

Университет ИТМО

Методы генерации детерминированных конечных автоматов с использованием сокращения пространства поиска при решении задачи выполнимости

Закирзянов Илья Тимурович

Специальность 05.13.17 — Теоретические основы информатики

Диссертация на соискание учёной степени кандидата технических наук

Научный руководитель:
канд. техн. наук
Ульянцев Владимир Игоревич

16 декабря 2020
Санкт-Петербург

Актуальность

Конечно-автоматные модели:

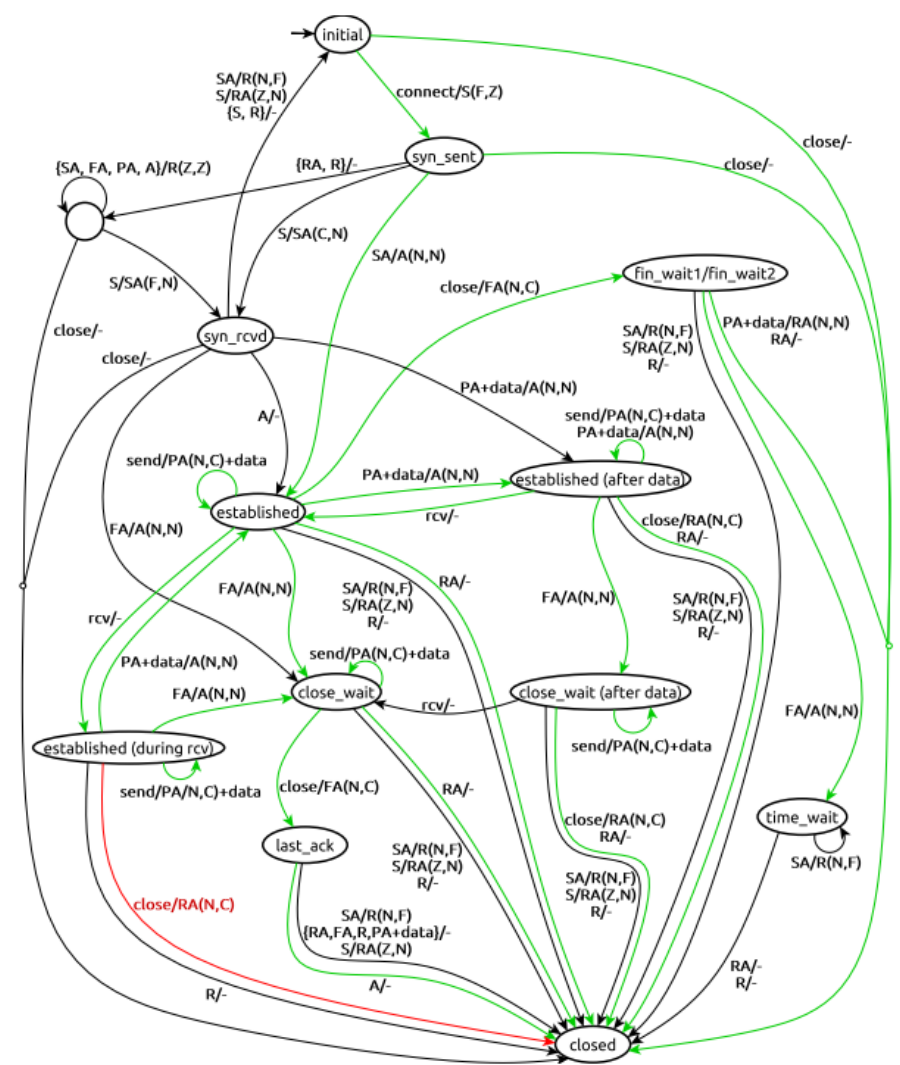
- анализ поведения сложных программных систем [Heule et al., 2013];
- анализ поведения сетевых протоколов [Sivakorn et al., 2017; Fiterau-Brostean P., 2018];
- построение формальных моделей объектов управления [Buzhinsky I et al., 2017];
- ...

Проектирование автоматной модели:

- Ручное построение.
- Автоматическая генерация.

Детерминированные конечные автоматы:

- могут сами выступать в качестве моделей [Heule et al., 2013];
- можно адаптировать методы для более сложных моделей [Ульянцев В., 2015; Chivilikhin D., 2020].



Модель TCP клиента Windows 8, сгенерированная по примерам поведения [Fiterau-Brostean P., 2018]

Heule M., Verwer S. Software model synthesis using satisfiability solvers, 2013.

Ульянцев В., Генерация конечных автоматов с использованием программных средств решения задач выполнимости и удовлетворения ограничений, 2015.

Sivakorn S. et al., HVLearn: Automated Black-box Analysis of Hostname Verification in SSL/TLS Implementations, 2017.

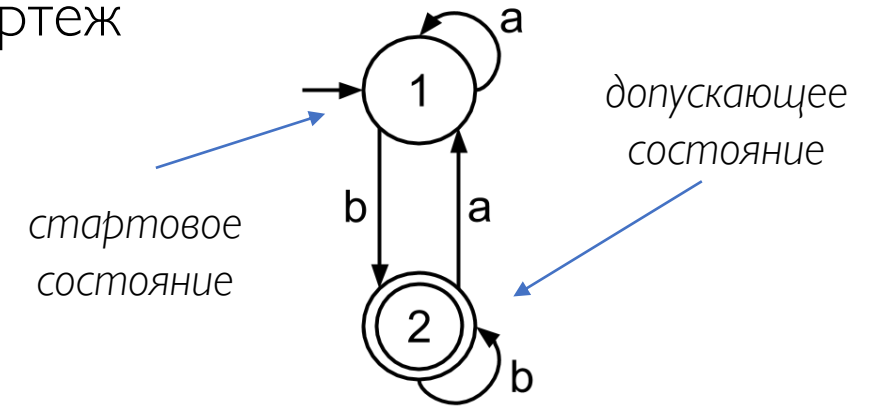
Buzhinsky I., Vyatkin V. Automatic inference of finite-state plant models from traces and temporal properties, 2017.

Fiterau-Brostean P. Active Model Learning for the Analysis of Network Protocols, 2018.

Chivilikhin D. et al., Automatic State Machine Reconstruction From Legacy Programmable Logic Controller Using Data Collection and SAT Solver, 2020

Введение [1/4]

- Детерминированный конечный автомат (ДКА) — это кортеж $A = (Q, \Sigma, \delta, q_1, Q^+)$, где:
 - Q — конечное множество состояний;
 - Σ — конечный алфавит;
 - $\delta: Q \times \Sigma \rightarrow Q$ — функция переходов;
 - $q_s \in Q$ — стартовое состояние;
 - $Q^+ \subset Q$ — множество допускающих состояний.
- Язык $\mathcal{L}(A)$ — множество всех слов, допускаемых автоматом.
- Подходы к генерации (построению, синтезу) ДКА.
 - Активный — доступны запросы к оракулу.
 - Пассивный — по известным заранее примерам поведения.



Введение [2/4]

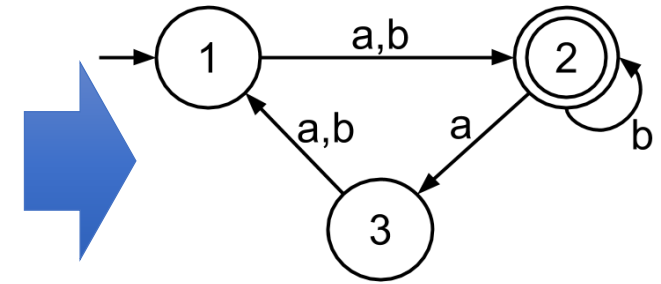
- S_+ — слова, которые должны допускаться автоматом.
- S_- — слова, которые не должны допускаться автоматом.

S_+

- a
- bb
- aaaa

S_-

- bab
- aa



- Генерация ДКА по примерам поведения: обобщение знаний — по конечному набору примеров поведения строится автомат, допускающий бесконечное число слов.
- Минимальность важна ввиду принципа Бритвы Оккама. Чем меньше автомат — тем лучше он обобщает известные данные.
- Задача генерации ДКА конкретного размера (как следствие, минимального) по известным примерам поведения NP-полна [Gold, 1978].

Введение [3/4]

Задача выполнимости булевых формул (*Boolean satisfiability, SAT*):

- Исторически первая NP-полная задача [Cook, 1971; Левин, 1973].
- Существует ли выполняющая подстановка для булевой формулы?
- Обычно, конъюнктивная нормальная форма (КНФ) — конъюнкция дизъюнктов и/или литералов.

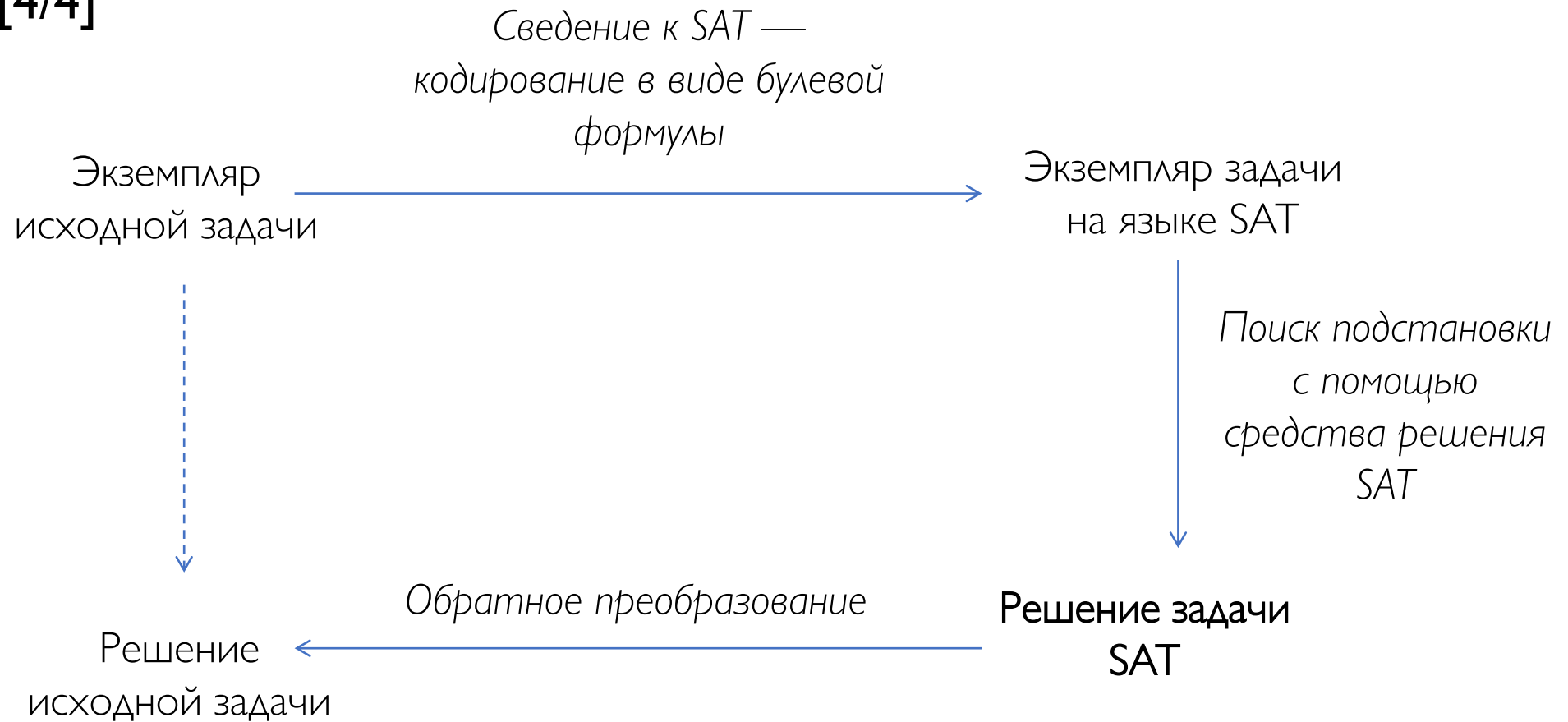
$$(a \vee b) \wedge (a \vee c) \wedge (\neg b \vee \neg c \vee d) \wedge (\neg a \vee \neg d)$$



$$a = 0; b = 1; c = 1; d = 1$$

- Популярный и эффективный подход решения NP задач — сводить к SAT.

Введение [4/4]



Соревнования по выявлению лучших программных средств для решения SAT.
Улучшение производительности методов генерации ДКА без изменения алгоритма!

Степень разработанности темы исследования [1/3]

- Неточные методы не гарантируют, что автомат будет найден, а если и найден, то не гарантируется минимальность.
 - Эвристические алгоритмы слияния состояний (*state merging*) [Lang et al., 1998].
 - Метаэвристические алгоритмы:
 - эволюционные стратегии [Lucas et al., 2007];
 - генетические алгоритмы [Pawar et al., 2002];
 - муравьиные алгоритмы [Chivilikhin D. et al., 2012].
- Точные методы всегда находят автомат минимального размера. Основаны на сведениях к SAT.
 - DFASAT [Heule, Verwer, 2010].
 - DFA-Inductor [Ulyantsev et al., 2015] — программный комплекс, реализующий методы, предложенные научным руководителем при участии автора диссертации.

Lang K. et al. Results of the Abbadingo One DFA Learning Competition and a New Evidence-Driven State Merging Algorithm, 1998.

Lucas S., Reynolds J. Learning Finite State Transducers: Evolution versus Heuristic State Merging, 2007.

Pawar P., Nagaraja G. Regular Grammatical Inference: A Genetic Algorithm Approach, 2002.

Chivilikhin D., Ulyantsev V. Learning Finite-State Machines with Ant Colony Optimization, 2012.

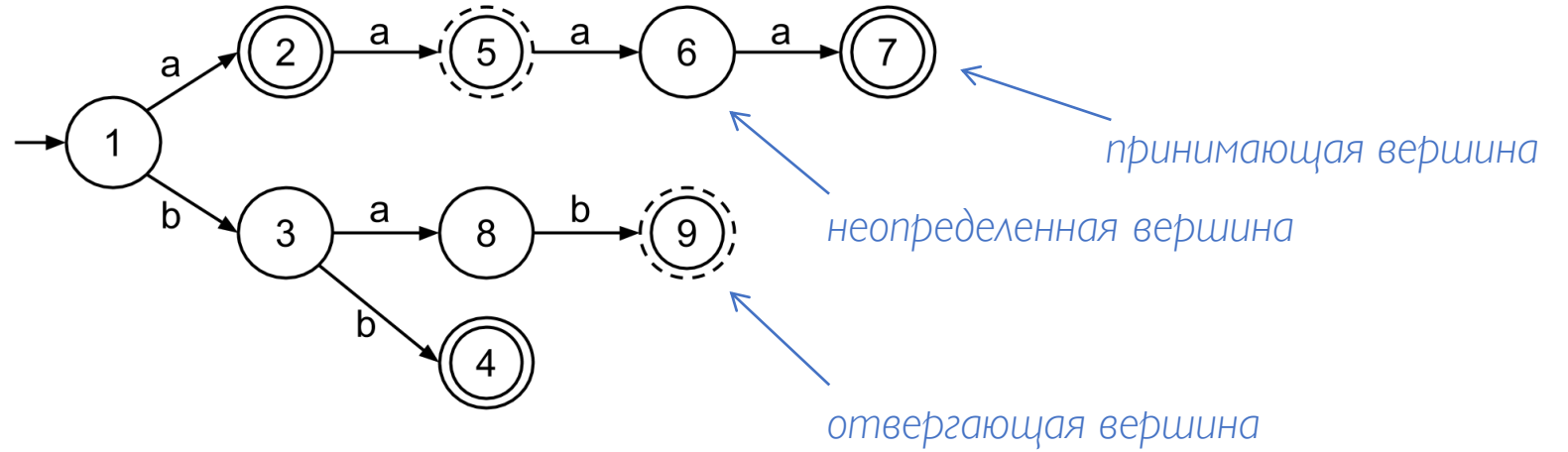
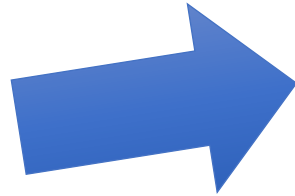
Heule M., Verwer S. Exact DFA identification using SAT solvers, 2010

Ulyantsev, V., **Zakirzyanov, I.**, Shalyto, A.: BFS-based symmetry breaking predicates for DFA identification, 2015.

Степень разработанности темы исследования [2/3]

DFASAT [Heule et al., 2013]

- | | |
|--------|-------|
| S_+ | S_- |
| • a | • bab |
| • bb | • aa |
| • aaaa | |

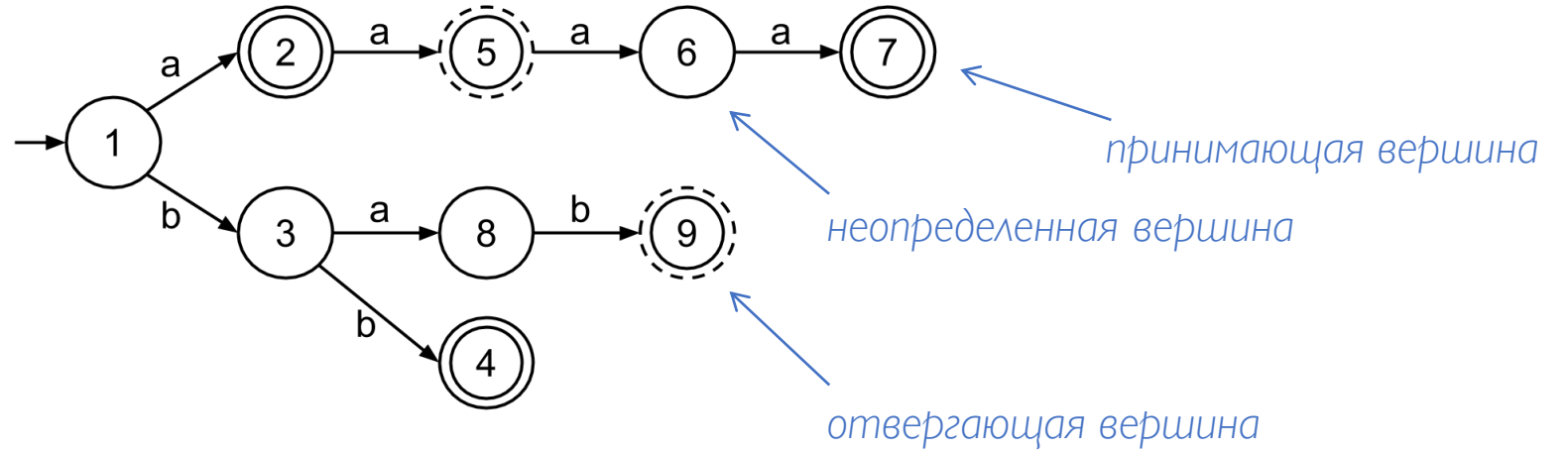
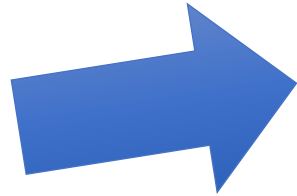


Расширенное префиксное дерево

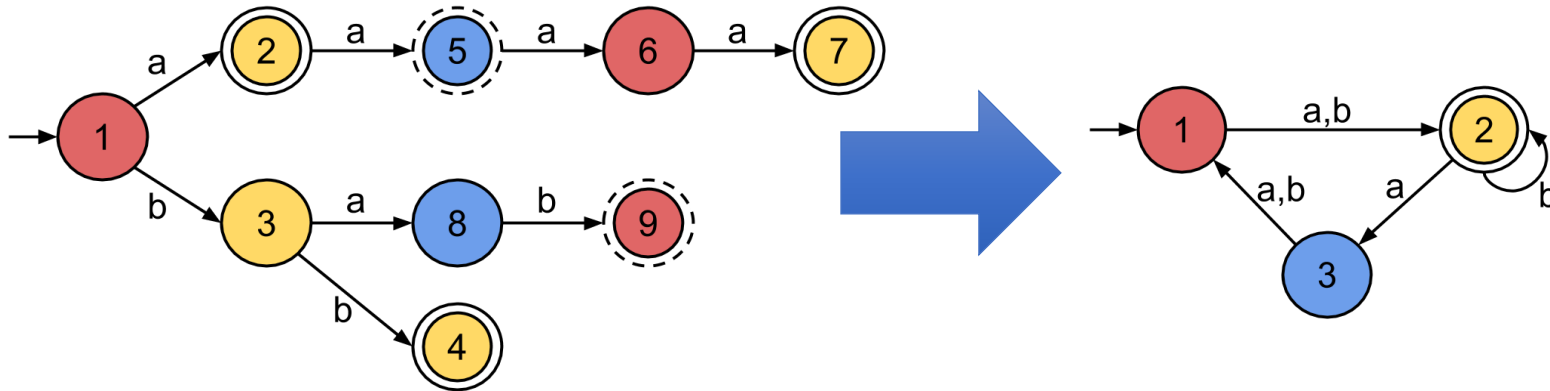
Степень разработанности темы исследования [2/3]

DFASAT [Heule et al., 2013]

- | | |
|--------|-------|
| S_+ | S_- |
| • a | • bab |
| • bb | • aa |
| • aaaa | |



Расширенное префиксное дерево

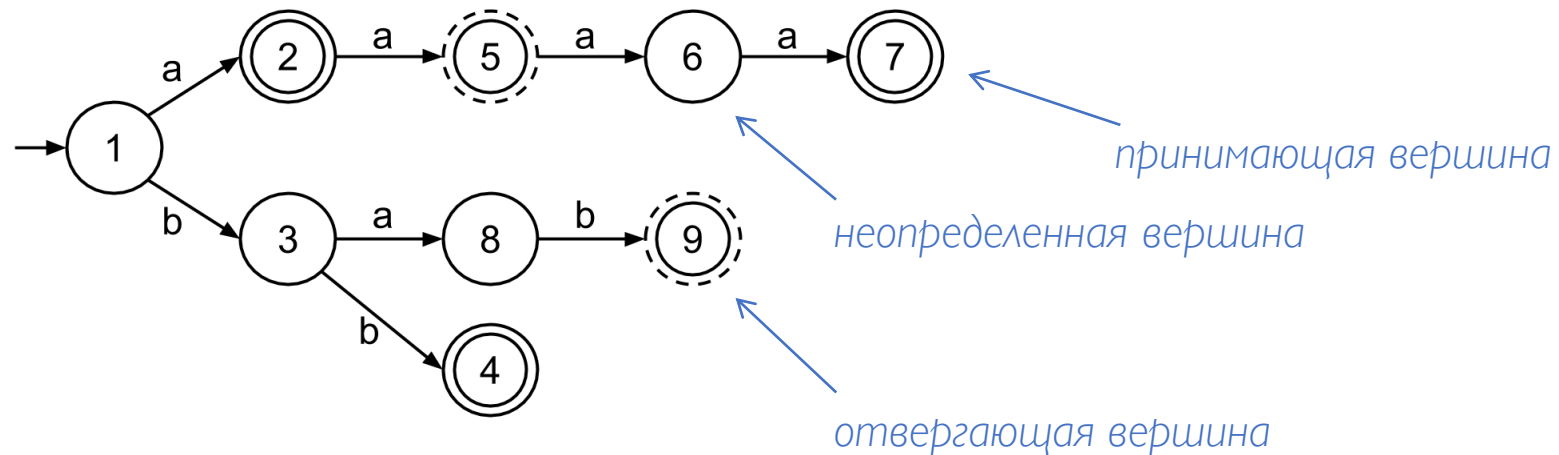
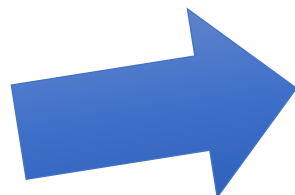


Итеративно увеличиваем число цветов для гарантии минимальности.

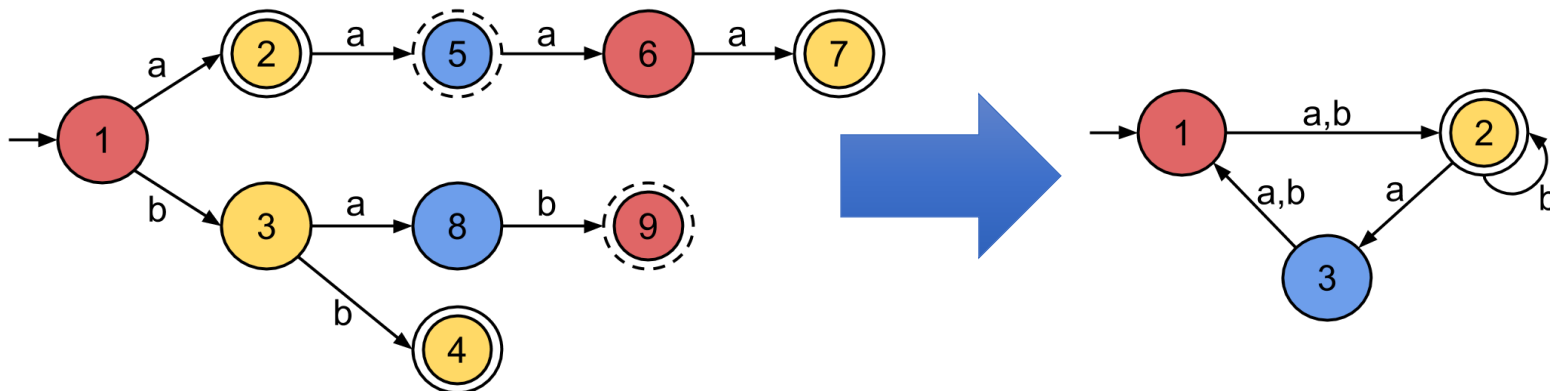
Степень разработанности темы исследования [2/3]

DFASAT [Heule et al., 2013]

- | | |
|--------|-------|
| S_+ | S_- |
| • a | • bab |
| • bb | • aa |
| • aaaa | |



Расширенное префиксное дерево



Итеративно увеличиваем число цветов для гарантии минимальности.

$x_{v,i} = 1 \Leftrightarrow$ вершина v префиксного дерева соответствует состоянию i (покрашена в цвет i).

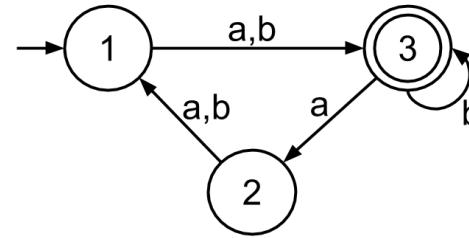
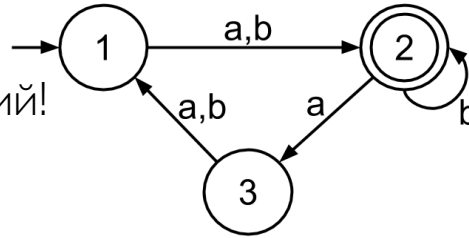
$y_{i,l,j} = 1 \Leftrightarrow$ в автомате есть переход по символу l из состояния i в состояние j .

$z_i = 1 \Leftrightarrow$ в автомате состояние i является допускающим.

Степень разработанности темы исследования [3/3]

Симметрия — изоморфные автоматы.

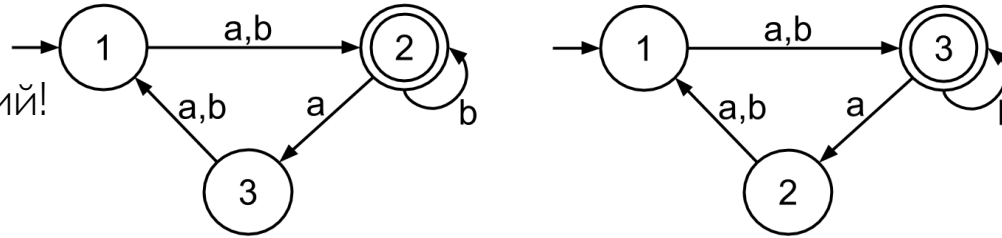
Факториал перестановок номеров состояний!



Степень разработанности темы исследования [3/3]

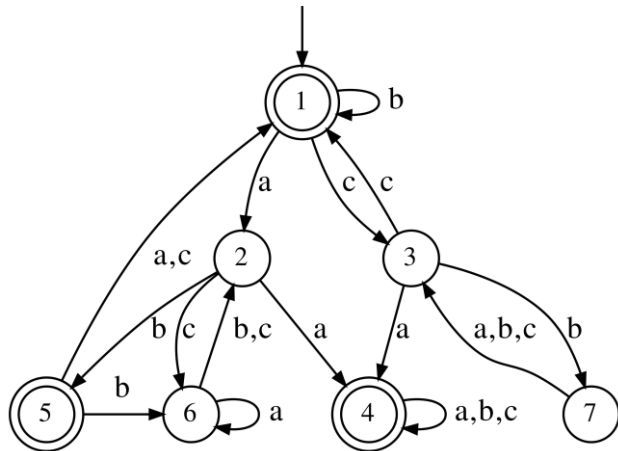
Симметрия — изоморфные автоматы.

Факториал перестановок номеров состояний!

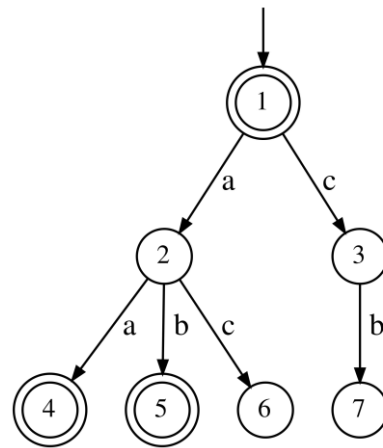


DFA-Inductor [Ulyantsev et al., 2015].

Предикаты нарушения симметрии, допускающие единственного представителя для каждого класса эквивалентности по изоморфизму, — предикаты нарушения симметрии на основе алгоритма обхода графа в ширину (BFS).



BFS пронумерованный ДКА



дерево BFS

- $t_{i,j} = 1 \Leftrightarrow$ в автомате есть переход из состояния i в состояние j .
- $p_{j,i} = 1 \Leftrightarrow$ состояние i является родителем состояния j в дереве BFS.
- $m_{i,l,j} = 1 \Leftrightarrow l$ является лексикографически меньшим символом, который встречается на переходах из состояния i в состояние j .

Недостатки существующих методов

1. Наивное кодирование предикатов нарушения симметрии на языке SAT не оптимально.
 $O(M^3 + M^2 \times L^2)$ дизъюнктов, где M — размер ДКА, L — мощность алфавита.
2. Существующие методы генерации ДКА неприменимы при избыточности имеющихся данных, так как размер булевой формулы зависит от размера данных — длины и числа примеров поведения.
 $O(N \times M^2)$ дизъюнктов, где N — размер префиксного дерева, M — размер ДКА.
3. Существующие методы не приспособлены для генерации всех ДКА минимального размера по заданным примерам поведения. Как следствие, не существует эффективного способа доказать единственность решения.

Цель и задачи исследования

Цель — повышение эффективности точных методов генерации детерминированных конечных автоматов по заданным примерам поведения посредством сокращения пространства поиска при решении задачи выполнимости.

Задачи:

1. Разработка предикатов нарушения симметрии, основанных на кодировании алгоритмов обхода графа в ширину и в глубину, для сокращения пространства поиска при решении задачи выполнимости. Разработка и реализация точных методов генерации ДКА по заданным примерам поведения, использующих данные предикаты, проведение экспериментальных исследований с ними.
2. Разработка и реализация точного метода генерации ДКА по избыточному набору примеров поведения с использованием сведения к задаче выполнимости и подхода уточнения абстракции по контрпримерам. Проведение экспериментальных исследований с методом.
3. Разработка и реализация метода генерации всех неизоморфных ДКА минимального размера, удовлетворяющих заданным примерам поведения, с использованием предикатов нарушения симметрии и программных средств решения задачи выполнимости. Проведение экспериментальных исследований с методом.

Выносимые на защиту положения

1. Предикаты нарушения симметрии, построенные на основе кодирования алгоритмов обхода графа в ширину и в глубину, предназначенные для сокращения пространства поиска при решении задачи выполнимости. Точные методы генерации ДКА по заданным примерам поведения, использующих данные предикаты.
2. Точный метод генерации ДКА по избыточному набору примеров поведения с использованием сведения к задаче выполнимости и подхода уточнения абстракции по контрпримерам.
3. Метод генерации всех неизоморфных ДКА минимального размера, удовлетворяющих заданным примерам поведения, с использованием предикатов нарушения симметрии и программных средств решения задачи выполнимости.

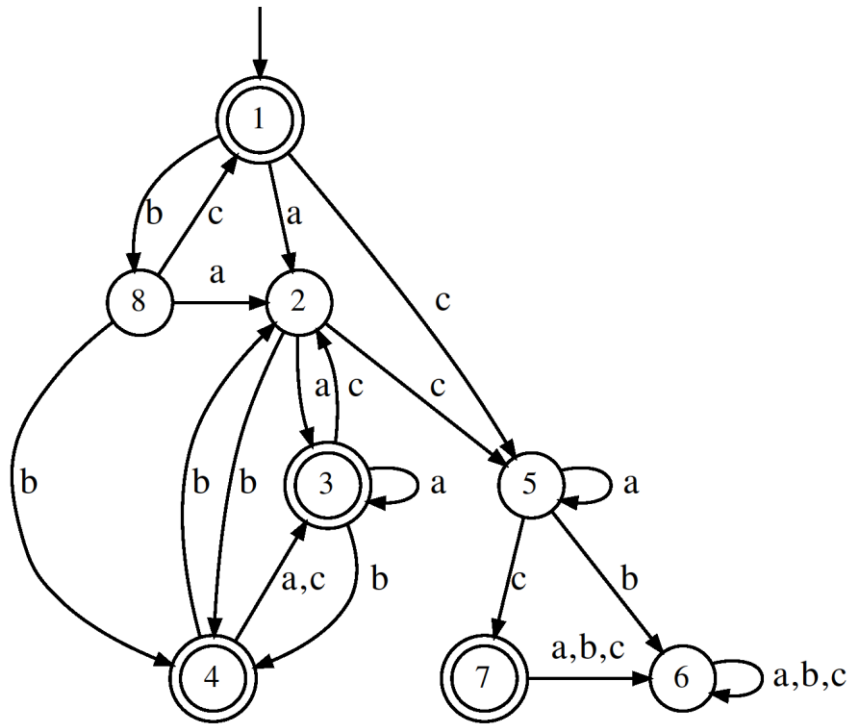
Соответствие паспорту специальности

Специальность 05.13.17 — Теоретические основы информатики

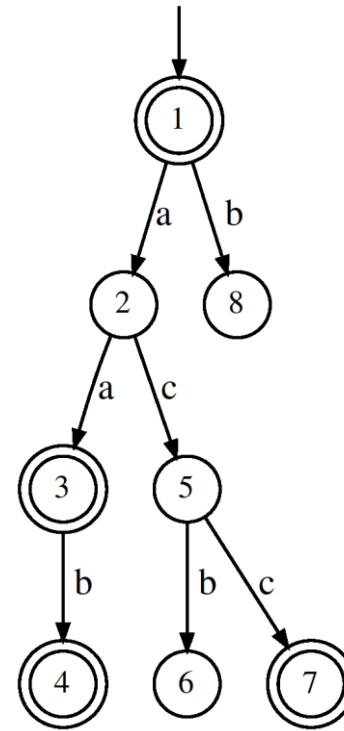
10. Разработка основ математической теории языков и грамматик, теории конечных автоматов и теории графов.

I. Предикаты нарушения симметрии, основанные на кодировании алгоритма обхода графа в глубину

- Depth-first search — DFS.
- $O(M^4 + M^3L^2)$ ДИЗЬЮНКТОВ.



DFS пронумерованный ДКА



дерево DFS

I. Экспериментальные исследования разработанного метода генерации ДКА с использованием DFS-предикатов нарушения симметрии

Число состояний	DFS предикаты, с	BFS предикаты, с	DFASAT, с
10	20,9	20,5	23,3
12	40,4	37,6	240,3
14	82,2	62,4	—
16	205,1	114,1	—
18	601,7	181,9	—
20	2501,6	293,7	—
22	—	453,3	—
24	—	625,1	—
26	—	925,8	—
28	—	1314,4	—
30	—	1635,5	—

- 100 автоматов каждого размера.
- Лимит по времени — час.
- Дальнейшая разработка DFS-предикатов малоперспективна.

I. Предикаты нарушения симметрии, основанные на кодировании алгоритма обхода графа в ширину [1/4]

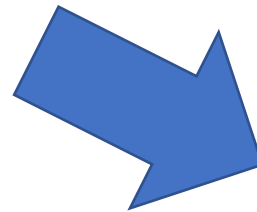
- M — размер искомого ДКА.
- Проанализируем особенности имеющих в сведении переменных и ограничений и разработаем кодировку, использующую эти особенности!

$$\underbrace{\underbrace{\bigwedge_{1 < j \leq M} p_{j,i}}_{O(M)} \bigwedge_{1 \leq i \leq j} p_{j,i}}_{O(M^3)} \leftrightarrow (t_{i,j} \wedge \underbrace{\neg t_{i-1,j} \wedge \dots \wedge \neg t_{1,j}}_{O(M)})$$



$$p_{j,i} = 1 \Leftrightarrow t_{i,j} = 1 \wedge (\forall k < i. t_{k,j} = 0)$$

$$t_{*,j}: (0 \quad \dots \quad 0 \quad 1)$$



$$nt_{i,j} = 1 \Leftrightarrow \forall k < i. t_{k,j} = 0$$

Зададим рекурсивно!

$$\underbrace{\bigwedge_{1 < j \leq M} nt_{0,j}}_{O(M)}$$

$$\underbrace{\bigwedge_{1 < j \leq M} \bigwedge_{1 \leq i < j} nt_{i,j} \leftrightarrow nt_{i-1,j} \wedge \neg t_{i,j}}_{O(M^2)}$$

$$\underbrace{\bigwedge_{1 < j \leq M} \bigwedge_{1 \leq i < j} p_{j,i} \leftrightarrow (t_{i,j} \wedge nt_{i-1,j})}_{O(M^2)}$$

I. Предикаты нарушения симметрии, основанные на кодировании алгоритма обхода графа в ширину [1/4]

- M — размер искомого ДКА.
- Проанализируем особенности имеющих в сведении переменных и ограничений и разработаем кодировку, использующую эти особенности.

$$\underbrace{\underbrace{\bigwedge_{1 < j \leq M} p_{j,i}}_{O(M)} \bigwedge_{1 \leq i \leq j} p_{j,i}}_{O(M^3)} \leftrightarrow (t_{i,j} \wedge \underbrace{\neg t_{i-1,j} \wedge \dots \wedge \neg t_{1,j}}_{O(M)})$$



$$p_{j,i} = 1 \Leftrightarrow t_{i,j} = 1 \wedge (\forall k < i. t_{k,j} = 0)$$

$$t_{*,j}: (0 \quad \dots \quad 0 \quad 1)$$

+ короткие дизъюнкты лучше длинных для средств решения SAT! [Marques-Silva et al, 2009]

$$nt_{i,j} = 1 \Leftrightarrow \forall k < i. t_{k,j} = 0$$

Зададим рекурсивно!

$$\underbrace{\bigwedge_{1 < j \leq M} nt_{0,j}}_{O(M)}$$

$$\underbrace{\bigwedge_{1 < j \leq M} \bigwedge_{1 \leq i < j} nt_{i,j} \leftrightarrow nt_{i-1,j} \wedge \neg t_{i,j}}_{O(M^2)}$$

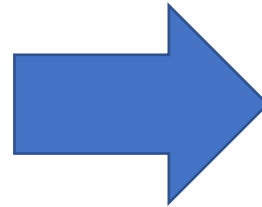
$$\underbrace{\bigwedge_{1 < j \leq M} \bigwedge_{1 \leq i < j} p_{j,i} \leftrightarrow (t_{i,j} \wedge nt_{i-1,j})}_{O(M^2)}$$

I. Предикаты нарушения симметрии, основанные на кодировании алгоритма обхода графа в ширину [2/4]

- M — размер искомого ДКА; L — размер алфавита.

$\mathcal{O}(M^3 + M^2L^2)$ дизъюнктов.

Базовые предикаты нарушения симметрии на основе BFS [Ulyantsev et al., 2015].

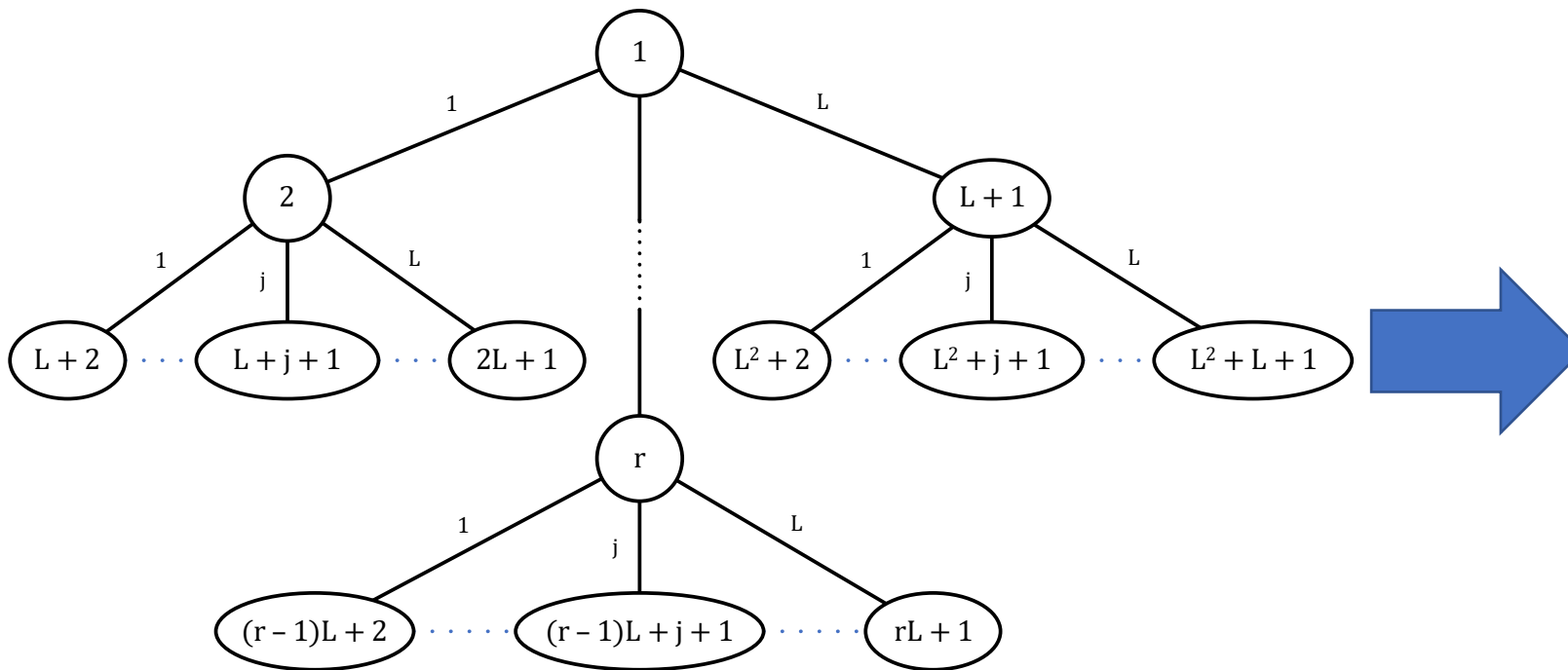


$\mathcal{O}(M^2L)$ дизъюнктов.

Разработанные предикаты нарушения симметрии на основе BFS.

+ помощь средству решения SAT!

I. Предикаты нарушения симметрии, основанные на кодировании алгоритма обхода графа в ширину [3/4]

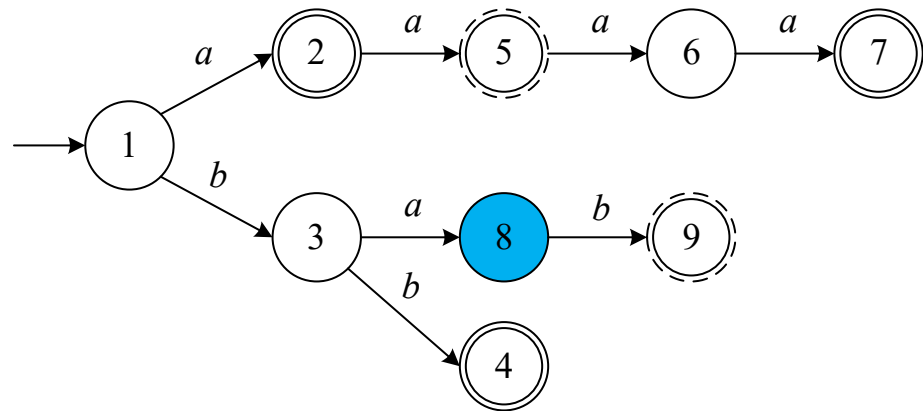


Полное BFS дерево

Дополнительные ограничения:

- У состояния r дети в BFS дереве могут иметь номер от $r + 1$ до $rL + 1$.
- У состояния r не более чем L детей.
- Дети состояния r имеют последовательные номера.

I. Предикаты нарушения симметрии, основанные на кодировании алгоритма обхода графа в ширину [4/4]

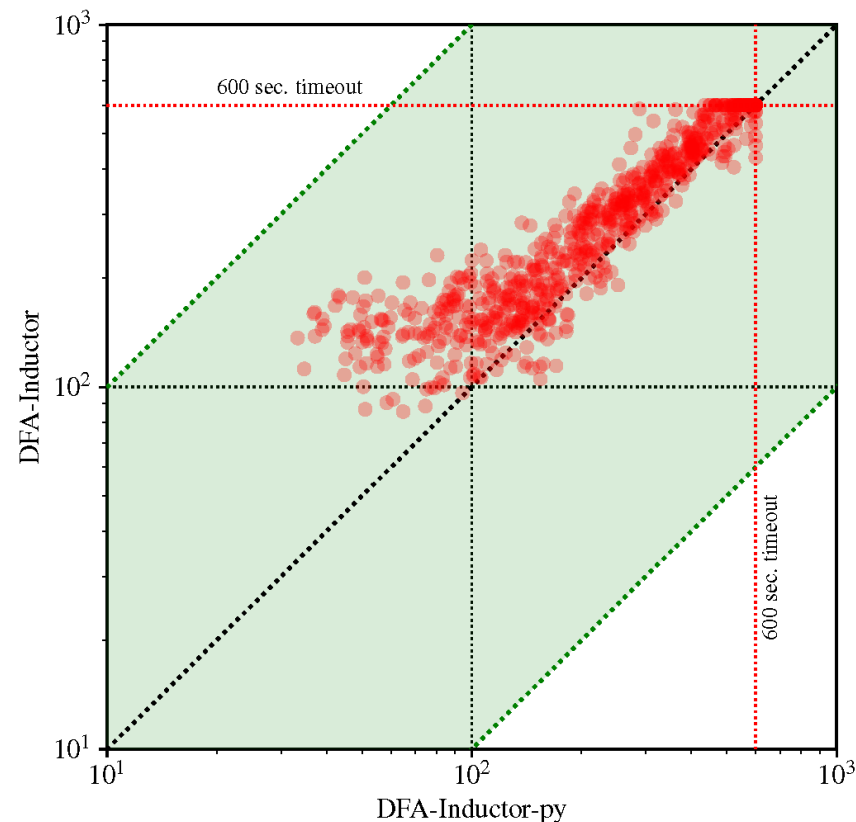
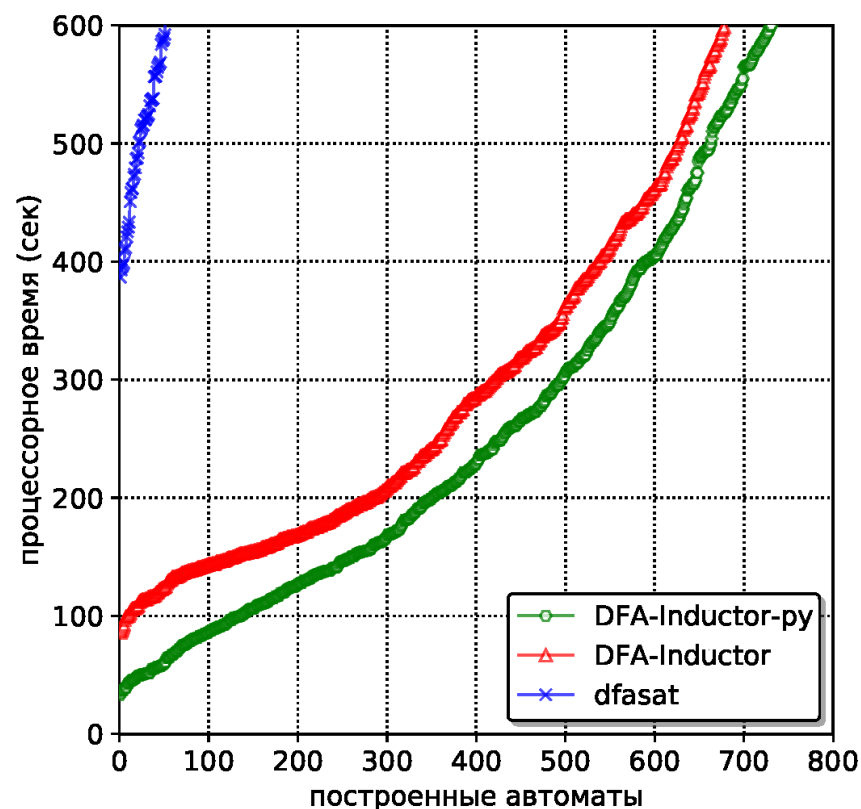


Префиксное дерево

Глубина вершины 8 — $depth(v_8) = 2$.

Значит, если вершина 8 соответствует состоянию q в ДКА, то должен существовать путь из стартового состояния до состояния q длины не более двух.

I. Экспериментальные исследования разработанного метода генерации ДКА с использованием BFS-предикатов нарушения симметрии



- Решена 731 задача (DFA-Inductor-py) против 678 (DFA-Inductor) за 600 секунд.
- 91 % задач решаются быстрее с помощью DFA-Inductor-py.

II. Точный метод генерации ДКА по избыточному набору примеров поведения с использованием сведения к задаче выполнимости и подхода уточнения абстракции по контрпримерам [1/3]

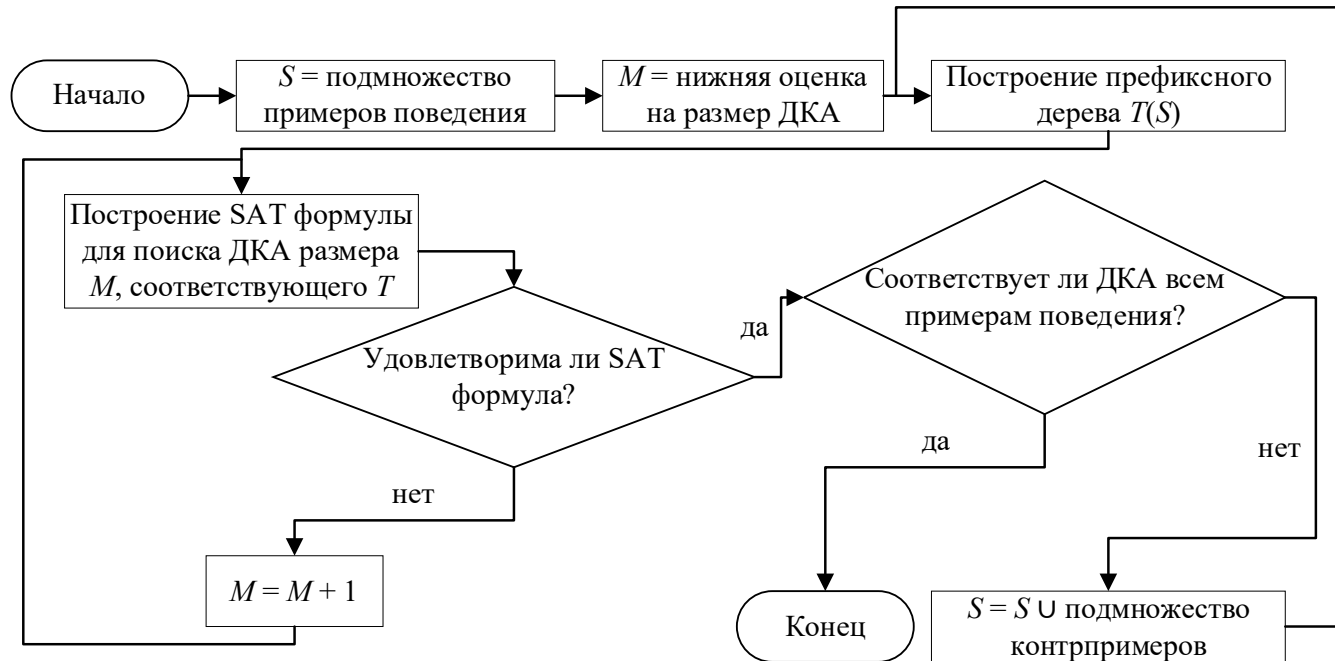
- N — размер префиксного дерева; M — размер искомого ДКА; L — размер алфавита.
- $\mathcal{O}(N \times M + M^2 \times L)$ — булевых переменных.
- $\mathcal{O}(N \times M^2 + M^2 \times L)$ — дизъюнктов.
- Для конкретного автомата M и L фиксированы. N — зависит от числа и длины примеров поведения.
- Обычно, больше примеров — лучше. Но слишком много примеров — плохо!



II. Точный метод генерации ДКА по избыточному набору примеров поведения с использованием сведения к задаче выполнимости и подхода уточнения абстракции по контрпримерам [2/3]

Рассмотреть только часть примеров поведения, но как их выбрать?

- *Случайно?* Можем отбросить значимые примеры поведения.
- *Эвристика?* Аналогично.
- *Алгоритм уточнения абстракции по контрпримерам (CEGAR)* [Clarke et al., 2003]?



Инкрементальное средство решения SAT

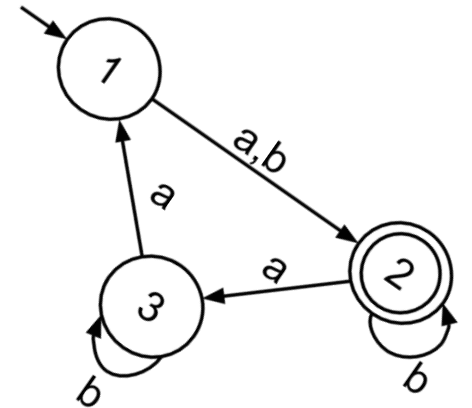
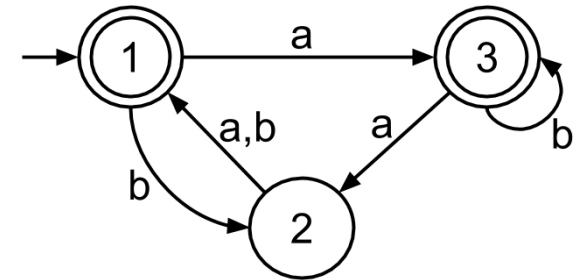
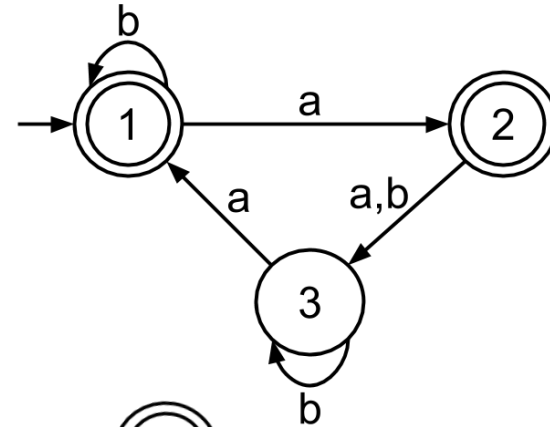
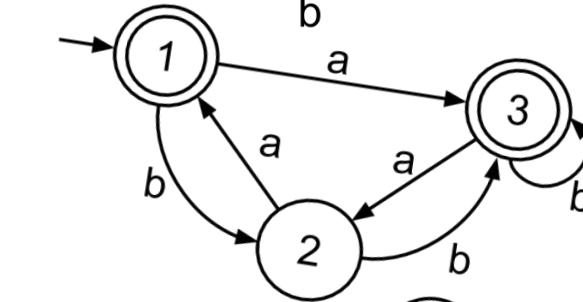
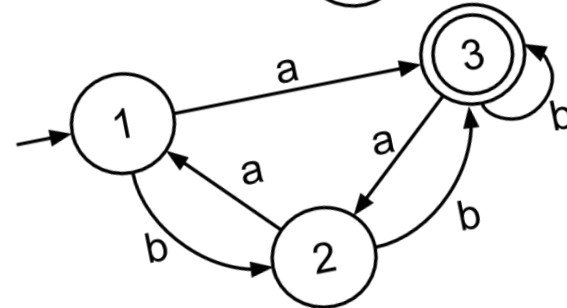
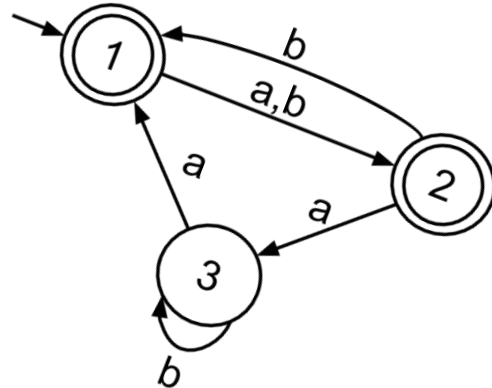
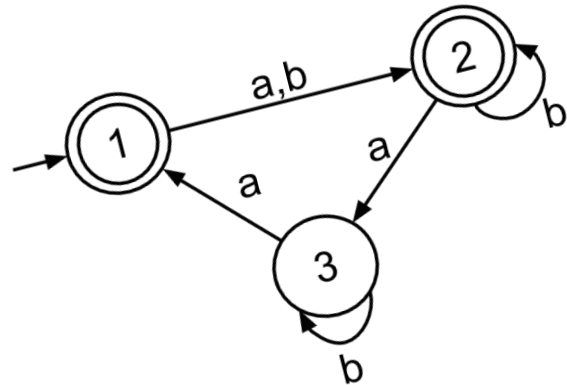
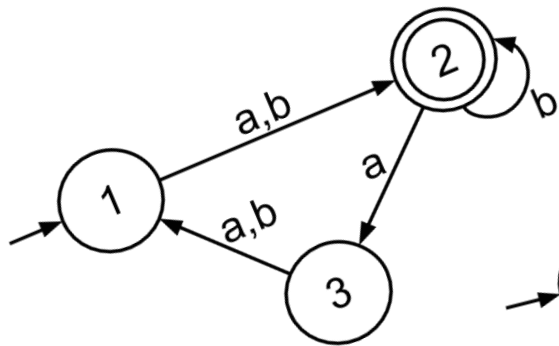
II. Точный метод генерации ДКА по избыточному набору примеров поведения с использованием сведения к задаче выполнимости и подхода уточнения абстракции по контрпримерам [3/3]

- При большом числе примеров поведения ($|S_+| + |S_-| \geq 200 \times M$) выигрыш по скорости работы метода от двух раз и тем больше, чем больше число примеров.

III. Метод генерации всех неизоморфных ДКА минимального размера, удовлетворяющих заданным примерам поведения [1/3]

Можно ли найти все различные (неизоморфные) автоматы минимального размера, соответствующие имеющимся примерам поведения?

- | | |
|--------|-------|
| S_+ | S_- |
| • a | • bab |
| • bb | • aa |
| • aaaa | |



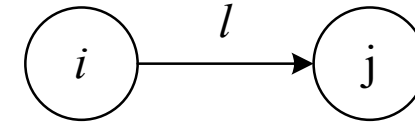
III. Метод генерации всех неизоморфных ДКА минимального размера, удовлетворяющих заданным примерам поведения [2/3]

Можно ли найти все различные (неизоморфные) автоматы минимального размера, соответствующие имеющимся примерам поведения? Эффективного решения не предлагалось ранее!

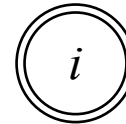
- Неточными методами — нельзя в принципе.
- Точными методами — можно, но перебирать придется слишком много.
- С BFS-предикатами можно решить эффективно!
- Бонус: можно доказать единственность существующего решения.

III. Метод генерации всех неизоморфных ДКА минимального размера, удовлетворяющих заданным примерам поведения [3/3]

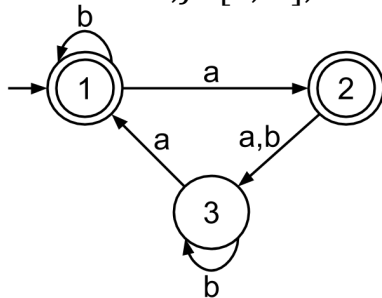
- $y_{i,l,j} = 1 \Leftrightarrow$ в автомате есть переход по символу l из состояния i в состояние j .



- $z_i = 1 \Leftrightarrow$ в автомате состояние i является допускающим.



- Множества $\{y_{i,l,j} = 1\}_{i,j \in [1;M]; l \in \Sigma}$ и $\{z_i = 1\}_{i \in [1;M]}$ определяют найденный ДКА.



$$\begin{array}{lll}
 y_{1,a,2} = 1; & y_{2,b,3} = 1; & z_1 = 1; \\
 y_{1,b,1} = 1; & y_{3,a,1} = 1; & z_2 = 1; \\
 y_{2,a,3} = 1; & y_{3,b,3} = 1; & z_3 = 0.
 \end{array}$$

- Запретим средству решения SAT данный автомат, добавим блокирующее ограничение!

$$\neg y_{1,a,2} \wedge \neg y_{1,b,1} \wedge \neg y_{2,a,3} \wedge \neg y_{2,b,3} \wedge \neg y_{3,a,1} \wedge \neg y_{3,b,3} \wedge \neg z_1 \wedge \neg z_2$$

- Так как BFS-предикаты допускают единственного представителя для каждого класса эквивалентности по изоморфизму, то после добавления блокирующего ограничения, изоморфные автоматы рассматриваться не будут!

III. Экспериментальные исследования разработанного метода генерации всех неизоморфных ДКА минимального размера, удовлетворяющих заданным примерам поведения

Число состояний	Нашлось более одного ДКА	Перезапуск средства, с	Инкрементальное средство, с	Переборный алгоритм, с
10	89/100	15,7	5,3	2756,2
11	94/100	19,9	7,3	—
12	90/100	28,0	9,9	—
13	92/100	185,5	18,1	—
14	87/100	408,5	49,0	—
15	95/100	571,1	174,1	—

100 экземпляров задач для каждого числа состояний.

Результаты

- Предикаты нарушения симметрии, основанные на кодировании алгоритмов обхода графа в ширину и в глубину, для сокращения пространства поиска при решении задачи выполнимости
 - DFS-предикаты ранее не предлагались, но оказались неэффективны.
 - Булева формула меньшего размера.
 - Повышение эффективности за счет прироста производительности (сгенерирован 731 автомат против 678 за 10 минут; 91 % задач решается быстрее).
- Точный метод генерации ДКА по избыточному набору примеров поведения с использованием сведения к задаче выполнимости и подхода уточнения абстракции по контрпримерам.
 - Повышение эффективности за счет возможности генерировать ДКА по избыточному множеству примеров поведения (выигрыш от двух раз).
- Метод генерации всех неизоморфных ДКА минимального размера, удовлетворяющих заданным примерам поведения, с использованием предикатов нарушения симметрии и программных средств решения задачи выполнимости.
 - Решение возможно только с использованием BFS-предикатов.
 - Повышение эффективности относительно переборных алгоритмов.
 - Можно доказать единственность существующего решения.

Результаты

Разработанные методы реализованы в рамках программного комплекса DFA-Inductor-py [<https://github.com/ctlab/DFA-Inductor-py>].

Для работы с SAT используется библиотека PySAT — <https://pysathq.github.io/> [Ignatiev A. et al., 2018].

- Генерация ДКА минимального размера по заданным примерам поведения.
 - Предикаты нарушения симметрии на основе поиска большой клики.
 - Граф несовместимости (полный или заданной глубины).
 - Предикаты нарушения симметрии на основе BFS.
 - Предикаты нарушения симметрии на основе DFS.
 - Уточнение абстракции по контрпримерам.
- Генерация всех ДКА минимального размера.
 - Сведение к SAT.
 - Переборный алгоритм с возвратами.

Публикации

- Scopus:

1. Ulyantsev V., **Zakirzyanov I.**, Shalyto A. BFS-Based Symmetry Breaking Predicates for DFA Identification // Proceedings of 9th International Conference on Language and Automata Theory and Applications (LATA). — 2015, pp. 611–622.
2. **Zakirzyanov I.**, Shalyto A., Ulyantsev V. Finding All Minimum-Size DFA Consistent with Given Examples: SAT-Based Approach // Proceedings of 6th International Symposium “From Data to Models and Back (DataMod)”. — 2017, pp. 117–131.
3. Ovsiannikova P., Chivilikhin D., Ulyantsev V., Stankevich A., **Zakirzyanov I.**, Vyatkin V., Shalyto A. Active Learning of Formal Plant Models For Cyber-Physical Systems // Proceedings of 16th IEEE International Conference on Industrial Informatics (INDIN). — 2018. pp. 719–724.
4. **Zakirzyanov I.**, Morgado A., Ignatiev A., Ulyantsev V., Marques-Silva J. Efficient Symmetry Breaking for SAT-Based Minimum DFA Inference // Proceedings of 13th International Conference on Language and Automata Theory and Applications (LATA). — 2019, pp. 159–173.

- ВАК:

5. Закирзянов И. Т. Построение детерминированных конечных автоматов по примерам поведения с использованием подхода уточнения абстракции по контрпримерам // Научно-технический вестник информационных технологий, механики и оптики. — 2020. — Т. 20, № 3. — С. 394–401.

Публикации

- Другие:

6. Закирзянов, И. Т. Разработка предикатов нарушения симметрии на основе поиска в ширину для построения детерминированных конечных автоматов // Сборник тезисов докладов Всероссийского конгресса молодых ученых. — 2015.
7. Закирзянов, И. Т. Эмпирическая оценка производительности программных средств решения задачи SAT для синтеза ДКА // Сборник тезисов докладов Всероссийского конгресса молодых ученых. — 2016.
8. Закирзянов, И. Т., Ульянов В.И. Сравнительный анализ методов задания ограничений типа at-most-one на примере задачи построения ДКА с использованием программных средств решения SAT // Сборник тезисов докладов Всероссийского конгресса молодых ученых. — 2017.
9. Закирзянов, И. Т. Разработка предикатов нарушения симметрии для построения автоматных моделей программного обеспечения по заданной спецификации в виде темпоральных формул. — 2018.
10. Закирзянов, И. Т. Построение минимального ДКА на основе сведения к SAT с использованием предположений // Сборник тезисов докладов конгресса молодых ученых. — 2020.

Публикации не по теме диссертации

- Scopus:

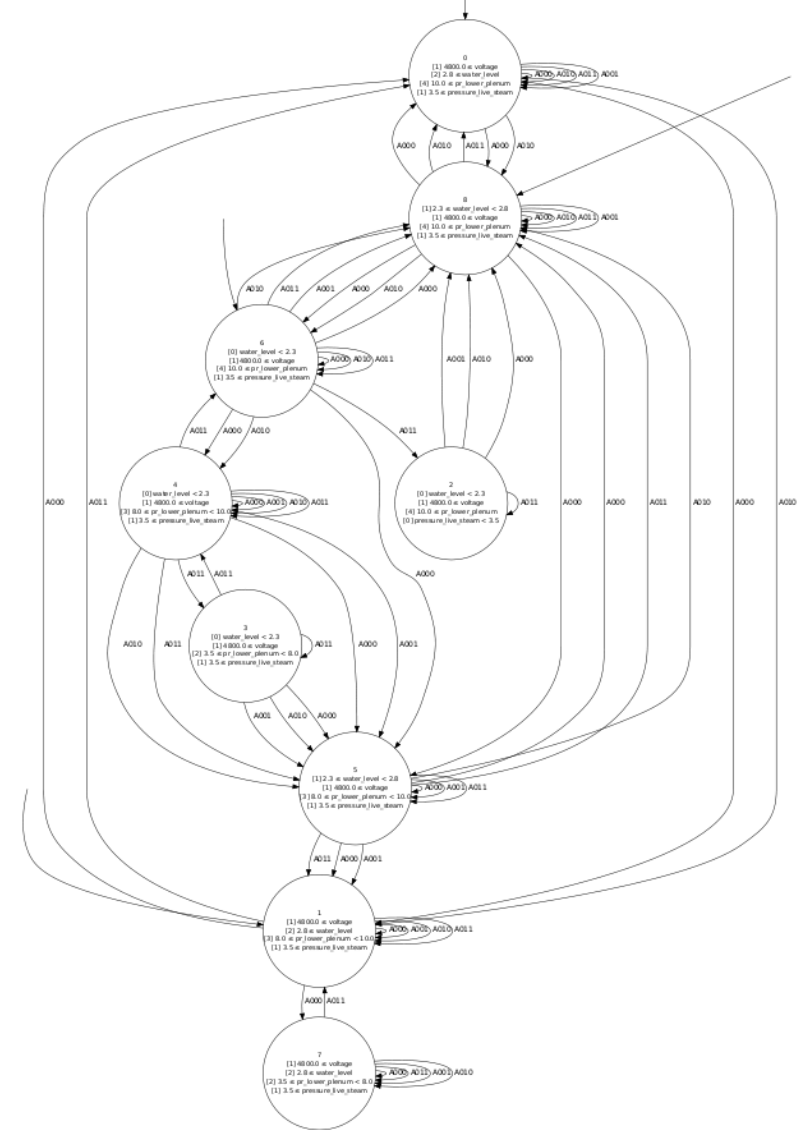
Kachalsky I., **Zakirzyanov I.**, Ulyantsev V. Applying Reinforcement Learning and Supervised Learning Techniques to Play Hearthstone // Proceedings of 16th IEEE International Conference on Machine Learning and Applications (ICMLA). — 2017. pp. 1145–1148. [*разработка методов машинного обучения для создания агента, способного играть в видеоигру*]

Апробация работы

- Международные конференции:
 1. 9th International Conference on Language and Automata Theory and Applications (LATA 2015). 2015, Ницца, Франция.
 2. 6th International Symposium “From Data to Models and Back (DataMod). 2017, Тренто, Италия.
 3. 16th IEEE International Conference on Industrial Informatics (INDIN 2018). 2018, Порту, Португалия.
 4. 13th International Conference on Language and Automata Theory and Applications (LATA 2019). 2015, Санкт-Петербург.
- Всероссийские конференции:
 5. IV-VII Всероссийский Конгресс молодых ученых. 2015-2018, Санкт-Петербург.
 6. IX Конгресс молодых ученых. 2020, Санкт-Петербург.
 7. XLVI Научная и учебно-методическая Конференция Университета ИТМО. 2017, Санкт-Петербург.
 8. XLVIII Научная и учебно-методическая Конференция Университета ИТМО. 2019, Санкт-Петербург.

Внедрение результатов работы

- Проект SAUNA (“Integrated safety assessment and justification of nuclear power plant automaton”) выполненный, исследовательской группой “IT in Automation” кафедры электротехники и автоматике университета Аалто, Финляндия, в рамках Финской программы исследований безопасности атомных электростанций — SAFIR2018.
 - Метод генерации моделей подсистем атомных электростанций.
 - Подход уточнения абстракции по контрпримерам.
 - Нарушение симметрии.
- Учебный процесс факультета ИТиП Университета ИТМО, курс “Проектирование автоматных программ” программы бакалавриата “Математические модели и алгоритмы в разработке программного обеспечения”.
- Грант РФФИ 18-37-00425 по теме «Разработка эффективных методов машинного обучения для построения детерминированных конечных автоматов на основе решения задачи выполнимости».



Модель одной из защитных подсистем атомной электростанции [Buzhinskii I., 2018]

Спасибо за внимание!

III. Метод генерации всех неизоморфных ДКА минимального размера, удовлетворяющих заданным примерам поведения [4/5]

- Как добавлять блокирующее ограничение?
 - Перезапускать средство решения SAT с новой формулой — **каждый раз поиск будет начинаться заново.**
 - Использовать средство решения SAT в инкрементальном режиме — **используются результаты предыдущих поисков!**

В инкрементальном режиме средство после нахождения решения, приостанавливает свое выполнение, ждет новых ограничений и продолжает поиск дальше.