Национальный исследовательский университет ИТМО
(Университет ИТМО)

На правах рукописи

Чухарев Константин Игоревич

Методы декомпозиции задачи булевой выполнимости для синтеза и верификации моделей автоматных программ

Специальность 2.3.5

Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей

Научный доклад об основных результатах диссертации
на соискание учёной степени
кандидата технических наук

Научный руководитель:
канд. техн. наук, доцент
Семёнов Александр Анатольевич

Санкт-Петербург, Россия
2024

Реферат

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. В современном информационном обществе автоматизированные системы стали неотъемлемой частью различных областей науки и техники, что подчеркивает важность проблемы эффективной верификации и синтеза автоматных программ. Под *автоматизированными системами* понимается широкий класс объектов, решающих вычислительные задачи путем выполнения конкретных функций. Они находят применение в таких разнообразных областях, как программирование, инженерия, робототехника, управление производственными процессами и многое другое. Для анализа и разработки таких систем используются *абстрактные модели*, которые позволяют формализовать их поведение и свойства.

Одним из ключевых понятий в этой области является понятие *состояния* вычисляющей модели, которое было впервые введено Аланом Тьюрингом[1]. Это понятие является основой для построения абстрактных моделей и играет важную роль в различных прикладных областях, таких как разработка языков программирования, трансляторов, компиляторов, микроконтроллеров и микропроцессоров.

Концепция «*автоматного программирования*», предлагает подход к построению программ, основанный на конечно-автоматной парадигме. Этот подход предполагает разбиение программы на более простые модули, которые могут рассматриваться как отдельные вычислительные единицы, находящиеся в различных состояниях. При этом задачи верификации и синтеза для автоматных программ сводятся к аналогичным задачам для формальных моделей.

В настоящей диссертации рассматриваются два класса таких моделей: конечные автоматы и булевы схемы. Несмотря на различия между этими моделями, между ними существует тесная взаимосвязь: конечные автоматы описывают функции, которые могут принимать входные данные произвольной длины, в то время как булевы схемы оперируют данными конкретной длины. Соответственно, каждой функции, задаваемой конечным автоматом, соответствует счетное число функций, задаваемых булевыми схемами. Для решения конкретных вычислительных задач приходится рассматривать конечные входные данные и, таким образом, переходить к булевым схемам.

[1] *Turing A. M.* On Computable Numbers, with an Application to the Entscheidungsproblem // Proceedings of the London Mathematical Society. 1937. Vol. s2–42, no. 1. P. 230–265.

Задачи верификации и синтеза для конечных автоматов и булевых схем являются вычислительно сложными и относятся к классу NP-трудных задач. Это означает, что они не могут быть решены известными алгоритмами за полиномиальное время. В таких случаях, как и во многих других ситуациях, касающихся NP-трудных задач, для решения конкретных экземпляров рассматриваемых проблем используются комбинаторные задачи с хорошо развитой алгоритмической базой. Одной из таких является задача булевой выполнимости (Boolean Satisfiability Problem — SAT), для решения которой за последние 20 лет разработаны весьма эффективные на практике эвристические алгоритмы, применяемые для решения задач символьной верификации[2], компьютерной безопасности и криптографии[3], построению расписаний и планированию[4] и многим другим прикладным областям. В применении к перечисленным задачам программные реализации алгоритмов решения SAT — так называемые SAT-решатели — дают мощные вычислительные инструменты, позволяющие решать частные случаи рассматриваемых задач таких размерностей, перед которыми другие подходы оказываются бессильны.

Таким образом, актуальной является проблема разработки алгоритмов и программных комплексов, основанных на решении задачи SAT, для верификации и синтеза формальных моделей автоматных программ, таких как конечные автоматы и булевые схемы. Одной из ключевых проблем при решении этой задачи является отсутствие априорных оценок времени работы SAT-решателя на сложных формулах, кодирующих рассматриваемые задачи. Грубо говоря, решатель, получив на вход формулу, может работать час, неделю, месяц или даже больше, и нет общего способа определить, сколько времени ему потребуется для завершения работы, притом что формально данный алгоритм является полным и на любой формуле завершает свою работу за конечное время. Это явление известно как *heavy-tailed behavior phenomenon* (HTB)[5], при котором время работы SAT-решателя на некоторых формулах может быть непредсказуемо длинным. В рамках данной диссертации для борьбы с явлением HTB предлагаются специальные декомпозиционные представления булевых формул. Разработанные алгоритмы и методы показали высокую эффективность

[2] *Kroening D.* Software Verification // Handbook of Satisfiability. Vol. 336 / ed. by A. Biere [et al.]. 2nd ed. IOS Press, 2021. P. 791–818. (Frontiers in Artificial Intelligence and Applications).

[3] *Bard G. V.* Algebraic Cryptanalysis. Boston, MA : Springer US, 2009.

[4] *Prestwich S.* CNF Encodings // Handbook of Satisfiability. Washington, 2021. P. 75–100.

[5] *Gomes C. P.*, *Sabharwal A.* Exploiting Runtime Variation in Complete Solvers // Handbook of Satisfiability. Vol. 185 / ed. by A. Biere [et al.]. IOS Press, 2009. P. 271–288. (Frontiers in Artificial Intelligence and Applications).

при решении сложных задач, связанных с синтезом и верификацией конкретных примеров автоматных программ.

Учитывая всё сказанное выше, можно утверждать, что разработка новых методов декомпозиции задачи булевой выполнимости (SAT) для синтеза и верификации моделей автоматных программ является актуальной и важной задачей, имеющей значительные теоретические и практические приложения.

Цель работы. Целью настоящей диссертации является повышение эффективности (снижение времени работы) полных алгоритмов решения задачи булевой выполнимости (SAT) применительно к синтезу и верификации моделей автоматных программ за счет разработки оригинальных методов и техник декомпозиции булевых формул.

Задачи работы. Для достижения поставленной цели были решены следующие научно-технические задачи:

1. Разработка методов кодирования в SAT задач синтеза конечно-автоматных моделей с заданным поведением и свойствами. Новые методы включают в себя кодирование структуры охранных условий в виде деревьев разбора соответствующих формул, что отличает их от существующих решений.
2. Разработка методов кодирования в SAT задач синтеза модульных конечно-автоматных моделей. Эти методы включают автоматизированное модульное разбиение, что улучшает их адаптивность и эффективность.
3. Создание методов кодирования в SAT задач синтеза булевых схем и булевых формул по заданной таблице истинности. В отличие от существующих методов, новые подходы позволяют использовать произвольные элементарные гейты, что расширяет их применение.
4. Разработка методов декомпозиции булевых формул, кодирующих задачи синтеза конечно-автоматных моделей и верификации булевых схем. Новые методы позволяют строить оценки декомпозиционной трудности, что улучшает прогнозируемость времени работы SAT-решателей.
5. Разработка программной библиотеки `kotlin-satlib`, обеспечивающей взаимодействие с SAT-решателями через программный интерфейс. Библиотека предоставляет широкий выбор различных SAT-решателей, контроль за различными этапами построения SAT-кодировок и возможность манипуляции переменными с произвольными конечными доменами.

6. Создание программного комплекса fbSAT для синтеза и верификации конечно-автоматных моделей с использованием SAT-решателей. Этот комплекс интегрирует разработанные методы и алгоритмы, предоставляя удобный инструмент для практического применения.
7. Проведение масштабных вычислительных экспериментов для подтверждения практической эффективности всех разработанных методов.

Основные положения, выносимые на защиту.

1. Методы декомпозиции булевых формул, применяемые к задачам тестирования и верификации моделей автоматных программ и использующие SAT-решатели, отличающиеся от известных методов тем, что с целью получения более точных верхних оценок трудности формул в предлагаемых методах используются специальные конструкции SAT-разбиений.
2. Метод синтеза минимальных представлений булевых функций в виде формул и схем, использующий сведение к задаче выполнимости (SAT), отличающийся от существующих подходов тем, что с целью достижения более высокой эффективности (относительно времени и точности решения) предлагаемый метод использует инкрементальные SAT-решатели.
3. Методы синтеза и верификации монолитных и модульных конечно-автоматных моделей по примерам поведения и формальной спецификации, использующие сведения к задаче выполнимости (SAT) и контрпримеры (Counterexample-Guided Inductive Synthesis — CEGIS), отличающиеся от существующих подходов тем, что с целью повышения эффективности (относительно времени решения) применяется техник явного кодирования структуры охранных условий.
4. Программная библиотека `kotlin-satlib`[6] для взаимодействия с SAT-решателями и обеспечения контроля над всеми этапами построения SAT-кодировок, отличающаяся от известных библиотек тем, что с целью расширения области применимости разработанная библиотека предоставляет широкий выбор различных SAT-решателей и возможность манипулировать переменными с произвольными конечными доменами.
5. Программный комплекс fbSAT[7] для синтеза и верификации конечно-автоматных моделей с помощью SAT-решателей, включающий в себя реализацию всех предложенных методов.

[6] `https://github.com/Lipen/kotlin-satlib`

[7] `https://github.com/ctlab/fbSAT`

Научная новизна. Новыми являются все основные результаты, полученные в диссертации, в том числе:

1. Методы декомпозиции булевых формул, применяемые к задачам тестирования и верификации моделей автоматных программ с использованием SAT-решателей. Отличие от известных методов заключается в применении специальных конструкций SAT-разбиений, что позволяет получать более точные верхние оценки трудности формул.
2. Метод синтеза минимальных представлений булевых функций в виде формул и схем, основанный на сведении к задаче выполнимости (SAT). В отличие от существующих подходов, предлагаемый метод использует инкрементальные SAT-решатели, что позволяет достичь более высокой эффективности по времени и точности решения.
3. Методы синтеза и верификации монолитных и модульных конечно-автоматных моделей, разработанные на основе сведений к задаче выполнимости (SAT) и использования контрпримеров (Counterexample-Guided Inductive Synthesis — CEGIS). Отличие состоит в применении техники явного кодирования структуры охранных условий, что значительно повышает эффективность по времени решения.
4. Программная библиотека `kotlin-satlib` для взаимодействия с SAT-решателями и обеспечения контроля над всеми этапами построения SAT-кодировок. В отличие от существующих библиотек, разработанная библиотека предоставляет широкий выбор различных SAT-решателей и возможность манипулировать переменными с произвольными конечными доменами.
5. Программный комплекс FBSAT для синтеза и верификации конечно-автоматных моделей с помощью SAT-решателей, включающий реализацию всех предложенных методов. Этот комплекс позволяет эффективно решать экстремально трудные задачи синтеза моделей автоматных программ.

Соответствие специальности. Содержание диссертации соответствует паспорту специальности 2.3.5 «Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей» в пунктах «1. Модели, методы и алгоритмы проектирования, анализа, трансформации, верификации и тестирования программ и программных систем» и «3. Модели, методы, архитектуры, алгоритмы,

языки и программные инструменты организации взаимодействия программ и программных систем» в следующих частях:

- в диссертации представлено семейство методов и алгоритмов, применяемых для трансформации автоматных программ в булевы схемы и формулы с целью вычислительного решения задач синтеза и верификации автоматных программ (пункт 1);
- представлены алгоритмы решения задач синтеза, верификации и тестирования моделей автоматных программ при помощи декомпозиционных представлений булевых формул, кодирующих исходные задачи (пункт 1);
- разработанная программная библиотека `kotlin-satlib` обеспечивает взаимодействие между алгоритмами кодирования в SAT задач синтеза и верификации автоматных программ и современными эффективными SAT-решателями (пункт 3).

Теоретическая значимость диссертации заключается в разработке новых методов и алгоритмов для синтеза и верификации моделей автоматных программ. В работе предложены инновационные подходы к декомпозиции булевых формул и построению оценок их декомпозиционной трудности, что расширяет существующие теоретические основы в области применения SAT-решателей и предоставляет более точные инструменты для анализа сложных задач.

Практическая значимость работы проявляется в разработке программной библиотеки `kotlin-satlib` и программного комплекса FBSAT, которые позволяют эффективно применять новые методы и алгоритмы к реальным задачам проектирования и верификации программного обеспечения. Эти инструменты демонстрируют высокую эффективность и могут быть интегрированы в существующие программные системы, что подтверждается лучшими результатами по сравнению с известными подходами и инструментами.

Методы и инструменты исследования. Теоретическая часть работы основана на методологии дискретной математики, математической логики и теории вычислительной сложности. Для синтеза конечно-автоматных моделей применялся программный комплекс FBSAT, разработанный в рамках данной диссертации. Верификация этих моделей осуществлялась с помощью символьного верификатора NUSMV[8]. При построении вычислительных задач из области проверки логической

[8] `https://nusmv.fbk.eu`

эквивалентности схем использовалась программная система Transalg[9]. Для решения экземпляров задачи SAT применялись различные современные SAT-решатели, такие как MiniSAT[10], Glucose[11], Kissat[12], CaDiCaL[13]. Взаимодействие с SAT-решателями осуществлялось через программную библиотеку `kotlin-satlib`, разработанную в рамках данной диссертации. Вычислительные эксперименты проводились с использованием вычислительного кластера.

Достоверность научных достижений диссертации подтверждается обоснованностью постановок задач, теоретической корректностью предложенных алгоритмов, а также результатами масштабных вычислительных экспериментов, проведенных для проверки и демонстрации эффективности разработанных методов.

Апробация работы. Основные результаты диссертации докладывались на следующих конференциях:

- VIII Конгресс молодых ученых, Университет ИТМО, Санкт-Петербург, 2019.
- Конференция СПИСОК-2019, СПбГУ, Санкт-Петербург, 2019.
- IX Конгресс молодых ученых, Университет ИТМО, Санкт-Петербург, 2020.
- Конференция ППС 2021, Университет ИТМО, Санкт-Петербург, 2021.
- X Конгресс молодых ученых, Университет ИТМО, Санкт-Петербург, 2021.
- Воркшоп SAT/SMT Solvers: Theory and Practice, Санкт-Петербург, 2021.
- XI Конгресс молодых ученых, Университет ИТМО, Санкт-Петербург, 2022.
- Конференция MIPRO 2024, Опатия, Хорватия, 2024.

Диссертационная работа была выполнена при поддержке грантов и проектов:

- Грант РФФИ №19-07-01195 А «Разработка методов машинного обучения на основе SAT-решателей для синтеза модульных логических контроллеров киберфизических систем».
- Грант №19-37-51066 Научное наставничество «Разработка методов синтеза конечно-автоматных алгоритмов управления для программируемых логических контроллеров в распределенных киберфизических системах».
- НИР-ФУНД 77051 «Исследование алгоритма тестирования на основе обучения и улучшения его эффективности», 2020-2021.

[9] `https://gitlab.com/transalg/transalg`
[10] `https://github.com/niklasso/minisat`
[11] `https://github.com/audemard/glucose`
[12] `https://github.com/arminbiere/kissat`
[13] `https://github.com/arminbiere/cadical`

- НИР-ПРИКЛ 222004 «Алгоритмы решения SAT для логических схем и анализа программ», 2021-2023.
- НИР-ПРИКЛ 223099 «Алгоритмы решения SAT для логических схем и анализа программ», 2023-2024.

Публикации по теме диссертации. Основные результаты по теме диссертации изложены в 11 публикациях. Из них четыре изданы в изданиях, индексируемых в базе цитирования Scopus, и одна в журнале, рекомендованном ВАК.

Личный вклад автора. Методы синтеза монолитных и модульных конечно-автоматных моделей по примерам поведения и формальной спецификации, основанные на сведении к задаче SAT, разработаны соискателем в соавторстве с Чивилихиным Д. С. Методы синтеза и верификации модульных конечно-автоматных моделей по примерам поведения и формальной спецификации, основанные на сведении к задаче SAT и использовании контрпримеров, разработаны соискателем в соавторстве с Чивилихиным Д. С. и Суворовым Д. М. Программный комплекс fbSAT разработан лично соискателем. Реализация всех разработанных методов синтеза и верификации конечно-автоматных моделей в программном комплексе fbSAT выполнена лично соискателем. Метод синтеза булевых формул и схем по заданной таблице истинности, основанный на сведении к задаче SAT, предложен и разработан лично соискателем. Прототип программной библиотеки `kotlin-satlib` для взаимодействия с SAT-решателями через унифицированный программный интерфейс разработан в соавторстве с Гречишкиной Д. С. Дальнейшая разработка и расширение программной библиотеки `kotlin-satlib`, что включает в себя поддержку дополнительных SAT-решателей (например, Kissat) и разработку модуля для манипуляции переменными с конечными доменами, производилась лично соискателем. Общая стратегия оценивания трудности формул, кодирующих проверку эквивалентности (задача верификации) булевых схем, разработана соискателем в соавторстве с Семёновым А. А., Кондратьевым В. С., Кочемазовым С. Е. и Тарасовым Е. А. Описанные конструкции декомпозиций формул, кодирующих эквивалентность булевых формул, предложены лично соискателем. Теоретическое обоснование корректности предложенных конструкций выполнены соискателем в соавторстве с Семёновым А. А.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обоснована актуальность темы диссертации, сформулированы цель и задачи исследования, описаны научная новизна и практическая значимость работы, представлены методы исследования, сформулированы основные положения, выносимые на защиту, а также приведены данные об апробации работы и личном вкладе автора.

В **Главе 1** представлен обзор предметной области исследования, охватывающий основные понятия, концепции и методы, используемые при синтезе и верификации моделей автоматных программ — конечных автоматов и булевых схем. Подробно проанализированы существующие методы и алгоритмы решения задачи выполнимости (SAT), а также их применение к задачам синтеза и верификации моделей автоматных программ. Затронуты вопросы декомпозиции булевых формул, а также оценки декомпозиционной трудности конкретных экземпляров задачи SAT. Совокупно, эта глава закладывает теоретическую основу для последующих исследований и предложенных методов.

В разделе 1.1 обсуждаются конечные автоматы, которые представляют собой математическую модель, используемую для описания дискретных систем с конечным числом состояний. Приводятся определения детерминированных и недетерминированных конечных автоматов, рассматриваются их основные свойства и примеры использования в различных областях, таких как моделирование поведения программного обеспечения и систем управления.

Раздел 1.2 посвящён булевым схемам, которые являются основой для цифровых систем и вычислений. Описываются элементы булевых схем, такие как логические гейты (AND, OR, NOT и другие), и способы их комбинирования для реализации сложных логических функций. Рассматриваются методы кодирования булевых схем в виде булевых формул в конъюктивной нормальной форме (КНФ) для последующего решения задачи SAT с помощью SAT-решателей с целью тестирования и верификации исходных схем. Также, вводится лемма о том, что применение правила единичного дизъюнкта (Unit Propagation) к КНФ-формуле с подставленными значениями входных переменных схемы эквивалентно процессу вычисления (интерпретации) всей схемы, в частности, выходных значений.

В разделе 1.3 рассматривается международный стандарт IEC 61499, который определяет архитектуру для разработки распределённых систем управления. Описываются основные компоненты стандарта, такие как функциональные блоки,

и их использование для моделирования и реализации промышленных систем автоматизации.

Раздел 1.4 посвящён описанию формальной модели базового функционального блока, которая используется в стандарте IEC 61499. Описываются структура и поведение базового функционального блока, способы его конфигурации и взаимодействия с другими блоками. Рассматриваются примеры использования базовых функциональных блоков для построения сложных систем управления. Представленная абстрактная модель впоследствии используется в качестве основы при разработке методов синтеза и верификации конечно-автоматных моделей.

В разделе 1.5 обсуждаются сценарии выполнения, которые представляют собой последовательности действий или событий в системе. Описывается, как сценарии выполнения могут использоваться для спецификации требований к системе и её верификации. Рассматриваются различные методы описания сценариев выполнения и их использование в процессе синтеза и тестирования моделей.

Раздел 1.6 посвящён линейной темпоральной логике (LTL), которая используется для спецификации и верификации временных свойств систем. Описываются основные операторы LTL и способы их использования для выражения различных временных аспектов поведения систем. Приводятся примеры спецификаций на основе LTL и методы их проверки.

В разделе 1.7 рассматриваются методы формальной верификации, основанные на проверке моделей (Model Checking). Описываются основные этапы процесса проверки моделей и инструменты, используемые для этой цели. Подробно рассматриваются и сравниваются два основных подхода к проверке моделей: символьный (symbolic) и ограниченный (bounded).

В разделе 1.8 рассматриваются различные методы синтеза конечно-автоматных моделей, которые используются для автоматического построения моделей на основе заданных требований и поведения. Описываются как традиционные, так и современные подходы к синтезу, а также их сравнительный анализ.

Раздел 1.9 посвящён задаче проверки эквивалентности булевых схем, которая заключается в проверке, производят ли две булевы схемы одинаковые выходные значения для всех возможных входных данных. Описываются методы и алгоритмы решения этой задачи, а также примеры их применения в различных областях.

В разделе 1.10 обсуждается задача генерации тестовых шаблонов, используемых для верификации булевых схем. Описываются методы генерации тестов, которые позволяют обнаруживать ошибки в схемах и проверять их корректность. Приводятся

примеры применения тестовых шаблонов в практике проектирования цифровых систем.

Раздел 1.11 посвящён задаче булевой выполнимости (SAT), которая является фундаментальной проблемой в теории вычислительной сложности и играет ключевую роль в синтезе и верификации логических схем. Описываются формулировка задачи SAT, её значение и примеры применения в различных областях.

В разделе 1.12 рассматриваются основные алгоритмы решения задачи булевой выполнимости (SAT). Описываются как традиционные, так и современные методы, включая полные и неполные алгоритмы. Рассматриваются как полные алгоритмы решения SAT, основанные на алгоритме DPLL, которые гарантируют нахождение решения или доказательство его отсутствия, так и неполные, которые не гарантируют нахождение решения, но могут быть эффективны на практике для больших и сложных задач. Отдельно рассматривается концепция CDCL (Conflict-Driven Clause Learning), которая является усовершенствованием алгоритма DPLL и лежит в основе современных SAT-решателей.

В разделе 1.13 обсуждаются ограничения на кардинальность, которые часто встречаются в задачах SAT. Ограничения на кардинальность представляют собой условия, которые ограничивают количество переменных, принимающих значение истинности. Описывается один из методов, используемых для обработки таких ограничений — кодирование в КНФ с помощью метода *totalizer*, который заключается в кодировании унарной записи числа, задающего сумму переменных, в виде булевой формулы. На итоговое число в унарном представлении накладываются ограничения в виде единичных дизъюнктов, которые обеспечивают выполнение условий на кардинальность, например, вхождение числа истинных переменных в заданный диапазон.

Раздел 1.14 посвящён методам разбиения задачи SAT на подзадачи. Описываются различные подходы к разбиению и их преимущества. Важным аспектом при решении задач SAT является декомпозиционная трудность булевых формул. В этом контексте рассматриваются методы оценки трудности и их применение на практике. Также обсуждаются основы теории вероятности, необходимые для понимания вероятностных методов оценки трудности задач SAT, включая основные понятия и методы, используемые в этой области. Дополнительно рассматривается вероятностный подход к оцениванию трудности булевых формул, который позволяет более точно прогнозировать сложность их решения. Описываются методы и алгоритмы, использующие вероятностные модели, и примеры их применения.

В **главе 2** рассматриваются методы синтеза моделей автоматных программ на основе сведения задач к задаче булевой выполнимости (SAT). Основное внимание уделено разработке и применению методов кодирования задач синтеза в формате SAT, что позволяет значительно улучшить эффективность и точность синтеза конечно-автоматных моделей. Глава начинается с описания программной библиотеки `kotlin-satlib`, разработанной для взаимодействия с SAT-решателями и упрощающей процесс построения SAT-кодировок. Далее рассматриваются методы синтеза булевых формул и методы синтеза конечно-автоматных моделей монолитных логических контроллеров по примерам поведения. Особое внимание уделено алгоритмам синтеза минимальных моделей, индуктивному синтезу на основе контрпримеров, а также разработке программного средства fbSAT для синтеза и верификации конечно-автоматных моделей. В главе 2 также приведены результаты экспериментальных исследований, демонстрирующих практическую значимость и эффективность предложенных методов.

В разделе 2.1 описывается программная библиотека `kotlin-satlib`, разработанная для взаимодействия с SAT-решателями. Эта библиотека предназначена для упрощения процесса синтеза и верификации моделей автоматных программ за счёт предоставления удобного интерфейса для работы с SAT-решателями. Библиотека включает несколько модулей, каждый из которых имеет специфические функции. Первый модуль обеспечивает взаимодействие с SAT-решателями посредством технологии JNI, что позволяет использовать различные SAT-решатели без существенных изменений в коде. Второй модуль занимается записью ограничений с использованием преобразований Цейтина, что оптимизирует процесс конвертации логических выражений в формат SAT. Третий модуль позволяет манипулировать переменными с ограниченным доменом, что расширяет возможности моделирования. Наконец, четвёртый модуль поддерживает работу с массивами SAT переменных, предоставляя средства для эффективного управления большими логическими структурами.

Раздел 2.2 посвящён методам синтеза булевых формул. В нём рассматриваются различные подходы к построению булевых формул, которые затем могут быть использованы для синтеза моделей автоматных программ. Этот раздел также включает результаты экспериментального исследования, которое демонстрирует эффективность предложенных методов синтеза. Были проведены многочисленные эксперименты для оценки производительности различных подходов к синтезу булевых формул, результаты которых подтверждают высокую эффективность разработанных методов.

В разделе 2.3 описывается метод синтеза конечно-автоматных моделей монолитных логических контроллеров по примерам поведения. Этот метод включает несколько ключевых этапов: сначала происходит кодирование структуры автомата, затем вводятся BFS-предикаты нарушения симметрии для состояний автомата, что помогает избежать избыточных вычислений. Далее, кодируется отображение позитивного дерева сценариев и устанавливаются ограничения на количество переходов между состояниями. В разделе также подробно рассматриваются различные алгоритмы, такие как Basic, Extended и Complete, каждый из которых предлагает свои подходы к синтезу моделей с учётом различных ограничений и требований. Этот метод позволяет значительно сократить время синтеза и повысить точность моделирования.

Раздел 2.4 фокусируется на методах синтеза минимальных монолитных моделей. Здесь рассматриваются алгоритмы Basic-min, Extended-min и Complete-min, которые оптимизируют процесс синтеза с целью получения минимальных по размеру и сложности моделей. В этом разделе также описывается алгоритм Extended-min-UB, который использует дополнительные ограничения для улучшения производительности.

В разделе 2.5 представлен индуктивный синтез, основанный на контрпримерах (CEGIS). Этот метод включает алгоритмы CEGIS и CEGIS-min, которые позволяют улучшить процесс синтеза за счёт использования контрпримеров, выявленных в ходе верификации моделей. Эти алгоритмы помогают итеративно улучшать модели, корректируя их на основе найденных ошибок.

В разделе 2.6 описывается программное средство fbSAT, предназначенное для синтеза и верификации конечно-автоматных моделей с использованием SAT-решателей. Это средство интегрирует разработанные методы и алгоритмы, предоставляя удобный инструмент для практического применения. fbSAT обеспечивает полный цикл разработки и верификации моделей, от начального синтеза до окончательной проверки корректности.

В разделе 2.7 приводится экспериментальное исследование на примере Pick-and-Place манипулятора. В рамках этого исследования синтезировались минимальные конечно-автоматные модели по примерам поведения, а также по примерам поведения и LTL-спецификации. Эти эксперименты демонстрируют практическую значимость разработанных методов и их применимость к реальным задачам.

В разделе 2.8 рассматривается экспериментальное исследование, проведённое в рамках соревнований SYNTCOMP. Результаты этих экспериментов подтверждают

высокую эффективность и конкурентоспособность предложенных методов синтеза и верификации.

Глава 3 посвящена методам синтеза модульных конечно-автоматных моделей. Рассматриваются подходы к параллельной, последовательной и произвольной композиции модулей по примерам поведения. Описаны методы сведения задач к SAT и приведены алгоритмы, оптимизирующие процесс синтеза для модульных моделей. Также обсуждаются методы синтеза минимальных модульных моделей и переход от монолитного к распределённому синтезу. Глава завершается результатами экспериментальных исследований, которые подтверждают эффективность предложенных методов синтеза модульных моделей и их применимость к реальным задачам.

В разделе 3.1 описан метод синтеза модульных конечно-автоматных моделей с параллельной композицией модулей по примерам поведения. Этот метод включает этапы сведения задачи к SAT, где сначала определяются переменные, затем вводятся ограничения, необходимые для корректного функционирования моделей. В разделе также подробно описываются алгоритмы Parallel-Basic и Parallel-Extended, которые используют эти ограничения для создания эффективных модульных моделей. Эти алгоритмы помогают снизить сложность синтеза за счёт параллельной обработки различных компонентов модели, что ускоряет процесс и улучшает масштабируемость.

В разделе 3.2 представлен метод синтеза конечно-автоматной модели модульного логического контроллера с последовательной композицией модулей по примерам поведения. Здесь также используется сведение к SAT, где переменные и ограничения определяются таким образом, чтобы обеспечить последовательное выполнение различных модулей. Описаны алгоритмы Consecutive-Basic и Consecutive-Extended, которые оптимизируют процесс синтеза для последовательных композиций, улучшая тем самым точность и эффективность моделей.

В разделе 3.3 рассматривается метод синтеза модульных конечно-автоматных моделей с произвольной композицией модулей по примерам поведения. Этот метод позволяет создавать более гибкие модели, которые могут включать произвольное количество и комбинации модулей. В разделе подробно описаны этапы сведения задачи к SAT, включая определение переменных и введение ограничений, а также алгоритмы Arbitrary-Basic и Arbitrary-Extended, которые оптимизируют процесс синтеза для таких моделей.

Раздел 3.4 посвящён методам синтеза минимальных модульных моделей. Здесь рассматриваются различные подходы к минимизации размера и сложности

модульных моделей, что позволяет создавать более эффективные и компактные решения.

В разделе 3.5 описан переход от монолитного к распределённому синтезу. Этот метод включает сведение к SAT для распределённого синтеза по примерам поведения, что позволяет разбивать задачу на более мелкие и управляемые компоненты. Также рассматриваются составное негативное дерево сценариев и его отображение, что помогает улучшить процесс синтеза распределённых контроллеров. В разделе приводятся методы нахождения минимального распределённого контроллера, что позволяет оптимизировать распределённые системы.

В разделе 3.6 приводятся результаты экспериментального исследования модульного синтеза. Эти исследования демонстрируют эффективность предложенных методов и подтверждают их применимость к реальным задачам. Эксперименты показывают, что модульный синтез позволяет значительно улучшить производительность и точность моделей.

В **главе 4** рассматриваются методы оценивания декомпозиционной трудности булевых формул, которые применяются к задачам тестирования и верификации моделей автоматных программ с использованием SAT-решателей. Описаны методы декомпозиции булевых формул, позволяющие строить более точные верхние оценки трудности формул. Конкретно, предложены оригинальные конструкции SAT-разбиений, которые улучшают прогнозируемость времени работы SAT-решателей. Глава включает в себя теоретические аспекты, алгоритмы и результаты вычислительных экспериментов, демонстрирующих эффективность предложенных подходов в различных сценариях.

В разделе 4.1 рассматривается концепция трудности булевых формул относительно разбиения, которая является ключевым фактором при решении задач SAT. Обсуждается, как структура и сложность формулы влияют на эффективность её решения, и вводится понятие *декомпозиционной трудности*.

Вероятностный алгоритм оценки трудности представляет собой метод, который использует вероятностные модели для прогнозирования трудности SAT-разбиения булевых формул. Описываются основные этапы алгоритма: построение вероятностной модели, определение времени решения подзадач, оценка на основе статистических данных.

В разделе 4.2 представляются два новых метода разбиения SAT, разработанные специально для задачи CircuitSAT, которая является частным случаем задачи булевой выполнимости, применимым к логическим схемам. Эти методы направлены на

повышение эффективности решения за счёт более оптимального разбиения исходной формулы на подформулы.

Раздел 4.3 посвящён описанию вычислительных экспериментов, которые были проведены для подтверждения практической эффективности предложенных методов декомпозиции булевых формул. В этом разделе детально рассматриваются тестовые данные, методология проведения экспериментов и анализ полученных результатов.

Эксперименты по оценке декомпозиционной трудности булевых формул проводились с использованием предложенного вероятностного алгоритма. Методология проведения этих экспериментов включала в себя сбор и обработку данных о сложност решения задач SAT при различных вариантах разбиения формул. Полученные результаты сравнены с теоретическими оценками, что позволяет проверить точность и надёжность алгоритма. Статистический анализ данных показал, что вероятностный алгоритм успешно предсказывает декомпозиционную трудность формул, что подтверждается корреляцией между теоретическими оценками и реальными измерениями трудности.

Кроме того, были проведены эксперименты по поиску прообразов криптографической хеш-функции MD4 с использованием разработанных методов разбиения. В рамках этих экспериментов задача поиска прообразов MD4 была закодирована в виде задачи SAT, и применены новые методы разбиения для её решения. Результаты показали, что предложенные методы существенно повышают эффективность решения таких задач. Анализ временных характеристик и сравнение с другими подходами продемонстрировали, что новые методы разбиения обеспечивают лучшее качество разбиения и более высокую производительность.

В общем, вычислительные эксперименты, описанные в разделе 4.3, демонстрируют высокую практическую значимость предложенных методов декомпозиции булевых формул. Эти методы не только теоретически обоснованы, но и подтверждены на практике, что показывает их применимость для решения сложных задач синтеза и верификации логических схем и конечно-автоматных моделей.

Вся глава 4 подробно иллюстрирует, как предложенные методы и алгоритмы могут применяться для решения сложных задач синтеза и верификации, предоставляя как теоретическое обоснование, так и практическую реализацию, поддержанную экспериментальными данными.

В заключении сформулированы основные результаты научной работы.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ ДИССЕРТАЦИИ И ВЫВОДЫ

В данной диссертации были получены следующие основные результаты и выводы:

- Разработаны алгоритмы декомпозиции булевых формул, которые могут применяться к задачам верификации моделей автоматных программ. Эти алгоритмы позволяют строить оценки декомпозиционной трудности рассматриваемых формул.
- Предложены новые алгоритмы кодирования в SAT задачи синтеза булевых схем и булевых формул по заданной таблице истинности. Эти алгоритмы отличаются от существующих возможностью использования произвольных элементарных гейтов, что расширяет их применение.
- Разработаны алгоритмы кодирования в SAT монолитных и модульных конечно-автоматных моделей для решения задач синтеза и верификации. Эти алгоритмы включают явное кодирование структуры охранных условий в виде деревьев разбора соответствующих формул, что позволяет их минимизировать, что в свою очередь повышает человеко-читаемость полученных выражений.
- Созданы программные библиотеки для взаимодействия с различными SAT-решателями через специально разработанный программный интерфейс. Эти библиотеки отличаются возможностью контроля всех этапов построения SAT-кодировок и манипулирования переменными с произвольными конечными доменами.
- Проведены масштабные вычислительные эксперименты, подтвердившие практическую эффективность всех разработанных методов. Эти эксперименты продемонстрировали значительное улучшение в решении трудных примеров синтеза и верификации моделей автоматных программ — конечных автоматов и булевых схем.

Таким образом, разработанные методы и алгоритмы значительно повышают эффективность (снижают время работы) полных алгоритмов решения задачи булевой выполнимости (SAT) и вносят важный вклад в области синтеза и верификации моделей автоматных программ.

ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Chivilikhin D., Patil S., **Chukharev K.**, Cordonnier A., Vyatkin V. Automatic State Machine Reconstruction from Legacy PLC Using Data Collection and SAT Solver // IEEE Transactions on Industrial Informatics. 2020. Vol. 16, issue 12. P. 7821–7831. (**Scopus, Q1**)
2. **Chukharev K.**, Suvorov D., Chivilikhin D. SAT-based Counterexample-Guided Inductive Synthesis of Distributed Controllers // IEEE Access. 2020. Vol. 8. P. 207485–207498. (**Scopus, Q1**)
3. **Chukharev K.**, Chivilikhin D. fbSAT: Automatic Inference of Minimal Finite-State Models of Function Blocks Using SAT Solver // IEEE Access. 2022. Vol. 10. P. 131592–131610. (**Scopus, Q1**)
4. Andreev A., **Chukharev K.**, Kochemazov S., Semenov A. Solving Influence Maximization Problem under Deterministic Linear Threshold Model using Metaheuristic Optimization // MIPRO, 2024. (**Scopus**)
5. **Чухарев К.** Применение инкрементальных SAT-решателей для решения NP-трудных задач на примере задачи синтеза минимальных булевых формул // Научно-технический вестник информационных технологий, механики и оптики. 2020. Т. 20, 6(130). С. 841—847. (**ВАК**)
6. Semenov A., **Chukharev K.**, Tarasov E., Chivilikhin D., Kondratiev V. Estimating the Hardness of SAT Encodings for Logical Equivalence Checking of Boolean Circuits // arXiv, 2022.
7. **Чухарев К.**, Чивилихин Д. Построение минимальных конечно-автоматных моделей функциональных блоков по обучающим примерам // Сборник тезисов докладов конгресса молодых ученых. Электронное издание. СПб: Университет ИТМО, 2018.
8. **Чухарев К.** Построение конечно-автоматных моделей функциональных блоков по примерам поведения и темпоральным свойствам // Сборник тезисов докладов конгресса молодых ученых. Электронное издание. СПб: Университет ИТМО, 2019.
9. **Чухарев К.** Автоматический синтез минимальных конечно-автоматных моделей функциональных блоков по примерам поведения и темпоральным свойствам // Материалы 8-й всероссийской научной конференции по проблемам информатики СПИСОК-2019. СПб.: ВВМ, 2019.
10. **Чухарев К.** Синтез конечно-автоматных моделей модульных логических контроллеров по примерам поведения с помощью SAT-решателей // Сборник тезисов

докладов конгресса молодых ученых. Электронное издание. СПб: Университет ИТМО, 2020.

11. Гречишкина Д., **Чухарев К.** Программный интерфейс для SAT-решателей на основе технологии JNI // Сборник тезисов докладов конгресса молодых ученых. Электронное издание. СПб: Университет ИТМО, 2020.