

Ульянцев Владимир Игоревич

**Генерация конечных автоматов
с использованием программных средств решения задач
выполнимости и удовлетворения ограничений**

Специальность 05.13.11 – Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата технических наук

Санкт-Петербург – 2015

Работа выполнена в Санкт-Петербургском национальном исследовательском университете информационных технологий, механики и оптики

Научный руководитель: доктор технических наук, профессор
Шалыто Анатолий Абрамович

Официальные оппоненты: **Волков Михаил Владимирович,**
доктор физико-математических наук,
профессор, заведующий кафедрой алгебры и
дискретной математики ФГАОУ ВПО
«Уральский федеральный университет имени
первого Президента России Б.Н. Ельцина»

Семенов Александр Анатольевич,
кандидат технических наук, доцент,
заведующий лабораторией дискретного
анализа и прикладной логики Института
динамики систем и теории управления имени
В.М. Матросова Сибирского отделения РАН

Ведущая организация: Федеральное государственное бюджетное
учреждение науки Институт проблем
управления им. В.А. Трапезникова Российской
академии наук

Защита состоится 24 декабря 2015 г. в 15 часов 00 минут на заседании диссертационного совета Д 212.227.06 при Санкт-Петербургском национальном исследовательском университете информационных технологий, механики и оптики по адресу: 197101, г. Санкт-Петербург, Кронверкский просп., д. 49., ауд. 431.

С диссертацией можно ознакомиться в библиотеке Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики по адресу: 197101, г. Санкт-Петербург, Кронверкский просп., д. 49. и на сайте ipro.ifmo.ru.

Автореферат разослан 10 ноября 2015 года.

Ученый секретарь
диссертационного совета
Д 212.227.06, к. ф.-м. н., доц.

Холодова Светлана Евгеньевна

Общая характеристика работы

Актуальность проблемы. Теория автоматов является одним из важнейших разделов дискретной математики и информатики. Многие компоненты аппаратного и программного обеспечения моделируются конечными автоматами, а при проектировании событийных систем все чаще программный код для целевой платформы генерируется по созданной автоматной модели.

Основными достоинствами при проектировании и анализе программ с использованием конечных автоматов являются наглядность их представления в виде диаграмм состояний и повышение уровня автоматизации процесса верификации методом проверки моделей (model checking). Данная проверка применяется при создании ответственных систем, программирование которых недостаточно сопровождать лишь экспертным анализом, статическим и динамическим тестированием.

Для повышения надежности и уменьшения степени влияния человеческого фактора на этапе разработки используемых в ответственных системах автоматных моделей применяются методы их *генерации*. Особенно актуальна разработка методов генерации автоматных моделей для тех задач, в которых их построение вручную затруднительно или невозможно. Существующие методы генерации можно классифицировать по типу обучения (с заранее заданными данными и интерактивные), по типам ограничений на искомый автомат (достижение им заданного значения целевой функции приспособленности, его полное или частичное соответствие заданным примерам поведения или темпоральным свойствам, и т. д.), по типу искомой автоматной модели (детерминированная, недетерминированная, вероятностная).

Настоящая диссертация посвящена решению задач автоматизированной генерации детерминированных автоматных моделей по заранее заданным *примерам поведения*, которым должен соответствовать искомый автомат. Помимо задач генерации и анализа программ такие задачи встречаются в грамматическом выводе, биоинформатике, лингвистике, распознавании речи, робототехнике. Заданные примеры поведения при этом могут быть получены как автоматизированно (протоколы работы существующей программы, биологические последовательности, обработанные аудиозаписи речи, примеры поведения робота, и т. д.), так и построены вручную специалистом.

Известно, что в общем случае задача генерации детерминированного конечного автомата (ДКА) с наименьшим числом состояний по *обучающим словарям* (автомат должен допускать слова из S_+ и не допускать слова из S_-) полна в классе NP. Задача нахождения ДКА с числом состояний, приближенным к наименьшему, также относится к классу NP-полных. Это указывает на актуальность разработки и реализации практически применимых методов генерации. Существует три основных подхода к генерации автоматных моделей по заданным примерам поведения: эвристические алгоритмы слияния состояний (state merging), применение метаэвристических алгоритмов и использование программных средств решения классических NP-полных задач. Первые два подхода являются *неточными* – в общем случае ими не гарантируется нахождение искомого ответа за конечное время или его отсутствие. Рассмотрим более подробно третий подход.

Использование существующего программного средства решения некоторой задачи A возможно после сведения к ней решаемой задачи B – однозначного представления (кодирования) экземпляра x задачи B в виде экземпляра $\mathcal{F}(x)$ задачи A . По определению, каждая NP-полная задача за полиномиальное время сводится к другой, однако построение конкретной функции \mathcal{F} является нетривиальной задачей, а существующие сведения зачастую носят теоретический характер и неэффективны. Данное сведение применяется на практике, если задача A «комбинаторно проще» задачи B , и для решения задачи A существуют эффективные алгоритмы и программные средства. В качестве таких задач используются такие классические задачи, как, например, задачи выполнимости булевой формулы, удовлетворения ограничений, выполнимости формул в теориях (satisfiability modulo theories – SMT), раскраски графа.

Задача *выполнимости булевых формул* (задача выполнимости, Boolean satisfiability – SAT), для которой и была впервые доказана NP-полнота, заключается в нахождении выполняющей подстановки значений переменных для заданной булевой формулы. Для решения этой задачи уже несколько десятилетий продолжается развитие алгоритмов и программных средств, ежегодно проходят конференции, посвященные теоретическим и практическим вопросам их разработки, а также соревнования среди существующих реализаций методов решения. В настоящее время программные средства способны находить за минуты работы персонального компьютера выполняющую подстановку (или продемонстрировать ее отсутствие) для формул, соответствующих практическим задачам. Такие формулы при этом могут состоять из миллионов дизъюнктов.

Указанная производительность программных средств делает актуальной разработку алгоритмов решения иных NP-трудных задач, основанных на сведении к задаче выполнимости (*пропозициональном кодировании*). При таком подходе сначала для экземпляра x решаемой задачи строится соответствующая булева формула $\mathcal{F}(x)$, затем она подается на вход программному средству для решения SAT, и, наконец, в случае нахождения выполняющей подстановки формируется ответ на исходную задачу x . Во-первых, при таком подходе границы применимости алгоритма, основанного на сведении, будут со временем расширяться *без его изменения* пропорционально производительности программных средств для решения SAT. Во-вторых, в отличие от неточных алгоритмов подход обладает точностью – в случае отсутствия искомого решения программное средство, основанное на *поиске с возвратом* (backtracking), сообщит о невыполнимости соответствующей булевой формулы.

Аналогичным образом используются программные средства решения задачи *удовлетворения ограничений* (обобщенная задача выполнимости, constraint satisfaction problem – CSP), которая заключается в подборе таких допустимых значений переменных x_1, \dots, x_n (не обязательно булевых), что выполняются все заданные ограничения на эти переменные. Существующие программные средства поддерживают ограничения многих типов, что позволяет использовать выразительную силу языка CSP и представлять комбинаторные задачи более наглядно и «естественно», в отличие, например, от языка SAT – булевых формул.

Также стоит отметить, что некоторые возникающие на практике ограничения на целочисленные переменные затруднительно выразить в виде булевой формулы полиномиальной длины, что актуализирует для данных задач применение

программных средств решения CSP. Задача SAT является частным случаем задачи CSP, однако за счет простоты модели позволяет применять более эффективные эвристики решения. Таким образом, достоинством сведения к SAT является более высокая эффективность (на настоящий момент) программных средств, а достоинствами сведения к CSP являются наглядность представления в виде ограничений и применимость для более широкого класса задач за счет выразительной силы языка.

Вернемся к задачам генерации конечных автоматов, решаемых в настоящей диссертации. Известно сведение, строящее булеву формулу $\mathcal{F}(x)$, выполняемую только в случае существования ДКА с заданным числом состояний C , удовлетворяющего заданным обучающим словарям S_+ и S_- (в данном случае $x = \langle S_+, S_-, C \rangle$). Помимо гарантированной точности генерации, которой не обладают эвристические и метаэвристические алгоритмы решения задачи, подход, основанный на сведении, превосходит их по скорости работы на практических задачах.

В настоящее время известно лишь описанное в предыдущем абзаце применение программных средств решения SAT для генерации конечных автоматов. В то же время для других задач генерации конечных автоматов по примерам поведения известны только неточные алгоритмы решения. Такими задачами являются, например, задача генерации ДКА по *зашумленным* обучающим словарям (с конечным числом ошибок в метках допуска/недопуска слов), а также задачи генерации *управляющих конечных автоматов* по сценариям работы. Сценарии могут быть как безошибочными, так и зашумленными – содержать конечное число ошибочных последовательностей выходных воздействий. Решение таких задач актуально не только при моделировании, но и при построении программных систем с использованием технологии *автоматного программирования*, а также систем визуального проектирования управляющих программ (например, *MATLAB/Stateflow*, *IBM Rational Rhapsody*).

Из изложенного следует, что развитие существующих и разработка новых точных методов генерации конечных автоматов по примерам поведения с использованием программных средств решения SAT и CSP **актуальны**.

В соответствии с паспортом специальности 05.13.11 – «Математическое обеспечение вычислительных машин, комплексов и компьютерных сетей» диссертация относится к области исследований «1. Модели, методы и алгоритмы проектирования и анализа программ и программных систем, их эквивалентных преобразований, верификации и тестирования».

Цель диссертационной работы – разработка точных методов генерации конечных автоматов по примерам поведения с использованием программных средств решения задач выполнимости и удовлетворения ограничений.

Основные задачи диссертационной работы:

1. Доказать NP-трудность задачи генерации управляющих конечных автоматов заданного размера по сценариям работы. Данный теоретический результат указывает на целесообразность разработки эффективных методов для задач генерации управляющих автоматов, основанных на сведении к задаче CSP.
2. Разработать точные методы генерации ДКА по безошибочным и зашумленным обучающим словарям с использованием программных средств решения задачи выполнимости. Реализовать методы в программном средстве *DFAInducer* с

открытым кодом. Выполнить сравнение программных средств *DFAIInducer* и *DFASAT* (Делфтский технический университет, Нидерланды).

3. Разработать точные методы генерации управляющих автоматов по безошибочным и зашумленным сценариям работы с использованием программных средств решения задачи удовлетворения ограничений. Реализовать методы в программном средстве *EFSMTools* с открытым кодом.
4. Внедрить разработанные методы генерации ДКА после их публикации в *DFASAT*.

Научная новизна. В работе получены следующие новые научные результаты, которые выносятся на защиту.

1. Доказательство NP-трудности задачи построения управляющих конечных автоматов по сценариям работы.
2. Точные методы генерации ДКА по безошибочным и зашумленным обучающим словарям. Новизна метода генерации по безошибочным словарям заключается в разработанных *предикатах нарушения симметрии*, обеспечивающих статистически значимое преимущество скорости работы по сравнению с *DFASAT*. Метод построения ДКА по зашумленным словарям является *точным*, в отличие от известных.
3. Точные методы генерации управляющих конечных автоматов по сценариям работы. Метод генерации управляющих автоматов по безошибочным сценариям работы включает в себя новые алгоритмы построения дерева сценариев и его графа совместимости. Известные методы генерации, как по безошибочным, так и по зашумленным сценариям, являются неточными.

Методология и методы исследования. Методологическую основу диссертации составили принципы формализации, обобщения, дедуктивного и индуктивного обоснования утверждений, проведение экспериментальных исследований и анализ их результатов. В работе используются методы теории автоматов, дискретной математики, теории сложности и математической статистики.

Достоверность научных положений, выводов и практических рекомендаций, полученных в диссертации, подтверждается корректным обоснованием постановок задач, точной формулировкой критериев, а также результатами вычислительных экспериментов по использованию предложенных в диссертации методов.

Теоретическое значение работы состоит в доказанной NP-трудности задачи генерации управляющих конечных автоматов по сценариям работы и разработанных сведениях задач генерации конечных автоматов к задачам SAT и CSP.

Практическое значение работы. Разработанные методы и реализованные программные средства позволяют автоматизированно и *точно* решать задачи генерации конечных автоматов. С их использованием можно повысить надежность процессов проектирования и анализа программ, уменьшить влияние человеческого фактора.

Внедрение результатов работы. Разработанное сведение задачи генерации ДКА по обучающим словарям было использовано при реализации программного средства *DFASAT*. Также результаты использовались в учебном процессе кафедры «Компьютерные технологии» Университета ИТМО в рамках курса «Теория автоматов и программирование», при руководстве и выполнении выпускных квалификационных работ студентов кафедры.

Апробация результатов работы. Основные результаты диссертационной работы докладывались на следующих научных и научно-практических конференциях:

- Всероссийская научная конференция по проблемам информатики СПИСОК. 2011-2014, Матмех СПбГУ;
- Всероссийский конгресс молодых ученых. 2011-2013, Университет ИТМО;
- Всероссийское совещание по проблемам управления. 2014, Институт проблем управления РАН, Москва;
- 14th IFAC Symposium «Information Control Problems in Manufacturing – INCOM'12». 2012, Бухарест, Румыния;
- International Conference on Machine Learning and Applications. 2011 – Гонолулу, США. 2014 – Детройт, США;
- 9th International Conference on Language and Automata Theory and Applications. 2015, Ницца, Франция.

Личный вклад автора. Решение задач диссертации, разработанные методы и алгоритмы принадлежат лично автору.

Публикации. По теме диссертации опубликовано 14 работ, в том числе шесть публикаций в изданиях из перечней ВАК или Scopus.

Свидетельства о регистрации программ для ЭВМ. В рамках диссертационной работы получены четыре свидетельства о регистрации программ для ЭВМ:

- № 2012 616462 от 18.07.2012 г. «Программное средство для построения графа совместимости вершин дерева сценариев работы программы»;
- № 2012 660438 от 20.11.2012 г. «Программное средство для построения КНФ-формулы по графу совместимости вершин дерева сценариев работы программы»;
- № 2013 619840 от 17.10.2013 г. «Программный комплекс для построения и тестирования управляющих конечных автоматов»;
- № 2015 619224 от 27.08.2015 г. «Программное средство преобразования полученных методами машинного обучения управляющих автоматов в формат *MATLAB/Stateflow*».

Участие в научно-исследовательских работах. Результаты диссертации использовались при выполнении следующих НИР, которыми руководил автор:

- «Разработка методов автоматизированного построения надежного программного обеспечения на основе автоматного подхода по обучающим примерам и темпоральным свойствам» (грант РФФИ «Мой первый грант», 2014, 2015);
- «Построение управляющих автоматов с помощью методов решения задачи удовлетворения ограничений» (программа «У.М.Н.И.К.», 2012).

Полученные результаты также использовались при проведении НИР:

- «Разработка метода машинного обучения на основе алгоритмов решения задачи о выполнимости булевой формулы для построения управляющих конечных автоматов» (ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009-2013 годы, 2011-2013);
- «Разработка методов построения управляющих конечных автоматов по обучающим примерам на основе решения задачи удовлетворения ограничений»

(ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009-2013 годы, 2012, 2013);

- «Технология разработки программного обеспечения систем управления ответственными объектами на основе методов машинного обучения и конечных автоматов» (научно-исследовательская работа в рамках проектной части государственного задания в сфере научной деятельности, Минобрнауки, 2014-2016).

Структура диссертации. Диссертация изложена на 129 страницах и состоит из введения, пяти глав, заключения и двух приложений. Список источников содержит 101 наименование. Работа проиллюстрирована 20 рисунками и тремя таблицами.

Содержание работы

В первой главе приводится обзор работ, посвященных задачам выполнимости булевой формулы и удовлетворения ограничений, алгоритмам и методам их решения, а также конечным автоматам и методам их генерации по примерам поведения.

Описываются основные методы решения задачи SAT, гарантирующие нахождение искомого ответа за конечное время. Эти методы основаны на алгоритме *DPLL* – поиске с возвратом, ускоряющем нахождение решения за счет применения правил распространения переменных и исключения «чистых» переменных. В современных программных средствах решения задачи SAT используется стратегия управляемого конфликтами обучения дизъюнктов (*conflict-driven clause learning, CDCL*). На указанных алгоритмах основаны и современные средства решения задачи удовлетворения ограничений CSP.

Анализируются результаты соревнований *SAT Competition 2014* и *MiniZinc Challenge 2015*. Среди программных средств решения SAT для применения в диссертации были выбраны *lingeling* и *CryptoMiniSat*, а для решения CSP – *Opturion CPX* и *OR-tools*.

Проводится анализ работ, посвященных методам генерации конечных автоматов по примерам поведения. Рассматриваются детерминированные конечные автоматы (ДКА) и управляющие конечные автоматы (управляющие автоматы) – модели, соответствующие многим компонентам аппаратного и программного обеспечения. На рисунке 1а приведен пример ДКА, моделирующего работу блока обработки входящих звонков телефона, на рисунке 1б – пример управляющего автомата для торгового аппарата.

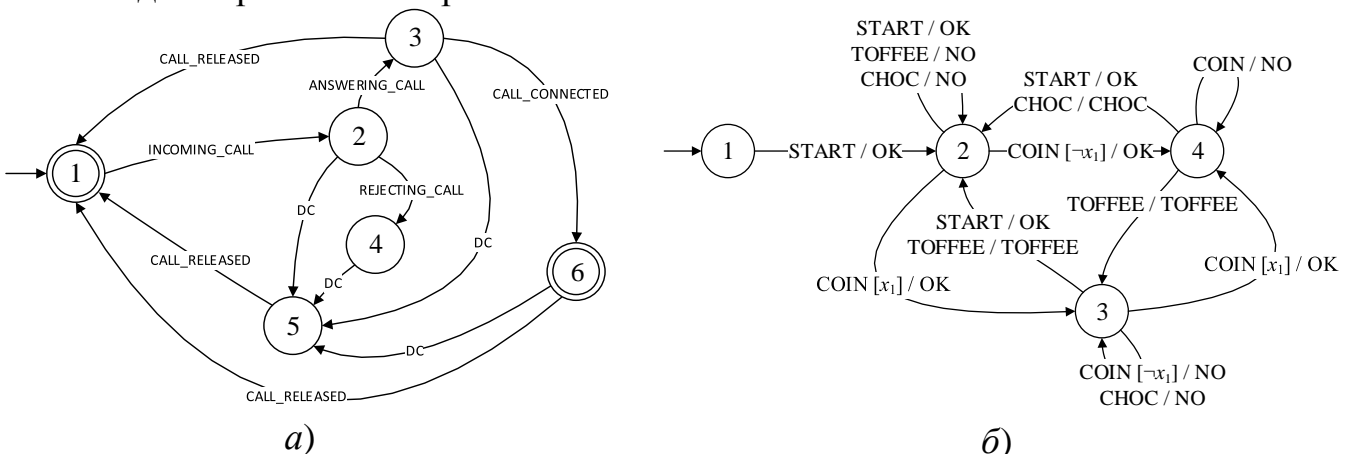


Рисунок 1 – Примеры ДКА (а) и управляющего автомата (б)

Для конечных управляющих автоматов примерами поведения являются *сценарии работы* – последовательности троек $T_i = \langle e_i; f_i; A_i \rangle$, где e_i – входное событие, f_i – булева формула от входных переменных, задающая *охранное условие*, A_i – последовательность выходных воздействий. Управляющий конечный автомат торгового аппарата (рисунок 1б) *удовлетворяет*, например, следующим сценариям:

- $\langle \text{START}; 1; \text{OK} \rangle, \langle \text{COIN}; x_1; \text{OK} \rangle, \langle \text{COIN}; \neg x_1; \text{NO} \rangle, \langle \text{COIN}; x_1; \text{OK} \rangle$;
- $\langle \text{START}; 1; \text{OK} \rangle, \langle \text{START}; 1; \text{OK} \rangle, \langle \text{TOFFEE}; 1; \text{NO} \rangle, \langle \text{CHOC}; 1; \text{NO} \rangle$;
- $\langle \text{START}; 1; \text{OK} \rangle, \langle \text{COIN}; \neg x_1; \text{OK} \rangle, \langle \text{TOFFEE}; 1; \text{TOFFEE} \rangle$.

Выделяются три группы методов генерации автоматов, основанных на: эвристических алгоритмах слияния состояний (State merging); метаэвристических алгоритмах; сведении задач генерации к классическим NP-полным задачам. Первые два подхода являются неточными, а третий – точным. Отмечается, что известны только точные методы генерации ДКА по незашумленным обучающим словарям, основанные на сведении к задачам SAT и раскраски графов. При этом самым производительным точным методом является *DFASAT*, задача улучшения которого стоит в настоящей диссертации. Отмечается, что для решения задач генерации ДКА по зашумленным обучающим словарям, а также управляющих автоматов по сценариям работы не известно точных методов генерации.

На основании результатов обзора ставятся цели диссертационного исследования и формулируются задачи генерации конечных автоматов, решаемые в диссертации.

Во второй главе проводится доказательство принадлежности задач генерации управляющих автоматов классу NP-трудных. Данной задаче соответствует язык L – множество пар $\langle S, C \rangle$ (S – набор сценариев работы программы, C – натуральное число, записанное в унарной системе счисления), для каждой из которых существует управляющий автомат \mathcal{A} размера C , удовлетворяющий сценариям из S .

Доказана принадлежность языка L классу NP-трудных. Доказательство проведено с использованием результата из теории грамматического вывода: задача генерации ДКА с заданным числом состояний C по обучающим словарям S_+ и S_- является NP-полной. Также приводится условие полноты языка L задачи генерации управляющих автоматов в классе NP – необходимо, чтобы содержащиеся в сценариях работы охранные условия были или заданы таблицами истинности, или имели ограниченную константой длину.

В третьей главе описываются разработка, реализация и численные экспериментальные исследования двух методов генерации ДКА по обучающим словарям.

Разработан точный метод генерации ДКА с наименьшим числом состояний по безошибочным обучающим словарям S_+ и S_- . Метод основан на сведении задачи генерации к SAT и состоит из следующих шагов.

1. Построение префиксного дерева \mathcal{T} по S_+ и S_- (заданных над алфавитом Σ).
2. Построение графа совместимости \mathcal{G} для дерева \mathcal{T} .
3. Генерация булевой формулы $\mathcal{F}_{\text{SAT}}(\mathcal{T}, \mathcal{G}, C)$ для текущего числа состояний ДКА C (увеличивающегося до выполнимости формулы).
4. Запуск стороннего программного средства решения SAT для нахождения выполняющей подстановки сгенерированной формулы \mathcal{F}_{SAT} и генерация ДКА в случае существования подстановки.

Основным отличием разработанного метода от лучшего известного в мире метода *DFASAT* решения задачи генерации ДКА является подход к сокращению пространства поиска. Предложены *предикаты нарушения симметрии*, задающие BFS-нумерацию ДКА: номера состояний автомата должны соответствовать порядку обхода автомата в ширину. При таком подходе для каждого класса эквивалентных по отношению изоморфизма автоматов будет рассмотрен только один представитель. Данные предикаты добавляются к формуле сведения \mathcal{F}_{SAT} и не влияют на ее разрешимость: если формула выполнима, то после добавления предикатов она останется таковой.

Вводятся дополнительные переменные трех типов: переменные *наличия перехода* $t_{i,j}$, переменные *предка в BFS-обходе* $p_{j,i}$, переменные *наименьшего символа* $m_{l,i,j}$. С использованием данных переменных, а также переменных $y_{a,i,j}$, задающих функцию переходов генерируемого автомата, составляется набор дополнительных предикатов $\mathcal{F}_{\text{SB}} = \mathcal{F}_t \wedge \mathcal{F}_p \wedge \mathcal{F}_{\text{ALO}(p)} \wedge \mathcal{F}_{\text{BFS}(p)} \wedge \mathcal{F}_m \wedge \mathcal{F}_{\text{BFS}(m)}$, где:

1. $\mathcal{F}_t = \bigwedge_{1 \leq i < j \leq c} (t_{i,j} \Leftrightarrow \bigvee_{l \in \Sigma} y_{l,i,j})$;
2. $\mathcal{F}_p = \bigwedge_{1 \leq i < j \leq c} (p_{j,i} \Leftrightarrow t_{i,j} \wedge \bigwedge_{1 \leq k < i} \neg t_{k,j})$;
3. $\mathcal{F}_{\text{ALO}(p)} = \bigwedge_{1 < j \leq c} \text{ALO}(\{p_{j,i}\}_{i=1}^{j-1})$;
4. $\mathcal{F}_{\text{BFS}(p)} = \bigwedge_{1 \leq k < i < j < c} (p_{j,i} \Rightarrow \neg p_{j+1,k})$;
5. $\mathcal{F}_m = \bigwedge_{1 \leq i < j \leq c, l \in \Sigma} (m_{l,i,j} \Leftrightarrow y_{l,i,j} \wedge \bigwedge_{l^* \in \Sigma, l^* < l} \neg y_{l^*,i,j})$;
6. $\mathcal{F}_{\text{BFS}(m)} = \bigwedge_{1 \leq i < j < c} \bigwedge_{l, l^* \in \Sigma, l^* < l} (p_{j,i} \wedge p_{j+1,i} \wedge m_{l,i,j} \Rightarrow \neg m_{l^*,i,j+1})$.

Выражение $\text{ALO}(X)$ обозначает ограничение вида *at-least-one*: в заданном наборе переменных X хотя бы одна переменная должна выполняться. Суммарное число добавляемых дизъюнктов составляет $\mathcal{O}(C^2|\Sigma|^2 + C^3)$. Они используют $\mathcal{O}(C^2|\Sigma|)$ дополнительных переменных трех типов. Для случая двухсимвольного алфавита ($|\Sigma| = 2$) предложен упрощенный вариант предикатов.

На основе описанного метода разработан точный метод генерации ДКА по зашумленным (не более K ошибочных пометок) обучающим словарям S_+ и S_- . Наличие ошибок не позволяет использовать граф совместимости и, соответственно, не позволяет применять подход к нарушению симметрии, основанный на клике данного графа и используемый в методе *DFASAT*. Разработанные предикаты нарушения симметрии не чувствительны к наличию ошибок в пометках слов и использованы при разработке метода.

Предложены *предикаты обработки ошибочных пометок*, идея которых заключается в следующем. Для каждой вершины префиксного дерева \mathcal{T} с допускающей или недопускающей меткой будем хранить булеву переменную f_v , истинность которой соответствует возможности того, что метка вершины v ошибочна. Ошибок может быть допущено не более K , и предикаты обработки ошибочных пометок ограничивают сверху совокупное число истинных значений переменных f_v . Механизм ограничения оперирует с дополнительными переменными $r_{k,v}$ и $o_{k,v}$, которые хранят положения ошибочных меток в префиксном дереве. Предикаты состоят из $\mathcal{O}((C + K) \cdot (|S_+| + |S_-|))$ дизъюнктов. На рисунке 2 приведена схема взаимодействия указанных переменных при $K = 2$.

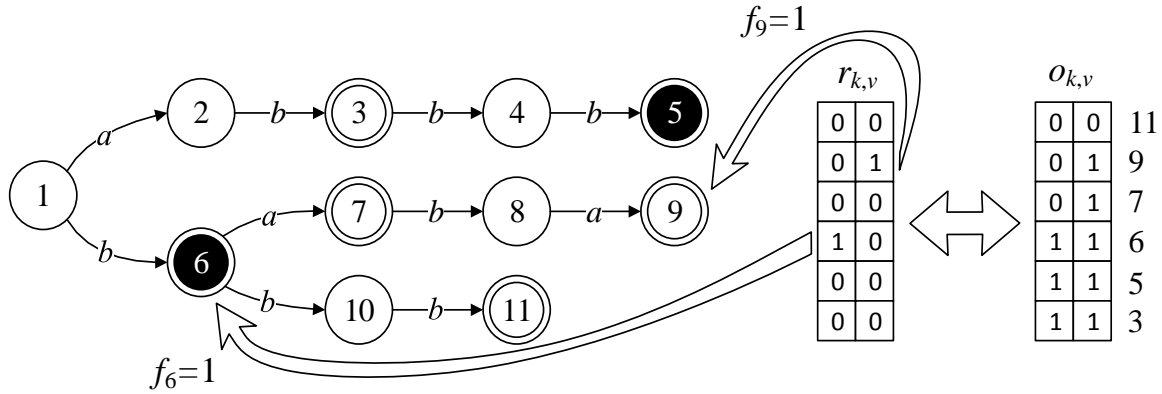


Рисунок 2 – Схема взаимодействия переменных f_v , $r_{k,v}$ и $o_{k,v}$

Предложенные методы генерации ДКА реализованы на языке *Java*. Создано программное средство *DFAInducer* с открытым кодом. Средство в качестве параметра принимает путь к программному средству решения задачи SAT. Время работы *DFAInducer* напрямую зависит от производительности данного средства. Таким образом, с развитием методов решения SAT будет уменьшаться время работы *DFAInducer*.

Проведены экспериментальные исследования разработанных и реализованных методов генерации ДКА и получен следующий практический результат: время работы предложенного метода генерации ДКА по незашумленным данным значительно меньше времени работы метода *DFASAT*, являющегося лучшим из известных методов решения этой задачи. Для анализа времени работы было проведено 16000 запусков разработанного средства на сгенерированных данных. Каждый из экземпляров задачи был решен менее чем за час работы (здесь и далее приводится время работы одного ядра процессора с тактовой частотой 2,4 ГГц). Медианное время работы *DFASAT* превышало час при $C > 13$. На рисунке 3 приведены ящичные диаграммы для вычисленных времен работы средства *DFAInducer*.

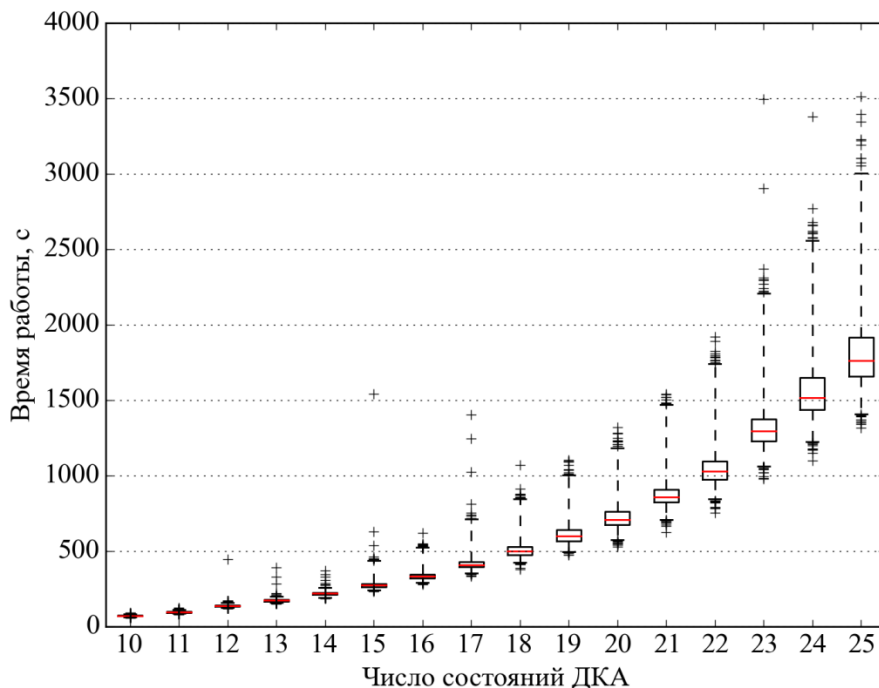


Рисунок 3 – Ящичные диаграммы для времен работы предложенного метода генерации ДКА

Приведенные диаграммы показывают, что предложенный метод решает каждую из задач генерации при $C \leq 14$ менее чем за 10 минут работы персонального компьютера. Метод применим при $C \leq 25$ и доступном часе процессорного времени. Быстрая генерация автоматов актуальна при частом обновлении обучающих словарей.

В четвертой главе описываются разработка, реализация и проведенные экспериментальные исследования двух методов генерации управляющих конечных автоматов по сценариям работы S .

Разработан точный метод генерации управляющих конечных автоматов по безошибочным сценариям работы. Метод основан на сведении к задаче удовлетворения ограничений и имеет структуру, аналогичную методу генерации ДКА по обучающим словарям. Разработаны алгоритмы построения дерева сценариев \mathcal{T} и его графа совместимости \mathcal{G} . Предложено сведение, состоящее из $\mathcal{O}(C(C + |\Sigma_X|) + |\mathcal{T}|^2)$ ограничений на переменные цвета x_v , переменные переходов $y_{i,e}$ и переменные использования переходов $u_{i,e}$:

1. $x_v \neq x_u$ для всех $(v, u) \in \mathcal{G}$.
2. $(x_v = i) \Rightarrow (x_u = y_{i,e})$ для $i \in 1 \dots C$ и каждого ребра $(v, u) \in \mathcal{T}$.
3. $(u_{i,e} = 1) \Leftrightarrow \bigvee_{v \in V(e)} (x_v = i)$ для $i \in 1 \dots C, e \in \Sigma_X$.
4. $\sum_{e \in \Sigma_X: \text{event}(e)=e_o} (u_{i,e} \cdot c(e)) \in \{0, 2^m\}$ для $i \in 1 \dots C, e_o \in \Sigma$.

Ограничения последнего указанного вида задают требование полноты генерируемого автомата: для любой последовательности входных данных система поставит в соответствие выходную последовательность, подтвержденную заданными сценариями работы. Именно для задания такого вида требований и целесообразно использовать язык CSP, а не SAT. Также модифицируются и используются предикаты нарушения симметрии, задающие BFS-нумерацию.

Рассмотрен пример работы предложенного метода на примере разработки автомата управления торговым аппаратом по 25 сценариям, некоторые из которых приведены выше. Для них на первом шаге сгенерировано дерево сценариев \mathcal{T} , состоящее из 340 вершин. На втором шаге для \mathcal{T} сгенерирован граф совместимости \mathcal{G} , содержащий 12884 ребра. На третьем шаге при $C = 4$ по \mathcal{T} и \mathcal{G} сгенерировано 15240 ограничений на 1870 целочисленных переменных (в том числе упомянутые переменные цвета x_v , переменные переходов $y_{i,e}$ и переменные использования переходов $u_{i,e}$). После нахождения программным средством значений переменных (время работы средства *Opturion CPX* составило 8,7 секунд) по ним был построен управляющий автомат, приведенный на рисунке 1б.

Разработан точный метод генерации управляющих конечных автоматов по сценариям работы, которые могут содержать зашумленные последовательности выходных воздействий. Приведем пример сценария работы, которому автомат управления торговым аппаратом удовлетворяет только после исправления одного выходного воздействия: $\langle \text{START}; 1; \text{OK} \rangle, \langle \text{COIN}; \neg x_1; \text{OK} \rangle, \langle \text{TOFFEE}; 1; \text{NO} \rangle$. В последнем элементе сценария вместо действия TOFFEE «выдать ириску» содержится действие NO «ничего не выдать».

Метод, в отличие от ранее предложенных, не использует древовидное представление входных данных. Основой метода является сведение к задаче CSP. При этом суммарное число ограничений на семь типов переменных составляет $\mathcal{O}(C^2 + |V|)$ (где $|V|$ – суммарная длина сценариев S).

Разработанные методы реализованы в программном средстве *EFSMTools* с открытым кодом. Реализация проведена на языках программирования *Java* и *Python*, ограничения на целочисленные переменные записаны на языке *MiniZinc*. Для решения задачи удовлетворения ограничений используется средство *Opturion CPX*, являющееся одним из лучших в настоящее время.

Экспериментальные исследования метода генерации управляющих автоматов по безошибочным сценариям работы показали статистически меньшее время работы предложенного метода по сравнению с точным алгоритмом поиска с возвратом при числе состояний автомата $C \in [6, 13]$. Для сравнения было проведено 1000 экспериментов. Точный метод генерации управляющих автоматов по зашумленным сценариям работы был экспериментально сравнен с точным алгоритмом поиска с возвратом и метаэвристическим методом *MuACO*. Для сравнения каждый метод был запущен 3000 раз. Результаты вычислительных экспериментов показали, что предложенный метод работает значительно быстрее поиска с возвратом при росте размерности задачи и не уступает при этом *MuACO*.

Пятая глава посвящена внедрению результатов диссертации. Описанные предикаты нарушения симметрии для задания требования BFS-нумерации были использованы при разработке программного средства *DFASAT*. Средство разрабатывается М. Хеуле (Техасский Университет в Остине, США) и С. Вервером (Делфтский технический университет, Нидерланды). В рамках диссертации был сделан доклад на конференции по языкам, теории и приложениям автоматов *LATA 2015* (Ницца, Франция). После выступления диссертанта и публикации работы «BFS-Based Symmetry Breaking Predicates for DFA Identification» был установлен контакт с упомянутыми исследователями и в дальнейшем предлагаемые предикаты нарушения симметрии были внедрены в программное средство *DFASAT*. Получен акт внедрения из Делфтского технического университета.

Результаты диссертации использовались в учебном процессе кафедры «Компьютерные технологии» Университета ИТМО при руководстве двумя бакалаврскими и выполнении двух магистерских выпускных квалификационных работ студентов, а также в рамках курса «Теория автоматов и программирование».

Заключение

В диссертационном исследовании получены следующие результаты.

1. Теоретическое доказательство NP-трудности задачи генерации управляющих конечных автоматов по сценариям работы.
2. Точные методы генерации детерминированных конечных автоматов по безошибочным и зашумленным обучающим словарям. Методы основаны на сведении к задаче выполнимости булевой формулы. Создано программное средство *DFAInducer* с открытым кодом, реализующее разработанные методы. В случае безошибочных данных время работы предложенного метода меньше времени работы существующих. Для генерации автоматов по зашумленным данным ранее точных методов не предлагалось.
3. Точные методы генерации управляющих конечных автоматов по безошибочным и зашумленным сценариям работы. В основе методов лежит сведение к задаче удовлетворения ограничений. Методы реализованы в программном средстве *EFSMTools* с открытым кодом. Вычислительные экспериментальные исследования показали преимущество по сравнению с поиском с возвратом.

4. Внедрение разработанных методов, осуществленное при разработке программного средства *DFASAT* и выполненное после публикации работы диссертанта «BFS-Based Symmetry Breaking Predicates for DFA Identification». Имеется акт внедрения из Делфтского технического университета. Также результаты диссертации использовались в учебном процессе кафедры «Компьютерные технологии» Университета ИТМО.

Полученные результаты рекомендуется использовать для решения практических задач генерации конечных автоматов. Перспективы дальнейшей разработки темы включают в себя развитие предложенных методов, их модификацию для применения к другим автоматным моделям.

Статьи в рецензируемых изданиях из перечней ВАК или Scopus

1. *Панченко Е. В., Ульянов В. И.* Применение методов решения задачи о выполнимости квантифицированной булевой функции для построения управляющих конечных автоматов по сценариям работы и темпоральным свойствам // Научно-технический вестник информационных технологий, механики и оптики. – 2013. – 4(86). – С. 151–153. – 0,188 п. л. / 0,1 п. л.
2. *Ульянцев В. И., Царев Ф. Н.* Применение методов решения задачи о выполнимости булевой формулы для построения управляющих конечных автоматов по сценариям работы // Научно-технический вестник информационных технологий, механики и оптики. – 2012. – 1(77). – С. 96–100. – 0,313 п. л. / 0,16 п. л.
3. *Chivilikhin D., Ulyantsev V., Shalyto A.* Combining Exact And Metaheuristic Techniques For Learning Extended Finite-State Machines From Test Scenarios and Temporal Properties // Proceedings of the 13th International Conference on Machine Learning and Applications (ICMLA'14). – 2014. – P. 350–355. – 0,375 п. л. / 0,15 п. л.
4. *Ulyantsev V., Tsarev F.* Extended Finite-State Machine Induction using SAT-Solver // Proceedings of the Tenth International Conference on Machine Learning and Applications. – IEEE, 2011. – Vol. 2. – P. 346-349. – 0,25 п. л. / 0,125 п. л.
5. *Ulyantsev V., Zakirzyanov I., Shalyto A.* BFS-Based Symmetry Breaking Predicates for DFA Identification // Language and Automata Theory and Applications. – Springer, 2015. – P. 611-622. – (Lecture Notes in Computer Science; 8977). – 0,375 п. л. / 0,2 п. л.
6. *Ulyantsev V., Tsarev F.* Extended Finite-State Machine Induction using SAT-Solver // Proceedings of the 14th IFAC Symposium “Information Control Problems in Manufacturing – INCOM'12”. – IFAC, 2012. – P. 512–517. – 0,375 п. л. / 0,25 п. л.

Другие публикации по теме

7. *Ведерников Н. В., Демьянюк В. Ю., Кротков П. А., Ульянов В. И., Шалыто А. А.* Применение методов машинного обучения для автоматизированного построения управляющих автоматов в высокоуровневых средствах проектирования систем // Труды XII Всероссийского совещания по проблемам управления ВСПУ-2014. – М.: Институт проблем управления им. В.А. Трапезникова РАН, 2014. – С. 3159–3166. – 0,5 п. л. / 0,2 п. л.
8. *Ульянцев В. И.* Применение методов решения задачи о выполнимости булевой формулы для построения управляющих конечных автоматов по сценариям работы // Аннотированный сборник научно-исследовательских выпускных

квалификационных работ бакалавров и специалистов НИУ ИТМО. – СПб.: НИУ ИТМО, 2011. – С. 35–38. – 0,25 п. л.

9. *Ульянцев В. И.* Применение методов решения задачи удовлетворения ограничениям для построения управляющих конечных автоматов по сценариям работы // Сборник тезисов докладов конгресса молодых ученых, Выпуск 1. Труды молодых ученых. – СПб.: НИУ ИТМО, 2012. – С. 230, 231. – 0,125 п. л.
10. *Ульянцев В. И.* Метод построения управляющих конечных автоматов по сценариям работы на основе решения задачи удовлетворения ограничений // Научные работы участников конкурса «Молодые ученые НИУ ИТМО» 2012 года. – СПб.: НИУ ИТМО, 2013. – С. 256–260. – 0,313 п. л.
11. *Ульянцев В. И., Царев Ф. Н., Шалыто А. А.* Применение методов решения задачи о выполнимости булевой формулы для построения управляющих конечных автоматов по сценариям работы // Сборник докладов XIV Международной конференции по мягким вычислениям и измерениям (SCM'2011). – 2011. – Т. 2. – С. 69–75. – 0,438 п. л. / 0,2 п. л.
12. *Ульянцев В. И., Царев Ф. Н.* Применение методов решения задачи удовлетворения ограничениям для построения управляющих конечных автоматов по сценариям работы // Список-2012: материалы всероссийской научной конференции по проблемам информатики. – СПб.: ВВМ, 2012. – С. 444, 445. – 0,125 п. л. / 0,07 п. л.
13. *Ульянцев В. И., Вихарев А. К., Шалыто А. А.* Применение алгоритма EDSM для построения управляющих конечных автоматов по сценариям работы // Сборник докладов XIV Международной конференции по мягким вычислениям и измерениям SCM'2011. – 2011. – Т. 2. – С. 76–80. – 0,313 п. л. / 0,16 п. л.
14. *Ульянцев В. И., Царев Ф. Н.* Применение методов решения задачи о выполнимости булевой формулы для построения управляющих конечных автоматов по сценариям работы // Список-2011: материалы межвузовской научной конференции по проблемам информатики. – СПб.: ВВМ, 2011. – С. 356–358. – 0,188 п. л. / 0,1 п. л.

Тиражирование и брошюровка выполнены в учреждении
«Университетские телекоммуникации»
197101, Санкт-Петербург, Саблинская ул., 14
Тел. (812) 233 46 69 Объем 1,0 у.п.л.
Тираж 100 экз.