимизации временного автомата по состояниям, т.е. минимизации числа состояний временного автомата. Подобно алгоритму минимизации для классических детерминированных автоматов, алгоритм построения приведенной по состояниям формы детерминированного временного автомата основан на построении разбиения множества состояний автомата на классы эквивалентности. Каждый класс эквивалентности исходного автомата соответствует состоянию приведённой формы. Предлагаемый в работе алгоритм использует конечно автоматную абстракцию для построения такого разбиения. Однако, в отличие от классических автоматов, приведённая по состояниям форма временного автомата (эквивалентный временной автомат с минимальным числом состояний) не является единственной с точностью до изоморфизма, что и демонстрируется в третьем разделе. Для получения канонической минимальной формы детерминированного временного автомата временные аспекты также должны быть минимизированы. Соответственно, вводится понятие приведённой по времени формы временного автомата и доказывается, что приведенная по состояниям и времени форма временного автомата единственна с точностью до изоморфизма. В четвёртом разделе предложенный метод используется для минимизации частных случаев

временных автоматов, т.е. для минимизации автоматов только с таймаутами и автоматов только с временными ограничениями. Подобно минимизации классических детерминированных автоматов [4], алгоритм минимизации временного детерминированного автомата использует линейное число шагов относительно числа состояний конечно автоматной абстракции, и пятый раздел содержит результаты компьютерных экспериментов по времени минимизации случайно сгенерированных временных автоматов.

В третьей главе исследуется задача оптимизации числа временных переменных в системах временных детерминированных автоматов, взаимодействующих в диалоговом режиме, а именно, вводится операция параллельной композиции инициальных детерминированных автоматов с временными ограничениями и выделяются классы временных автоматов, замкнутые относительно этой операции, т.е. классы автоматов, для которых параллельная композиция является детерминированным временным автоматом. Параллельная композиция (временных) автоматов, компоненты которой работают в режиме диалога, активно изучалась и изучается в настоящее время [9, 30, 51, 53, 62, 63, 83]. Такая композиция хорошо соответствует функционированию программных компонент сложной системы, в частности, позволяет достаточно адекватно описывать взаимодействие клиентской и серверной частей сетевых сервисов. В первом разделе вводятся определения бинарной параллельной композиции для классических и временных автоматов и рассматриваются существующие методы построения композиций. В ряде случаев, композицию из n взаимодействующих автоматов можно свести к случаю бинарной композиции, и соответственно, полученные для бинарных композиций результаты можно расширить на многокомпонентные композиции. Для классических конечных автоматов параллельная композиция может быть построена путём формирования дерева преемников или посредством классических операций над соответствующими полуавтоматами. Множество детерминированных конечных автоматов замкнуто относительно операции параллельной композиции при условии так называемой «медленной внешней среды», когда следующий входной

символ подается на композицию только после того, как композиция произвела выходной сигнал на предыдущий входной символ. Метод построения параллельной композиции для автоматов только с таймаутами был предложен в ряде работ [9, 51], и в настоящей работе мы предлагаем метод синтеза параллельной композиции для инициальных автоматов с временными ограничениями на основе построения глобального полуавтомата. Если автоматы не имеют временных ограничений, то результат операции совпадает с таковым для параллельной композиции классических конечных автоматов. Мы показываем, что в отличие от классических автоматов, такая операция, в общем случае, не является замкнутой в множестве детерминированных временных автоматов при условии «медленной внешней среды». Предложенный алгоритм синтеза параллельной композиции возвращает автомат-композицию как детерминированный и полностью определённый автомат, если это возможно, или выдаёт сообщения о недетерминированном поведении композиции и/или наличии осцилляций в противном случае. Во втором разделе исследуются причины появления недетерминизма композиции и определяются классы временных автоматов, для которых композиция является временным детерминированным автоматом.

Четвёртая глава посвящена разработке метода синтеза проверяющих тестов с гарантированной полнотой для детерминированных автоматов с таймаутами и временными ограничениями [27]. Как отмечалось в первой главе, в ряде случаев, поведение временного автомата может быть адекватно описано классическим конечным автоматом (конечно автоматной абстракцией) [27]. Предложенные модель неисправности и соответствующий метод синтеза тестов основаны на такой конечно автоматной абстракции и представлены в первом разделе. Методы синтеза тестов для временного автомата по соответствующему конечному автомату были рассмотрены в работах [37, 60], однако, многие вопросы остались нерешенными. Наиболее просто адаптировать классические методы синтеза тестов с гарантированной полнотой, используя синтез тестов на основе абстракции, можно для автоматов с временными ограничениями [37]. В таком случае, проверяющий тест может быть построен непосредственно по конечно автоматной абстракции,

поскольку между состояниями автомата с временными ограничениями и состояниями его абстракции может быть установлено взаимно однозначное соответствие. Таким образом, модель неисправности для временного автомата может быть легко преобразована в соответствующую модель для конечно автоматной абстракции и построенный по абстракции тест будет полным относительно модели неисправности временного автомата. При наличии таймаутов во временном автомате число состояний конечно автоматной абстракции может превышать число состояний исходного временного автомата, что порождает вопрос о соответствии между моделями неисправности на основе исходного временного автомата и его абстракции. Кроме того, в отличие от неинициальных автоматов с таймаутами и временными ограничениями, рассмотренных в главе 2, минимальная форма инициального детерминированного временного автомата не является единственной и две минимальные формы могут иметь различное число состояний. В [27] установлено необходимое и достаточное условие эквивалентности детерминированных инициальных временных автоматов, согласно которому такие временные автоматы эквивалентны, если и только если эквивалентны их конечно автоматные абстракции. Предлагаемая модель неисправности для детерминированных временных автоматов содержит спецификацию, представленную автоматом с таймаутами и временными ограничениями, отношение эквивалентности и область неисправности, включающую каждый полностью определённый детерминированный автомат с таймаутами и временными ограничениями с таким же входным алфавитом, как и у спецификации, и конечно автоматной абстракцией не более чем с m состояниями. Соответствующий проверяющий тест строится по конечно автоматной абстракции классическими «W-подобными» методами и является полным относительно заданной модели неисправности. Второй раздел содержит экспериментальные результаты по оценке длины проверяющих тестов для случайно сгенерированных детерминированных автоматов с таймаутами и временными ограничениями и реальных сервисных систем.

В пятой главе предлагается алгоритм синтеза проверяющих тестов для недетерминированных автоматов с таймаутами и временными ограничениями относительно редукции, при условии, что проверяемый автомат является детерминированным, и рассматриваются возможные способы оптимизации длины таких тестов для классических и временных автоматов. Как отмечалось выше, тесты относительно редукции используются на практике, когда недетерминизм в спецификации является следствием ее опциональности [36, 38, 46, 49, 61, 64, 65, 89]. Аналогично подходу, предложенному в предыдущей главе, предлагаемые модель неисправности и метод синтеза тестов для недетерминированных временных автоматов основаны на конечно автоматной абстракции и представлены в первом разделе. Мы исследуем свойства конечно автоматной абстракции для недетерминированных автоматов с таймаутами и временными ограничениями и устанавливаем, что подобно отношению эквивалентности, отношение редукции между инициальными временными автоматами также можно полностью проверить на основе соответствующих конечно автоматных абстракций. Методы синтеза тестов на основе автоматных моделей для недетерминированных автоматов хорошо известны; в частности, HSI-метод позволяет построить полные тесты для недетерминированных конечных автоматов относительно редукции [64]. Однако, тесты, построенные для недетерминированных автоматов-спецификаций относительно редукции, практически всегда оказываются очень длинными и, достаточно часто – избыточными. Как известно, длина полных тестов, построенных HSI-методом, в общем случае экспоненциально зависит от числа состояний недетерминированного автомата-спецификации. Ввиду последнего, во втором разделе мы исследуем возможные подходы к оптимизации проверяющих тестов для классических и временных недетерминированных автоматов, возможно, с сокращением области неисправности. В первую очередь, мы предлагаем использовать адаптивные тестовые последовательности [38, 49, 65, 87, 88, 89], поскольку адаптивность является одной из возможностей сокращения длины различающих последовательностей [54, 89], и соответственно, длины полного проверяющего

теста в «W-подобных» методах синтеза таких тестов. Следующий входной символ в адаптивной входной последовательности, которая обычно представляется ациклическим автоматом, достаточно часто называемым тестовым примером [38], зависит от реакции тестируемой системы на предыдущий входной символ. При существовании (адаптивной) последовательности, различающей все состояния автомата-спецификации (аналог (адаптивной) диагностической последовательности в детерминированном автомате), и (адаптивной) передаточной последовательности для каждого состояния автомата из начального состояния, длина проверяющего теста в значительной степени определяется длиной различающей последовательности. Известно, что адаптивные различающие последовательности обычно короче безусловных [65, 89] и существуют чаще, что позволяет сократить длину проверяющего теста относительно моделей «белого и черного» ящиков. Мы исследуем свойства адаптивных различающих последовательностей для классических недетерминированных автоматов и предлагаем алгоритм построения кратчайшей адаптивной различающей последовательности для состояний недетерминированного автомата, основанный на построении различающего автомата [38]. Другим подходом к оптимизации проверяющих тестов относительно редукции для полностью определённых недетерминированных конечных автоматов является сокращение спецификации. Согласно описанным выше результатам, проверяющие тесты имеют меньшую длину, когда спецификация обладает различающей последовательностью полиномиальной длины и передаточными последовательностями для каждого состояния. Мы предлагаем удалять из спецификации минимальное количество переходов, чтобы полученная сокращённая спецификация удовлетворяла таким свойствам, и соответствующий тест имел полиномиальную длину, таким образом, уменьшая опциональность в автоматном описании для оптимизации проверяющего теста.

В **заключении** подводятся итоги о проделанной работе и обсуждаются возможные направления дальнейших исследований.

1 Основные определения и обозначения

Настоящая глава содержит используемые в работе определения и обозначения из теории конечных автоматов [4]. Кроме того, вводится понятие временного автомата как конечного автомата с одной временной переменной. Рассматриваются также конечно автоматные абстракции, которые используются для описания ряда свойств временного автомата [27].

1.1 Конечные автоматы

Конечный автомат [4] представляет собой модель с конечным числом состояний, которая используется для описания реактивных или реагентных систем [90], выполняющих переходы между состояниями под действием входного воздействия, выдавая при этом некоторую выходную реакцию, и в настоящее время достаточно часто называемых *трансдьюсерами*. При поступлении входного воздействия автомат выполняет переход в следующее состояние и выдает выходной сигнал (реакцию) в соответствии с текущим состоянием и поступившим входным сигналом, после чего ожидает следующее входное воздействие. В настоящей работе конечный автомат рассматривается как преобразователь потенциально бесконечного множества входных последовательностей в множество выходных последовательностей такой же длины [4, 82]. Конечный автомат позволяет описать поведение дискретной системы, в которой выделены конечные непустые множества состояний, входных воздействий и выходных реакций. Необходимо отметить, что в случае наличия недетерминизма в описываемой системе, в одном состоянии автомата может существовать несколько переходов по одному и тому же входному воздействию.

Под *конечным автоматом* [4] понимается четвёрка $\mathsf{S} = (S, I, O, h_{\mathsf{S}})$, где I – конечное непустое множество *входных символов* (входной алфавит), O – конечное непустое множество *выходных символов* (выходной алфавит), $I \cap O = \varnothing$, S – конечное непустое множество *состояний*, $h_{\mathsf{S}} \subseteq (S \times I \times O \times S)$ – *отношение переходов*. Кортеж (s, i, o, s') описывает *переход* из состояния s в состояние s' под

действием входного символа i с выдачей выходного символа o. Для наглядности, конечный автомат часто представляется ориентированным графом (диаграммой переходов), вершины которого соответствуют состояниям автомата, а дуги помечены парой (входной символ/выходной символ). Таким образом, если в автомате существует переход (s, i, o, s'), то в соответствующем графе существует дуга из вершины s в вершину s', помеченная парой i/o.

Пример. Многие современные интернет-сервисы используют двух-шаговую аутентификацию для получения доступа к учётной записи (также называемую *аккаунтом*), требуя не только ввести пароль, но и подтвердить личность пользователя, введя специальный код, отправленный по электронной почте. В качестве примера мы рассмотрим фрагмент серверной реализации такой системы, автоматное описание которого (рисунок 1.1) содержит два состояния, два входных и два выходных символа. В состоянии “Ожидание запроса” система ждёт запрос от пользователя на разрешение входа в учётную запись; когда поступает входной символ “Запрос” автомат отправляет пользователю письмо с кодом доступа и переходит в состояние “Ожидание кода”. Далее система ждёт введения правильного кода пользователем (или возможно повторения запроса), в ответ на который предоставляет или не предоставляет доступ к учётной записи. Для простоты на рисунке мы не рассматриваем случай, когда введенный код неправильный; в этом случае система не разрешает доступ к учетной записи.

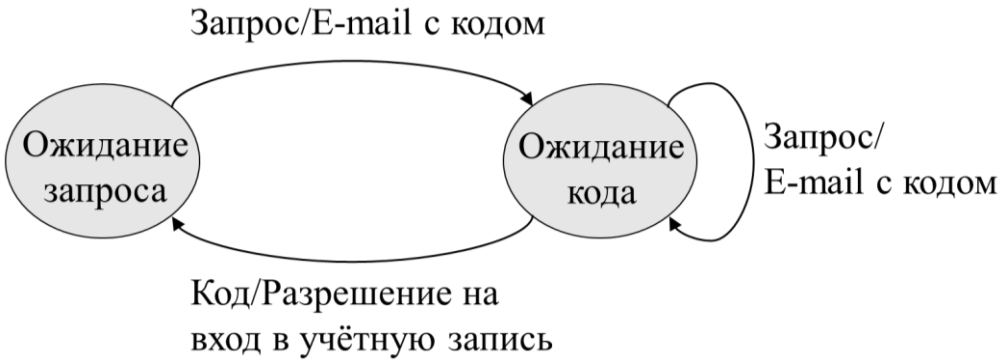

Рисунок 1.1 – Конечный автомат для системы аутентификации

Если в автомате S для любой пары $(s, i) \in S \times I$ существует переход $(s, i, o, s') \in h_{\mathsf{S}}$, то конечный автомат называется *полностью определённым* (по входным

символам), иначе – *частичным*. Частичность автомата существенно повышает сложность решения многих задач теории автоматов [87]. В ряде случаев полностью определённый автомат может быть достаточно просто получен из частичного доопределением неопределенных переходов, механизм которого существенно зависит от интерпретации таких переходов [2, 48]. Различные виды доопределений широко используются при анализе систем, в частности, при анализе реализаций телекоммуникационных протоколов, и в настоящей работе мы рассматриваем только полностью определённые автоматы. Заметим, что автомат на рисунке 1.1 является частичным, но может быть доопределён до полностью определённого переходом по входному символу “Код” в состоянии “Ожидание запроса” с выходным символом NULL. При таком доопределении разрешение на вход в учетную запись будет по-прежнему дано только после соответствующего запроса и правильно введенного кода (рисунок 1.2).

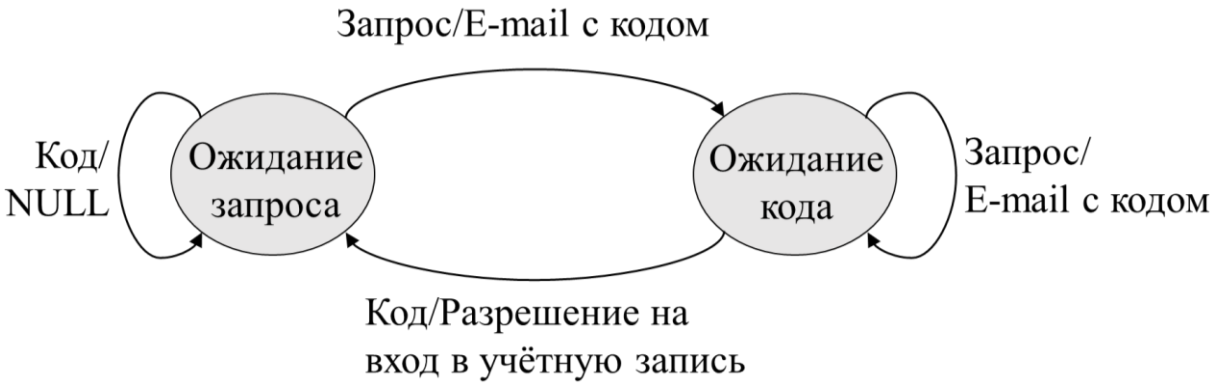

Рисунок 1.2 – Доопределение конечного автомата для системы аутентификации

Если в автомате S для любой пары $(s, i) \in S \times I$ существует не более одного перехода $(s, i, o, s') \in h_{\mathsf{S}}$, то конечный автомат называется *детерминированным*, иначе – *недетерминированным*. Например, в спецификацию системы на рисунке 1.2 можно внести недетерминизм, добавив возможность отправлять коды пользователю не только по электронной почте, но и посредством SMS-сообщений. Соответствующий автомат изображён на рисунке 1.3 и является примером недетерминизма как следствия опциональности в спецификации. Полностью определённый недетерминированный автомат называется

наблюдаемым, если для каждой тройки $(s, i, o) \in S \times I \times O$ существует единственный переход $(s, i, o, s') \in h_S$, т.е. следующее состояние определяется единственным образом. Заметим, что автомат на рисунке 1.3 является наблюдаемым.

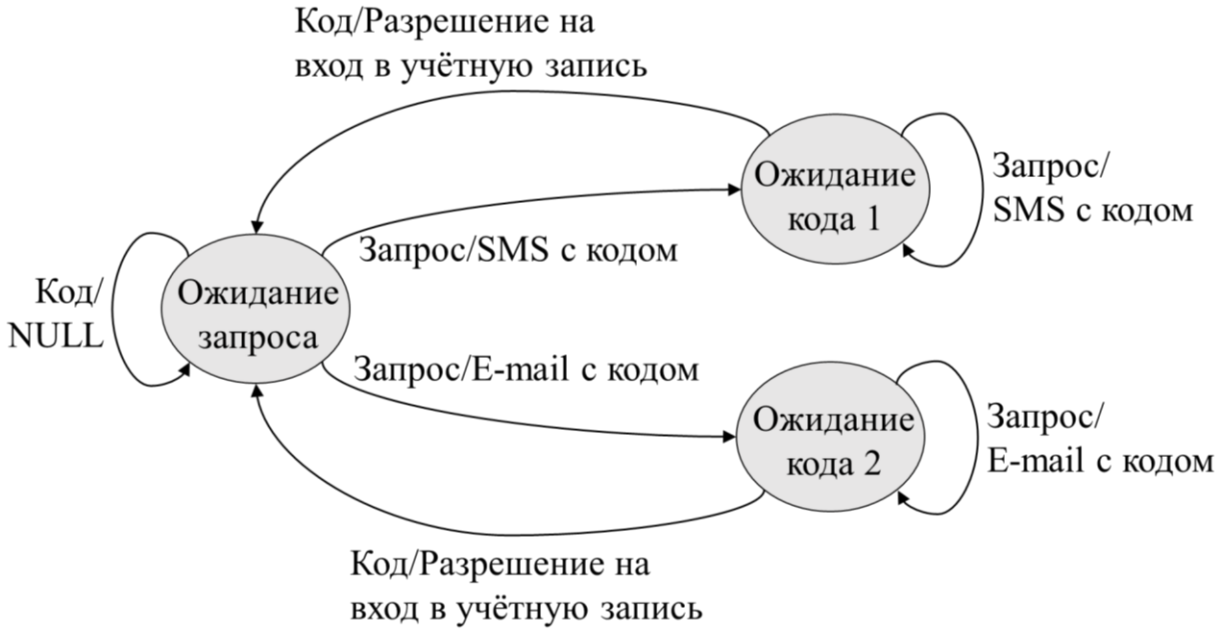

Рисунок 1.3 – Полностью определённый недетерминированный конечный автомат для системы аутентификации

Если в автомате зафиксировано единственное начальное состояние, то автомат называется *инициальным*. Инициальные автоматы обычно используются для описания поведения систем, обладающих надежным сигналом сброса из любого состояния в некоторое (начальное) состояние. Инициальный автомат есть пятёрка $S = (S, I, O, h_S, s_0)$, где s_0 – выделенное начальное состояние. Если любое состояние автомата может быть выбрано в качестве начального, то автомат называется *неинициальным*.

Для автомата S допустимая последовательность пар входной символ/выходной символ i_1/o_1, i_2/o_2, …, i_n/o_n в состоянии s_1 называется *входо-выходной последовательностью* в состоянии s_1, если в S существует соответствующая последовательность переходов (s_1, i_1, o_1, s_2), (s_2, i_2, o_2, s_3), …, (s_n, i_n, o_n, s_{n+1}). При этом $\alpha = i_1, i_2, \ldots, i_n$ есть *входная последовательность*, $\gamma = o_1, o_2, \ldots,$

o_n – соответствующая *выходная последовательност*ь, а входо-выходная последовательность обозначается α/γ. Для входо-выходной последовательности α/γ входную последовательность α иногда называют *входной проекцией*, а соответствующую выходную последовательность γ – (*выходной*) *реакцией*. Отношение поведения полностью определенного автомата обычным образом распространяется на входо-выходные последовательности: если автомат в состоянии s может выдать выходную реакцию γ на входную последовательность α и перейти в состояние s', то $(s, \alpha, \gamma, s') \in h_S$. Под длиной входной (выходной) последовательности понимается количество входных (выходных) символов, входящих в последовательность.

Пусть S – полностью определенный автомат. *Префиксом* последовательности $\alpha = i_1, i_2, \ldots, i_n$ называется начальный отрезок данной последовательности, т.е. последовательность $\alpha' = i_1, i_2, \ldots, i_k$, где $0 < k \leq n$. Последовательность α в состоянии s автомата S *покрывает* состояние s', если существует префикс α′ последовательности α, такой что $(s, \alpha', \gamma, s') \in h_S$. Если $(s, \alpha, \gamma, s') \in h_S$ в наблюдаемом автомате S, то состояние s' называется α/γ-преемником состояния s по входо-выходной последовательности α/γ. Если α/γ-преемником состояния s является пустое множество, то иногда говорят, что α/γ-преемник состояния s не существует. Вообще говоря, для последовательностей $\alpha = i_1, i_2, \ldots, i_n$ и $\gamma = o_1, o_2, \ldots, o_n$ запись α/γ является сокращением записи для входо-выходной последовательности $i_1/o_1, i_2/o_2, \ldots, i_n/o_n$. Если автомат является недетерминированным, то для входной последовательности α может существовать несколько выходных реакций $\{\gamma_1, \ldots, \gamma_n\}$, и соответственно, в некоторых (всех) состояниях существует несколько входо-выходных последовательностей с одной и той же входной проекцией. Кроме того, α/γ-преемником состояния s в ненаблюдаемом автомате может быть множество состояний, однако для наблюдаемого, в частности, детерминированного автомата, которые рассматриваются в настоящей работе, α/γ-преемник состояния либо не

существует, либо является синглетоном. Для входной последовательности α можно рассматривать α-преемник состояния s, которым является множество состояний, в которые автомат может перейти из состояния s под действием входной последовательности α. Отметим также, что в наблюдаемом автомате S α/γ-преемником подмножества состояний S' является множество α/γ-преемников состояний $s \in S'$.

Под множеством входо-выходных последовательностей инициального автомата понимается множество таких последовательностей в начальном состоянии автомата, и в ряде случаев это множество называется *языком* автомата.

Пример. Для автомата на рисунке 1.3 в состоянии “Ожидание запроса” входной последовательности α = “Запрос”,“Запрос” соответствуют выходные последовательности “SMS с кодом”,“SMS с кодом” и “E-mail с кодом”,“E-mail с кодом”, а множество α-преемников содержит состояния “Ожидание кода 1” и “Ожидание кода 2”. Однако для любой выходной последовательности $\gamma \in$ {“SMS с кодом”,“SMS с кодом”; “E-mail с кодом”,“E-mail с кодом”} начальное состояние имеет только один α/γ-преемник.

В ряде литературных источников входная последовательность называется *адаптивной* [49], если следующий входной символ зависит от реакции автомата на предыдущие входные символы. Адаптивная последовательность обычно представляется *тестовым примером* (над входным алфавитом I и выходным алфавитом O) [38, 65], т.е. инициальным автоматом с ациклическим графом переходов, в каждом состоянии которого либо определен переход только по одному входному символу со всеми выходными символами, либо не определён ни один переход. Состояние, из которого нет переходов, будем называть *тупиковым* или *терминальным*. Высота адаптивного тестового примера равна длине максимальной входной последовательности, которая соответствует переходу из начального состояния в терминальное. Заметим, что, если множество I содержит больше одного входного символа, то тестовый пример над входным алфавитом I и выходным алфавитом O является частичным автоматом.

Пример. На рисунке 1.4 представлен тестовый пример для системы аутентификации (Рисунок 1.3), в соответствии с которым сначала будет подан входной символ “Запрос”. Далее, в случае получения реакции “E-mail с кодом” будет также подано следующее воздействие “Код”, в то время как при получении реакции “SMS с кодом” тестовый пример перейдет в тупиковое состояние и подача адаптивной входной последовательности завершится.

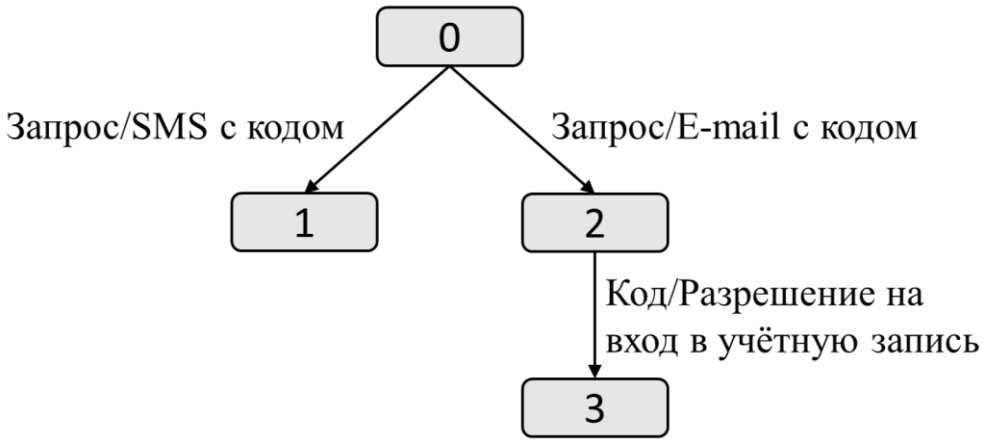

Рисунок 1.4 – Тестовый пример для системы аутентификации

1.2 Временные автоматы

Временные автоматы, или автоматы с одной временной переменной, являются, с одной стороны, расширением конечного автомата, а с другой стороны, ограничением модели временного полуавтомата с множеством временных переменных (англ. ‘timed automaton’ [25]). Как показано в работах [25, 32, 75], многие задачи анализа и синтеза для временных полуавтоматов являются алгоритмически неразрешимыми, и модель конечного автомата с одной временной переменной в ряде случаев оказывается «хорошим компромиссом» между сложностью и выразительностью модели, описывая поведение систем с временными аспектами.

Кроме дискретных множеств состояний, входных и выходных символов, временной автомат принимает во внимание временные аспекты, влияющие на поведение системы, и соответственно, содержит временную переменную, которая может принимать неотрицательные рациональные или целочисленные значения.

Временная переменная постоянно увеличивает своё значение подобно секундомеру, ведущему отсчёт времени, и сбрасывается в ноль при выполнении автоматом перехода (механизм сброса времени будет подробнее рассмотрен при формальном описании модели). Для расширения конечного автомата временными аспектами существует несколько подходов. Временной автомат может содержать временные ограничения [27, 35], таймауты [9, 27, 43] и выходные задержки [45, 50, 60]. Таким образом, поведение временного автомата определяется не только текущим состоянием и входным символом, но и значением временной переменной. Далее будет определён тип временного автомата, содержащий все три вида расширения и его частные случаи, также используемые на практике.

Под *автоматом с таймаутами и временными ограничениями* понимается пятерка $\mathsf{S} = (S, I, O, \lambda_{\mathsf{S}}, \Delta_{\mathsf{S}})$, где S – конечное непустое множество состояний, I – входной алфавит, O – выходной алфавит, $I \cap O = \varnothing$, $\lambda_{\mathsf{S}} \subseteq (S \times I \times O \times S \times \Pi \times Z)$ – отношение переходов, $\Delta_{\mathsf{S}}: S \to S \times (N \cup \{\infty\})$ – функция *таймаута*, определяющая для каждого состояния максимальное время ожидания входного символа (входной таймаут). В этих определениях N – множество натуральных чисел, Z – множество целых неотрицательных чисел, Π – множество входных временных интервалов из промежутка $[0; T)$ вида $\langle a, b\rangle$, где $\langle \in \{[, (\}, \rangle \in \{],)\}$, $a, b \in Z$, а T есть величина максимального таймаута в автомате.

Таким образом, *кортеж* (s, i, o, s', g, d) описывает переход из состояния s в состояние s' под действием входного символа i, поступившего на автомат, временная переменная которого в этот момент имела значение t, $t \in g$, с выдачей выходного символа o через d тактов времени после поступления входного символа. Иногда время d обработки входного символа называют *выходным таймаутом*. Временная переменная измеряет время, проведённое автоматом в текущем состоянии до перехода по входному символу или таймауту, и время обработки входного воздействия между поступлением входного символа и выдачей выходной реакции. Таким образом, временная переменная сбрасывается в 0 при поступлении входного символа и при выдаче выходной реакции. Если в некотором состоянии

автомата входной сигнал не поступает до достижения временной переменной входного таймаута в текущем состоянии, то автомат изменяет своё состояние в соответствии с функцией таймаута. Переход по таймауту выполняется мгновенно и «сбрасывает» значение временной переменной в 0. Например, если $\Delta_S(s) = (s', T)$ и в состоянии s в течение T единиц времени на автомат не было подано ни одного входного символа, то автомат переходит в состояние s'. Если $s = s'$, то после достижения таймаута в состоянии s отсчет времени начинается с 0. Отметим, что таймаут не выполняется между подачей входного символа и ожиданием соответствующей реакции. Если $\Delta_S(s) = (s', \infty)$, то автомат может ожидать входной символ в состоянии s бесконечно долго. Для временного автомата S через B_S будем обозначать наибольшую конечную границу для входных временных интервалов, которая совпадает с максимальной величиной таймаутов.

Временной автомат называется *полностью определённым по входным символам*, если для любой пары $(s, i) \in S \times I$, то есть для любого входного символа i, поступающего на вход автомата в состоянии s, существует кортеж $(s, i, o, s', g, d) \in \lambda_S$. Временной автомат называется *полностью определённым*, если автомат полностью определён по входным символам, и для каждой пары $(s, i) \in S \times I$ объединение всех временных интервалов g из переходов (s, i, o, s', g, d) в состоянии s по входному символу i равно $[0; T)$, где $\Delta_S(s) = (s', T)$. Аналогично классическим автоматам, частичный автомат может быть доопределён до полностью определённого различными способами в зависимости от интерпретации неопределенных переходов, однако вопросы доопределения остаются за пределами нашей работы. Если для любых двух кортежей $(s, i, o_1, s_1, g_1, d_1)$, $(s, i, o_2, s_2, g_2, d_2) \in \lambda_S$ справедливо, что $g_1 \cap g_2 = \varnothing$, то временной автомат называется *детерминированным*, иначе – *недетерминированным*.

Полностью определённый недетерминированный временной автомат называется *наблюдаемым*, если для каждой пары переходов (s, i, o, s_1, g_1, d), $(s, i, o, s_2, g_2, d) \in \lambda_S$, такой что $g_1 \cap g_2 \neq \varnothing$, справедливо, что $s_1 = s_2$, т.е. следующее состояние определяется единственным образом.

Для каждого состояния s временного автомата введём функцию $time(s, t) = s'$, которая определяет, в каком состоянии s' находится автомат через t тактов времени, при условии, что входной символ не был подан в течение этого времени.

Временным входным символом называется пара (i, t), где i – символ входного алфавита, t – время поступления входного символа, отсчитываемое от момента выдачи автоматом последнего выходного символа или начала эксперимента для первого входного символа. *Временным выходным символом* называется пара (o, d), где o – символ выходного алфавита, d – число единиц времени между подачей входного символа и выдачей выходного символа (выходной таймаут). Для нахождения реакции временного автомата на временной входной символ (i, t) в состоянии s на первом шаге определяется состояние $s' = time(s, t)$, в котором находится автомат в момент времени t. В состояние s' автомат переходит по цепочке таймаутов $\Delta_S(s) = (s_1, T_1)$, $\Delta_S(s_1) = (s_2, T_2)$, …, $\Delta_S(s_{n-1}) = (s', T_n)$, такой что сумма таймаутов $\Sigma = T_1 + T_2 + \ldots + T_n$ не превышает значения t и $t - \Sigma < T_{n+1}$, где $\Delta_S(s') = (s'', T_{n+1})$, т.е таймаут в состоянии s' не будет достигнут. Далее автомат выполняет переход (s', i, o, s'', g, d), где $t - \Sigma \in g$, и, соответственно, имеет место входо-выходная пара $(i, t) / (o, d)$ в состоянии s и следующее состояние s''. Вычитание величины Σ из времени поступления входного символа отражает тот факт, что отсчёт времени начинается с 0 при выполнении автоматом перехода по таймауту. Время поступления t' следующего входного символа (i', t'), отсчитывается от момента выдачи автоматом последнего выходного символа o. Отметим, что временная переменная автомата может сбрасываться таймаутами между выдачей выходного символа и поступлением следующего входного символа и, таким образом, время поступления входного символа является глобальным и отсчитывается во внешней среде независимо от внутренних часов автомата.

Для временного автомата S и временной входной последовательности $\alpha = (i_1, t_1), (i_2, t_2) \ldots, (i_n, t_n)$ соответствующая временная выходная последовательность $\gamma = (o_1, d_1), (o_2, d_2) \ldots, (o_n, d_n)$ в состоянии s называется (*выходной*) *реакцие*й. При

этом последовательность пар «временной входной символ / временной выходной символ» называется *временной входо-выходной* последовательностью и обозначается α/γ. Реакция временного автомата на временную входную последовательность в состоянии s вычисляется итеративно, начиная с состояния s, как было показано выше. В случае, когда автомат является недетерминированным, для временной входной последовательности α может существовать несколько выходных реакций $\gamma_1, \ldots, \gamma_n$, которые являются множеством временных выходных последовательностей. Понятия преемников по временным входо-выходным последовательностям вводятся аналогично таковым для классических конечных автоматов.

В ряде работ и приложений рассматриваются автоматы только с временными ограничениями или только с таймаутами. Под *автоматом с временными ограничениями* понимается четвёрка $\mathsf{S} = (S, I, O, \lambda_\mathsf{S})$, где S – конечное непустое множество состояний, I – входной алфавит, O – выходной алфавит, $\lambda_\mathsf{S} \subseteq (S \times I \times O \times S \times \Pi \times Z)$ – отношение переходов, Π – множество интервалов из промежутка $[0; \infty)$ вида $\langle a, b \rangle$, где $\langle \in \{[, (\}, \rangle \in \{],)\}$, и Z – множество целых неотрицательных чисел. Соответственно, *кортеж* (s, i, o, s', g, d) описывает переход из состояния s в состояние s' под действием входного символа i, поступившего в момент времени t, $t \in g$, после перехода автомата в текущее состояние с выдачей выходного символа o через d тактов времени после поступления входного символа. Полнота (частичность) и детерминизм (недетерминизм) для автомата с временными ограничениями определяются аналогично временному автомату общего вида, полагая что отсутствие таймаутов идентично бесконечными таймаутам.

Автоматом с таймаутами [45] называется пятёрка $\mathsf{S} = (S, I, O, h_\mathsf{S}, \Delta_\mathsf{S})$, где S – конечное непустое множество *состояний*, I – *входной* алфавит, O – *выходной* алфавит, $h_\mathsf{S} \subseteq (S \times I \times O \times S \times Z)$ – *отношение переходов*, Z – множество целых неотрицательных чисел, Δ_S: $S \to S \times (N \cup \{\infty\})$ – функция *таймаута*, определяющая для каждого состояния максимальное время ожидания входного символа, N – множество натуральных чисел. Аналогично автоматам с таймаутами

и временными ограничениями, если в некотором состоянии автомата входной сигнал не поступает в течение определенного времени (таймаута), то автомат может изменить своё состояние. Если в автомате есть переход (s, i, o, s', d) и входной символ будет подан в состоянии s менее чем через T единиц времени, где $\Delta_S(s) = (s', T)$, то автомат перейдет в состояние s' и через d единиц времени выдаст выходной символ. Полнота (частичность) и детерминизм (недетерминизм) для автомата с таймаутами определяются аналогично классическому конечному автомату.

Отметим, что для частных случаев временного автомата понятия входной и входной временной последовательностей вводятся аналогично общему случаю.

Пример. На рисунке 1.5 изображена система аутентификации (рисунок 1.2), расширенная таймаутами (*а*) и временными ограничениями (*b*). В этом случае, система ожидает код в течении 5 минут, после чего сбрасывает аутентификацию и возвращается в состояние “Ожидание запроса”.

а)

б)

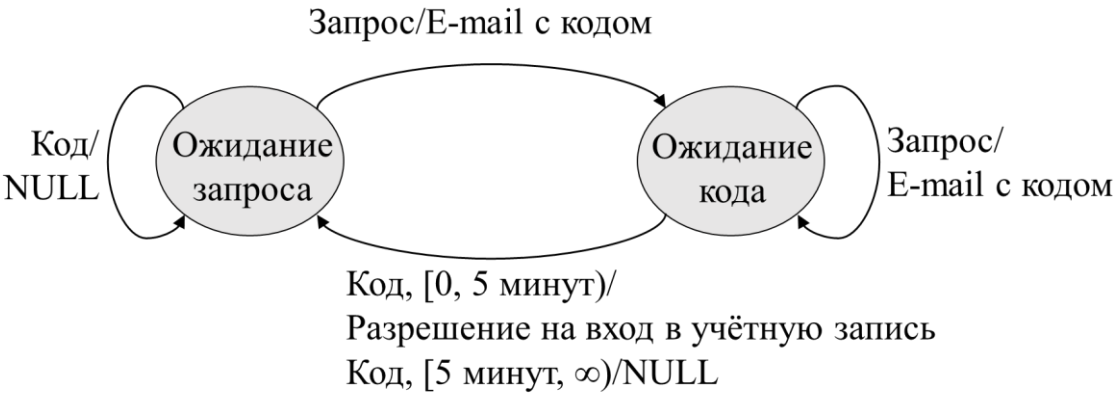

Рисунок 1.5 – Система аутентификации с таймаутами (*а*) и с временными ограничениями (*б*)

1.3 Отношения конформности между (временными) автоматами

При анализе и синтезе систем, поведение которых описано конечным автоматом, используют подходящие отношения соответствия (или конформности). Далее мы представляем отношения конформности для классических и временных автоматов, используемые в настоящей работе.

Состояния s и p полностью определённых детерминированных (временных) автоматов S и P называются *эквивалентными* ($s \cong p$), если реакции автоматов в этих состояниях совпадают на любую (временную) входную последовательность [27]. В таком случае, говорят, что поведение автоматов S и P в состояниях s и p совпадает. Если состояния s и p не являются эквивалентными, то они называются *различимыми*, а входная последовательность, реакции на которую в этих состояниях не совпадают, называется *различающей последовательностью* автоматов в состояниях s и p. Аналогичным образом определяются отношение эквивалентности и различающая последовательность для состояний одного автомата.

Инициальные (временные) автоматы S и P *эквивалентны*, если эквивалентны их начальные состояния. Неинициальные (временные) автоматы S и P *эквивалентны*, если для каждого состояния автомата S существует эквивалентное состояние в автомате P, и для каждого состояния автомата P существует эквивалентное состояние в автомате S. Полностью определенный автомат называется *приведённым*, если любые два различные его состояния различимы.

Пусть s и p – состояния полностью определённых (временных), возможно, недетерминированных автоматов S и P. Состояние p есть *редукция* [64] состояния s ($p \leq s$), если множество (временных) входо-выходных последовательностей автомата P в состоянии p содержится в множестве (временных) входо-выходных последовательностей автомата S в состоянии s. Инициальный автомат P есть редукция инициального автомата S, если отношение редукции выполняется для начальных состояний этих автоматов, т.е. $p_0 \leq s_0$.

Автомат P является подавтоматом автомата S, если $P \subseteq S$, $p_0 = s_0$ и $h_{\mathsf{P}} \subseteq h_{\mathsf{S}}$.

Конечные автоматы S и P с одинаковыми входными и выходными алфавитами *изоморфны* [4], если между множествами состояний и переходов этих автоматов можно установить взаимно однозначное соответствие, т.е. существует взаимно однозначное отображение H: $S \to P$, такое что $(s_i, i, o, s_j) \in h_S$, если и только если $(H(s_i), i, o, H(s_j)) \in h_P$. Временные автоматы S и P с одинаковыми входными и выходными алфавитами *изоморфны*, если между множествами состояний и переходов этих автоматов можно установить взаимно однозначное соответствие, т.е. существует взаимно-однозначное отображение H: $S \to P$, такое что $(s_i, i, o, s_j, g, d) \in \lambda_S$, если и только если $(H(s_i), i, o, H(s_j), g, d) \in \lambda_P$ и $\Delta_S(s_i) = (s_j, T)$, если и только если $\Delta_P(H(s_i)) = (H(s_j), T)$. Временной автомат, изоморфный заданному временному автомату, может быть получен переименованием состояний исходного автомата. Для инициальных изоморфных (временных) автоматов S и P имеет место $H(s_0) = p_0$.

1.4 Модели неисправности и проверяющие тесты

Конечно автоматные методы синтеза проверяющих тестов основаны на моделях неисправности [33], определяющих эталонное поведение тестируемой системы (спецификацию), множество проверяемых систем (обычно конечное) и отношение конформности между спецификацией и корректно реализованной проверяемой системой.

1.4.1 Модель неисправности

В настоящей работе мы рассматриваем задачу синтеза тестов для инициальных детерминированных и недетерминированных полностью определенных временных автоматов и, соответственно, вводим две модели неисправности для этих случаев. Отметим, что в данном разделе мы приводим классические для теории автоматов модели неисправности [33], в то время как вопросы их адаптации к временным автоматам будут рассмотрены в главах 4 и 5. При построении тестов для детерминированных инициальных полностью определенных автоматов мы полагаем, что тестируемая реализация является

«правильно» функционирующей, т.е. *конформна* спецификации, если автомат-реализация и автомат-спецификация эквивалентны, и, соответственно, для каждой входной последовательности реакции реализации и спецификации в начальных состояниях совпадают. Если наблюдаемые реакции проверяемой системы не совпадают с ожидаемыми или не выдаются в соответствующие моменты времени, то реализация содержит неисправность, т.е. реализация *неконформна* спецификации. В таком случае рассматривается *модель неисправности* FM^{DFSM} = <S, ≅, $\aleph_m$>, где S – инициальный полностью определённый детерминированный автомат-спецификация, ≅ – отношение эквивалентности и $\aleph_m$ – область неисправности, содержащая каждый инициальный полностью определённый детерминированный автомат с таким же входным алфавитом, как и у спецификации, и не более чем с m состояниями.

Если в спецификация присутствует опциональность, т.е. возможность конформной реализации различных вариантов функционирования, то ставится задача построения тестов для недетерминированных спецификаций. В этом случае автомат-реализация является инициальным детерминированным автоматом, в то время как инициальный автомат-спецификация может быть недетерминированным. При этом полагается, что реализация конформна спецификации, если автомат-реализация есть редукция спецификации, т. е. для каждой входной последовательности α реакция реализации в начальном состоянии содержится в множестве реакций спецификации на α. Для построения тестов по недетерминированной спецификации мы рассматриваем *модель неисправности* FM^{NDFSM} = <S, ≤, $\aleph_m$>, где S – инициальный полностью определённый недетерминированный автомат-спецификация, ≤ – отношение редукции и $\aleph_m$ – область неисправности, содержащая каждый инициальный полностью определённый детерминированный автомат с таким же входным алфавитом, как и у спецификации, и не более чем с m состояниями.

1.4.2 Проверяющие тесты

Проверяющий тест для конечного автомата представляет собой конечное множество входных последовательностей автомата-спецификации, позволяющее определить, конформна (соответствует) ли тестируемая реализация спецификации; соответственно, для временного автомата тест содержит временные входные последовательности. Под *длиной* теста понимается сумма длин всех (временных) входных последовательностей. Если тест содержит адаптивные входные последовательности, то каждая последовательность теста представляется тестовым примером. В этом случае под *длиной* теста понимается сумма высот всех тестовых примеров.

Проверяющий тест называется *полным* относительно модели неисправности $FM^{NDFSM} = <\mathsf{S}, \leq, \aleph_m>$, если для каждого автомата $\mathsf{P} \in \aleph_m$ такого что P есть редукция S, выходная реакция на каждую последовательность теста содержится в множестве выходных реакций S на эту последовательность, в то время как для каждого автомата $\mathsf{P} \in \aleph_m$ такого что P не является редукцией S, в тесте есть последовательность, выходная реакция на которую не принадлежит множеству выходных реакций S на эту последовательность. Аналогично определяется полнота теста относительно модели неисправности $FM^{DFSM} = <\mathsf{S}, \cong, \aleph_m>$, где каждая не эквивалентная автомату S реализация из $\aleph_m$ обнаруживается полным тестом, в то время как каждая конформная реализация «проходит» тест (англ. *pass* [31]).

Определение модели неисправности для инициальных детерминированных и недетерминированных временных автоматов более подробно обсуждается в главах 4 и 5 соответственно.

1.4.3 Методы синтеза тестов для конечных автоматов

В настоящем разделе мы кратко представляем известные конечно автоматные методы синтеза полных проверяющих тестов [3, 7, 31, 33, 42, 64, 86] для инициальных автоматов. Отметим, что задача синтеза тестов обычно рассматривается для связных автоматов, каждое состояние которых достижимо из

начального. В этом случае, для автомата S с начальным состоянием s_0, последовательность α называется передаточной последовательностью для состояния *s*, если *s* есть α-преемник состояния s_0.

Одним из наиболее распространенных методов синтеза тестов для инициальных детерминированных автоматов является обход графа переходов автомата-спецификации. Такой тест является полным относительно выходных ошибок и для каждого перехода спецификации содержит входную последовательность, проходящую по этому переходу.

Для того чтобы построить тест, гарантирующий проверку не только выходных реакций автомата-реализации, но и переход тестируемой системы в состояние, предписанное спецификацией, используются более сложные методы, наиболее известным из которых является W-метод [3, 31], а также его модификации: *Wp*-метод [42], *HIS*-метод [6], *UIO*-метод [86], *SC*-метод [64], *SCR*-метод [64] и другие подобные методы. W-метод и большинство его производных можно разделить на два этапа. На первом этапе устанавливается соответствие между состояниями реализации (проверяемого автомата) и спецификации, для чего автомат сначала переводится в проверяемое состояние, и затем достигнутое состояние идентифицируется. Соответствующие тестовые последовательности формируются конкатенацией передаточной (-ых) и различающей (-их) последовательностей. Если реакции реализации и спецификации на такие последовательности не совпадают, то соответствия между состояниями реализации и спецификации не существует и проверяемый автомат не эквивалентен спецификации. На втором этапе проверяется соответствие между переходами спецификации и реализации. Для этого, после достижения реализацией каждого состояния на неё подаются все возможные входные символы, и достигнутые таким образом состояния идентифицируются. Соответствующие тестовые последовательности формируются конкатенацией передаточной последовательности, входного символа и различающей (-их) последовательности (-ей). Основным отличием разновидностей W-подобных

методов является подход к идентификации состояний. Так например, W-метод использует различающее множество, содержащее для каждой пары различных состояний автомата-спецификации входную последовательность их различающую; в *DS*-методе различающее множество состоит из единственной диагностической последовательности, т.е. последовательности, реакции на которую не совпадают для любой пары различных состояний; *HIS*-метод использует идентификаторы состояний, различающие заданное состояние со всеми другими состояниями.

Для недетерминированных автоматов существуют W-подобные методы построения полных тестов относительно редукции [64], которые более подробно рассматриваются в главе 5.

1.5 Конечно автоматные абстракции

Известно [27], что поведение временных автоматов в ряде случаев можно адекватно описать при помощи конечного автомата, т.е. с использованием *конечно автоматной абстракции*. Соответственно, ряд задач анализа и синтеза для временных автоматов можно решать на основе известных подходов классической теории автоматов.

Для автомата с таймаутами и временными ограничениями соответствующая конечно автоматная абстракция может быть построена подобно методу из [27]. Заметим, что мы расширяем предложенное в этой работе описание, добавив в модель временного автомата выходные таймауты.

Пусть $\mathsf{S} = (S, I, O, \lambda_{\mathsf{S}}, \Delta_{\mathsf{S}})$ – полностью определённый, возможно, недетерминированный временной автомат и B_{S} – наибольшая конечная граница для временных входных интервалов, которая не превышает максимального конечного таймаута по определению временного автомата, в то время как D определяет наибольшую выходную задержку. Построим *конечно автоматную абстракцию временного автомата*, которая является полностью определённым конечным автоматом $\mathsf{A}_{\mathsf{S}} = (S_A, I_A, O_A, \lambda_{A\mathsf{S}})$, где $S_A \subseteq \{(s, 0), (s, (0, 1)), \ldots, (s, (B_{\mathsf{S}} - 1, B_{\mathsf{S}})), (s, B_{\mathsf{S}}),$

$(s, (B_S, \infty))$: $s \in S\}$, $I_A = I \cup \{\mathtt{I}\}$, $O_A = \{(o, 0), (o, 1), \ldots, (o, D)$: $o \in O\} \cup \{\mathtt{I}\}$, а $\mathtt{I}$ – специальный символ конечно автоматной абстракции. Для состояния (s, t_j), $t_j = 0, \ldots, B_S$, автомата A_S и входного символа i, множество λ_{AS} содержит переход $((s, t_j), i, (o, d), (s', 0))$, если и только если существует переход $(s, i, o, s', g, d) \in \lambda_S$, такой что $t_j \in g$. Для состояния (s, g_j), $g_j = (0, 1) \ldots, (B_S - 1, B_S), (B_S, \infty)$, автомата A_S и входного символа i, множество λ_{AS} содержит переход $((s, g_j), i, (o, d), (s', 0))$, если и только если существует переход $(s, i, o, s', g, d) \in \lambda_S$, такой что $g_j \subseteq g$. Переходы по специальному символу $\mathtt{I}$ отражают изменение временной переменной между вещественными значениями из открытого интервала вида (a, b) и целочисленными значениями, либо переходы по таймаутам между состояниями. Переходы $((s, n), \mathtt{I}, \mathtt{I}, (s, (n, n + 1)))$, $((s, (n, n + 1)), \mathtt{I}, \mathtt{I}, (s, n + 1)) \in \lambda_{AS}$, если и только если $n < B_S$, где $\Delta_S(s) = (s', \infty)$. Переходы $((s, n), \mathtt{I}, \mathtt{I}, (s, (n, n + 1)))$, $((s, (n, n + 1)), \mathtt{I}, \mathtt{I}, (s, n + 1))$, $((s, T - 1), \mathtt{I}, \mathtt{I}, (s, (T - 1, T))) \in \lambda_{AS}$, если и только если $n < T - 1$, где $\Delta_S(s) = (s', T)$. Переход $((s, (n - 1, n)), \mathtt{I}, \mathtt{I}, (s', 0)) \in \lambda_{AS}$, если и только если $n = T < \infty$, где $\Delta_S(s) = (s', T)$. Для каждого состояния (s, B_S) абстракции переход $((s, B_S), \mathtt{I}, \mathtt{I}, (s, (B_S, \infty))) \in \lambda_{AS}$, если $\Delta_S(s) = (s, \infty)$. Для каждого состояния $(s, (B_S, \infty))$ абстракции переход $((s, (B_S, \infty)), \mathtt{I}, \mathtt{I}, (s, (B_S, \infty))) \in \lambda_{AS}$. Таким образом, для сохранения информации о временной переменной в конечно автоматную абстракцию вносятся дополнительные абстрактные состояния, которые представляют собой пару (состояние временного автомата; значение временной переменной). Для инициального временного автомата с начальным состоянием s_0, конечно автоматная абстракция также является инициальным автоматом с начальным состоянием $(s_0, 0)$.

Поясним, как временная входная последовательность α может быть переведена в соответствующую последовательность входных символов α_{FSM} для конечно автоматной абстракции. Каждый временной входной символ (i, t) представляется последовательностью входных символов абстракции вида $\mathtt{I}, \mathtt{I}, \ldots, \mathtt{I}, i$, где число символов $\mathtt{I}$ равно числу переходов временной переменной из целого числа в интервал и обратно за время t. В то же время выходная реакция

абстракции на последовательность I, I, …, I, i, имеет вид I, I, …, I, (o, d), где число символов I во входной и выходной последовательности совпадает, а (o, d) есть реакция исходного временного автомата на (i, t). Таким образом, выходная последовательность абстракции γ_{FSM} может быть переведена в соответствующую временную выходную последовательность γ удалением всех символов I.

Для автоматов с таймаутами конечно автоматная абстракция вводится похожим способом [27], с той лишь разницей, что состояния абстракции соответствуют только целочисленным значениям временной переменной. Соответственно, для конечно автоматной абстракции $\mathsf{A_S} = (S_A, I \cup \{\mathtt{I}\}, O \cup \{\mathtt{I}\}, \lambda_{A\mathsf{S}}, s_0)$ автомата с таймаутами $\mathsf{S} = (S, I, O, \lambda_\mathsf{S}, \Delta_\mathsf{S}, s_0)$ множество состояний абстракции, порождённых состоянием s временного автомата, имеет вид $\{(s, 0), (s, 1), \ldots, (s, T-1))\}$, где $\Delta_\mathsf{S}(s) = (s', T)$.

Конечно автоматная абстракция автомата с временными ограничениями $\mathsf{S} = (S, I, O, \lambda_\mathsf{S}, s_0)$ с наибольшей конечной границей для временных входных интервалов B_S и наибольшей выходной задержкой D может быть построена как полностью определённый конечный автомат $\mathsf{A_S}(B_\mathsf{S}) = (S, I_A, O_A, \lambda_{A\mathsf{S}}, s_0)$, где $I_A = \{(i, 0), (i, (0, 1)), \ldots, (i, (B_\mathsf{S} - 1, B_\mathsf{S})), (i, B_\mathsf{S}), (i, (B_\mathsf{S}, \infty)): i \in I\}$, $O_A = \{(o, 0), (o, 1), \ldots, (o, D): o \in O\}$. Для состояния s автомата $\mathsf{A_S}(B_\mathsf{S})$ и входного символа (i, t_j), $t_j = 0, \ldots, B_\mathsf{S}$, множество $\lambda_{A\mathsf{S}}$ содержит переход $(s, (i, t_j), (o, d), s')$, если и только если существует переход $(s, i, o, s', g, d) \in \lambda_\mathsf{S}$, такой что $t_j \in g$. Для состояния s автомата $\mathsf{A_S}(B_\mathsf{S})$ и входного символа (i, g_j), $g_j = (0, 1) \ldots, (B_\mathsf{S} - 1, B_\mathsf{S}), (B_\mathsf{S}, \infty)$, множество $\lambda_{A\mathsf{S}}$ содержит переход $(s, (i, g_j), (o, d), s')$, если и только если существует переход $(s, i, o, s', g, d) \in \lambda_\mathsf{S}$, такой что $g_j \subseteq g$. Таким образом, для построения абстракции к каждому входному символу добавляются интервалы времени или целочисленные значения временной переменной. Отметим, что между состояниями автомата с временными ограничениями и его конечно автоматной абстракцией может быть установлено взаимно однозначное соответствие, что в некоторой степени упрощает построение полных проверяющих тестов для такого класса временных автоматов.

Временная входная последовательность α для автомата с временными ограничениями может быть переведена в соответствующую последовательность входных символов α_{FSM} для конечно автоматной абстракции. При этом каждый временной входной символ (i, t) представляется входным символом (i, t), если t целое, либо символом (i, (a, b)), где $t \in (a, b)$, в противном случае. Аналогично, входная последовательность конечно автоматной абстракции может быть переведена во временную входную последовательность. При этом, для входного символа (i, (a, b)) значением временной переменной соответствующего входного временного символа может быть выбрано любое значение $t \in (a, b)$. В то же время выходная реакция абстракции на входной символ имеет вид (o, d), т.е. соответствует реакции исходного временного автомата, выданной через d тактов времени после поступления входного символа. Таким образом, выходная последовательность абстракции γ_{FSM} может быть переведена в соответствующую временную выходную последовательность γ.

Отметим [27], что конечно автоматная абстракция «наследует» такие свойства временного автомата как полнота (частичность) и детерминированность (недетерминированность). Таким образом, если временной автомат S является полностью определённым (частичным), то и его конечно автоматная абстракция $\mathsf{A_S}$ является полностью определённым (частичным) автоматом. Если временной автомат S является детерминированным (недетерминированным), то и его конечно автоматная абстракция $\mathsf{A_S}$ является детерминированным (недетерминированным) автоматом.

2 Минимизация временных автоматов

В настоящей главе рассматривается задача минимизации детерминированных полностью определенных автоматов с таймаутами и временными ограничениями (в данной главе, *временной автомат*). Задача минимизации является одной из классических задач теории автоматов [4]. Минимальная форма полностью определенного детерминированного автомата единственна (с точностью до изоморфизма) и может рассматриваться как каноническая форма классического конечного автомата. В настоящей главе задача минимизации расширяется на временные детерминированные автоматы, и показывается, что две приведенные по состояниям формы временного детерминированного полностью определенного автомата не обязательно изоморфны. Соответственно, мы вводим понятие приведенной по времени формы, как временного автомата с максимально возможными временными входными интервалами и минимальными таймаутами, и показываем, что две приведенные по состояниям и времени формы временного детерминированного полностью определенного неинициального автомата изоморфны, т.е. такую приведенную форму можно рассматривать как каноническую форму временного автомата. Однако, как будет продемонстрировано в главе 4, если временной автомат является инициальным и содержит таймауты, то две приведенные по состояниям и времени формы автомата не обязательно изоморфны; более того, такие минимальные формы могут иметь различное число состояний. Результаты главы 2 опубликованы в [13, 14, 15, 17, 18, 28].

2.1 Свойства конечно автоматных абстракций

В настоящей главе рассматриваются только неинициальные детерминированные полностью определённые автоматы и, соответственно, отношение эквивалентности для таких автоматов. Предлагаемый метод минимизации временного детерминированного автомата основан на построении соответствующей конечно автоматной абстракции, и в настоящем разделе

устанавливается ряд свойств такой абстракции, необходимых для анализа возможностей минимизации временного автомата. В разделе 1.5 было показано, как временные входные и выходные последовательности для временного автомата могут быть преобразованы в соответствующие последовательности для конечно автоматной абстракции. Для временных входной и выходной последовательностей α и γ временного автомата $\mathtt{S}$ соответствующие последовательности конечно автоматной абстракции $\mathtt{A_S}$ обозначаются как α_{FSM} и γ_{FSM}. Справедливость такого преобразования отражает следующее утверждение, которое более подробно доказано в [27].

Утверждение 2.1. Временная входо-выходная последовательность α/γ существует в состоянии s временного автомата $\mathtt{S}$, если и только если входо-выходная последовательность $\alpha_{FSM}/\gamma_{FSM}$ существует в состоянии $(s, 0)$ конечно автоматной абстракции $\mathtt{A_S}$.

Доказательство. Рассмотрим состояние s временного автомата $\mathtt{S}$ и множество состояний $(s, 0)$, $(s, (0, 1))$, …, $(s, (T - 1, T))$ конечно автоматной абстракции $\mathtt{A_S}$, где $\Delta_{\mathtt{S}}(s) = (s', T)$. Каждое состояние $\mathtt{A_S}$ соответствует целочисленному моменту времени либо интервалу вида (a, b), в котором находится значение временной переменной автомата $\mathtt{S}$, а переходы по входному символу $\mathtt{I}$ отражают изменение значения временной переменной автомата $\mathtt{S}$ из целого числа в интервал вида (a, b) и обратно. Таким образом, по построению множества переходов абстракции входо-выходная пара $(i, t) / (o, d)$ существует в состоянии s автомата $\mathtt{S}$, если и только если существует входо-выходная последовательность $\mathtt{I}, \mathtt{I}, \ldots, \mathtt{I}, i / \mathtt{I}, \mathtt{I}, \ldots, \mathtt{I}, (o, d)$ в состоянии (s, g) автомата $\mathtt{A_S}$, где число символов $\mathtt{I}$ равно числу m переходов временной переменной из целого числа в интервал вида (a, b) и обратно за время $t < T$ и автомат $\mathtt{A_S}$ переходит из состояния $(s, 0)$ в состояние (s, g) при наличии последовательности из m входных символов $\mathtt{I}$. В силу того, что переходу по таймауту $\Delta_{\mathtt{S}}(s) = (s', T)$ автомата $\mathtt{S}$ соответствует переход $((s, (T - 1, T)), \mathtt{I}, \mathtt{I}, (s', 0))$ автомата $\mathtt{A_S}$, аналогичные рассуждения можно провести для состояния s' и множества $(s', 0)$, $(s', (0, 1))$, …,

(s', (T – 1, T)), а также всех последующих состояний, в которые автомат S может перейти по таймауту. В то же время, по правилам построения переходов конечно автоматной абстракции, входной временной символ (i, t) переведёт автомат S в состояние s'', если и только если входная последовательность I, I, …, I, i переводит автомат A_S в состояние (s'', 0). Далее, для пары состояний s'' и (s'', 0) можно провести аналогичные рассуждения.

Утверждение 2.1 показывает, что эквивалентность состояний временного автомата можно проверить по соответствующим состояниям конечно автоматной абстракции.

Утверждение 2.2. Состояния s_1 и s_2 временного автомата S эквивалентны, если и только если в соответствующем конечном автомате A_S эквивалентными являются состояния (s_1, 0) и (s_2, 0).

Доказательство. Пусть состояния s_1 и s_2 временного автомата S эквивалентны. Предположим, что состояния (s_1, 0) и (s_2, 0) соответствующего конечного автомата A_S не являются эквивалентными, т.е. для них существует различающая последовательность α_{FSM}. Тогда в силу утверждения 2.1, последовательность α будет различающей для состояний s_1 и s_2, что противоречит утверждению, что состояния s_1 и s_2 эквивалентны.

Пусть теперь состояния (s_1, 0) и (s_2, 0) эквивалентны в построенном конечном автомате A_S, однако состояния s_1 и s_2 временного автомата S не являются эквивалентными, т.е. для них существует различающая последовательность α. Тогда согласно утверждению 2.1, последовательность α_{FSM} будет различающей для состояний (s_1, 0) и (s_2, 0), что противоречит утверждению, что состояния (s_1, 0) и (s_2, 0) эквивалентны.

Таким образом, вывод об эквивалентности или неэквивалентности состояний временного автомата (различных временных автоматов) может быть сделан на основе свойств соответствующих состояний конечно автоматной абстракции (конечно автоматных абстракций). Отметим, что утверждения 2.1. и 2.2

справедливы также для автоматов только с таймаутами и автоматов только с временными ограничениями.

2.2 Минимизация временных автоматов по состояниям

Задача минимизации в теории автоматов заключается в синтезе приведённого конечного автомата, эквивалентного исходному неинициальному автомату, т.е. в построении автомата с заданным поведением, в котором любые два различные состояния различимы. В [4] представлен алгоритм минимизации конечных полностью определенных детерминированных автоматов, на который и опирается предложенный нами подход для минимизации временных автоматов. Отношение эквивалентности на множестве состояний полностью определенного детерминированного автомата индуцирует разбиение, которое далее называется *разбиением по отношению эквивалентности* и обозначается E. Любые два состояния, принадлежащие одному классу (блоку) разбиения E, являются эквивалентными; любые два состояния, принадлежащие различным классам разбиения E, являются различимыми. Классический метод основан на разбиении множества состояний по отношению эквивалентности, построение которого содержит следующие этапы.

На первом шаге строится разбиение по отношению 1-эквивалентности. Два состояния принадлежат одному классу 1-разбиения, если реакции автомата в этих состояниях на любой входной символ совпадают. Два состояния принадлежат различным классам 1-разбиения, если реакции автомата в этих состояниях не совпадают хотя бы на один входной символ.

Затем выполняется построение разбиения по отношению (k+1)-эквивалентности на основе разбиения по отношению k-эквивалентности. Два состояния из одного блока k-разбиения принадлежат одному блоку (k+1)-разбиения, если преемники этих состояний по любому входному символу находятся в одном классе k-разбиения. Два состояния из одного блока k-разбиения принадлежат разным блокам (k+1)-разбиения, если преемники этих состояний хотя бы по одному входному символу находятся в разных блоках k-

разбиения. Этот шаг выполняется до тех пор, пока текущее и следующее разбиения не совпадут, последнее означает, что построенное разбиение представляет собой разбиение по отношению эквивалентности.

Далее, на основе полученного разбиения строится приведённая форма автомата. Классы разбиения по отношению эквивалентности соответствуют состояниям приведённой формы, а переходы из состояния приведённой формы совпадают с таковыми для любого состояния из соответствующего класса.

Для временного автомата S приведённая по состояниям форма представляет собой временной автомат эквивалентный S, в котором любые два различные состояния различимы. Аналогично классическому алгоритму минимизации, мы строим приведённую по состояниям форму временного автомата на основе разбиения множества состояний по отношению эквивалентности. В соответствии с результатами предыдущего раздела, такое разбиение может быть получено из разбиения множества состояний конечно автоматной абстракции по отношению эквивалентности.

Алгоритм 2.1 построения приведенной по состояниям формы временного неинициального полностью определённого детерминированного автомата.

Вход: Временной полностью определённый детерминированный автомат S.

Выход: Приведённая по состояниям форма B автомата S.

Шаг 1. По временному автомату S строится соответствующая конечно автоматная абстракция A_S, для которой выполняется построение разбиения E_{FSM} множества состояний по отношению эквивалентности.

Шаг 2. Строится разбиение E множества состояний автомата S по отношению эквивалентности: состояния s_1 и s_2 автомата S принадлежат одному блоку разбиения E, если и только если состояния $(s_1, 0)$ и $(s_2, 0)$ абстракции A_S принадлежат одному блоку разбиения E_{FSM}.

Шаг 3. Строится приведенная по состояниям форма B автомата S. Входной и выходной алфавиты автомата B совпадают с таковыми для автомата S, состояния $b_1, b_2, \ldots, b_l$ автомата B соответствуют блокам $B_1, B_2, \ldots, B_l$ разбиения

E. Множество переходов λ_B автомата B формируется следующим образом. Из каждого блока B_i выбирается состояние s_i. Для состояний b_i и b_j существует кортеж $(b_i, i, o, b_j, g, d) \in \lambda_B$, если и только если существует состояние $s_j \in B_j$, такое что кортеж $(s_i, i, o, s_j, g, d) \in \lambda_S$. Для функции таймаута $\Delta_B(b_i) = (b_j, T)$, если и только если $\Delta_S(s_i) = (s_j, T)$, где $s_j \in B_j$.

Конец алгоритма.

Следующее утверждение подтверждает корректность построения приведенной по состояниям формы временного автомата по описанному выше алгоритму.

Утверждение 2.3. Пусть B – автомат, построенный для временного автомата S по алгоритму 2.1. Автомат B является детерминированным полностью определенным автоматом, и состояние b_i автомата B эквивалентно состоянию s автомата S, если и только если $s \in B_i$.

Доказательство. Согласно утверждению 2.2, состояния s_1 и s_2 временного автомата S принадлежат одному блоку разбиения E, если и только если состояния $(s_1, 0)$ и $(s_2, 0)$ конечно автоматной абстракции A_S принадлежат одному блоку разбиения E_{FSM}. Таким образом, разбиение E, полученное на втором шаге алгоритма, есть разбиение множества состояний автомата S по отношению эквивалентности.

Автомат B является полностью определенным и детерминированным автоматом, поскольку автомат S обладает этими свойствами и для каждого состояния b_i автомата B множество переходов по входным символам и таймауту «повторяет» таковые для некоторого состояния $s \in B_i$.

Покажем, что состояние $s_i \in B_i$ автомата S и состояние b_i автомата B, соответствующее блоку B_i разбиения E, эквивалентны. Все переходы из состояния b_i дублируют переходы из некоторого состояния $s_i' \in B_i$, эквивалентного любому состоянию s_i из B_i в силу того, что E – разбиение по отношению эквивалентности. В то же время, если $\Delta_S(s_i') = (s_k, T)$ и $\Delta_B(b_i) = (b_k, T)$, $T < \infty$, то $s_k \in B_k$ есть состояние, эквивалентное состоянию s_k', переходы которого

«дублируются» в состоянии b_k. Аналогичные рассуждения можно провести и для возможных преемников по таймаутам состояний s_k и b_k. Таким образом, реакции на любой временной входной символ в состояниях s_i и b_i совпадают. Поскольку переходу (s_i, i, o, s_i', g, d) исходного автомата соответствует переход (b_i, i, o, b_i', g, d) приведённой формы, где $s_i' \in B_i'$, то для состояний s_i' и b_i', и для всех их преемников по входным временным символам, можно провести аналогичные рассуждения. Таким образом, реакции автоматов B и S в состояниях b_i и s_i на любую временную входную последовательность совпадают. Покажем теперь, что состояния $s_j \in B_j$ и b_i, где $i \neq j$, не эквивалентны. Состояние $s_i \in B_i$ эквивалентно состоянию b_i согласно первой части доказательства. Состояния s_i и s_j не эквивалентны, так как находятся в различных блоках разбиения E. Таким образом, состояния s_j и b_i не эквивалентны, и справедлива следующая теорема.

Теорема 2.4. Пусть S – полностью определённый детерминированный автомат с таймаутами и временными ограничениями. Полностью определённый детерминированный автомат B, построенный по алгоритму 2.1, является приведённой по состояниям формой автомата S.

В качестве примера, на рисунке 2.1 представлены временной автомат S (слева) и его приведённая по состояниям форма (справа).

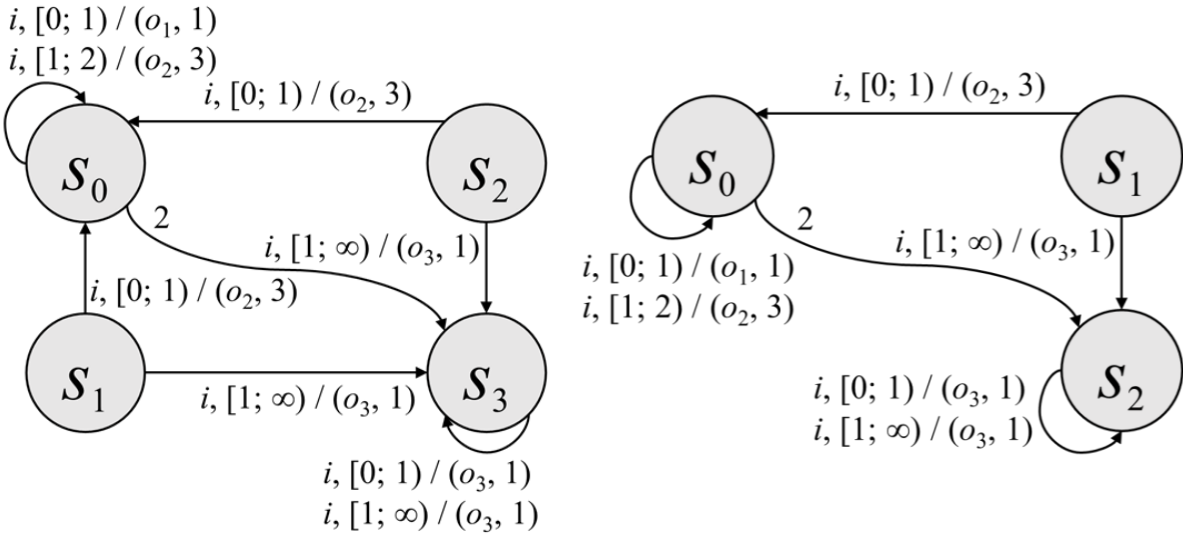

Рисунок 2.1 – Временной автомат S и его приведенная форма

2.3 Единственность минимальной формы для временного автомата

Известно, что приведённая по состояниям форма конечного полностью определенного детерминированного автомата единственна с точностью до изоморфизма [4], однако, для временного автомата могут существовать неизоморфные приведенные по состояниям формы. Рассмотрим временные автоматы на рисунке 2.2. Непосредственной проверкой можно убедиться, что оба эти автомата являются приведёнными по состояниям формами автомата S на рисунке 2.1 (слева), но не являются изоморфными.

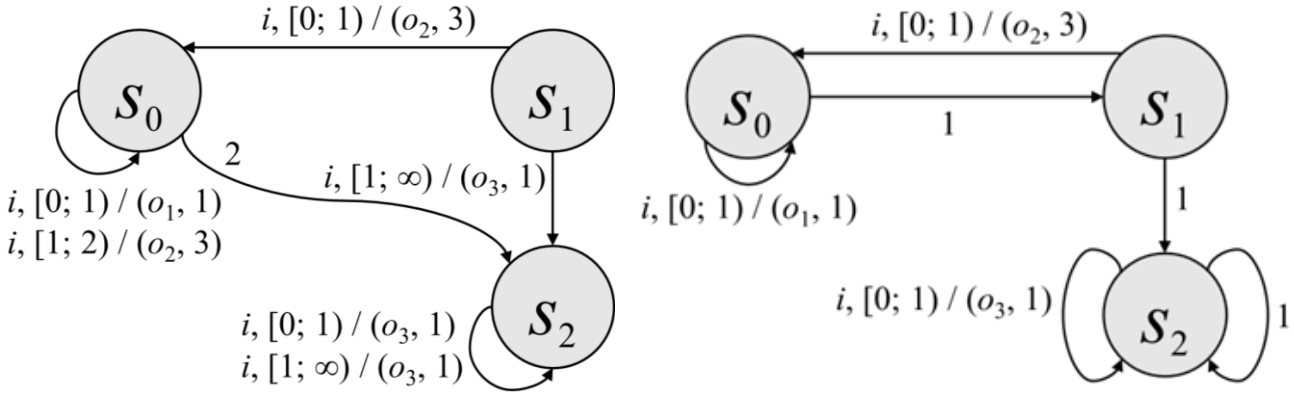

Рисунок 2.2 – Приведённые по состояниям формы автомата S

Причиной описанной выше ситуации является тот факт, что для изоморфизма двух временных автоматов должно иметь место необходимое соответствие между временными аспектами. В настоящем разделе мы показываем, как можно привести таймауты и временные ограничения автоматов с одной временной переменной для построения единственной минимальной формы временного автомата. Автомат с таймаутами и временными ограничениями S называется *приведённым по времени*, если для любых двух кортежей (s, i, o, s', g_1, d), $(s, i, o, s', g_2, d) \in \lambda_S$ справедливо, что g_1 и g_2 нельзя объединить в один интервал и для любого состояния s, такого что $\Delta_S(s) = (s', T)$, не существует состояния s'' и целого неотрицательного числа $T' < T$, таких что автомат S', полученный изменением функции таймаута состояния s автомата S на $\Delta_S(s) = (s'', T')$, эквивалентен исходному автомату S.

Если для переходов (s, i, o, s', g_1, d), $(s, i, o, s', g_2, d) \in \lambda_S$ временного автомата S интервалы g_1 и g_2 могут быть объединены в один интервал, то эти два перехода можно заменить одним переходом вида $(s, i, o, s', g_1 \cup g_2, d)$. Поиск таймаута меньшей величины может быть выполнен по конечно автоматной абстракции A_S. Для поиска минимального таймаута в состоянии s автомата S необходимо рассмотреть состояния абстракции вида (s, n), соответствующие поведению временного автомата в состоянии s в целочисленные моменты времени. Если для некоторого состояния (s, j) существует эквивалентное состояние вида $(s', 0)$, то переход по специальному входному символу $((s, (j-1, j)), I, I, (s, j))$ может быть заменён на переход $((s, (j-1, j)), I, I, (s', 0))$. Соответственно, таймаут в состоянии s автомата S может быть заменён на $\Delta_S(s) = (s', j)$. Таким образом, для каждого состояния может быть выбран минимально возможный таймаут, который не изменяет поведение временного автомата.

Для построения приведённой по времени формы допустимые переходы по входным символам с временными ограничениями объединяются, а каждый таймаут приводится к минимальному значению. Мы предлагаем следующий алгоритм построения приведённой по времени формы временного автомата.

Алгоритм 2.2 построения приведённой по времени формы автомата с таймаутами и временными ограничениями S.

Вход: Полностью определённый детерминированный временной автомат S.

Выход: Приведённая по времени форма автомата S.

Шаг 1. Каждая пара переходов (s, i, o, s', g_1, d), $(s, i, o, s', g_2, d) \in \lambda_S$, такая, что g_1 и g_2 могут быть объединены в один интервал, заменяется на переход $(s, i, o, s', g_1 \cup g_2, d)$.

Шаг 2. По исходному автомату с таймаутами и временными ограничениями S строится соответствующая конечно автоматная абстракция A_S, для которой выполняется построение разбиения по отношению эквивалентности.

Шаг 3. Для каждого состояния s автомата S, такого что $\Delta_S(s) = (s', T)$, среди состояний (s, j) автомата A_S, где $j = 1, \ldots, T-1$, определяется состояние с

минимальным j, для которого существует эквивалентное состояние вида $(s'', 0)$. Если такое состояние s'' существует, то таймаут в состоянии s заменяется на $\Delta_S(s) = (s'', j)$, иначе таймаут в состоянии s не изменяется.

Шаг 4. Для каждого состояния s автомата S, такого что $\Delta_S(s) = (s', \infty)$, определяется состояние вида $(s'', 0)$ в автомате A_S, эквивалентное состоянию $(s, (j, \infty))$. Если такое состояние s'' существует, то таймаут в состоянии s заменяется на $\Delta_S(s) = (s'', j + 1)$, иначе таймаут в состоянии s не изменяется.

Шаг 5. Для каждого состояния s, где $\Delta_S(s) = (s', T)$ и $T < \infty$, удаляются все переходы (s, i, o, s', g, d), такие что $g \cap [0, T) = \varnothing$. Если для перехода $(s, i, o, s', \langle a, b\rangle, d)$ справедливо, что $\langle a, b\rangle \cap [0, T) \neq \varnothing$ и неверно, что $\langle a, b\rangle \subseteq [0, T)$, то такой переход $(s, i, o, s', \langle a, b\rangle, d)$ заменяется на переход $(s, i, o, s', \langle a, T), d)$.

Конец алгоритма.

Следующее утверждение подтверждает корректность алгоритма 2.2.

Утверждение 2.5. Пусть P – временной автомат, построенный для полностью определённого детерминированного автомата с таймаутами и временными ограничениями S по алгоритму 2.2. Автомат P является приведённым по времени и эквивалентен автомату S.

Доказательство. Покажем, что автоматы S и P эквивалентны. Объединение переходов (s, i, o, s', g_1, d), (s, i, o, s', g_2, d) в переход $(s, i, o, s', g_1 \cup g_2, d)$ (Шаг 1) не изменяют поведение автомата. Рассмотрим процедуру замены таймаута на третьем шаге алгоритма. Пусть для состояния p временного автомата P найдено состояние p'' автомата P, такое, что существует состояние (p, j) автомата A_P, эквивалентное состоянию $(p'', 0)$ автомата A_P. Здесь $\Delta_P(p) = (p', T)$ и $j < T$. Последнее означает, в силу утверждения 2.1, что реакция автомата P в состоянии p'' на любую временную входную последовательность совпадает с таковой для автомата в состоянии p в момент времени j. Таким образом, при замене функции таймаута $\Delta_P(p) = (p', T)$ в состоянии p на $\Delta_P(p) = (p'', j)$ поведение автомата P в состоянии p не меняется. Аналогично, замена таймаута на четвёртом шаге алгоритма не изменяет поведение автомата. На шаге 5 алгоритма удаляются

переходы (p, i, o, p', g, d), временные интервалы которых не пересекаются с интервалом [0, T), где $\Delta_P(p) = (p', T)$, соответственно, такие переходы не могут быть выполнены автоматом. Переходы, временные интервалы которых пересекаются с [0, T), но выходят за границу таймаута, ограничиваются сверху величиной таймаута, что также не изменяет поведение автомата.

Покажем теперь, что автомат P приведённый по времени. Допустим, что для некоторого состояния p автомата P таймаут $\Delta_P(p) = (p_1, T_1)$ может быть заменён на таймаут $\Delta_P(p) = (p_2, T_2)$, где $T_1 > T_2$, таким образом, что поведение автомата в состоянии p не изменится. Последнее означает, что в конечно автоматной абстракции A_P существует либо пара эквивалентных состояний (p, T_2) и (p_2, 0), либо (p, ($T_2 - 1$, ∞)) и (p_2, 0). Существование эквивалентных состояний (p, T_2) и (p_2, 0) невозможно по третьему шагу алгоритма. Существование эквивалентных состояний (p, ($T_2 - 1$, ∞)) и (p_2, 0) отражает тот факт, что реакция на любую входную временную последовательность автомата P в состоянии p в момент времени $T > (T_2 - 1)$ совпадает с таковой для состояния p_2 в момент времени 0. В таком случае, состояния (p, T_2) не существует по построению абстракции, однако, по шагу 4 алгоритма, таймаут был бы заменён на $\Delta_P(p) = (p_2, T_2)$. Таким образом, существование эквивалентных состояний (p, ($T_2 - 1$, ∞)) и (p_2, 0) невозможно.

В то же время, любые два перехода вида (s, i, o, s', g_1, d), (s, i, o, s', g_2, d), которые возможно объединить в один переход (s, i, o, s', $g_1 \cup g_2$, d), объединяются на шаге 1 алгоритма 2.2.

На основе полученных результатов, можно установить необходимое и достаточное условие эквивалентности временных автоматов.

Теорема 2.6. Детерминированные полностью определённые приведённые по состояниям и времени неинициальные временные автоматы S и P эквивалентны, если и только если эти автоматы изоморфны.

Необходимость. Пусть S и P – детерминированные полностью определённые приведённые по состояниям и времени временные автоматы,

которые являются эквивалентными. Покажем, что автоматы S и P изоморфны. Рассмотрим отображение H: $S \to P$, при котором $H(s)$ есть состояние автомата P, эквивалентное состоянию s. Отображение H является взаимно однозначным, поскольку автоматы S и P являются приведёнными по состояниям. Покажем, что для любой пары состояний s и $p = H(s)$, такой что $\Delta_S(s) = (s', T_s)$, $\Delta_P(p) = (p', T_p)$, справедливо, что $T_s = T_p$ и $p' = H(s')$. Допустим, что $T_p < T_s$; тогда в силу эквивалентности S и P, существует состояние s'', эквивалентное состоянию p'. А поскольку состояния s и p также являются эквивалентными, то таймаут в состоянии s может быть заменён на $\Delta_S(s) = (s'', T_s')$, где $T_s' = T_p < T_s$. Последнее невозможно, поскольку S является приведённым по времени автоматом. Пусть теперь $T_p > T_s$; тогда в силу эквивалентности S и P, существует состояние p'', эквивалентное состоянию s'. А поскольку состояния s и p также являются эквивалентными, то таймаут в состоянии p может быть заменён на $\Delta_P(p) = (p'', T_p')$, где $T_s = T_p' < T_p$. Последнее невозможно, поскольку P – приведённый по времени автомат. В силу равенства таймаутов $T_p = T_s$ и эквивалентности состояний s и p, справедливо, что состояния $s' = time(s, T_s)$ и $p' = time(p, T_p)$ эквивалентны, и, соответственно, $p' = H(s')$. Покажем далее, что для любого перехода $(s_1, i, o, s_2, g, d) \in \lambda_S$ существует переход $(p_1, i, o, p_2, g, d) \in \lambda_P$, где $p_1 = H(s_1)$ и $p_2 = H(s_2)$. Предположим обратное, пусть в автомате S существует переход (s_1, i, o, s_2, g, d), но перехода (p_1, i, o, p_2, g, d) не существует в автомате P. Поскольку состояния s и p эквиваленты, а также имеют таймауты одинаковой величины, поведение автомата P в состоянии p в интервале времени g совпадает с таковым для автомата S в состоянии s. Соответственно, в автомате P существует переход (p_1, i, o, p_2, g', d), где $g \subset g'$, ввиду того что автомат P является приведённым по времени и состояниям. Однако, последнее невозможно, поскольку поведение автомата S в состоянии s, а соответственно и автомата P в состоянии p в интервале времени g отличается от поведения в соседних интервалах по тому же входному символу ввиду того, что автоматы S и P являются приведёнными по времени.

Достаточность. Поскольку изоморфные автоматы совпадают с точностью до переименования состояний, то изоморфные автоматы эквивалентны.

Соответственно, для временных автоматов вводится определение минимальной формы, учитывающей не только минимизацию числа состояний, но и сведение временных аспектов к приведённой форме. *Минимальной формой* полностью определённого детерминированного неинициального автомата с таймаутами и временными ограничениями S называется приведённый по состояниям и времени автомат, эквивалентный S. По определению минимальной формы и согласно теореме 2.6, справедлива следующая теорема.

Теорема 2.7. Минимальная форма неинициального автомата с таймаутами и временными ограничениями единственна с точностью до изоморфизма.

Для приведённой по состояниям формы автомата S (рисунок 2.2 слева) приведённая по времени форма построена на рисунке 2.2 (справа). Как видно из рисунка 2.2, таймаут $\Delta_{\mathsf{S}}(s_0) = (s_2, 2)$ был заменён на таймаут $\Delta_{\mathsf{S}}(s_0) = (s_1, 1)$, а переход $(s_0, i, o_2, s_0, [1, 2), 3)$ удалён. В то же время, два перехода из состояния s_2 были объединены, а таймауты $\Delta_{\mathsf{S}}(s_1) = (s_1, \infty)$ и $\Delta_{\mathsf{S}}(s_2) = (s_2, \infty)$ заменены на $\Delta_{\mathsf{S}}(s_1) = (s_2, 1)$ и $\Delta_{\mathsf{S}}(s_2) = (s_2, 1)$ соответственно.

2.4 Частные случаи автоматов с таймаутами и временными ограничениями

Очевидно, что алгоритм 2.1 позволяет построить приведённую по состояниям форму для автоматов с таймаутами и автоматов с временными ограничениями, как частных случаев рассмотренного выше временного автомата. Определение приведённой по времени формы временных автоматов, предложенное в предыдущем разделе, содержит два условия, которые обеспечивают единственность минимальной формы. Определения приведённой по времени единственной формы для автоматов с временными ограничениями и автоматов с таймаутами являются частными случаями соответствующего определения для рассмотренной выше модели временного автомата.

Первое условие связано с невозможностью объединения переходов по входным символам и полностью совпадает с определением приведённой по времени единственной формы для автоматов с временными ограничениями, введённой нами в [14]. Очевидно, что приведённая по состояниям форма автомата с временными ограничениями не единственна, поскольку временные интервалы и соответствующе переходы часто могут быть объединены различными способами. Например, это продемонстрировано в эквивалентных приведённых по состояниям автоматах на рисунке 2.2. Таким образом, автомат с временными ограничениями S называется приведённым по времени, если для любых двух кортежей (s, i, o, s', g_1, d), $(s, i, o, s', g_2, d) \in \lambda_S$ интервалы g_1 и g_2 не могут быть объединены в один интервал и справедливо следующее утверждение.

Утверждение 2.8. Детерминированные полностью определенные приведённые по времени и состояниям автоматы с временными ограничениями S и P эквивалентны, если и только если эти автоматы изоморфны.

Рассмотрим далее два эквивалентных приведенных по состояниям автомата с таймаутами (рисунок 2.3).

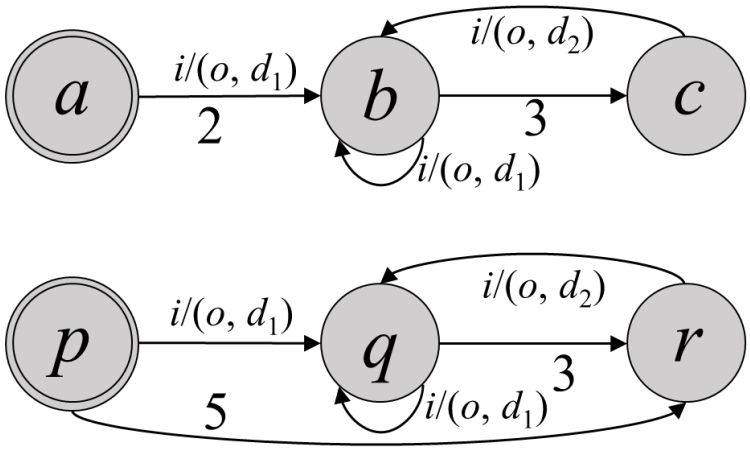

Рисунок 2.3 – Два эквивалентных приведённых, но неизоморфных автомата с таймаутами

Этот пример иллюстрирует, что приведённые по состояниям формы автомата с таймаутами не всегда являются изоморфными, поскольку некоторые цепочки переходов по таймаутам могут быть объединены в один переход, или, напротив,

один таймаут может быть разделён на несколько. В частности, в состоянии p отсутствует переход по таймауту 2, который есть в состоянии a другого автомата, эквивалентном p. Далее мы вводим класс минимальных форм для автоматов с таймаутами, которые единственны с точностью до изоморфизма.

Второе условие единственности для временных автоматов накладывается на таймауты и требует, чтобы таймаут в каждом состоянии был минимален. Это условие отличается от определения приведённых по времени автоматов с таймаутами, введённого нами в [17] и предполагавшего поиск максимального таймаута для каждого состояния, однако также может быть использовано для получения единственной формы данной модели. Таким образом, автомат с таймаутами S называется приведённым по времени, если для любого состояния s, такого что $\Delta_S(s) = (s', T)$, не существует состояния s'' и целого неотрицательного числа $T' < T$, таких что автомат S', полученный изменением функции таймаута состояния s автомата S на $\Delta_S(s) = (s'', T')$, эквивалентен исходному автомату S, и справедливо следующее утверждение.

Утверждение 2.9. Детерминированные полностью определенные приведённые по времени и состояниям автоматы с таймаутами S и P эквивалентны, если и только если эти автоматы изоморфны.

Данное утверждение доказывается аналогично теореме 2.6.

Аналогично автоматам с таймаутами и временными ограничениями, введём определение минимальной формы для других моделей с временными аспектами. *Минимальная форма* полностью определённого детерминированного автомата только с таймаутами (только с временными ограничениями) S есть приведённый по состояниям и времени автомат, эквивалентный S и, как следствие теоремы 2.7, справедлива следующая теорема.

Теорема 2.10. Минимальная форма автомата только с таймаутами (только с временными ограничениями) единственна с точностью до изоморфизма.

2.5 Экспериментальные результаты по оценке времени минимизации в зависимости от размеров автомата

Для классических конечных автоматов сложность алгоритма минимизации определяется как количество итераций для построения разбиения множества состояний по отношению эквивалентности и в худшем случае равна $n - 1$ [4], где n – число состояний минимизируемого автомата. Поскольку предлагаемый метод минимизации временных автоматов основан на построении такого разбиения для соответствующих конечно автоматных абстракций, то сложность алгоритма минимизации временных автоматов может быть оценена следующим образом. По определению, число состояний конечно автоматной абстракции не превышает величины $2B_S m$, где m – число состояний временного автомата, а B_S – наибольшая конечная граница для временных интервалов. Соответственно, для построения разбиения по отношению эквивалентности множества состояний такой абстракции потребуется не более чем $2B_S m - 1$ шагов. Таким образом, сложность алгоритма минимизации остаётся полиномиальной относительно числа состояний исходного временного автомата.

Мы также провели компьютерные эксперименты по оценке времени, затрачиваемого на выполнение задачи минимизации, в зависимости от числа состояний временного автомата. Эксперимент проводился для случайно сгенерированных временных автоматов [69] с десятью входными и выходными символами и двумя временными интервалами вида $[a, a + 2)$; число состояний изменялось от 10 до 100. Для каждого значения числа состояний генерировалась 1000 автоматов с таймаутами и временными ограничениями, для которых определялось среднее время построения минимальной по состояниям и времени формы автомата. Соответствующий график показан на рисунке 2.4.

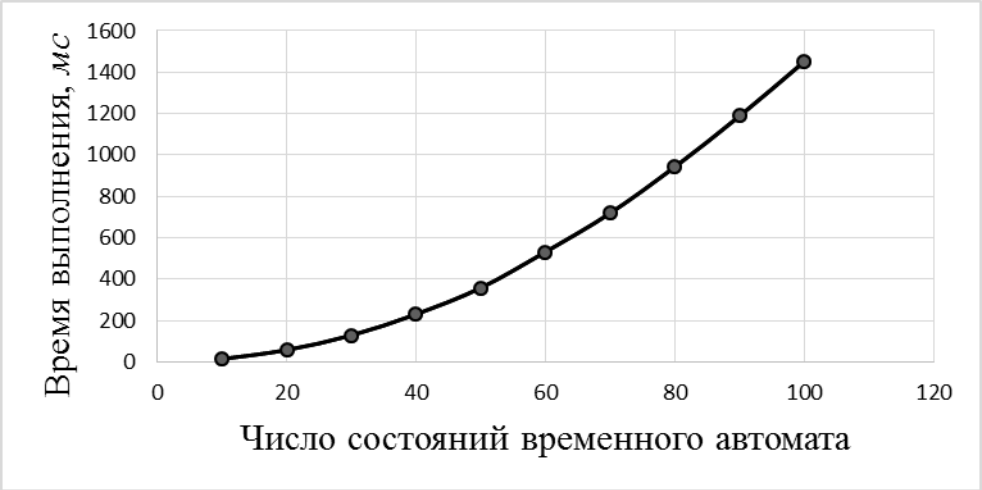

Рисунок 2.4 – Зависимость времени выполнения алгоритма минимизации от числа состояний временного автомата

Как видно из приведенных результатов, время, в течение которого выполняется алгоритм минимизации для автоматов с таймаутами и временными ограничениями, растет практически линейно относительно числа состояний автомата, когда число последних более 40 и граница B_{S} сравнима с числом состояний, несмотря на то, что теоретическая оценка пропорциональна произведению числа состояний на величину B_{S}.

2.6 Выводы по главе 2

Глава 2 содержит следующие основные результаты по минимизации неинициальных детерминированных полностью определенных автоматов с таймаутами и временными ограничениями:

1. Предложен метод построения приведённой по состояниям формы временного автомата, в которой любые два различные состояния являются различимыми. Показано, что в отличие от классических конечных автоматов, приведённая по состояниям форма временного автомата не является единственной с точностью до изоморфизма.

2. Введено понятие приведенной по времени формы временного автомата с максимально возможными временными входными интервалами и минимальными таймаутами. Показано, что две приведенные по состояниям и времени формы

временного детерминированного полностью определенного неинициального автомата изоморфны, т.е. такую приведенную форму можно рассматривать как каноническую форму временного автомата.

3. Сложность алгоритма минимизации для временных детерминированных автоматов пропорциональна произведению числа состояний на наибольшую конечную границу входных временных интервалов.

3 Композиции временных автоматов

В настоящей главе вводится операция параллельной композиции для инициальных детерминированных полностью определённых автоматов с временными ограничениями, результатом которой является автомат, описывающий поведение системы взаимодействующих в режиме диалога автоматов. Такая параллельная композиция широко используется, например, при описании совместной работы различных клиентских и серверных приложений. Вводимая нами операция является расширением параллельной композиции для классических автоматов и сводится к ней, если автоматы не имеют временных ограничений. Для классических автоматов операция параллельной композиции является замкнутой в множестве детерминированных автоматов при условии «медленной» внешней среды, т.е. при условии, что следующий водной символ поступает на композицию после того, как получен выходной сигнал на предыдущий входной символ. Мы исследуем замкнутость операции параллельной композиции в множестве детерминированных автоматов с временными ограничениями и показываем, что в отличие от классических инициальных автоматов, «медленная» внешняя среда не является достаточным условием для замкнутости множества временных детерминированных автоматов относительно операции параллельной композиции. Соответственно, мы определяем частные случаи, для которых параллельная композиция временных инициальных детерминированных автоматов является детерминированным автоматом. Результаты главы опубликованы в [21, 47, 76, 80].

3.1 Параллельная композиция классических и временных детерминированных автоматов

Модель взаимодействующих в режиме диалога классических конечных автоматов достаточно хорошо изучена и кратко описана в начале данного раздела. Множество детерминированных конечных автоматов является замкнутым относительно этой операции при наличии «медленной» внешней среды, и

существуют различные алгоритмы построения такой бинарной композиции [30, 62, 83]. В настоящем разделе мы определяем операцию параллельной композиции для временных автоматов и предлагаем соответствующий алгоритм синтеза для инициальных детерминированных полностью определенных автоматов с временными ограничениями.

3.1.1 Бинарная параллельная композиция конечных автоматов

Как отмечалось выше, в данной главе рассматриваются только инициальные автоматы. Система взаимодействующих автоматов (компонент) [9, 30, 51, 62, 63, 83] представляет собой конечное множество автоматов, которые могут обмениваться сообщениями (входными и выходными воздействиями) как с внешней средой, так и между собой. Такая структура позволяет описывать поведение сложных систем как композицию более простых в некотором смысле автоматов-компонент. В настоящей работе мы рассматриваем только параллельную композицию (далее часто просто композицию), т.е. систему, в которой автоматы (компоненты) обрабатывают входные воздействия в режиме диалога. В каждый момент времени в такой системе обрабатывается не более чем одно сообщение. Для классических автоматов операция параллельной композиции в ряде случаев является ассоциативной, поэтому в настоящем разделе мы рассматриваем только бинарную композицию. Схематично бинарная композиция (временных) автоматов изображена на рисунке 3.1, где I_1 и I_2 – внешние входные алфавиты, O_1 и O_2 – внешние выходные алфавиты, U и V – внутренние алфавиты, т.е. входной и выходной алфавиты компоненты C_2, а также, соответственно, выходной и входной алфавиты компоненты C_1. При поступлении на композицию внешнего входного воздействия компоненты поочерёдно обрабатывают входные символы до тех пор, пока не будет выдан внешний выходной символ, если диалог между компонентами заканчивается за конечное число шагов. Если диалог между компонентами при некотором входном символе является бесконечным, то говорят, что в композиции существует осцилляция, и поведение на данном входном символе обычно оставляют неопределенным.

Результатом операции композиции является автомат, описывающий работу системы взаимодействующих компонент, т.е. автомат, поведение которого совпадает с поведением композиции на любой последовательности определенных внешних входных символов. Подходы к построению параллельной композиции для классических автоматов достаточно хорошо изучены и обычно основаны на построении дерева преемников [30] или сведении операции композиции к хорошо известным операциям над соответствующими полуавтоматами [9, 51]. Например, композиция детерминированных конечных автоматов может быть построена на основе дерева преемников, вершины которого помечены парами состояний вида (s, p), где s и p – состояния первой и второй компоненты соответственно, и являются состояниями автомата-композиции [62]. Входной и выходной алфавиты композиции представляют собой объединения множеств внешних входных и выходных алфавитов компонент соответственно. Переход ((s, p), i, o, (s', p')) существует в вершине дерева (состоянии композиции), если композиция, компоненты которой находятся в состояниях s и p, выдаёт внешний выходной символ o в ответ на внешний входной символ i таким образом, что при обработке входного воздействия компоненты переходят в состояния s' и p'. До выдачи внешнего выходного сигнала компоненты композиции могут «проходить» по промежуточным состояниям, в которых компоненты принимают и выдают внутренние сигналы.

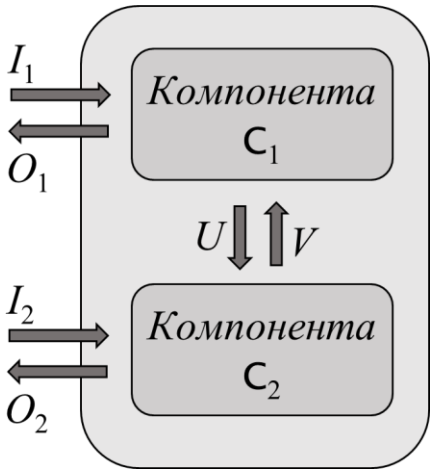

Рисунок 3.1 – Параллельная композиция двух (временных) автоматов

Известно, что если внешняя среда является «медленной» [83], то композиция детерминированных конечных автоматов является детерминированным автоматом, и, соответственно, множество классических детерминированных автоматов замкнуто относительно операции параллельной композиции. «Медленная» внешняя среда предполагает, что следующий входной символ подаётся на композицию только после получения реакции композиции на предыдущий внешний входной символ. Отметим, что при наличии осцилляции или частичных автоматов-компонент, композиция является частичным автоматом.

Далее мы вводим понятие параллельной композиции временных автоматов и определяем соответствующую операцию композиции.

3.1.2 Бинарная параллельная композиция временных автоматов

Параллельная композиция временных автоматов описывает поведение системы взаимодействующих в режиме диалога временных автоматов. Аналогично композиции классических автоматов, каждая из компонент может иметь внешние и внутренние входные и выходные символы. В то же время, в случае временных автоматов, предполагается, что компоненты обрабатывают входные воздействия не мгновенно, а с некоторой задержкой, и соответственно, бездействующая компонента «накапливает» значение временной переменной и может выполнить переход по таймауту или пересечь границу между временными интервалами, не дожидаясь воздействия от другой компоненты.

Пусть компоненты C_1 и C_2 на рисунке 3.1 являются автоматами с таймаутами и/или временными ограничениями. Опишем принцип работы такой композиции на примере обработки временного входного символа (i, t), поступившего на компоненту C_1, где время t отсчитывается от момента, когда значение временной переменной было равно 0 в каждой из компонент. В этом случае временные переменные компонент имеют значение t в момент поступления входного символа i, и компонента C_1 может отреагировать внутренним или внешним выходным символом. Если был произведён внешний выходной символ, то композиция готова принять следующее внешнее входное воздействие. В то же время, к значению

временной переменной компоненты C_2 прибавляется время, в течение которого компонента C_1 взаимодействовала с внешней средой. Если компонента C_1 выдает внутреннюю выходную реакцию, то компонента C_2 обрабатывает это воздействие в соответствии с предписанным ей поведением и производит внутреннее или внешнее выходное воздействие. Отметим, что величина задержки во внешних выходных временных символах представляет собой сумму задержек всех переходов, выполненных между подачей внешнего входного символа и выдачей внешнего выходного символа.

В качестве примера рассмотрим композицию автоматов с временными ограничениями на рисунке 3.2 и временные входные последовательности (i_1, 0.2), (i_1, 1.1) и (i_1, 0.2), (i_1, 1.9). Пусть временные переменные обеих компонент равны нулю. Если подать на композицию временной входной символ (i_1, 0.2), то компонента S произведёт внешний выходной символ o_{12} через один такт после подачи входного символа, временная переменная автомата S будет сброшена в 0, а временная переменная компоненты P достигнет значения 1.2, которое представляет собой сумму времени подачи входного символа и задержки на обработку входного символа автоматом S. После этого, при поступлении временного входного символа (i_1, 1.1), компонента S передаст второй компоненте входной символ u через один такт, который поступит на автомат P, когда временная переменная последнего достигнет значения 3.3. Далее компонента P возвращает символ v с задержкой в один такт компоненте S, которая, в свою очередь, выдаёт внешний выходной символ o_{11}. Таким образом, в ответ на внешний временной входной символ (i_1, 1.1) композиция вернула временной выходной символ (o_{11}, 3), где 3 – сумма задержек на всех выполненных переходах. Если же второй входной символ i_1 будет подан не в момент времени 1.1, а в момент 1.9, то композиция выдаст выходной символ o_{12} через шесть тактов.

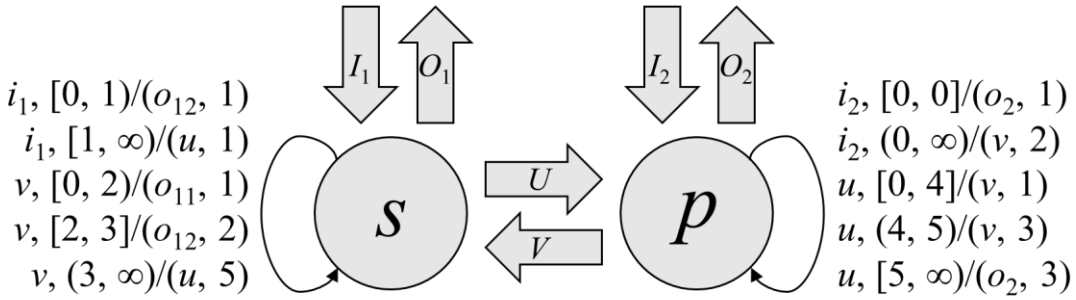

Рисунок 3.2 – Автоматы с временными ограничениями S и P

Таким образом, после обработки первого внешнего входного символа временная переменная второй компоненты имеет рациональное значение и может достичь меняющей поведение целочисленной границы временного интервала менее чем за один такт. Соответственно, для вычисления реакции композиции на второй внешний временной входной символ необходимо знать точное значение двух временных переменных, которые могут принимать рациональные значения.

Аналогично классическим автоматам, результатом операции композиции над двумя временными автоматами является временной автомат, т.е. автомат с одной временной переменной, описывающий работу системы взаимодействующих компонент. Таким образом, задача композиции временных автоматов может трактоваться как оптимизация числа временных переменных в системе. Однако, из примера на рисунке 3.2 явно следует, что количество временных переменных в модели в общем случае не может быть уменьшено, т.е. система взаимодействующих детерминированных временных автоматов не всегда может быть описана автоматом того же класса. Последнее согласуется с результатами в области временных полуавтоматов [75], где показано, что не для всякого временного полуавтомата с несколькими временными переменными существует эквивалентный полуавтомат с одной временной переменной. Тем не менее, далее мы предлагаем операцию композиции, которая возвращает детерминированный полностью определённый временной автомат-композицию в тех случаях, когда это возможно.

3.1.3 Операция параллельной композиции для автоматов с временными ограничениями

Поскольку поведение автомата с временными ограничениями может быть достаточно адекватно описано конечно автоматной абстракцией, мы далее предлагаем строить композицию двух временных автоматов на основе конечно автоматных абстракций компонент. Поведение композиции записывается в глобальный полуавтомат, в состояниях которого сохраняются состояния и значения временных переменных компонент, после чего полученный полуавтомат преобразуется в конечно автоматную абстракцию и, далее, во временной автомат [27]. Под *конечным полуавтоматом* [25, 85] понимается тройка $\mathsf{S} = (S, A, \eta_S)$, где S – конечное непустое множество *состояний*, A – конечное непустое множество *действий*, $\eta_S \subseteq (S \times A \times S)$ – *отношение переходов*, где кортеж (s, a, s') определяет переход полуавтомата из состояния s в состояние s' по действию a. Для композиции автоматов с временными ограничении S и P (Рисунок 3.3), конечно автоматные абстракции $\mathsf{A}_S(3)$ и $\mathsf{A}_P(3)$ изображены на рисунке 3.4.

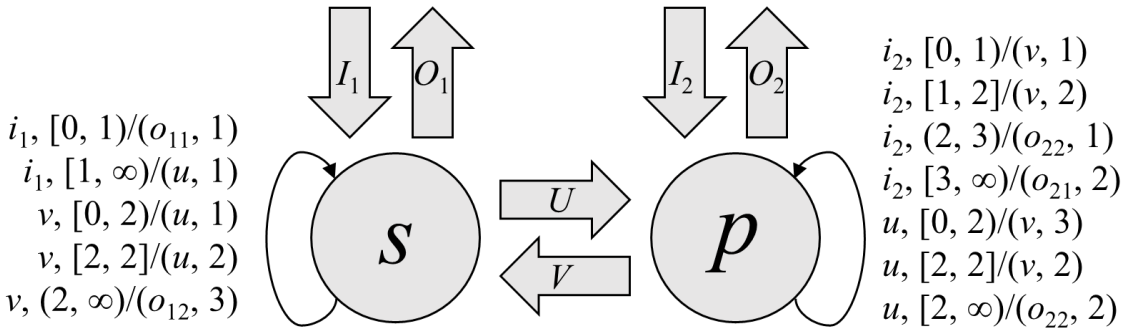

Рисунок 3.3 – Автоматы с временными ограничениями S и P

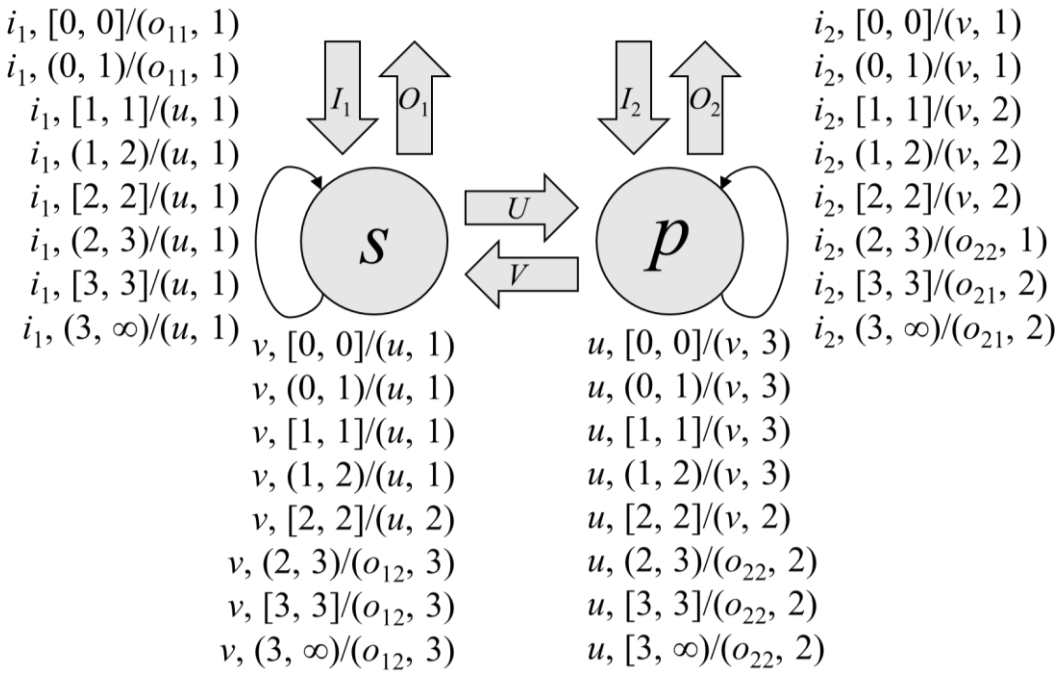

Рисунок 3.4 – Конечно автоматные абстракции $\mathsf{A_S}(3)$ и $\mathsf{A_P}(3)$

Пусть B_S – наибольшая конечная граница временных интервалов временного автомата S. Введём следующие обозначения. $G_B = \{[a, a], (a, a + 1), [B_\mathsf{S}, B_\mathsf{S}], (B_\mathsf{S}, \infty)$: $a \in N$ и $a < B_\mathsf{S}\}$, где N – множество неотрицательных целых чисел, представляет собой множество минимальных непересекающихся интервалов, в которых может находиться временная переменная в интервале $[0, B_\mathsf{S}]$, в объединении с (B_S, ∞). Для пары интервалов $g, g' \in G_B$, определим операцию $\circ$, результат которой вычисляется следующим образом. Если $g = [a, a]$ и $g' = [b, b]$, то $g \circ g' = [a + b, a + b]$; если $g = (a, b)$ и $g' = (a', b')$, то $g \circ g' = (a + a', b + b')$, где $b + \infty = \infty$; если $g = (a, a')$ и $g' = [b, b]$, то $g \circ g' = (a + b, a' + b)$. Для интервала g, границы которого суть целые неотрицательные числа, $F_B(g)$ представляет собой множество интервалов из G_B, пересекающихся с g, т.е. $F_B(g) = \{g' : g' \in G_B, g' \cap g \neq \varnothing\}$.

Алгоритм 3.1 построения параллельной композиции инициальных полностью определённых детерминированных автоматов с временными ограничениями S и P.

Вход: Инициальные детерминированные полностью определённые автоматы с временными ограничениями $\mathsf{S} = (S, I_\mathsf{S} = I_s \cup V, O_\mathsf{S} = O_s \cup U, \lambda_\mathsf{S}, s_0)$ и

P = (P, $I_P = I_p \cup U$, $O_P = O_p \cup V$, λ_P, p_0), где I_s и O_s (I_p и O_p) – внешние входной и выходной алфавиты компоненты S (P), V и U (U и V) – внутренние входной и выходной алфавиты компоненты S (P); алфавиты I_s, I_p, V, U попарно не пересекаются, $I_s \cup I_p \neq \varnothing$, $V \cup U \neq \varnothing$.

Выход: Параллельная композиция временных автоматов S и P в виде инициального полностью определённого детерминированного автомата с временными ограничениями T или одно из сообщений: «В системе имеется осцилляция», «Поведение системы не может быть описано детерминированным автоматом».

Шаг 1. Строятся конечно автоматные абстракции $A_S(B)$ и $A_P(B)$, где B – наибольшая из границ временных интервалов B_S и B_P.

Шаг 2. Формируется глобальный полуавтомат Q = (Q, {(i, g)/(r, d): $i \in I_S \cup I_P$, $r \in O_S \cup O_P$, $g \in G_B$, $d \in Z$}, q_0, λ_Q), где Z – множество целых неотрицательных чисел, и множество состояний Q содержит стабильные и промежуточные состояния. Под стабильным состоянием понимается четвёрка (s, g_s, p, g_p) $\in S \times G_B \times P \times G_B$; под промежуточным состоянием понимается пятёрка (s, g_s, p, g_p, (a, g)), где (s, g_s, p, g_p) $\in S \times G_B \times P \times G_B$, ($a$, g) $\in (U \cup V) \times G_B$. Состояние является стабильным, если и только если каждый входящий переход помечен действием (r, g)/(o, d), где $o \in O_s \cup O_p$, т.е. (o, d) является внешним временным выходным символом временного автомата S или P и соответствующим выходным символом конечно автоматной абстракции $A_S(B)$ или $A_P(B)$. В каждом стабильном состоянии определён переход по действию (i, g)/(r, d), где $i \in I_s \cup I_p$, т.е. (i, g) является внешним входным символом конечно автоматной абстракции $A_S(B)$ или $A_P(B)$. Начальное состояние q_0 = (s_0, [0, 0], p_0, [0, 0]) является стабильным. В каждом промежуточном (нестабильном) состоянии (s, g_s, p, g_p, (a, g)), определён единственный переход по действию (a, g)/(r, d), $r \in O_S \cup O_P$.

Для каждого стабильного состояния (s, [0, 0], p, g_p) глобального полуавтомата Q и перехода (s, (i, g), (r, d), s') конечно автоматной абстракции $A_S(B)$, такого что $i \in I_s$, $g \in G_B$

Если $r \in U$, **то** переход $((s, [0, 0], p, g_p), (i, g)/(r, d), (s', [0, 0], p, g_p', (r, [d, d])))$, где $g_p' \in F_B(g \circ g_p \circ [d, d])$, добавляется в Q

Иначе переход $((s, [0, 0], p, g_p), (i, g)/(r, d), (s', [0, 0], p, g_p'))$, где $g_p' \in F_B(g \circ g_p \circ [d, d])$, добавляется в Q;

Для каждого стабильного состояния $(s, g_s, p, [0, 0])$ глобального полуавтомата Q и перехода $(p, (i, g), (r, d), p')$ конечно автоматной абстракции $A_P(B)$, такого что $i \in I_p$, $g \in G_B$

Если $r \in V$, **то** переход $((s, g_s, p, [0, 0]), (i, g)/(r, d), (s, g_s', p', [0, 0], (r, [d, d])))$ где $g_s' \in F_B(g \circ g_s \circ [d, d])$, добавляется в Q;

Иначе переход $((s, g_s, p, [0, 0]), (i, g)/(r, d), (s, g_s', p', [0, 0]))$, где $g_s' \in F_B(g \circ g_s \circ [d, d])$, добавляется в Q;

Для каждого промежуточного состояния $(s, [0, 0], p, g_p, (r, [d, d]))$ глобального полуавтомата Q с входящим переходом, помеченным $(i, g)/(r, [d, d])$, определяется переход $(p, (r, g_p), (m, k), p')$ конечно автоматной абстракции $A_P(B)$;

Если $m \in V$, **то** переход $((s, [0, 0], p, g_p, (r, [d, d])), (r, [d, d])/(m, k), (s, [k, k], p', [0, 0], (m, [k, k])))$ добавляется в Q;

Иначе переход $((s, [0, 0], p, g_p, (r, [d, d])), (r, [d, d])/(m, k), (s, [k, k], p', [0, 0]))$ добавляется в Q;

Для каждого промежуточного состояния $(s, g_s, p, [0, 0], (r, [d, d]))$ глобального полуавтомата Q с входящим переходом по действию $(i, g)/(r, [d, d])$ определяется переход $(s, (r, g_s), (m, k), s')$ конечно автоматной абстракции $A_S(B)$;

Если $m \in U$, то переход $((s, g_s, p, [0, 0], (r, [d, d])), (r, [d, d])/(m, k), (s', [0, 0], p, [k, k], (m, [k, k])))$ добавляется в Q;

Иначе переход $((s, g_s, p, [0, 0], (r, [d, d])), (r, [d, d])/(m, k), (s', [0, 0], p, [k, k]))$ добавляется в Q;

Шаг 3. Если в глобальном полуавтомате Q существует цикл, помеченный только внутренними входными символами, **то** возвращается сообщение «В системе имеется осцилляция», **иначе** строится конечно автоматная абстракция A_T:

- состояния A_T соответствуют стабильным состояниям глобального полуавтомата Q;
- **для каждой последовательности** $\gamma = (i_1, g_1)/(o_1, d_1), \ldots, (i_n, g_n)/(o_n, d_n)$, которая переводит полуавтомат Q из стабильного состояния q_1 в стабильное состояние q_2 и содержит единственный внешний выходной символ (o_j, d_j), в A_T добавляется переход $(q_1, (i_1, g_1), (o_n, d), q_2)$, где $d = d_1 + \ldots + d_n$ является суммой выходных задержек, которыми помечены действия последовательности γ;

Если автомат A_T недетерминированный, **то** возвращается сообщение «Поведение системы не может быть описано детерминированным автоматом», **иначе** конечно автоматная абстракция A_T переводится во временной автомат T.

Конец алгоритма.

Таким образом, каждая последовательность переходов глобального полуавтомата между двумя стабильными состояниями соответствует последовательности переходов, выполненных парой взаимодействующих временных автоматов в ходе диалога между подачей внешнего входного символа и выдачей внешней реакции. Заметим также, что алгоритм 3.1 заканчивает работу за конечное число шагов, поскольку множества $S \times G_B \times P \times G_B$ и $(U \cup V) \times G_B$ являются конечными. Кроме того, предложенный алгоритм является расширением соответствующего алгоритма для классических конечных автоматов, т.е. справедливо следующее утверждение.

Утверждение 3.1. Результатом работы алгоритма 3.1 для полностью определённых детерминированных конечных автоматов S и P, является параллельная композиция этих автоматов при наличии «медленной» внешней среды и отсутствии осцилляций.

Доказательство. Поведение конечного автомата может быть описано соответствующим временным автоматом, для которого $\Pi = \{[0, \infty)\}$ и все выходные задержки равны нулю. Пусть S и P – конечные автоматы, а S_t и P_t – соответствующие временные автоматы, полученные добавлением интервалов $[0, \infty)$

и нулевых задержек в каждый переход автоматов S и P. Конечно автоматные абстракции таких автоматов имеют переходы только по входным символам (i, [0, 0]) и (i, (0, ∞)), реакции на которые совпадают для каждой пары «состояние, входной символ i». В таком случае, для каждого состояния q глобального полуавтомата Q, построенного для автоматов S_t и P_t на втором шаге алгоритма, и входного символа i временного автомата S или P, в Q существуют лишь действия (i, [0, 0])/(r, 0) и (i, (0, ∞))/(r, 0). Соответственно, построенный из полуавтомата Q конечный автомат A_T имеет переходы только по входным символам (i, [0, 0]) и (i, (0, ∞)), реакции на которые совпадают для каждой пары «состояние, входной символ i», и может быть преобразован во временной автомат T, для которого Π = {[0, ∞)} и все выходные задержки равны нулю. Таким образом, автомат T, построенный по алгоритму 3.1 для автоматов S и P является конечным автоматом. В то же время, процедура построения глобального полуавтомата идентична таковой для классического случая [30] и соответственно, построенный таким образом конечный автомат является параллельной композицией конечных автоматов S и P.

На рисунке 3.5 представлен фрагмент глобального полуавтомата, построенный на Шаге 2 алгоритма 3.1 для пары временных автоматов S и P (рисунок 3.3).

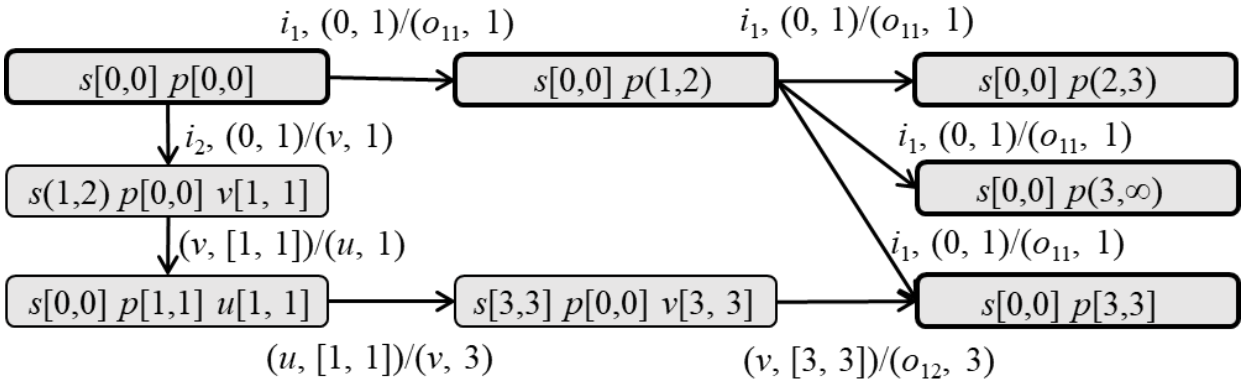

Рисунок 3.5 – Автоматы с временными ограничениями S и P

Фрагмент конечного автомата-композиции, соответствующий фрагменту глобального полуавтомата (Рисунок 3.5), изображён на рисунке 3.6.

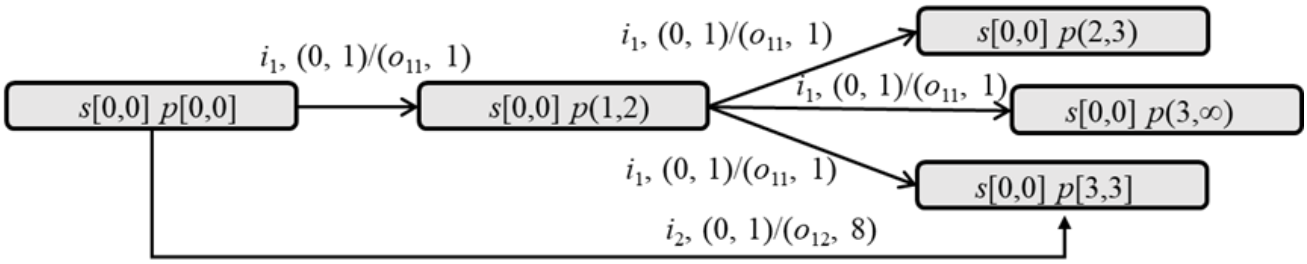

Рисунок 3.6 – фрагмент глобального полуавтомата Q композиции автоматов S и P

Отметим, что глобальный полуавтомат, фрагмент которого представлен на рисунке 3.5, и соответствующий конечный автомат на рисунке 3.6 имеют недетерминированные переходы, и, как следствие, композиция не может быть описана детерминированным временным автоматом.

Таким образом, множество детерминированных автоматов не является замкнутым относительно операции параллельной композиции, и в следующем разделе мы исследуем причины возникновения недетерминизма при построении композиции и определяем классы параллельных композиций детерминированных временных автоматов, которые могут быть описаны автоматом того же класса.

3.2 Замкнутость класса временных детерминированных автоматов относительно операции параллельной композиции

Рассмотрим параллельную композицию автоматов с временными ограничениями S и P (рисунки 3.3 - 3.6). Пусть композиция находится в состоянии (s, [0, 0], p, (1, 2)), и на компоненту S поступает входной символ i_1 в интервале (0, 1). Если компонента S выдаст внешний выходной символ, то состояние композиции изменится на одно из следующих состояний: (s, [0, 0], p, (2, 3)), (s, [0, 0], p, [3, 3]), (s, [0, 0], p, (3, ∞)). Таким образом, недетерминизм проявляется в композиции, когда значение временной переменной одной из компонент соответствует интервалу ненулевой длины. В таком случае, не всегда возможно однозначно определить реакцию соответствующей компоненты на входной символ, который также подаётся в ненулевом интервале.

В свою очередь, ненулевые интервалы появляются, когда одна из компонент может выдать внешнюю выходную реакцию в ответ на внешний входной символ, а именно, обработать внешнее воздействие в нецелочисленный момент времени без диалога со второй компонентой. В этом случае временная переменная не участвующей в обработке внешнего воздействия компоненты накапливает время из ненулевого интервала, которое не всегда может быть однозначно определено, и в состоянии глобального полуавтомата появляются интервалы длины больше единицы, которые в алгоритме 3.1 представляются несколькими состояниями с интервалами времени из G_B. Однако, если любое внешнее воздействие приводит к диалогу между компонентами, то такая ситуация не возникает, и можно сформулировать следующее утверждение.

Утверждение 3.2. Параллельная композиция автоматов с временными ограничениями S и P является детерминированным временным автоматом, если для каждого перехода $(sp, i, a, sp', g, d) \in \lambda_{\mathsf{S}} \cup \lambda_{\mathsf{P}}$, такого что i – внешний входной символ, a является внутренним выходным символом.

Доказательство. Рассмотрим инициальные детерминированные временные автоматы S и P, в которых для каждого внешнего входного символа в каждом состоянии соответствующий выходной символ является внутренним. В этом случае, каждая из компонент обязательно выполнит хотя бы один переход между поступлением внешнего входного символа и выдачей внешнего выходного символа и сбросит значение временной переменной в 0. Ввиду последнего, в момент выдачи внешнего выходного символа, значение временной переменной компоненты, выдавшей внешний выходной символ, «сбрасывается» в 0, в то время как значение временной переменной другой компоненты становится равным задержке последнего выполненного композицией перехода, т.е. имеет целочисленное значение. Поскольку в момент выдачи внешнего выходного символа значение временной переменной каждой из компонент целочисленное, то множество стабильных состояния глобального полуавтомата содержится в $S \times G_B^{int} \times P \times G_B^{int}$, где $G_B^{int} = \{[0, 0], [1, 1], \ldots, [B, B], (B, \infty)\}$.

Последнее исключает возможность недетерминизма в стабильных состояниях глобального полуавтомата, в то время как переходы из промежуточных состояний являются детерминированными по определению. Таким образом, автомат с временными ограничениями, соответствующий глобальному полуавтомату, является детерминированным.

Иными словами, если в композиции временных автоматов S и P каждый внешний входной символ приводит к диалогу между компонентами, то параллельная композиция автоматов S и P является временным детерминированным автоматом.

К данному классу, например, относятся последовательные композиции. В последовательной композиции лишь одна из компонент (головная компонента) принимает внешние входные символы, в то время как вторая компонента (хвостовая компонента) принимает входные воздействия только от первой компоненты, при этом внешние выходные реакции может выдавать лишь хвостовая компонента (рисунок 3.7). Поскольку головная компонента может выдавать лишь внутренние выходные реакции, условия утверждения 3.2 выполняются. Таким образом, множество инициальных временных детерминированных полностью определенных автоматов замкнуто относительно операции последовательной композиции.

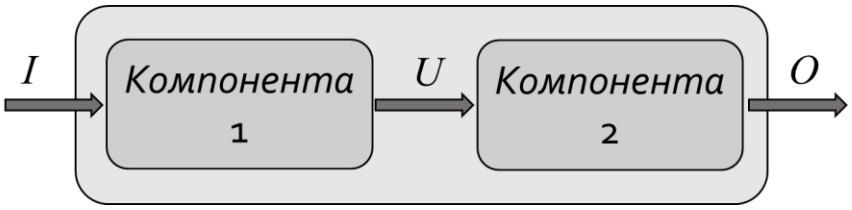

Рисунок 3.7 – Последовательная композиция

Аналогично рассмотренному выше утверждению 3.2, можно выделить другой класс детерминированных временных автоматов, для которых операция параллельной композиции является замкнутой.

Утверждение 3.3. Параллельная композиция автоматов с временными ограничениями S и P является детерминированным автоматом с временными

ограничениями, если внешние входные символы подаются на композицию только в целочисленные моменты времени.

Доказательство данного утверждения аналогично таковому для утверждения 3.2. Ввиду того, что момент поступления входного символа и выходные задержки являются целыми числами, множество стабильных состояния глобального полуавтомата содержится в $S \times G_B^{int} \times P \times G_B^{int}$, где $G_B^{int} = \{[0, 0], [1, 1], \ldots, [B, B], (B, \infty)\}$ и, более того, для промежуточных состояний вида $(s, g_s, p, g_p, (a, g))$, справедливо, что $(s, g_s, p, g_p) \in S \times G_B^{int} \times P \times G_B^{int}$ и $(a, g) \in (U \cup V) \times G_B^{int}$. Таким образом, глобальный полуавтомат и соответствующий автомат с временными ограничениями являются детерминированными.

Можно рассмотреть еще одно условие, при котором параллельная композиция временных детерминированных автоматов является детерминированным автоматом.

Утверждение 3.4. Параллельная композиция автоматов с временными ограничениями S и P является детерминированным автоматом с временными ограничениями, если для каждого перехода $(sp, i, a, sp', g, d) \in \lambda_S \cup \lambda_P$, справедливо что $d > B$, где B – максимальная из конечных границ временных интервалов B_S и B_P.

Действительно, если задержки переходов в автоматах-компонентах превышают верхнюю границу временных интервалов, то множество стабильных состояния глобального полуавтомата содержится в $S \times \{[0, 0], (B, \infty)\} \times P \times \{[0, 0], (B, \infty)\}$ и реакция композиции на внешнее входное воздействие может быть определена однозначно.

Еще одним классом композиций, поведение которых описывается детерминированным временным автоматом, является композиция контекста и встроенной компоненты (рисунок 3.8). В этом случае лишь одна из компонент (контекст) взаимодействует с внешней средой. Если поведение встроенной компоненты не зависит от времени, то значение временной переменной внешней компоненты (контекста) всегда «сбрасывается» в 0 при выдаче композицией внешнего выходного символа. Как известно, композиция контекста со

встроенной компонентой достаточно часто используется в различных приложениях [83], например, при композиции серверных и клиентских приложений.

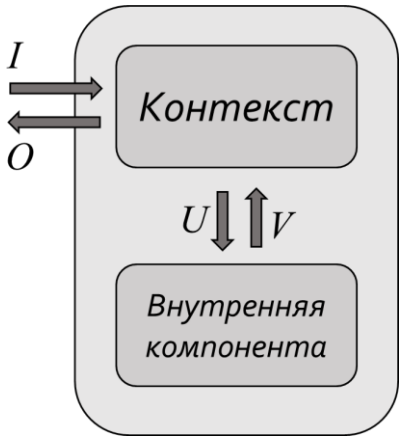

Рисунок 3.8 – Композиция с контекстом

В качестве примера рассмотрим композицию из двух автоматов с временными ограничениями S и P (рисунок 3.9), полученную из компонент композиции на рисунке 3.3 заменой переходов по входо-выходным парам из внешних входных и выходных символов на соответствующие переходы с внутренними выходными символами, и, таким образом, удовлетворяющую условиям утверждения 3.2.

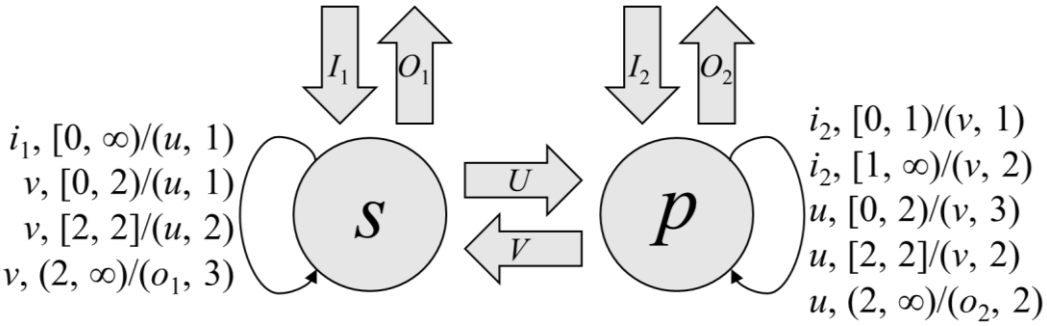

Рисунок 3.9 – Автоматы с временными ограничениями S и P

Конечно автоматные абстракции автоматов S и P представлены на рисунке 3.10. Отметим, что поскольку наибольшая из границ временных интервалов в композиции автоматов S и P равна 3, то переходы абстракций строятся до входного символа (i, (3, ∞)).

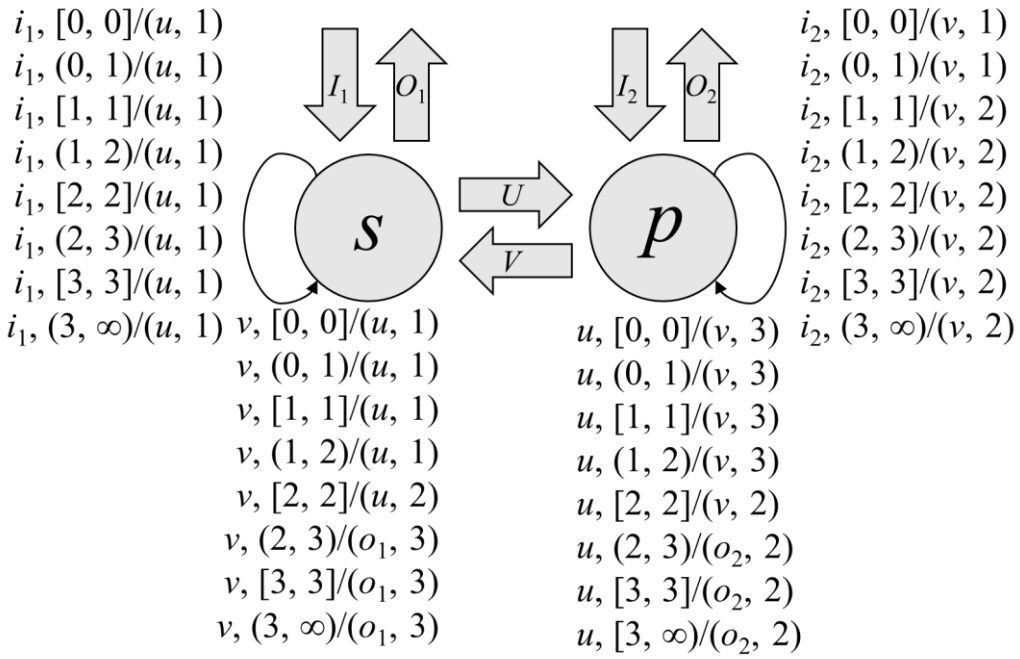

Рисунок 3.10 – Конечно автоматные абстракции $A_S(3)$ и $A_P(3)$ автоматов с временными ограничениями S и P на рисунке 3.9

Построим фрагмент глобального полуавтомата Q (рисунок 3.11), описывающий поведение композиции автоматов с временными ограничениями S и P.

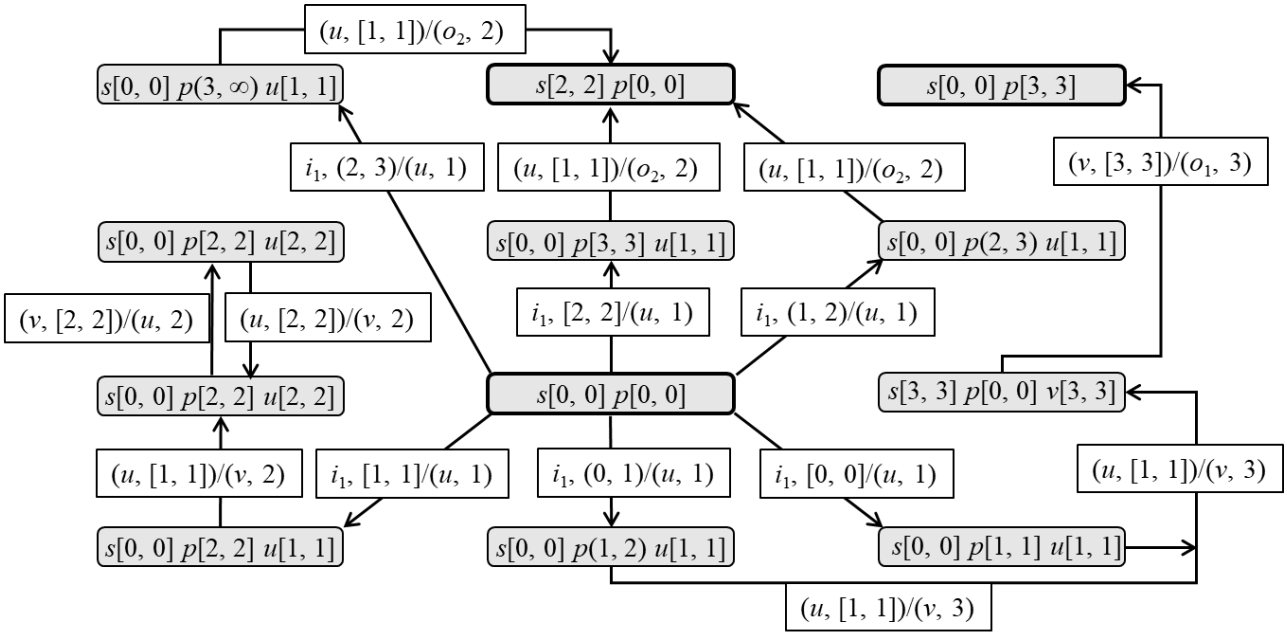

Рисунок 3.11 – Фрагмент глобального полуавтомата, построенного на втором шаге алгоритма 3.1 для временных автоматов S и P (рисунок 3.9)

Отметим, что глобальный полуавтомат имеет цикл, проходящий только через промежуточные состояния и соответствующий наличию осцилляции в композиции временных автоматов, и в таком случае, построить полностью определённый автомат-композицию невозможно. Однако, если заменить переход (*p*, *u*, *v*, *p*, [2, 2], 2) автомата P на переход (*p*, *u*, *v*, *p*, [2, 2], 3), то цикл исчезает, и соответствующий глобальному полуавтомату полностью определённый временной автомат-композиция может быть построен так, как это показано на шаге 3 алгоритма 3.1 (рисунок 3.12).

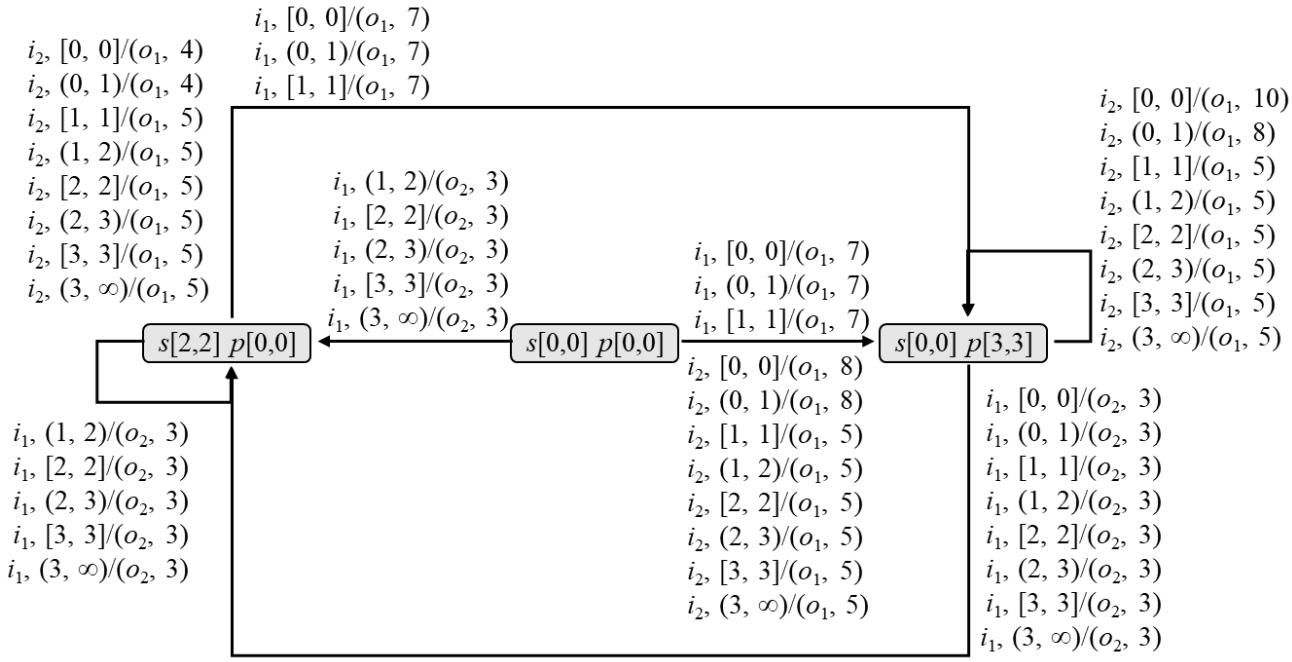

Рисунок 3.12 – Композиция автоматов S и P

Аналогично операции композиции над классическими конечными автоматами, можно доказать следующее утверждение.

Утверждение 3.5. Параллельная композиция полностью определённых автоматов с временными ограничениями S и P является полностью определённым автоматом с временными ограничениями, если соответствующий глобальный полуавтомат не имеет циклов, проходящих только через промежуточные состояния.

3.3 Выводы по главе 3

В настоящей главе была исследована задача бинарной параллельной композиции детерминированных временных автоматов и получены следующие основные результаты:

1. Операция параллельной композиции определена для инициальных детерминированных автоматов с временными ограничениями; результатом операции является временной автомат, описывающий поведение системы взаимодействующих временных автоматов. Введенная операция является расширением операции параллельной композиции над классическими конечными автоматами и сводится к ней, если компонентами композиции являются конечные автоматы без временных аспектов.

2. Показано, что в отличие от классических инициальных автоматов, «медленная» внешняя среда не является достаточным условием для описания параллельной композиции двух временных детерминированных автоматов автоматом этого же класса, т.е. множество инициальных детерминированных временных автоматов не является замкнутым относительно параллельной композиции.

3. Определены частные случаи систем взаимодействующих инициальных детерминированных временных автоматов, для которых параллельная композиция является детерминированным автоматом.

4 Построение проверяющих тестов для детерминированных временных автоматов

В настоящей главе мы предлагаем метод синтеза проверяющих тестов с гарантированной полнотой для инициальных детерминированных автоматов с таймаутами и временными ограничениями. Подобно известным конечно автоматным методам синтеза тестов, мы полагаем, что проверяемый автомат (автомат-реализация) и эталонный автомат (автомат-спецификация) являются связными, т.е. каждое состояние этих автоматов достижимо из начального. Как уже отмечалось в разделе 1.4, при построении тестов для инициальных детерминированных полностью определенных автоматов тестируемая реализация *конформна* спецификации, если автомат-реализация и автомат-спецификация эквивалентны, т.е. реакции реализации и спецификации в начальных состояниях совпадают для каждой (временной) входной последовательности. Вывод об эквивалентности двух временных автоматов может быть сделан на основе их конечно автоматных абстракций, что следует из утверждений 2.1 и 2.2 и мотивирует к использованию таких абстракций при построении тестов. Однако, последнее требует исследования свойств конечно автоматных абстракций для автоматов с таймаутами и временными ограничениями, позволяющих определить область неисправности и метод построения проверяющих тестов для временных автоматов с использованием классических конечно автоматных методов. Результаты данной главы опубликованы в [78, 81].

4.1 Методы синтеза тестов для детерминированных автоматов с таймаутами и временными ограничениями

Тесты с гарантированной полнотой для классических инициальных автоматов относительно модели неисправности $FM^{DFSM} = <\mathbf{S}, \cong, \aleph_m>$ могут быть построены обходом графа переходов, W-методом или его модификациями, которые были кратко рассмотрены в разделе 1.4.3. Большинство методов синтеза тестов для конечных автоматов разработаны в предположении, что автомат-

спецификация является полностью определенным и приведённым, т.е. любые два состояния эталонного автомата не являются эквивалентными. Отсутствие эквивалентных состояний в автомате гарантирует возможность идентификации состояний, и кроме того, приведённая форма детерминированного полностью определенного автомата единственна с точностью до изоморфизма [4]. Последнее позволяет гарантировать обнаружение каждой реализации, не изоморфной спецификации (не гомоморфной спецификации), если число состояний реализации не превышает число состояний автомата-спецификации (если число состояний реализации может превышать число состояний автомата-спецификации), т.е. позволяет проверить соответствующие свойства для конечных множеств состояний и входных символов.

В настоящем разделе мы исследуем возможность адаптации классических конечно автоматных методов синтеза тестов к инициальным детерминированным автоматам с таймаутами и временными ограничениями, определяем модель неисправности для построения проверяющих тестов с гарантированной полнотой для таких временных автоматов и предлагаем соответствующий алгоритм синтеза тестов.

4.1.1 Свойства конечно автоматных абстракций

В разделе 1.5 было показано, как поведение автомата с таймаутами и временными ограничениями может быть описано подходящей для ряда задач конечно автоматной абстракцией. Более того, в разделе 2.1, проверка эквивалентности состояний временных автоматов была сведена к проверке эквивалентности соответствующих состояний конечно автоматных абстракций. Тем не менее, как показывает следующий пример, приведенная по состояниям и времени форма инициального временного автомата не является единственной с точностью до изоморфизма. Рассмотрим два эквивалентных инициальных детерминированных автомата с таймаутами и временными ограничениями S и P на рисунке 4.1. Основываясь на результатах главы 2, можно убедиться, что каждый из этих автоматов является приведённым по состояниям и времени, а их начальные

состояния *a* и *e* эквивалентны. Однако, как можно видеть, временные автоматы S и P не являются изоморфными и более того, имеют различное число состояний.

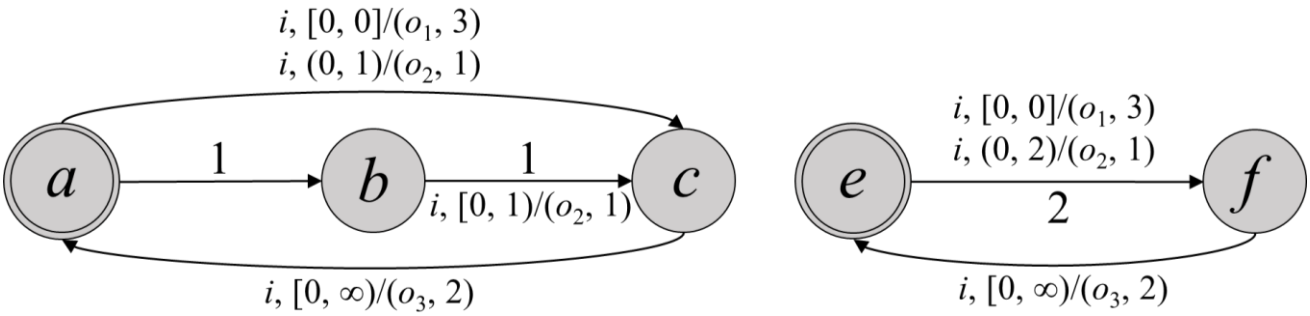

Рисунок 4.1 – Автоматы с таймаутами и временными ограничениями S (слева) и P (справа)

Таким образом, в отличие от классических конечных автоматов, приведённая форма инициального временного автомата в общем случае не является единственной с точностью до изоморфизма. Следовательно, тесты не могут быть построены W-подобными методами, которые основаны на проверке соответствия переходов автомата-спецификации и проверяемого автомата-реализации с использованием различающих и передаточных последовательностей.

В главе 1 было отмечено, что конечно автоматные абстракции «наследуют» такие свойства временного автомата как детерминированность и полнота. Более того, из определения переходов по специальной входо-выходной паре I/I следует, что конечно автоматная абстракция временного автомата «наследует» свойство связности. Однако, следует также отметить, что конечно автоматная абстракция приведённого по времени и состояниям автомата с таймаутами и временными ограничениями может не быть приведённым конечным автоматом. На рисунке 4.2 изображены конечно автоматная абстракция автомата S (рисунок 4.1 слева) и её минимальная форма. Тем не менее, минимальная форма конечно автоматной абстракции автомата P (рисунок 4.1 справа) изоморфна таковой для автомата S, т.е. минимальные формы конечно автоматных абстракций совпадают (с точностью до переименования состояний). Последнее явно следует из единственности минимальной формы для детерминированного полностью определенного конечного автомата.

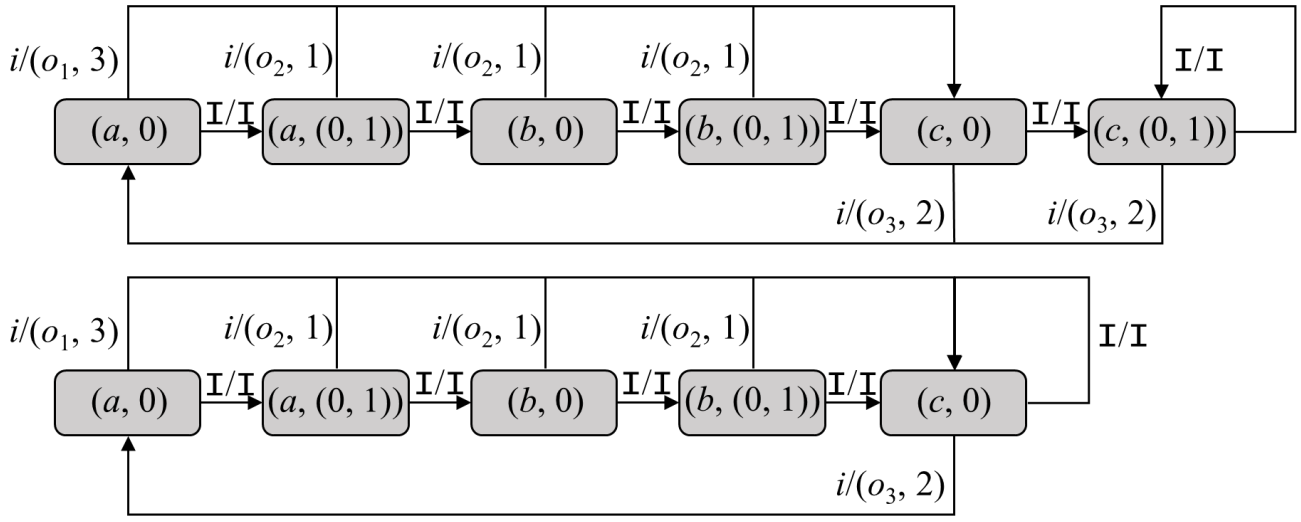

Рисунок 4.2 – Конечно автоматная абстракция A_S автомата S (вверху) и её минимальная форма (внизу)

Согласно утверждению 2.1, временная входо-выходная последовательность α/γ существует во временном автомате S, если и только если входо-выходная последовательность $\alpha_{FSM}/\gamma_{FSM}$ существует в конечно автоматной абстракции A_S. Соответственно, для инициальных временных автоматов можно сформулировать следующее утверждение, которое установлено и доказано в [27] для временных автоматов без выходных задержек.

Утверждение 4.1. Полностью определённые детерминированные инициальные автоматы с таймаутами и временными ограничениями S и P эквивалентны, если и только если эквивалентны их конечно автоматные абстракции A_S и A_P.

Далее, в силу единственности минимальной формы конечного автомата, может быть сформулировано необходимое и достаточное условие эквивалентности автоматов с таймаутами и временными ограничениями.

Утверждение 4.2. Инициальные связные полностью определённые детерминированные автоматы с таймаутами и временными ограничениями S и P эквивалентны, если и только если минимальные формы их конечно автоматных абстракций A_S и A_P изоморфны.

Напомним, что последовательность α называется различающей для состояний s и p автоматов S и P, если реакции автоматов в этих состояниях на

последовательность α отличаются. Последовательность α называется различающей для автомата S, если реакции автомата на α не совпадают для любой пары различных состояний.

Из утверждения 2.1 непосредственно следует, что различающая последовательность α существует для состояний *s* и *p* автомата с таймаутами и временными ограничениями S, если и только если соответствующая последовательность α_{FSM} является различающей для состояний (*s*, 0) и (*p*, 0) конечного автомата A_S. Аналогично, передаточная последовательность α существует для состояния *s* автомата с таймаутами и временными ограничениями S, если и только если соответствующая последовательность α_{FSM} является передаточной для состояния (*s*, 0) конечного автомата A_S. Таким образом, проверяющий тест для автоматов с таймаутами и временными ограничениями может быть построен по минимальной форме соответствующей конечно автоматной абстракции, и в следующем разделе мы определяем модель неисправности и предлагаем соответствующий алгоритм построения полного теста для временного автомата на основе такой абстракции.

4.1.2 Модель неисправности для синтеза тестов на основе временных автоматов

Пусть Q – детерминированный полностью определённый автомат с таймаутами и временными ограничениями с *n* состояниями и наибольшей конечной границей временных интервалов B_Q, конечно автоматная абстракция A_Q которого имеет m' и m состояний до и после минимизации соответственно. Поскольку проверяющий тест для временного автомата мы предлагаем строить на основе его конечно автоматной абстракции, необходимо установить соответствие между областями неисправности для временного и классического автоматов. Проверяющий тест для автомата A_Q, построенный W-методом относительно модели неисправности $FM^{DFDM} = <A_Q, \cong, \aleph_m>$, обнаружит каждую неисправную реализацию из $\aleph_m$, поведение которой описывается конечным

автоматом с тем же входным алфавитом, что и A_Q, и числом состояний не более m. Однако, множество $\aleph_m$ не является множеством конечно автоматных абстракций автоматов с таймаутами и временными ограничениями с таким же входным алфавитом, что и Q, наибольшей конечной границей временных интервалов B_Q и не более чем n состояниями. Например, внесём во временной автомат S (рисунок 4.1 слева) неисправность, не меняющую число состояний и границы B_S, но меняющую поведение автомата таким образом, что полученный автомат S′ (рисунок 4.3 сверху) не эквивалентен спецификации S. Соответствующая конечно автоматная абстракция $A_{S'}$, также изображенная на рисунке 4.3 (снизу), имеет 6 состояний и является приведённым автоматом, в то время как приведённая форма конечно автоматной абстракции спецификации S состояла из 5 состояний (рисунок 4.2 снизу). Соответственно, мы не можем гарантировать, что проверяющий тест, построенный относительно модели неисправности $<A_S, \cong, \aleph_5>$ обнаружит неэквивалентный автомат S′.

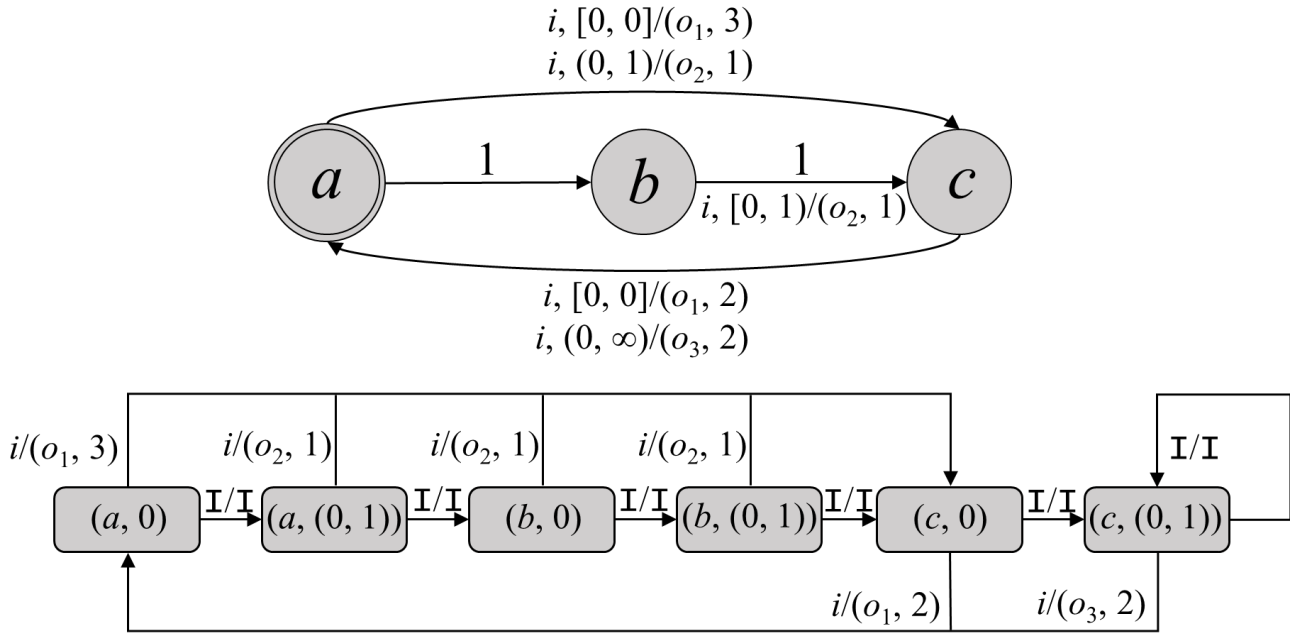

Рисунок 4.3 – Автомат с таймаутами и временными ограничениями S′ и его конечно автоматная абстракция $A_{S'}$

Таким образом, не каждый временной автомат-реализация с такими же входным алфавитом и наибольшей конечной границей временных интервалов,

как и спецификация, и числом состояний не превосходящим числа состояний временного автомата-спецификации, гарантированно может быть обнаружен тестом, построенным W-методом с заданным ограничением m на число состояний по конечно автоматной абстракции спецификации. Тем не менее, такой тест обнаруживает каждую неисправную реализацию с таймаутами и временными ограничениями, минимальная форма конечно автоматной абстракции которой имеет не больше m состояний.

В настоящей работе мы определяем модель неисправности для детерминированных полностью определённых автоматов с таймаутами и временными ограничениями следующим образом. Модель неисправности для временного автомата-спецификации есть $FM^{DTFSM} = <\mathsf{S}, \cong, \Im_m>$, где S – инициальный полностью определённый детерминированный автомат-спецификация с таймаутами и временными ограничениями, $\cong$ – отношение эквивалентности и $\Im_m$ – область неисправности, содержащая каждый инициальный полностью определённый детерминированный автомат с таймаутами и временными ограничениями с таким же входным алфавитом, как и автомат-спецификация, для которого минимальная форма конечно автоматной абстракцией имеет не более чем m состояний. В следующем разделе мы формально описываем предлагаемый алгоритм синтеза тестов относительно такой модели неисправности.

4.1.3 Синтез тестов с гарантированной полнотой для временных автоматов

Проверяющий тест для автомата с таймаутами и временными ограничениями представляет собой множество временных входных последовательностей и называется полным относительно модели неисправности $FM^{DTFSM} = <\mathsf{S}, \cong, \Im_m>$, если обнаруживает каждую неисправную реализацию из множества $\Im_m$. Отметим, что реализация считается неисправной, если наблюдаемые реакции спецификации и реализации на последовательности теста не совпадают или не выдаются в соответствующие моменты времени. Для конечно автоматной абстракции тест строится относительно модели

неисправности $FM^{DFSM} = <\mathsf{A_S}, \cong, \aleph_m>$ с использованием W-метода или одной из его модификаций и может быть преобразован в тестовые последовательности для соответствующего временного автомата согласно утверждению 2.1.

Алгоритм 4.1 синтеза тестов для детерминированных автоматов с таймаутами и временными ограничениями относительно эквивалентности.

Вход: инициальный детерминированный полностью определённый автомат-спецификация с таймаутами и временными ограничениями S.

Выход: Полный проверяющий тест *TS* относительно модели неисправности $FM^{DTFSM} = <\mathsf{S}, \cong, \Im_m>$, где m – число состояний минимальной формы конечно автоматной абстракции временного автомата S.

Шаг 1. Строится минимальная форма $\mathsf{A_S}^{min}$ конечно автоматной абстракции $\mathsf{A_S}$ временного автомата S.

Шаг 2. Строится проверяющий тест TS_A (с использованием W-метода или одной из его модификаций) относительно модели неисправности $FM^{DFSM} = <\mathsf{A_S}^{min}, \cong, \aleph_m>$, где $\aleph_m$ содержит каждый приведённый конечный автомат с таким же входным алфавитом, как $\mathsf{A_S}$, и не более чем m состояниями.

Шаг 3. Входные последовательности теста TS_A преобразуются в соответствующие временные входные последовательности и добавляются в тест *TS*.

Конец алгоритма.

Теорема 4.3. Проверяющий тест *TS*, построенный по алгоритму 4.1 для детерминированного полностью определённого автомата с таймаутами и временными ограничениями S, является полным относительно модели неисправности $<\mathsf{S}, \cong, \Im_m>$.

Доказательство. Пусть автомат с таймаутами и временным ограничениями P есть проверяемый автомат из $\Im_m$, т.е. реализация автомата-спецификации S из области неисправности, и P не эквивалентен спецификации S. Покажем, что такая реализация обнаруживается тестом *TS*. Поскольку временные автоматы S и P неэквивалентны, то их конечно автоматные абстракции $\mathsf{A_S}$ и $\mathsf{A_P}$ также

неэквивалентны (утверждение 4.1), и соответственно, минимальные формы конечных автоматов A_S и A_P неизоморфны (утверждение 4.2), и следовательно, неэквивалентны, поскольку детерминированные полностью определенные автоматы эквивалентны, если и только если они изоморфны. По определению области неисправности $\mathfrak{I}_m$, минимальная форма конечно автоматной абстракции A_P имеет не более m состояний, где m – число состояний минимальной формы A_S^{min} конечно автоматной абстракции A_S. Соответственно, тест TS_A, построенный W-подобным методом по автомату A_S^{min}, является полным относительно модели неисправности $<A_S^{min}, \cong, \aleph_m>$ и обнаруживает неэквивалентность конечных автоматов A_P и A_S. В силу утверждения 2.1, для каждой реакции γ_{FSM} автомата A_P на входную последовательность α_{FSM} теста TS_A существует аналогичная (с точностью до преобразования временных последовательностей в классические) реакция γ автомата P на соответствующую временную входную последовательность α теста TS, что гарантирует обнаружение неэквивалентности временного автомата P автомату-спецификации S. Таким образом, проверяющий тест TS обнаруживает каждый автомат из $\mathfrak{I}_m$, не эквивалентный автомату S. Аналогично можно показать, что каждая корректная реализация «проходит» тест.

Отметим, что при построении теста конечно автоматными методами для области неисправности, содержащей неисправные реализации с большим числом состояний, чем у спецификации, длина теста существенно увеличивается [70]. Соответственно, такой подход может быть использован лишь при незначительном превышении числа состояний реализации над числом состояний спецификации.

Предложенный выше алгоритм может быть также использован при синтезе проверяющих тестов для автоматов только с таймаутами и только с временными ограничениями. Для автоматов с временными ограничениями существует взаимно однозначное соответствие между состояниями временного автомата и его абстракции, и проверяющий тест может быть построен непосредственно по конечно автоматной абстракции классическими методами [19]. Для автоматов с

таймаутами область неисправности ограничивается числом состояний конечно автоматных абстракций временных автоматов, аналогично общему случаю [78].

4.2 Экспериментальные результаты

В настоящем разделе приводятся результаты компьютерных экспериментов, проведённых с программной реализацией предложенного выше метода синтеза теста (алгоритм 4.1). Мы приводим полученные оценки длины теста для случайно сгенерированных временных автоматов и некоторых реальных сервисных систем.

4.2.1 Исследование длины проверяющих тестов для временных автоматов

Для оценки длины проверяющего теста относительно временного автомата-спецификации, алгоритм 4.1 был реализован на языке программирования C++ [8, 12]. Компьютерные эксперименты проводились со случайно сгенерированными автоматами [69] с таймаутами и временными ограничениями, а также классическими конечными автоматами. Экспериментальная выборка генерировалась с параметрами $|I| = |O| = 10$, в то время как тест строился W и H методами [33] для классических автоматов и конечно автоматных абстракций временных автоматов [69]. Зависимости средней длины построенных тестов от числа состояний исследуемых автоматов представлены на рисунке 4.4, где W(FSM) и H(FSM) – тесты для классических конечных автоматов, W(TFSM) и H(TFSM) – тесты для временных автоматов. Согласно проведенным экспериментам, разница между длиной H(FSM) и H(TFSM) тестов не слишком большая, в то время как H(TFSM) тест обнаруживает все некорректные реализации, для которых минимальная форма конечно автоматной абстракции имеет не больше состояний, чем таковая для автомата-спецификации.

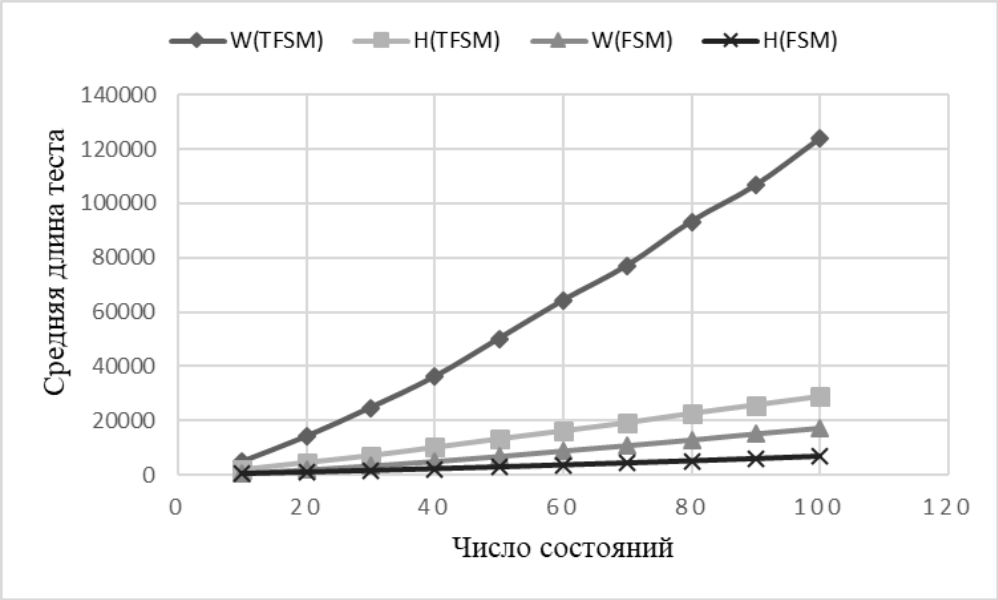

Рисунок 4.4 – Зависимость средней длины проверяющего теста от числа состояний временного автомата-спецификации

Мы также провели эксперименты по оценке влияния величины таймаутов на длину теста. Рассматривались таймауты величины 5, 10, и 15 для автоматов с 10 и 20 состояниями; результаты представлены на рисунке 4.5. Отметим, что длина H(TFSM) теста увеличивается вдвое, когда таймаут увеличивается в два раза.

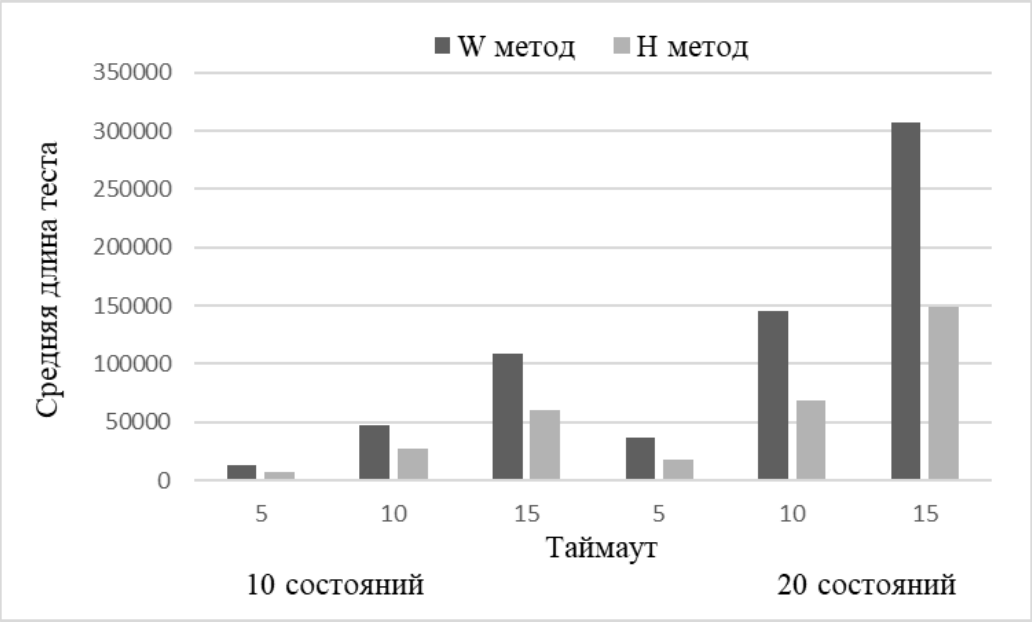

Рисунок 4.5 – Результаты экспериментов для автоматов-спецификаций с различными таймаутами

4.2.2 Синтез тестов для сервисных систем

Мы также оценили длину полных проверяющих тестов, построенных по алгоритму 4.1, для сервисных систем, поведение которых зависит от времени. Рассмотрим в качестве примера сервисную систему итальянской национальной железнодорожной компании, которая позволяет пассажирам получать компенсации в случае задержки поездов (рисунок 4.6).

Рисунок 4.6 – Временной автомат, описывающий сервисную систему итальянской национальной железнодорожной компании

Пассажиры информируются о задержках (переход “ожидаемое время отправления/уведомление о времени отправления”) и могут запросить компенсацию до отправления поезда при опоздании на 60 минут и более, пока автомат находится в состоянии “Отправление”. В этом случае полагается выплата полной суммы компенсации. После отправления поезда система переходит в состояние “В пути” и далее уведомляет пассажиров о времени прибытия. Если поезд опоздал на 60 минут и более, то автомат переходит в состояние “Задержка”, и пассажиры могут запросить компенсацию, в противном случае, система завершает обслуживание, выполнив переход “прибытие, [0, 60 мин)/уведомление об отсутствии задержки”. После поездки компенсация может быть получена в

период от 20 дней до 12 месяцев. Однако, в этот период можно получить лишь частичную компенсацию, в то время как через 12 месяцев после поездки получение компенсации невозможно. Построение тестовых последовательностей для временных автоматов с использованием описанных выше методов требует построения конечно автоматных абстракций, число состояний которых существенно зависит от значений таймаутов и временных ограничений на переходах. Соответственно, для того чтобы конечный автомат имел «разумные» размеры, мы сократили значения временных переменных в состоянии “Задержка” примерно в десять раз (с 20 дней и 12 месяцев до двух дней и одного месяца соответственно).

Длина построенного по алгоритму 4.1 теста с использованием HSI-метода, который является одной из модификаций W-метода, представлена в таблице 4.1 (столбец m). Построенный тест обнаруживает каждую неэквивалентную реализацию временного автомата, минимальная форма конечно автоматной абстракции которой имеет не больше состояний, чем минимальная форма соответствующей абстракции временного автомата-спецификации. Отметим также, что область неисправности может быть расширена реализациями (временными автоматами), конечно автоматные абстракции которых содержат больше состояний, чем конечно автоматная абстракция спецификации. Соответственно, мы также построили тесты, позволяющие обнаружить каждую неисправную реализацию (временной автомат), минимальная конечно автоматная абстракция которой содержит не более чем (m + 1) и (m + 2) состояния, где m – число состояний минимальной формы конечно автоматной абстракции временного автомата-спецификации. Длины проверяющих тестов для конечно автоматных абстракций и соответствующих временных автоматов, построенных методом HSI [6, 33], приведены в таблице 4.1.

В таблице 4.1 также представлены соответствующие результаты для системы управления лифтом, описание которой представлено на рисунке 4.7. В состоянии “Дверь закрыта” система ждёт вызова лифта, либо запроса на открытие двери. При поступлении запроса на открытие двери автомат переходит в состояние

“Дверь открыта” с выходным символом ←| |→. В состоянии “Дверь открыта”, реакция на входные символы зависит от времени подачи входного воздействия. Если входной символ “вверх” или “вниз” подан в промежутке [0, 7) система перейдёт в состояние “Закр. таймаут”, выждет пять тактов времени и начнёт закрывать двери перейдя в состояние “Закр. двери”. В то же время, в промежутке [7, 10), дверь уже начала закрываться и автомат сразу перейдёт в состояние “Закр. двери”, из которого перейдёт в состояние “Движение” после сообщения о закрытии двери. Если в состоянии “Дверь открыта” входное воздействие не было подано в течении 10 тактов, то дверь закроется и автомат перейдёт в соответствующее состояние по таймауту. Из состояния “Движение” система переходит в состояние “Дверь открыта” по команде открытия двери.

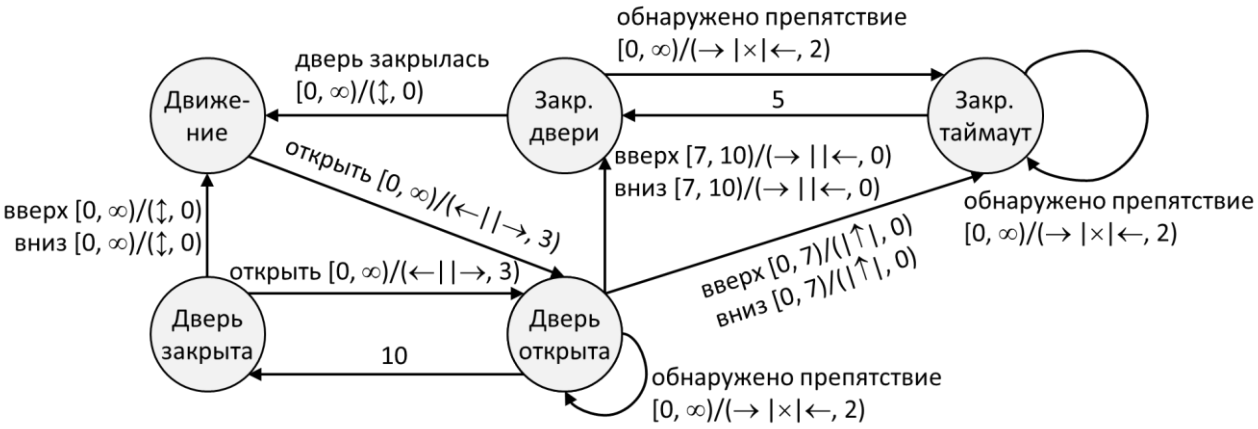

Рисунок 4.7 – Система управления лифтом

Таблица 4.1 – Длины тестов для сервисных систем, построенных HSI методом

	Число состояний реализации	m	m+1	m+2
Железнодорожная система возврата средств	Длина теста для временного автомата	20107	137681	932644
	Длина теста для соответствующей абстракции	112514	692006	4278194
Система управления лифтом	Длина теста для временного автомата	6181	46420	335504
	Длина теста для соответствующей абстракции	29388	184697	1167346

Можно заметить, что при близких значениях числа состояний и входных символов в рассмотренных примерах, тесты для первой из рассмотренных систем значительно длиннее, что является следствием более высоких значений таймаутов и/или временных ограничений. Тем не менее, тесты имеют достаточно разумную длину при очень высоком качестве, т.е. обнаружении большого количества неисправностей.

4.3 Выводы по главе 4

В настоящей главе была исследована задача синтеза проверяющих тестов для инициальных детерминированных временных автоматов и получены следующие основные результаты:

1. Поскольку минимальная форма инициальных детерминированных временных автоматов не является единственной с точностью до изоморфизма, задачу синтеза проверяющего теста с гарантированной полнотой для таких автоматов не удается свести к проверке соответствия между переходами спецификации и проверяемой реализации, что является основой синтеза таких тестов для классических автоматов.

2. Определена модель неисправности для инициальных детерминированных полностью определённых временных автоматов и предложен соответствующий метод синтеза тестов относительно эквивалентности с гарантированной полнотой покрытия неисправности. Область неисправности и метод синтеза тестов основаны на конечно автоматной абстракции временного автомата и позволяют свести проверку эквивалентности между временными автоматами к проверке изоморфизма между их абстракциями.

5 Синтез и оптимизация проверяющих тестов для недетерминированных временных автоматов

В настоящей главе предлагается метод синтеза проверяющих тестов с гарантированной полнотой для временных недетерминированных автоматов относительно редукции. Подход к построению тестов аналогичен таковому в предыдущей главе, т.е. модель неисправности и соответствующий тест строятся на основе конечно автоматной абстракции автомата-спецификации. Во втором разделе мы рассматриваем задачу оптимизации проверяющих тестов, а именно, предлагаем подходы к построению кратчайших адаптивных различающих последовательностей и оптимизации проверяющих тестов посредством модификации модели неисправности. Результаты главы опубликованы в [16, 19, 20, 22, 77, 79].

5.1 Синтез проверяющих тестов для недетерминированных автоматов с таймаутами и временными ограничениями

В данном разделе мы определяем модель неисправности для случая, когда поведение эталонной системы описано инициальным недетерминированным полностью определенным автоматом с таймаутами и временными ограничениями, проверяемый автомат (автомат-реализация) является временным детерминированным полностью определенным автоматом и отношением конформности является редукция, т.е. поведение конформной реализации должно содержаться в поведении автомата-спецификации. Для построения полного проверяющего теста относительно такой модели неисправности мы рассматриваем соответствующую модификацию классических конечно автоматных методов для недетерминированных автоматов [64]. Поскольку модель неисправности и тесты строятся на основе конечно автоматных абстракций, мы также устанавливаем некоторые свойства таких абстракций для недетерминированных временных автоматов.

5.1.1 Классические методы синтеза тестов для недетерминированных конечных автоматов

Полные проверяющие тесты относительно модели неисправности $FM^{NDFSM} = <\mathsf{S}, \leq, \aleph_m>$, когда автомат-спецификация является классическим инициальным недетерминированным наблюдаемым полностью определенным автоматом, могут быть построены на основе расширения W-метода, суть которого для классических автоматов была кратко рассмотрена в разделе 1.4.3. Общий алгоритм синтеза полного проверяющего теста относительно редукции для недетерминированного конечного автомата описан в [64] и основан на использовании *детерминировано*-передаточных и различающих (разделяющих) последовательностей. Входная последовательность α называется *детерминировано-передаточной* (*д-передаточной*) для состояния *s* конечного автомата S, если α переводит автомат из начального состояния в состояние *s* при любой выходной реакции. Входная последовательность α называется *разделяющей* для конечного автомата S, если множества реакций на α не пересекаются для двух различных состояний автомата S. Аналогично классическому W-методу, алгоритм синтеза тестов для недетерминированных автоматов доставляет достаточно короткие тесты при условии, что для каждого состояния существует *д*-передаточная последовательность, и в автомате существует не слишком длинная разделяющая последовательность. Если число состояний проверяемого автомата не превышает числа состояний автомата-спецификации, то построение полного проверяющего теста относительно модели неисправности $FM^{NDFSM} = <\mathsf{S}, \leq, \aleph_m>$ можно разделить на два этапа:

1. Для автомата S строятся *множество д-достижимости*, которое содержит *д*-передаточную последовательность для каждого состояния автомата, и разделяющая последовательность.

2. Каждая последовательность из множества *д*-достижимости конкатенируется с разделяющей последовательностью и каждым входным

символом; после каждого входного символа добавляется разделяющая последовательность.

Отметим также, что если в автомате не существует последовательности, разделяющей каждую пару состояний, то для синтеза тестов может быть использовано множество последовательностей, каждая из которых различает (разделяет) пару состояний автомата, что, вообще говоря, увеличивает длину теста. Однако, исследуемая система не всегда удовлетворяет требованиям описанного выше подхода, т.е. не каждое состояние соответствующего автомата является ∂-достижимым и/или не каждая пара состояний разделима, что побуждает к использованию других вариантов W-метода [61, 64]. Далее мы рассмотрим SCR-метод (State counting reduction-метод), который позволяет строить проверяющие тесты с гарантированной полнотой относительно модели неисправности $FM^{NDFSM} = <\mathsf{S}, \leq, \aleph_m>$, где не каждое состояние автомата S ∂-достижимо и/или S не обладает разделяющей последовательностью и не все состояния попарно разделимы.

1. Строится подмножество S_d состояний автомата S, которые являются ∂-достижимыми, и формируется ∂-покрытие автомата S, содержащее ∂-передаточную последовательность для каждого ∂-достижимого состояния из S_d.

2. Строится множество $R = \{R_1, R_2, \ldots, R_p\}$ максимальных подмножеств попарно разделимых состояний; для каждого множества $R_j \in R$, определяется подмножество R_{jd} состояний из R_j, которые являются ∂-достижимыми. Для каждого подмножества $R_j \in R$, строится разделяющее множество W_j, содержащее разделяющую последовательность для каждой пары различных состояний из R_j.

3. Для каждого состояния s_k из S_d, строится множество входных последовательностей N_k: входная последовательность $\alpha \in N_k$, если любая входо-выходная последовательность α/β в состоянии s_k покрывает состояния из множества $R_j \in R$ не менее $m - |R_{jd}| + 1$ раз и последнее не выполняется для префиксов последовательности α. Каждый префикс последовательности α конкатенируется с каждой последовательностью из множества W_j.

4. *∂*-передаточная последовательность для каждого состояния s_k конкатенируется с каждой последовательностью, полученной на шаге 3 при конкатенации последовательностей из множеств N_k и W_j.

В общем случае, длина построенного SCR-методом проверяющего теста значительно превышает таковую для случая, когда спецификация обладает разделяющей последовательностью и каждое её состояние *∂*-достижимо, и, соответственно, каждое множество W_j содержит одну разделяющую последовательность и $R = \{S\}$.

В следующем разделе мы исследуем возможности использования классических конечно автоматных методов синтеза тестов для автоматов с таймаутами и временными ограничениями.

5.1.2 Конечно автоматная абстракция для недетерминированных временных автоматов

Конечно автоматная абстракция недетерминированного полностью определённого автомата с таймаутами и временными ограничениями может быть построена по алгоритму, описанному в разделе 1.5. В этом случае абстракция также является недетерминированным полностью определённым автоматом.

Аналогично утверждениям 2.1 и 2.2 могут быть получены соответствующие результаты для недетерминированных автоматов с таймаутами и временными ограничениями.

Утверждение 5.1. Множество временных входо-выходных последовательностей α/γ_1, α/γ_2, …, α/γ_n существует в состоянии s недетерминированного временного автомата S, если и только если множество входо-выходных последовательностей $\alpha_{FSM}/\gamma_{1FSM}$, $\alpha_{FSM}/\gamma_{2FSM}$, …, $\alpha_{FSM}/\gamma_{nFSM}$ существует в состоянии $(s, 0)$ конечно автоматной абстракции $\mathsf{A_S}$.

Утверждение 5.2. Состояние s временного автомата S является редукцией состояния p временного автомата P, если и только если состояние $(s, 0)$ конечно

автоматной абстракции A_S является редукцией состояния $(p, 0)$ конечно автоматной абстракции A_P.

При построении проверяющих тестов для недетерминированного наблюдаемого временного автомата-спецификации мы полагаем, что проверяемый автомат (автомат-реализация) является детерминированным; в то же время, автомат-спецификация и автомат-реализация полагаются инициальными полностью определенными автоматами, если явно не сказано иное. Более того, аналогично примерам из предыдущей главы, можно показать, что для двух, возможно, недетерминированных автоматов с таймаутами и временными ограничениями S и P с n состояниями, количество состояний соответствующих конечно автоматных абстракций может отличаться. Таким образом, не каждый детерминированный временной автомат, не превосходящий недетерминированную спецификацию по числу состояний, имеющий такой же входной алфавит и конечную границу временных интервалов, гарантированно может быть обнаружен тестом, построенным описанной выше адаптацией W-метода по конечно автоматной абстракции спецификации. Тем не менее, такой тест обнаруживает каждую неисправную реализацию с таймаутами и временными ограничениями, конечно автоматная абстракция которой имеет не больше состояний, чем конечно автоматная абстракция спецификации, что формально доказывается в следующем разделе.

Отметим далее некоторые свойства конечно автоматных абстракций недетерминированных временных автоматов.

Рассмотрим временной автомат S на рисунке 5.1, для которого существует разделяющая последовательность $(i, 1)$. Однако, конечно автоматная абстракция A_S (рисунок 5.2) автомата S имеет пару неразделимых состояний $(b, 0)$ и $(b, (0, \infty))$, для которых множества реакций на любую входную последовательность совпадают.

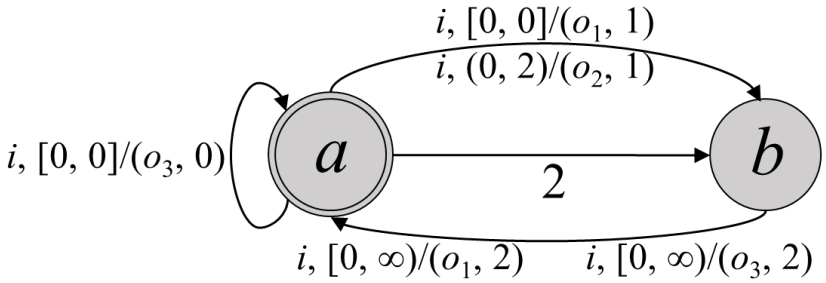

Рисунок 5.1 – Автомат с временными ограничениями S

Аналогично детерминированным конечным автоматам, недетерминированная конечно автоматная абстракция может быть минимизирована, например, состояния (b, 0) и (b, (0, ∞)) могут быть объединены в одно состояние [73]. Однако, в отличие от классических автоматов, для минимальной формы недетерминированных автоматов не каждая пара различных состояний является разделимой.

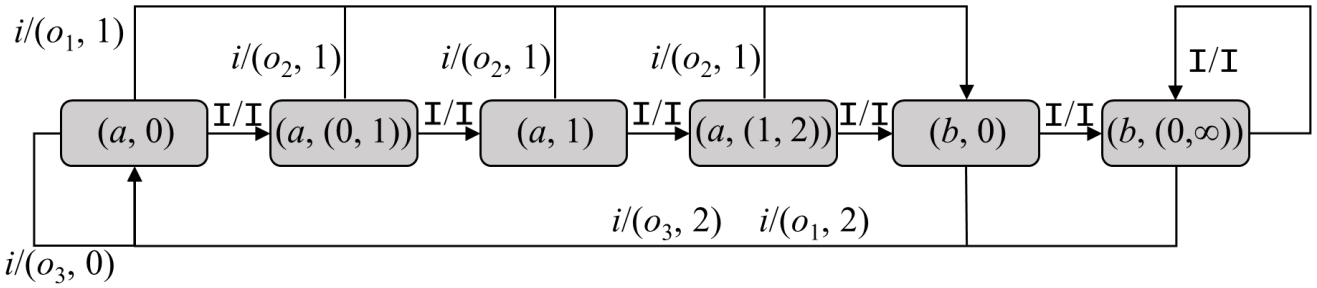

Рисунок 5.2 – Конечно автоматная абстракция A_S автомата S

Несмотря на то, что конечно автоматная абстракция «не наследует» свойство временного автомата обладать разделяющей последовательностью, состояния абстракции «наследуют» свойство ∂-достижимости, и справедливо следующее утверждение.

Утверждение 5.3. Состояния (s, 0), (s, (0, 1)), (s, 1), (s, (1, 2)) … конечно автоматной абстракции A_S ∂-достижимы, если и только если состояние s ∂-достижимо во временном автомате S.

Доказательство. Из утверждения 5.1 следует, что состояние (s, 0) ∂-достижимо в абстракции A_S, если и только если состояние s ∂-достижимо во временном автомате S. В то же время, состояния конечно автоматной абстракции A_S, содержащие состояние s (такие как (s, 0), (s, (0, 1)), (s, 1), (s, (1, 2)) …), ∂-

достижимы из состояния (s, 0) в силу детерминированности переходов по входо-выходной паре $\mathtt{I/I}$ (по определению абстракции) и, как следствие, $д$-достижимы из начального состояния.

Модель неисправности для недетерминированного полностью определённого автомата с таймаутами и временными ограничениями имеет вид $FM^{NDTFSM} = <\mathtt{S}, \leq, \Im_m>$, где $\mathtt{S}$ – полностью определённый недетерминированный наблюдаемый автомат-спецификация с таймаутами и временными ограничениями, $\leq$ – отношение редукции и $\Im_m$ – область неисправности, содержащая каждый полностью определённый детерминированный автомат с таймаутами и временными ограничениями с таким же входным алфавитом, как у спецификации, и конечно автоматной абстракцией не более чем с m состояниями. В следующем разделе мы формально представляем алгоритм синтеза тестов относительно такой модели неисправности.

5.1.3 Метод синтеза проверяющих тестов для недетерминированных автоматов с таймаутами и временными ограничениями

Проверяющий тест для недетерминированного автомата с таймаутами и временными ограничениями называется *полным* относительно модели неисправности $FM^{NDTFSM} = <\mathtt{S}, \leq, \Im_m>$, если тест обнаруживает каждую неконформную реализацию из множества $\Im_m$, т.е. каждый временной автомат, который не является редукцией автомата-спецификации. Иными словами, реализация считается неисправной или неконформной спецификации, если выходные реакции реализации на некоторую временную входную последовательность не содержатся в множестве соответствующих временных выходных последовательностей спецификации. Для конечно автоматной абстракции временного автомата-спецификации $\mathtt{S}$ тест строится одним из известных методов для недетерминированных автоматов относительно модели неисправности $FM^{NDFSM} = <\mathtt{A_S}, \leq, \aleph_m>$, где $\mathtt{A_S}$ – конечно автоматная абстракция автомата $\mathtt{S}$, $\leq$ – отношение редукции, $\aleph_m$ – множество конечных автоматов с таким

же входным алфавитом, как и A_S, и не более чем m состояниями. Отметим, что далее мы предлагаем использовать именно SCR-метод для синтеза тестов, поскольку конечно автоматная абстракция достаточно часто может содержать попарно неразделимые состояния.

Алгоритм 5.1 синтеза тестов для недетерминированных автоматов с таймаутами и временными ограничениями относительно редукции.

Вход: Недетерминированный наблюдаемый полностью определённый автомат с таймаутами и временными ограничениями S, описывающий поведение спецификации.

Выход: Полный проверяющий тест *TS* относительно модели неисправности $FM^{NDTFSM} = <S, \leq, \Im_m>$, где m – число состояний конечно автоматной абстракции временного автомата S.

Шаг 1. Строится конечно автоматная абстракция A_S временного автомата S.

Шаг 2. Строится полный проверяющий тест TS_A относительно модели неисправности $FM^{NDFSM} = <A_S, \leq, \aleph_m>$ SCR-методом.

Шаг 3. Входные последовательности теста TS_A преобразуются в соответствующие временные входные последовательности и добавляются в тест *TS*.

Конец алгоритма.

Теорема 5.4. Проверяющий тест *TS*, построенный по алгоритму 5.1 для недетерминированного полностью определённого автомата с таймаутами и временными ограничениями S, является полным относительно модели неисправности $FM^{NDTFSM} = <S, \leq, \Im_m>$.

Доказательство. Пусть автомат с таймаутами и временными ограничениями P есть автомат из множества $\Im_m$, неконформный спецификации S, т.е. проверяемый автомат P не является редукцией эталонного автомата S. Покажем, что такой автомат-реализация обнаруживается тестом *TS*. По определению области неисправности $\Im_m$, приведенная форма конечно автоматной абстракции A_P реализации P имеет не более m состояний. Поскольку P не является редукцией S,

то конечно автоматная абстракция A_P также не является редукцией абстракции A_S спецификации S в силу утверждения 5.2. Соответственно, тест TS_A, построенный на шаге 2 алгоритма и полный относительно модели неисправности $<A_S, \leq, \aleph_m>$, содержит входную последовательность, реакция на которую конечного автомата A_P не содержится в множестве соответствующих реакций автомата A_S. Из утверждения 5.1 следует, что для каждой реакции автомата A_P на последовательность γ_{FSM} теста TS_A существует аналогичная (с точностью до преобразования временных последовательностей в классические) реакция временного автомата P на соответствующую временную входную последовательность γ теста *TS*, что гарантирует обнаружение неконформной реализации P. Таким образом, проверяющий тест *TS* обнаруживает каждую неконформную реализацию из $\Im_m$. Аналогично можно показать, что каждая корректная реализация «проходит» тест.

Отметим, что даже при существовании разделяющей последовательности для конечно автоматной абстракции тест может достигать экспоненциальной длины относительно числа состояний спецификации, поскольку разделяющая последовательность может иметь экспоненциальную длину даже для детерминированных автоматов [54]. Для недетерминированных автоматов с n состояниями длина разделяющей последовательности может достигать величины $2(n^2/2)$ [10, 71]. В следующем разделе мы обсуждаем возможные подходы к оптимизации проверяющих тестов для полностью определённых возможно недетерминированных конечных автоматов.

5.2 Методы оптимизации проверяющих тестов

Как уже отмечалось выше, проверяющие тесты с гарантированной полнотой на основе недетерминированных автоматов-спецификаций могут быть достаточно длинными, что затрудняет использование таких моделей неисправности и соответствующих методов синтеза тестов при работе с реальными системами. В настоящем разделе мы исследуем свойства адаптивных различающих

последовательностей и предлагаем подход к оптимизации проверяющих тестов для недетерминированных автоматов посредством сокращения спецификации. Результаты данного раздела были получены для классических конечных автоматов и могут быть использованы при синтезе тестов относительно недетерминированных автоматов на основе конечно автоматной абстракции.

5.2.1 Адаптивные различающие тестовые последовательности для классических автоматов

Как известно [64, 88, 89], безусловные *d*-передаточные и разделяющие последовательности не всегда существуют для недетерминированных автоматов, что часто значительно увеличивает длину проверяющего теста. Однако, при построении тестов можно использовать адаптивные различающие и передаточные последовательности, которые существуют значительно чаще и могут иметь меньшую длину [38, 89].

Даже если использование адаптивных тестовых последовательностей не позволяет синтезировать для автомата-спецификации единственную различающую последовательность, различающие тестовые примеры могут сократить длину теста уменьшив мощность различающего множества и/или длину входящих в него последовательностей. Например, для рассмотренного в предыдущем разделе SCR-метода, использование адаптивных различающих и передаточных последовательностей может увеличить множества попарно различимых состояний W_j и, таким образом, сократить длину проверяющего теста.

Тестовый пример представляет *адаптивную различающую последовательность* [10, 38] для состояний s_1 и s_2 автомата S, если каждая входо-выходная последовательность, переводящая тестовый пример из начального состояния в терминальное, возможна не более чем в одном из состояний s_1 или s_2. В этом случае, состояния s_1 и s_2 автомата S *адаптивно различимы*. Если тестовый пример TC различает каждую пару состояний автомата S, то TC представляет адаптивную различающую последовательность для автомата S.

Тестовый пример представляет *адаптивную передаточную* последовательность для состояния *s* автомата S, если каждая входо-выходная последовательность тестового примера из начального состояния в терминальное переводит автомат S из начального состояния в состояние *s*. В этом случае состояние *s адаптивно достижимо* из начального состояния.

На рисунке 5.3 представлен недетерминированный конечный автомат S, начальным состоянием которого является состояние 0.

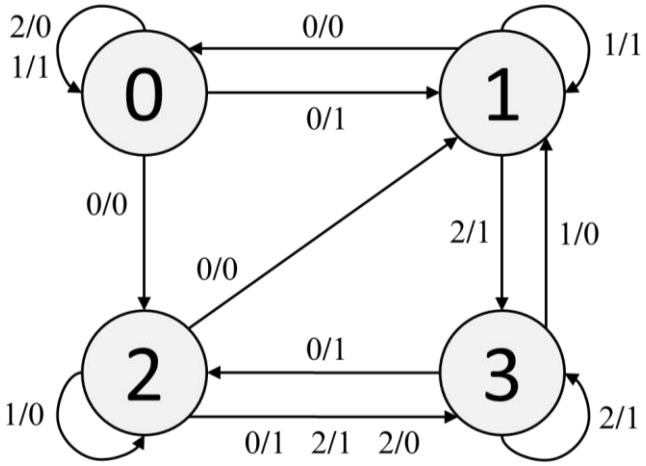

Рисунок 5.3 – Конечный автомат S

Тестовые примеры на рисунке 5.4 представляют *д*-передаточную (слева) и различающую (справа) адаптивные последовательности для автомата S (рисунок 5.3).

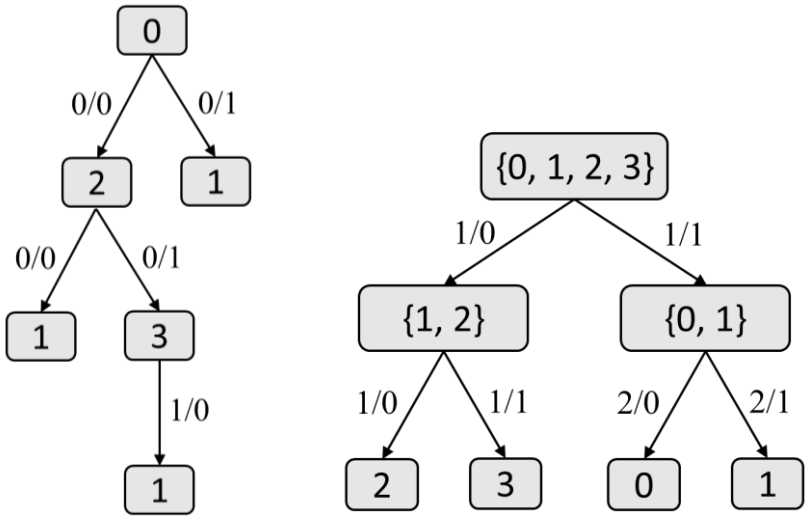

Рисунок 5.4 – Тестовые примеры, представляющие адаптивную различающую (справа) и передаточную (слева) последовательности для автомата S

Один из методов синтеза адаптивных различающих последовательностей основан на построении так называемого различающего автомата [38]. Различающий автомат S_{dist} для полностью определённого наблюдаемого возможно недетерминированного автомата $\mathsf{S} = (S, I, O, h_{\mathsf{S}}, s_0)$ строится по следующим правилам. Множества входных и выходных символов S_{dist} совпадает с таковыми для исходного конечного автомата S, в то время как состояния S_{dist} соответствуют непустым подмножествам состояний S, содержащим более одного элемента; кроме того, в различающем автомате присутствует специальное состояние F. Построение переходов S_{dist} начинается в его начальном состоянии, которое соответствует подмножеству, содержащему все состояния автомата S. Различающий автомат S_{dist} представляет собой минимальный автомат, построенный по следующим правилам.

Пусть b – состояние различающего автомата S_{dist}.

1. Переход (b, i, o, b') существует в автомате S_{dist}, если и только если b' не является одноэлементным подмножеством, для каждого $o' \in O$ непустые io'-*преемники* для любой пары различных состояний из b не совпадают, и b' – непустое множество io-преемников состояний из множества b. В таком случае, состояние b' добавляется в множество состояний S_{dist}.

2. Переход (b, i, o, F) существует в автомате S_{dist}, если и только если существует выходной символ $o' \in O$, такой что непустые io'-преемники для некоторой пары различных состояний из b совпадают (имеет место слияние переходов). В таком случае, состояние F добавляется в множество состояний S_{dist}.

3. Переход автомата S_{dist} из состояния b по входному символу i не определён, если и только если для каждого $o' \in O$, непустые io'-преемники любой пары различных состояний из b не совпадают, и каждый непустой io-преемник множества состояний из b является одноэлементным.

В состоянии F существует переход $(F, i, o\ F)$ для каждой входо-выходной пары i/o автомата S_{dist}.

Входной символ i различающего автомата S_{dist} называется *неопределённым* в состоянии b, если в автомате S_{dist} не существует переходов из состояния b по

входному символу i, т.е. для каждого $o' \in O$, непустые io'-преемники любой пары различных состояний из b не совпадают, и каждый io-преемник множества состояний из b является не более чем одноэлементным подмножеством. Для конечного автомата S на рисунке 5.3 соответствующий различающий автомат S_{dist}, представлен на рисунке 5.5 и имеет 10 состояний.

Для того чтобы определить, обладает ли конечный автомат S различающим тестовым примером, состояния различающего автомата S_{dist} с неопределённым входным символом удаляются из различающего автомата со всеми переходами в такие состояния. Состояния удаляются итеративно, пока либо в начальном состоянии не появляется неопределённый входной символ, либо в различающем автомате не заканчиваются состояния с неопределёнными входными символами. При удалении состояния b из S_{dist} мы сохраняем один из входных символов i, неопределённых в состоянии b, в специальном массиве UN, $UN(b) = i$. Если на некоторой итерации удаления состояний в начальном состоянии различающего автомата существует неопределённый входной символ, то различающий тестовый пример D может быть построен при помощи сохранённых в массиве UN входных символов. В этом случае начальное состояние d_0 автомата D соответствует начальному состоянию b_0 автомата S_{dist}. Входной символ $UN(b_0)$ представляет собой единственный входной символ, определённый в состоянии d_0 автомата D, и переход $(d_0, UN(b_0), o, d)$ существует в D, если и только если в S_{dist} (до удаления состояний) существует переход $(b_0, UN(b_0), o, b)$, где для состояний b и d, соответствующие подмножества состояний автомата S совпадают. Если для некоторого $o \in O$ не существует перехода $(b_0, UN(b_0), o, b)$ в автомате S_{dist}, то в D добавляется переход $(d_0, UN(b_0), o, F)$, где F – тупиковое состояние. Переходы для других состояний различающего тестового примера D формируются аналогично. Отметим, что для более удобного представления различающего тестового примера, вместо единственного состояния F, можно использовать множество тупиковых состояний, каждое из которых помечается состоянием, в котором находился автомат до начала эксперимента при наблюдении соответствующей входо-

выходной последовательности (если таковое существует). Для конечного автомата S на рисунке 5.3 соответствующий различающий тестовый пример представлен на рисунке 5.4 (справа), имеет длину 2 и может быть построен по различающему автомату S_{dist} (рисунок 5.5).

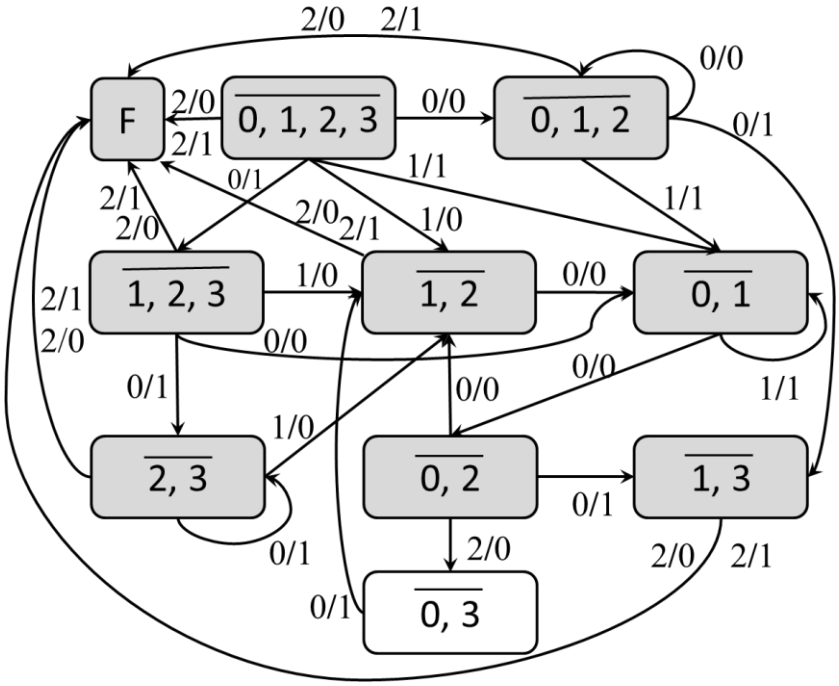

Рисунок 5.5 – Различающий автомат S_{dist}

5.2.2 Экспериментальные результаты по построению адаптивных различающих последовательностей для классических автоматов

Далее мы приводим результаты компьютерных экспериментов по оценке длины адаптивных различающих последовательностей.

Первая серия экспериментов проводилась со случайно сгенерированными конечными автоматами для оценки вероятности существования адаптивной различающей последовательности в зависимости от числа недетерминированных переходов автомата. Количество *nd* переходов автомата для каждой пары ‘состояние, входной символ’ задается заранее. Проведённые эксперименты (рисунок 5.6) показали, что для случайно сгенерированных автоматов различающий тестовый пример существует крайне редко, если *nd* > 3, и

соответственно, последующие эксперименты проводились с автоматами, для которых $nd \leq 3$.

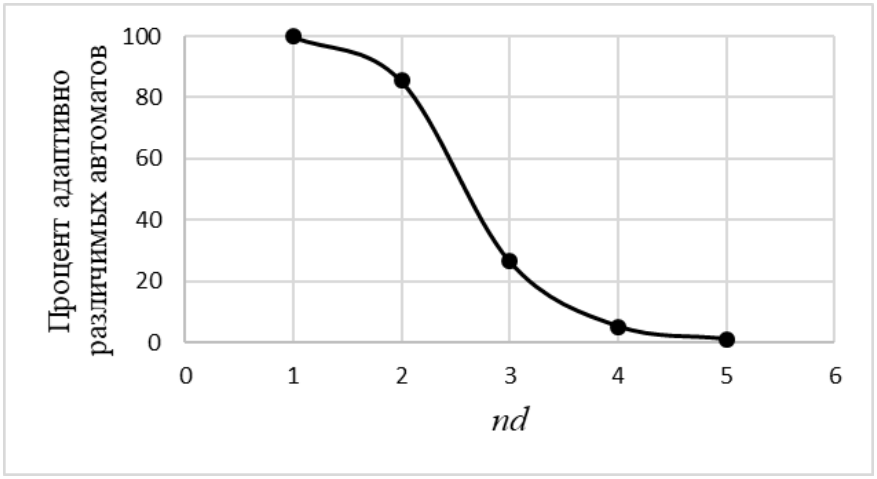

Рисунок 5.6 – Процент адаптивно различимых автоматов

На рисунке 5.7 представлена средняя высота различающего тестового примера в зависимости от числа состояний автомата. Эксперименты проводились с автоматами, для которых $|I| = |O| = 10$, при различном числе недетерминированных переходов. Нижняя кривая построена для детерминированных автоматов, средняя и верхняя кривые для значений *nd* равном 2 и 3 соответственно.

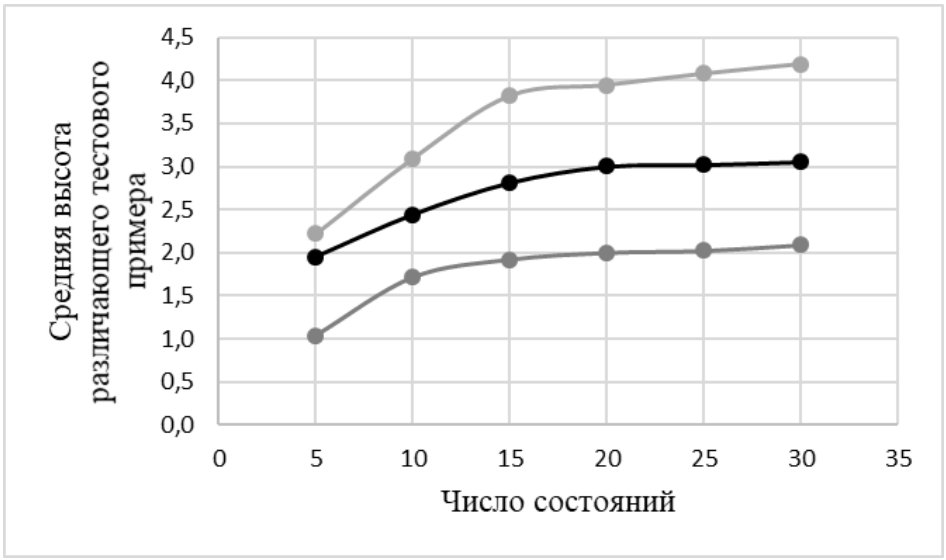

Рисунок 5.7 – Средняя высота различающего тестового примера для детерминированных и недетерминированных автоматов

Из проведённых экспериментов следует, что для случайно сгенерированных автоматов различающий тестовый пример либо имеет небольшую высоту, либо не существует. Однако, несмотря на небольшую высоту различающих тестовых примеров, требуемые для их построения различающие автоматы в некоторых случаях содержат большую избыточность, и кроме того, описанный выше подход не гарантирует построения кратчайшего различающего тестового примера (если такой пример существует). В следующем разделе мы предлагаем подход к синтезу кратчайших различающих тестовых примеров на основе оптимизации различающего автомата.

5.2.3 Оптимизация различающих тестовых примеров для классических автоматов

В [38] показано, что для полностью определённого наблюдаемого, возможно, недетерминированного автомата S существует различающий тестовый пример, если и только если в соответствующем различающем автомате S_{dist} не существует полностью определённого подавтомата. В той же работе показано, что длина различающего тестового примера может достигать значения $2^n - n - 1$ для полностью определённого недетерминированного наблюдаемого автомата S с n состояниями. В этом случае, все возможные подмножества состояний автомата S, не являющиеся одноэлементными, могут быть состояниями различающего автомата S_{dist}, и соответственно, процедура построения различающего автомата требует экспоненциального числа шагов. С другой стороны, как отмечалось выше, построение тестового примера по различающему автомату посредством итеративного удаления состояний с неопределенным входным символом не гарантирует построения различающего тестового примера наименьшей высоты. Для того чтобы построить кратчайшую адаптивную различающую последовательность, если таковая существует, за оптимальной число шагов, мы предлагаем ограничивать построение различающего автомата допустимой длиной адаптивной различающей последовательности. Сокращённый различающий автомат S^L_{dist} содержит лишь состояния, достижимые из начального состояния по

входо-выходным последовательностям длины не больше некоторого $L \geq 0$, и позволяет найти различающие тестовые примеры высоты не больше L. Предложенный подход согласуется с результатами из [38], где утверждается, что длина адаптивной различающей последовательности не превышает 15 для случайно сгенерированных конечных автоматов даже с большим (более 100) числом состояний. Ввиду того, что высота различающего тестового примера для большинства автоматов не слишком большая, имеет смысл ограничить длину, на которой осуществляется поиск тестового примера.

Таким образом, под автоматом S^{L}_{dist} понимается различающий автомат, содержащий лишь состояния, достижимые из начального состояния по входо-выходным последовательностям длины не больше L, где L – целое неотрицательное число, и состояние F. Отметим также, что все переходы S^{L}_{dist} из состояний, достижимых только по входной последовательности длины L ведут в состояние F. Например, для автомата на рисунке 5.3, различающий автомат S^{0}_{dist} содержит только начальное состояние {0, 1, 2, 3} и состояние F, в то время как S^{1}_{dist} содержит еще четыре состояния, достижимых из начального состояния по входо-выходной последовательности длины 1 (рисунок 5.5).

Аналогично утверждению 1 из [38], справедливо следующее утверждение.

Утверждение 5.5. Для полностью определённого наблюдаемого, возможно, недетерминированного конечного автомата S существует различающий тестовый пример высоты не больше L, если и только если различающий автомат S^{L}_{dist} не имеет полностью определённого подавтомата.

Доказательство. Пусть в автомате S существует различающий тестовый пример высоты L. Тогда по утверждению 1 [38], различающий автомат S_{dist} не имеет полностью определённого подавтомата. Итеративным удалением состояний, достижимых по последовательности длины не больше L, можно показать, что для автомата S^{L}_{dist} также не существует полностью определённого подавтомата. В то же время, если автомат S^{L}_{dist} не имеет полностью определённого подавтомата, то тестовый пример может быть непосредственно построен по приведённой выше процедуре, аналогичной процедуре из [38].

Таким образом, проверка существования различающего тестового примера высоты не больше L может быть проведена итеративным удалением состояний сокращённого различающего автомата S^{L}_{dist}. Если начальное состояние S^{L}_{dist} в результате такого удаления приобретает неопределённый входной символ, то S^{L}_{dist} не имеет полностью определённого подавтомата и соответствующий различающий тестовый пример может быть построен описанным выше методом.

Например, различающий автомат на рисунке 5.5 имеет 10 состояний, однако, различающий тестовый пример может быть построен на основе автомата S^{2}_{dist}, который имеет 9 состояний (выделены серым цветом). При удалении состояний с неопределённым входным символом можно убедиться, что различающий автомат S^{2}_{dist} не имеет полностью определённого подавтомата, и соответствующий различающий тестовый пример (рисунок 5.4 справа) может быть построен по описанной выше процедуре.

Таким образом, мы предлагаем следующую процедуру построения различающего тестового примера для полностью определённого, возможно, недетерминированного наблюдаемого конечного автомата.

Алгоритм 5.2 построения кратчайшего различающего тестового примера для классического конечного автомата.

Вход: полностью определённый, возможно, недетерминированный наблюдаемый автомат S, целое число $L \geq 1$

Выход: Различающий тестовый пример высоты не больше L для автомата S, сообщения ‘Автомат S не имеет различающего тестового примера высоты не больше L’ или ‘Автомат S не имеет различающего тестового примера’

$l := 0$, массив неопределённых входных символов $UN := \varnothing$, различающий автомат S^{l}_{dist} с начальным состоянием, совпадающим с множеством состояний автомата S, и состоянием F, в которое ведут переходы из начального состояния по всем входо-выходным парам.

Шаг 1. $l := l + 1$; В различающий автомат S^{l-1}_{dist} добавляются состояния, достижимые из начального состояния по входо-выходной последовательности длины l, т.е., строится автомат S^{l}_{dist}.

Если $\mathsf{S}^{l-1}_{dist} = \mathsf{S}^{l}_{dist}$**, то** вывести сообщение ‘Автомат S не имеет различающего тестового примера’ и **Конец алгоритма**;

иначе Шаг 2.

Шаг 2. Итеративно удалить из S^{l}_{dist} каждое состояние b, в котором существует неопределённый входной символ; выбрать неопределённый входной символ i, положив $UN(b) = i$.

Если в начальном состоянии S^{l}_{dist} существует неопределённый входной символ, **то** Шаг 3;

иначе:

Если $l = L$ **то** вывести сообщение ‘Автомат S не имеет различающего тестового примера высоты не больше L’ и **Конец алгоритма**;

иначе Шаг 1.

Шаг 3. Построить различающий тестовый пример по массиву неопределённых входных символов UN и различающему автомату S^{l}_{dist} до удаления состояний.

Конец алгоритма.

Теорема 5.6. Для полностью определённого наблюдаемого, возможно, недетерминированного автомата S и целого числа $L \geq 0$ алгоритм 5.2 возвращает кратчайший различающий тестовый пример, если и только если в автомате S существует различающий тестовый пример длины не больше L.

Доказательство. Из утверждения 5.5 и результатов [38] следует, что если автомат S имеет адаптивную различающую последовательность длины L, то этот различающий тестовый пример может быть построен по различающему автомату S^{L}_{dist}. Покажем далее, что удаление состояний на шаге 2 алгоритма 5.2 не влияет на построение различающего примера. Пусть для числа L был построен автомат S^{L}_{dist} и состояние b различающего автомата S^{L}_{dist} имеет неопределённый входной символ.

Последнее означает, что для подмножества состояний автомата S, соответствующего состоянию b, существует (адаптивная) различающая последовательность длины не более $L - L'$, где L' – длина кратчайшей входной последовательности из начального состояния S^{L}_{dist} в b. Таким образом, состояние b будет удалено, как только для него будет найдена кратчайшая (адаптивная) различающая последовательность, которая позже может быть восстановлена по множеству UN и включена в различающий тестовый пример.

5.2.4 Сокращение спецификации для классических автоматов

Если спецификация не имеет адаптивной различающей последовательности или не каждое её состояние является адаптивно достижимым, то полный проверяющий тест относительно $FM^{NDFSM} = <\mathsf{S}, \leq, \aleph_m>$ будет значительно длиннее. Далее, мы предлагаем метод построения теста, позволяющий получить тест полиномиальной длины, основанный на сокращении спецификации.

Проверяющий тест, построенный рассмотренными в разделе 5.1.1 методами, имеет полиномиальную длину относительно числа состояний автомата-спецификации S, если в последнем все состояния адаптивно детерминировано достижимы (свойство 1), и автомат обладает адаптивной различающей последовательностью, длина которой пропорциональна числу состояний автомата (свойство 2). Если исходный автомат-спецификация S не удовлетворяет свойствам 1 и 2, то мы предлагаем удалить из S такое подмножество переходов, чтобы полученный сокращенный автомат $\mathsf{S}^{\text{сокр}}$ обладал адаптивными различающей и передаточными последовательностями. Такое сокращение не всегда возможно и далее мы вводим необходимое и достаточное условие для построения сокращенного автомата $\mathsf{S}^{\text{сокр}}$.

Утверждение 5.7. Для полностью определённого связного, возможно, недетерминированного конечного автомата S существует полностью определённый связный подавтомат $\mathsf{S}^{\text{сокр}}$, удовлетворяющий свойствам 1 и 2, если

и только если для S существует детерминированный связный подавтомат, обладающий различающей последовательностью.

Доказательство. Если для автомата S существует детерминированный связный подавтомат P, обладающий различающей последовательностью, то существует подмножество переходов $Tr \subset h_S$, такое что $Tr \cap h_P = \varnothing$, удалив которое из множества переходов автомата S, будет получен автомат P удовлетворяющий свойствам 1 и 2.

Если для автомата S существует связный подавтомат $S^{сокр}$, удовлетворяющий свойствам 1 и 2, то для автомата $S^{сокр}$ существует детерминированный связный (в силу свойства 1) подавтомат P обладающий различающей последовательностью (в силу свойства 2). Соответственно, P является подавтоматом S в силу транзитивности отношения $\subseteq$ для отношений (множеств) переходов конечных автоматов.

В худшем случае результатом удаления переходов станет детерминированный подавтомат исходной спецификации. Если автомат-спецификация S не удовлетворяет условиям 1 или 2, то соответствующий автомат $S^{сокр}$ может быть построен по следующему алгоритму.

Алгоритм 5.3 построения сокращённой спецификации на основе перебора детерминированных подавтоматов.

Вход: недетерминированный полностью определённый конечный автомат S.

Выход: сокращённый автомат $S^{сокр}$, удовлетворяющий свойствам 1 и 2.

Шаг 1. Посредством перебора полностью определенных детерминированных подавтоматов с таким же числом состояний, как и S, определяется связный подавтомат S′ автомата S, для которого существует адаптивный различающий пример. Если такого подавтомат нет, то возвращается сообщение «Сокращение спецификации невозможно». Формируется множество *д*-достижимости и адаптивный различающий пример TC для автомата S′.

Шаг 2. Для каждой входной последовательности α из множества *д*-достижимости, среди переходов автомата S, которые выполняются при обработке последовательности α и отсутствуют в подавтомате S', удаляем те, которые нарушают свойство последовательности α быть передаточной последовательностью.

Шаг 3. Среди переходов автомата S, которые выполняются автоматом при обработке тестового примера TC в любом из состояний и отсутствуют в подавтомате S', удаляем те, которые приводят к нарушению свойства TC быть различающим тестовым примером.

Конец алгоритма.

Пусть S – недетерминированный автомат спецификация, $S^{сокр}$ – сокращённая спецификация, полученная из автомата S по алгоритму 5.3 и *TS* – полный тест, построенный для автомата $S^{сокр}$. Тест *TS* не является полным относительно модели неисправности $FM^{NDFSM} = <S, \leq, \aleph_m>$, поскольку может существовать автомат $P \in \aleph_m$, который является редукцией автомата S, но не является редукцией автомата $S^{сокр}$. Однако, полноту построенного теста можно определить формально и имеет место следующая теорема.

Теорема 5.8. Если реакция проверяемого автомата P на каждую входную последовательность из множества *TS* содержится в множестве выходных реакций сокращённого автомата-спецификации $S^{сокр}$, то проверяемый детерминированный автомат P есть редукция автомата S. Если выходная реакция проверяемого автомата P на некоторую последовательность из *TS* не содержится в соответствующем множестве выходных реакций автомата-спецификации S, то автомат P не является конформной реализацией.

Доказательство. Рассмотрим автомат-спецификацию S с начальным состоянием s_0, соответствующий автомат $S^{сокр}$, полученный из автомата S по алгоритму 5.3, с начальным состоянием $s_0^{сокр}$ и детерминированный автомат-реализацию $P \in \Im_m$ с начальным состоянием p_0. Поскольку автомат $S^{сокр}$ является связным, полностью определённым и получается удалением переходов из

автомата S без изменения числа состояний, множество реакций на любую входную последовательность автомата $S^{сокр}$ содержится в множестве реакций на эту последовательность автомата S, т.е. $s_0^{сокр} \leq s_0$. Следовательно, если реакции проверяемого автомата P на последовательности теста *TS* содержатся в соответствующих множествах выходных реакций $S^{сокр}$, то, поскольку тест *TS* является полным относительно модели неисправности $FM^{NDFSM} = <S^{сокр}, \leq, \aleph_m>$, автомат P является редукцией спецификации S. С другой стороны, если реакция автомата P на некоторую последовательность теста не содержится в соответствующем множестве реакций автомата S, то реализация P не является конформной.

Заметим также, что если реакции реализации P на все последовательности теста *TS* содержатся в соответствующих множествах выходных реакций S, но существует последовательность теста, реакция на которую автомата P не содержится в множестве выходных реакций автомата $S^{сокр}$, то мы не можем судить о том, является ли реализация P конформной или неконформной (так называемый «неопределенный вердикт» после выполнения тестирования).

Таким образом, предложенный подход позволяет построить тест меньшей длины для класса недетерминированных автоматов, обладающих детерминированным связным подавтоматом с адаптивной различающей последовательностью. При сокращении спецификации мы уменьшаем количество недетерминированных переходов автомата, сокращая тем самым опциональность спецификации.

Все описанные выше сокращения могут быть использованы в разделе 5.1 при построении тестов для временных автоматов на основе их конечно автоматных абстракций.

5.3 Выводы по главе 5

В настоящей главе была исследована задача синтеза проверяющих тестов для инициальных недетерминированных временных автоматов и получены следующие основные результаты:

1. Определена модель неисправности для инициальных недетерминированных наблюдаемых полностью определённых временных автоматов и предложен соответствующий метод синтеза тестов относительно редукции с гарантированной полнотой.

2. Исследована возможность построения адаптивных проверяющих тестов относительно редукции для недетерминированных наблюдаемых полностью определённых временных автоматов и предложены способы минимизации таких тестов; в частности, исследованы адаптивные различающие последовательности, использование которых позволяет сократить длину проверяющих тестов, и предложен подход к синтезу кратчайших различающих последовательностей.

3. Показано, как длина проверяющих тестов с гарантированной полнотой может быть уменьшена посредством сокращения области неисправности.

Заключение

Диссертация посвящена решению задач анализа и синтеза для конечных автоматов с одной временной переменной. Рассмотренные задачи синтеза направлены на оптимизацию описания системы в виде временного автомата, в то время как задачи анализа представлены проблемами генерации и оптимизации проверяющих тестов с гарантированной полнотой.

В диссертации получены следующие основные результаты:

1. Предложен метод построения минимальной формы полностью определенного детерминированного неинициального временного автомата, учитывающий как оптимизацию числа состояний, так и временных аспектов; показано, что минимальная форма единственна с точностью до изоморфизма.

2. Предложена и исследована операция параллельной композиции для системы взаимодействующих временных детерминированных автоматов, которая является расширением такой операции для классических конечных автоматов. Показано, что в отличие от классических конечных автоматов, класс детерминированных временных автоматов не является замкнутым относительно этой операции при наличии «медленной» внешней среды. Выделены классы детерминированных временных автоматов, замкнутые относительно операции параллельной композиции.

3. Предложены модель неисправности и соответствующий метод синтеза проверяющих тестов с гарантированной полнотой для детерминированных временных автоматов, обобщающий известные методы построения таких тестов для автоматов только с таймаутами или только с временными ограничениями.

4. Предложен алгоритм синтеза проверяющих тестов с гарантированной полнотой для недетерминированных временных автоматов относительно редукции; предложены алгоритмы минимизации адаптивных проверяющих тестов, возможно, с уменьшением полноты теста.

Отметим следующие возможные направления дальнейших исследований.

1. Минимальная по времени и состояниям форма неинициальных временных автоматов может рассматриваться как каноническая форма представления системы, однако, если в автомате выделено начальное состояние, то его минимальная форма не является единственной. Последнее, в частности, существенно затрудняет определение моделей неисправности для временных автоматов, для которых существуют методы синтеза проверяющих тестов с гарантированной полнотой без перехода к конечно автоматной абстракции, число состояний которой может существенно увеличиться по сравнению с исходным автоматом-спецификацией. Например, большие таймауты в сервисных системах могут вызвать существенный рост состояний в соответствующей абстракции (в качестве примера можно рассмотреть систему возврата денежных средств итальянской железнодорожной компании в разделе 4.2.2). Таким образом, интерес представляет определение и построение канонической минимальной формы для инициальных временных автоматов.

2. В работе рассмотрена лишь одна операция композиции временных автоматов, а именно, бинарная параллельная композиция детерминированных полностью определённых автоматов с временными ограничениями. Представляет интерес исследование этой операции для автоматов, которые имеют и таймаутами, и временные ограничения, как наиболее общей модели автомата с одной временной переменной. Другие операции композиции, в частности, операция синхронной композиции, которая в основном используется при композиции аппаратных средств, остались за рамками данной работы и требуют дополнительных исследований. Также представляет интерес оптимизация многомодульных композиций и/или их компонент на основе решения уравнений для временных автоматов.

3. Предложенные методы синтеза тестов для детерминированных и недетерминированных временных автоматов достаточно часто «возвращают» слишком длинные проверяющие тесты, и, соответственно, вопросы разработки новых моделей неисправности и методов синтеза тестов с гарантированной

полнотой относительно этих моделей для систем реального времени, по-прежнему, остаются актуальными. Интересной задачей также является адаптация предложенных моделей неисправности и подходов к синтезу тестов для других неклассических автоматных моделей, например, расширенных автоматов или входо-выходных полуавтоматов с конечным числом временных переменных.

Список использованной литературы

1. Агибалов Г. П. Лекции по теории конечных автоматов / Г. П. Агибалов, А. М. Оранов. – Томск : Издательство ТГУ, 1984. – 185 с.

2. Бурдонов И. Б. Теория соответствия для систем с блокировками и разрушениями / И. Б. Бурдонов, А. С. Косачев, В. В. Кулямин. – М. : ФИЗМАТЛИТ. – 2008. – 412 с.

3. Василевский М. П. О распознавании неисправности автоматов / М. П. Василевский // Кибернетика. – 1973. – № 4. – С. 93–108.

4. Гилл А. Введение в теорию конечных автоматов / А. Гилл. – М. : Издательство Наука, 1966. – 272 с.

5. Громов М. Л. Разработка методов синтеза условных тестов для автоматных моделей с недетерминированным поведением : дис. на соискание ученой степени канд. физ.-мат. наук / М. Л. Громов. – Томск, 2009. – 154 с.

6. Евтушенко Н. В. Метод построения эксперимента для произвольного детерминированного автомата / Н. В. Евтушенко, А. Ф. Петренко // Автоматика и вычислительная техника. – 1990. – № 5. – С. 73–76.

7. Евтушенко Н. В. Недетерминированные автоматы: анализ и синтез Ч. 1: Отношения и операции : учеб. пособие / Н. В. Евтушенко, А. Ф. Петренко, М. В. Ветрова. – Томск : Том. гос. ун-т, 2006. – 142 с.

8. Кнут Д. Искусство программирования. Т. 1 / Д. Кнут. – М. : Мир, 1976. – 734 с.

9. Кондратьева О.В. Параллельная композиция конечных автоматов с таймаутами / О. В. Кондратьева, Н. В. Евтушенко, А. Р. Кавалли // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2014. – № 2. – С. 73–81.

10. Кушик Н. Г. Методы выделения подклассов конечных автоматов с пониженными оценками сложности умозрительных экспериментов : диссертация на соискание ученой степени доктора физико-математических наук / Кушик Н. Г. – Томск, 2016. – 287 с.

11. Матросова А. Ю. Алгоритмические методы синтеза тестов / А. Ю. Матросова. – Томск: Изд-во Томского университета, 1990. – 207 с.

12. Страуструп Б. Язык программирования C++ / Б. Страуструп. – М. : Радио и связь, 1991. – 349 с.

13. Твардовский А. С. О минимизации временных автоматов / А. С. Твардовский // Новые информационные технологии в исследовании сложных структур : материалы 10-й Российской конференции с международным участием. – пос. Катунь Алтайского края, 9-13 июня 2014. – С. 41–42.

14. Твардовский А. С. К минимизации автоматов с временными ограничениями / А. С. Твардовский, Н. В. Евтушенко // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2014. – № 4(29). – С. 77–83.

15. Твардовский А. К минимизации автоматов с таймаутами / А. С. Твардовский // Труды Института системного программирования РАН. – Т. 26, вып. 6. – 2014. – С. 77–84.

16. Твардовский А. С. Синтез полных проверяющих тестов для недетерминированных автоматов с временными ограничениями / А. С. Твардовский, Н. В. Евтушенко // Известия вузов. Физика. – 2015. – Т. 58, № 11/2. – С. 107–110.

17. Твардовский А. С. О единственности минимильной формы для автоматов с таймаутами / А. С. Твардовский, Н. В Евтушенко // Известия вузов. Физика. – 2016. – Т. 59, № 8/2. – С. 101–103.

18. Твардовский А.С. Минимизация автоматов с таймаутами и временными ограничениями / А. С. Твардовский, Н. В Евтушенко, М. Л. Громов // Труды Института системного программирования РАН. – 2017. – Т. 29, вып. 4. – С. 139–154.

19. Твардовский А. С. Синтез тестов с гарантированной полнотой для недетерминированных временных автоматов / А. С. Твардовский, К. Эль-Факи, М.Л. Громов, Н. В Евтушенко // Моделирование и анализ информационных систем. – 2017. – Т. 24, № 4. – С. 496–507.

20. Твардовский А. С. К построению подавтоматов без слияний для недетерминированных конечных автоматов / А. С. Твардовский, Н. В. Евтушенко // Новые информационные технологии в исследовании сложных структур : материалы 12-й Российской конференции с международным участием. – пос. Катунь Алтайского края, 4-8 июня 2018. – С. 85–86.

21. Твардовский А. С. О возможностях автоматного описания параллельной композиции временных автоматов / А. С. Твардовский, А. В. Лапутенко // Труды Института системного программирования РАН. – 2018. – Т. 30, вып. 1. – С. 25–40.

22. Твардовский А. С. К синтезу адаптивных различающих последовательностей для конечных автоматов / А. С. Твардовский, Н. В Евтушенко // Труды Института системного программирования РАН. – 2018. – Т. 30, вып. 4. – С. 139–154.

23. Убар Р. Р. Проектирование контролепригодных дискретных систем : Учебное пособие / Р. Р. Убар. – Таллин : Таллинский политехнический институт, 1988. – 68 с.

24. Alur R. Minimization of timed transition systems / R. Alur, C. Courcoubetis, N. Halbwachs, D. L. Dill, H. Wong-Toi // Proceedings of the 3rd Conference on Concurrency Theory CONCUR ‘92. Lecture Notes in Computer Science. – 1992. – Vol. 630. – P. 340–354.

25. Alur R. A theory of timed automata / R. Alur, D. L. Dill // Theoretical Computer Science. – 1994. – Vol. 126, is. 2. – P. 183–235.

26. Bochmann G. V. Protocol testing: review of methods and relevance for software testing / G. V. Bochmann, A. Petrenko // Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis. – USA, 1994. – P. 109–124.

27. Bresolin D. Deterministic Timed Finite State Machines: Equivalence Checking and Expressive Power / D. Bresolin, K. El-Fakih, T. Villa, N. Yevtushenko // Proceedings of the Fifth International Symposium on Games, Automata, Logics and Formal Verification (GANDALF 2014). EPTCS. – 2014. – Vol. 161. – P. 203–216.

28. Bresolin D. Minimizing Deterministic Timed Finite State Machines / D. Bresolin, A. Tvardovskii, N. Yevtushenko, T. Villa, M. Gromov // IFAC-PapersOnline. – 2018. – Vol. 51, № 7. – P. 486–492.

29. Cavalli A. R. Testing Methods for SDL Systems / A. R. Cavalli, B.-M. Chin, K. Chon // Computer Networks and ISDN Systems. – 1996. – Vol. 28, is. 12. – P. 1669–1683.

30. Cavalli A. Fault detection within a component of a system communicating FSMs / A. Cavalli, S. Prokopenko, N. Yevtushenko // Proceedings of the 14th International conference TestCom. – 2002. – P. 317–332.

31. Chow T. S. Test software design modeled by finite state machine / T. S. Chow // IEEE Transactions on Software Engineering. – 1978. – Vol. 4, № 3. – P. 178–187.

32. Dima C. Real-time automata / C. Dima // Journal of Automata, Languages and Combinatorics. – 2001. – 6(1). – P. 3–23.

33. Dorofeeva R. FSM-based conformance testing methods: A survey annotated with experimental evaluation / R. Dorofeeva, K. El-Fakih, S. Maag, A. Cavalli, N. Yevtushenko // Information & Software Technology. – 2010. – 52(12). – P. 1286–1297.

34. El-Fakih K. Fault diagnosis in extended finite state machines / K. El-Fakih, S. Prokopenko, N. Yevtushenko, G. Bochmann // Lecture Notes in Computer Science. – 2003. – Vol. 2644. – P. 197–210.

35. El-Fakih K. Testing Timed Finite State Machines with Guaranteed Fault Coverage / K. El-Fakih, N. Yevtushenko, H. Fouchal // Proceedings of TestCom/FATES 2009. Lecture Notes in Computer Science. – 2009. – Vol. 5826. – P. 66–80.

36. El-Fakih K. Distinguishing Experiments for Timed Nondeterministic Finite State Machines / K. El-Fakih, M. Gromov, N. Shabaldin, N. Yevtushenko // Acta Cybernetica. – 2013. – № 2. – P. 205–222.

37. El-Fakih K. A practical approach for testing timed deterministic finite state machines with single clock / K. El-Fakih, N. Yevtushenko, A. Simão // Science of Computer Programming. – 2014. – Vol. 80. – P. 343–355.

38. El-Fakih K. Adaptive distinguishing test cases of nondeterministic finite state machines: test case derivation and length estimation / K. El-Fakih, N. Yevtushenko, N. Kushik // Formal Aspects of Computing. – 2018. – Vol. 30, is. 2. – P. 319–332.

39. En-Nouaary A. Timed Wp-Method: Testing Real-Time Systems / A. En-Nouaary, R. Dssouli, F. Khendek // IEEE Transactions on Software Engineering. – 2002. – Vol. 28, is. 11. – P. 1023–1038.

40. En-Nouaary A. A Guided Method for Testing Timed Input Output Automata / A. En-Nouaary, R. Dssouli // Proceedings of TestCom 2003. Lecture Notes in Computer Science. – 2003. – Vol. 2644. – P. 211–225.

41. En-Nouaary A. A Boundary Checking Technique for Testing Real-Time Systems Modeled as Timed Input Output Automata / A. En-Nouaary, Hamou-Lhadj Abdelwahab // The Eighth International Conference on Quality Software. – IEEE, 2008. – P. 209–215.

42. Fujiwara S. Test selection based on finite state models / S. Fujiwara, G. v. Bochmann, F. Khendek, M. Amalou, A. Ghedamsi // IEEE Transactions. – 1991. – Vol. 17, №. 6. – P. 591–603.

43. Giambiasi Norbert. An Introduction to Timed Sequential Machines / Giambiasi Norbert // Simulation. – 2014. – Vol. 90, is. 3. – P. 337–352.

44. Gonenc G. A method for the design of fault detection experiments / G. Gonenc // IEEE Trans. Computers. – 1970. – Vol. C-19, № 6. – P. 551–558.

45. Gromov M. Deriving test suites for timed Finite State Machines / M. Gromov, D. Popov, N. Yevtushenko // Proceedings of IEEE East-West Design & Test Symposium (EWDTS’2008). – 2008. – P. 339–343.

46. Gromov M. Distinguishing non-deterministic timed finite state machines / M. Gromov, K. El-Fakih, N. Shabaldina, N. Yevtushenko // International Conference on Formal Techniques for Distributed Systems 2009. Lecture Notes in Computer Science. – 2009. – Vol. 5522. – P. 137–151.

47. Gromov M. Testing Components of Interacting Timed Finite State Machines / M. Gromov, A. Tvardovskii, N. Yevtushenko // Proceedings of IEEE East-West Design & Test Symposium (EWDTS’2016). – 2016. – P. 193–196.

48. Hartmanis J. Algebraic structure theory of sequential machines / J. Hartmanis, R. Stearns. – Prentice-Hall, New York, 1966. – 210 p.

49. Hierons R. M. Adaptive testing of a deterministic implementation against a nondeterministic finite state machine / R.M. Hierons // The Computer Journal. – 1998. – Vol. 41, is. 5. – P. 349–355.

50. Hierons R. M Testing from a Stochastic Timed System with a Fault Model / R.M. Hierons, M.G. Merayo, M. Nunez // Journal of Logic and Algebraic Programming. – 2009. – Vol. 72, is. 8. – P. 98–115.

51. Kondratyeva O. To the Parallel Composition of Timed Finite State Machines / O. Kondratyeva, M. Gromov // Proceedings of the 5th Spring/Summer Young Researches' Colloquium on Software Engineering. – 2011. – P. 94–99.

52. Krichen M. Conformance testing for real-time systems / M. Krichen S. Tripakis // Formal Methods in System Design. – 2009. – Vol. 34, № 3. – P 238–304.

53. Larsen K. G. Testing real-time embedded software using UPPAAL-TRON: an industrial case study / K. G. Larsen, M. Mikucionis, B. Nielsen, A. Skou // Proceedings of the 5th ACM international conference on Embedded software. – 2005. – P. 299–306.

54. Lee D. Testing finite-state machines: state identification and verification / D. Lee, M. Yannakakis // IEEE Transactions on Computers. – 1994. – Vol. 43, is. 3. – P. 306–320.

55. Lee D. Principles and methods of testing finite state machines – a survey / D. Lee, M. Yannakakis // Proceedings of the IEEE. – 1996. – Vol. 84(8). – P. 1090–1123.

56. Lee E. The L-map method for sequential machine state minimization / E. Lee // Kybernetes, 1998. – Vol. 27, is. 9. – P. 1036–1052.

57. Lima L. P. A pragmatic approach to generating test sequences for embedded systems / L. P. Lima, A. R. Cavalli // Testing of Communicating Systems. – IFIP — The International Federation for Information Processing. – 1997. – P. 288–307.

58. Lynch N. Hybrid I/O Automata. Hybrid Systems III. / N. Lynch, R. Segala, F. Vaandrager, H. B. Weinberg // Lecture Notes in Computer Science. – 1996. – Vol. 1066. – P. 496–510.

59. Maslennikova M. Complexity of checking whether two automata are synchronized by the same language / M. Maslennikova // Lecture Notes in Computer Science. – 2014. – Vol. 8614. – P. 306–317.

60. Merayo M. G. Formal Testing from Timed Finite State Machines / M. G. Merayo, M. Nunez, I. Rodriguez // Computer Networks. – 2008. – Vol. 52, № 2. – P. 432–460.

61. Petrenko A. Nondeterministic state machines in protocol conformance testing / A. Petrenko, N. Yevtushenko, A. Lebedev, A. Das // Proceedings of IFIP TC6 Sixth International Workshop on Protocol Test Systems. – 1993. – P. 363–378.

62. Petrenko A. Testing in context: framework and test derivation / A. Petrenko, N. Yevtushenko, G. v. Bochmann, and R. Dssouli // Computer communications. – 1996. – Vol. 19. – P. 1236–1249.

63. Petrenko A. Fault Models for Testing in Context / A. Petrenko, N. Yevtushenko, G. v. Bochmann // Formal Description Techniques IX. – IFIP Advances in Information and Communication Technology. – 1996. – P. 163–178.

64. Petrenko A. Conformance Tests as checking experiments for partial nondeterministic FSM / A. Petrenko, N. Yevtushenko // Proceedings of the 5th International Workshop on Formal Approaches to Testing of Software (FATES). Lecture Notes in Computer Science. – 2005. – Vol. 3997. – P. 118–133.

65. Petrenko A. Adaptive Testing of Deterministic Implementations Specified by Nondeterministic FSMs / A. Petrenko, N. Yevtushenko // Proceedings of ICTSS. – Lecture Notes in Computer Science. – 2011. – Vol. 7019. – P. 162–178.

66. Rho J. Exact and heuristic algorithms for the minimization of incompletely specified state machines. IEEE Trans / J. Rho, G. D. Hachtel, F. Somenzi, R. M. Jacoby // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 1994. – Vol. 13, is. 2. – P. 167–177.

67. Rütz C. An experience report on an industrial case-study about timed model-based testing with UPPAAL-TRON / C. Rütz, J. Schmaltz // Proceedings of the IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops. – 2011. – P. 39–46.

68. Sabnani K. A protocol test generation procedure / K. Sabnani, A. Dahbura // Computer Networks and ISDN Systems. – 1988. – Vol. 15, № 4. – P. 285–297.

69. Shabaldina N. FSMTest-1.0: a manual for researches / N. Shabaldina, M. Gromov // Proceedings of the 13th Intern symposium on IEEE EAST-WEST DESIGN & TEST SYMPOSIUM (EWDTS'15). – 2015. – P. 216–219.

70. Simao A. Generating Reduced Tests for FSMs with Extra States / A. Simao, A. Petrenko, N. Yevtushenko // Lecture Notes in Computer Science. – 2009. – Vol. 5826 – P. 129–145.

71. Spitsyna N. Studying the separability relation between finite state machines / N. Spitsyna, K. El-Fakih, N. Yevtushenko // Software Testing, Verification & Reliability. – 2007. – Vol. 17, is. 4. – P. 227–241.

72. Springintveld J. Testing Timed Automata / J. Springintveld F. Vaandrager P. D'Argenio // Theoretical Computer Science. – 2001. – Vol. 254, is. 1-2. – P. 225–257.

73. Starke P. Abstract Automata / P. Starke. – American Elsevier, 1972. – 419 p.

74. Tretmans J. Model Based Testing with Labelled Transition Systems / J. Tretmans // Formal Methods and Testing. Lecture Notes in Computer Science. – 2008. – Vol. 2791. – P. 1–38.

75. Tripakis S. Folk theorems on the determinization and minimization of timed automata / S. Tripakis // Formal Modeling and Analysis of Timed Systems. Lecture Notes in Computer Science. – 2004. – Vol. 2791. – P. 182–188.

76. Tvardovskii A. Testing Systems of Interacting Timed Finite State Machines with the Guaranteed Fault Coverage / Aleksandr S. Tvardovskii, Maxim L. Gromov, Nina V. Yevtushenko // Proceedings of 17th International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices. – Novosibirsk State Technical University. – Novosibirsk, 2016. – P. 96–99.

77. Tvardovskii Aleksandr. Refining the Specification FSM When Deriving Test Suites w.r.t. the Reduction Relation // Proceedings of the International Conference on Testing Software and Systems (ICTSS 2017). Lecture Notes in Computer Science. – 2017. – Vol. 10533. – P. 333–339.

78. Tvardovskii A. Deriving Tests with Guaranteed Fault Coverage for Finite State Machines with Timeouts / A. Tvardovskii, K. El-Fakih, N. Yevtushenko // Proceedings of the International Conference on Testing Software and Systems (ICTSS 2018). Lecture Notes in Computer Science. – 2018. – Vol. 11146. – P. 149–154.

79. Tvardovskii A. Deriving test suites with guaranteed fault coverage against nondeterministic Finite State Machines with timed guards and timeouts / A. Tvardovskii, N. Yevtushenko // Proceedings of the 13th Spring/Summer Young Researches' Colloquium on Software Engineering. – The Institute for System Programming of the Russian Academy of Sciences. – Saratov, 2019. – P. 198–205.

80. Tvardovskii A. S. On parallel composition of finite state machines with timed guards / A. S Tvardovskii, N. V. Yevtushenko // System Informatics. – 2019. – № 14. – С. 55–64.

81. Tvardovskii A. Experimental Evaluation of Timed Finite State Machine Based Test Derivation / A. Tvardovskii, E. Vinarskii, N. Yevtushenko // Proceedings of International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices. – Novosibirsk State Technical University. – Novosibirsk. – 2019. – P. 102–107.

82. Villa T. Synthesis of Finite State machines: Logic Optimization / T. Villa, T. Kam, R. K. Brayton, A. Sangiovanni Vincentelli. – Springer, 1997. – 520 p.

83. Villa T. The Unknown Component Problem: Theory and Applications / T. Villa, N. Yevtushenko, R. K. Brayton, A. Mishchenko, A. Petrenko, A. L. Sangiovanni Vincentelli. – Springer, 2012. – 311 p.

84. Villa T. Component-Based Design by Solving Language Equations / T. Villa, A. Petrenko, N. Yevtushenko, A. Mishchenko, R. K. Brayton // Proceedings of the IEEE. – 2015. – Vol. 103, is. 11. – P. 2152–2167.

85. Vinogradov R. A. On a class of high-level finite-state automata / R. A. Vinogradov, V. A. Sokolov // Automatic Control and Computer Sciences. – 2010. – Vol. 44, is. 7. – P. 398–406.

86. Vuong S. T. The UIOv-method for protocol test sequence generation / S. T. Vuong, W. W. L. Chan, M. R. Ito // Proceedings of the 2nd IFIP International Workshop on Protocol Test Systems. – 1989. – P. 161–175.

87. Yenigün H. The complexity of checking the existence and derivation of adaptive synchronizing experiments for deterministic FSMs / H. Yenigün, N. Yevtushenko, N. Kushik // Information Processing Letters. – 2017. – Vol. 127. – P. 49–53.

88. Yevtushenko N. Decreasing the length of Adaptive Distinguishing Experiments for Nondeterministic Merging-free Finite State Machines / N. Yevtushenko, N. Kushik // Proceedings of IEEE East-West Design & Test Symposium (EWDTS). – 2015. – P. 338–341.

89. Yevtushenko N. On-the-fly construction of adaptive checking sequences for testing deterministic implementations of nondeterministic specifications / N. Yevtushenko, K. El-Fakih, A. Ermakov // Testing Software and Systems (ICTSS 2016). Lecture Notes in Computer Science. – Vol. 9976. – 2016. – P. 139–152.

90. Zakharov V.A. On the Minimization of Finite State Transducers over Semigroups / V.A. Zakharov, G.G. Temerbekova // Automatic Control and Computer Sciences. – 2017. – Vol. 51, is. 7. – P. 523–530.

91. Zhigulin M. FSM-Based Test Derivation Strategies for Systems with Time-Outs / M. Zhigulin, N. Yevtushenko, S. Maag, A. Cavalli // Proceedings of the 11th International Conference On Quality Software. – Madrid, July 13-14 2011. – P. 141–150.