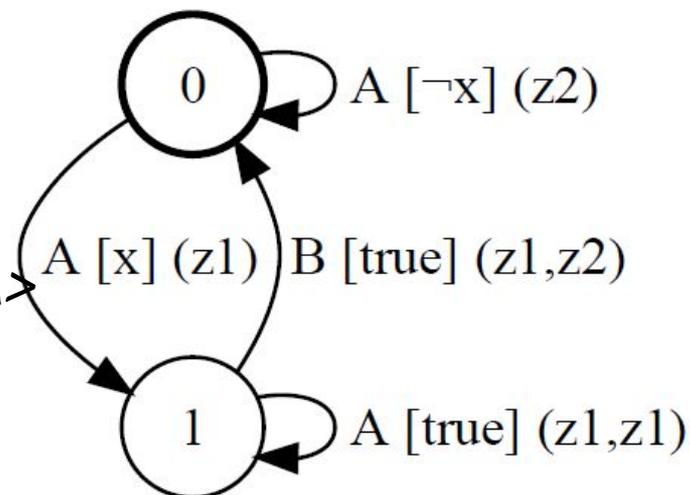


Применение методов решения задачи удовлетворения ограничениям для построения управляющих конечных автоматов по сценариям работы

Владимир Ульяновцев
Научный руководитель: Федор Царев

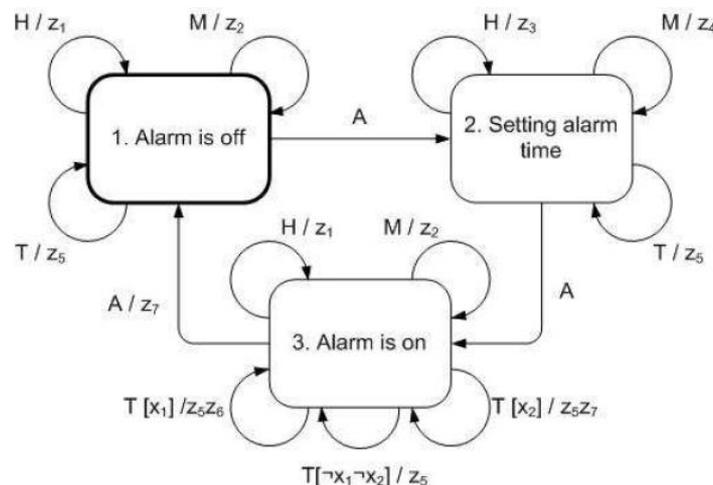
Управляющие автоматы и сценарии работы программы

- Управляющий автомат:
 - входные события
 - входные переменные
 - выходные воздействия
- Сценарий работы – последовательность троек $\langle e, f, A \rangle$
 - e – входное событие
 - f – охранный условие – Булева формула, зависящая от входных переменных
 - A – последовательность выходных воздействий
- Приведенный управляющий автомат:
 - Удовлетворяет $\langle A, \neg x, (z2) \rangle \langle A, x, (z1) \rangle$
 - Не удовлетворяет $\langle A, x, (z2) \rangle$



Пример управляющего автомата

- Часы с будильником
- Четыре события
 - Н – кнопка “Н” нажата
 - М – кнопка “М” нажата
 - А – кнопка “А” нажата
 - Т – происходит каждый момент времени
- Две входные переменные
- Семь выходных



Требования, предъявляемые к

автоматной программе

- **Непротиворечивость** – не должно быть двух переходов, исходящих из одного состояния автомата и одновременно выполнимых при некоторой комбинации события и входных переменных
- **Полнота** – любой комбинации события и входных переменных должен соответствовать переход в каждом состоянии

Постановка задачи

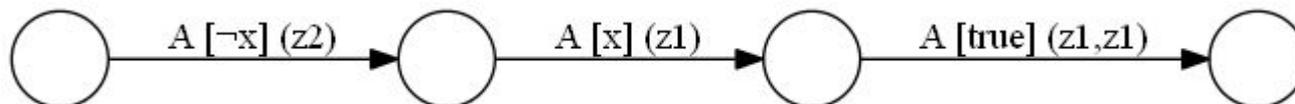
- Рассматриваются автоматные программы с единственным объектом управления
- Входные данные:
 - Набор сценариев работы программы (S_c)
 - Число состояний управляющего автомата (C)
- Необходимо найти управляющий автомат, состоящий из C состояний и удовлетворяющий всем сценариям работы

Предлагаемый метод

- Ulyantsev V., Tsarev F. Extended Finite-State Machine Induction using **SAT-Solver** / Proceedings of the Tenth International Conference on Machine Learning and Applications, ICMLA 2011, Honolulu, HI, USA, 18-21 December 2011. IEEE Computer Society, 2011. Vol. 2. P. 346–349
 - Не гарантируется полнота
- Выразить требование полноты в виде ограничений на целочисленные переменные (constraint satisfaction problem, **CSP**)

Основная идея

- Каждому сценарию работы соответствует «линейный» автомат



- «Раскраска» сценариев
 - Каждому «состоянию» каждого сценария работы необходимо поставить в соответствие состояние искомого управляющего автомата
 - Состояния управляющего автомата будем различать цветами

Этапы работы алгоритма

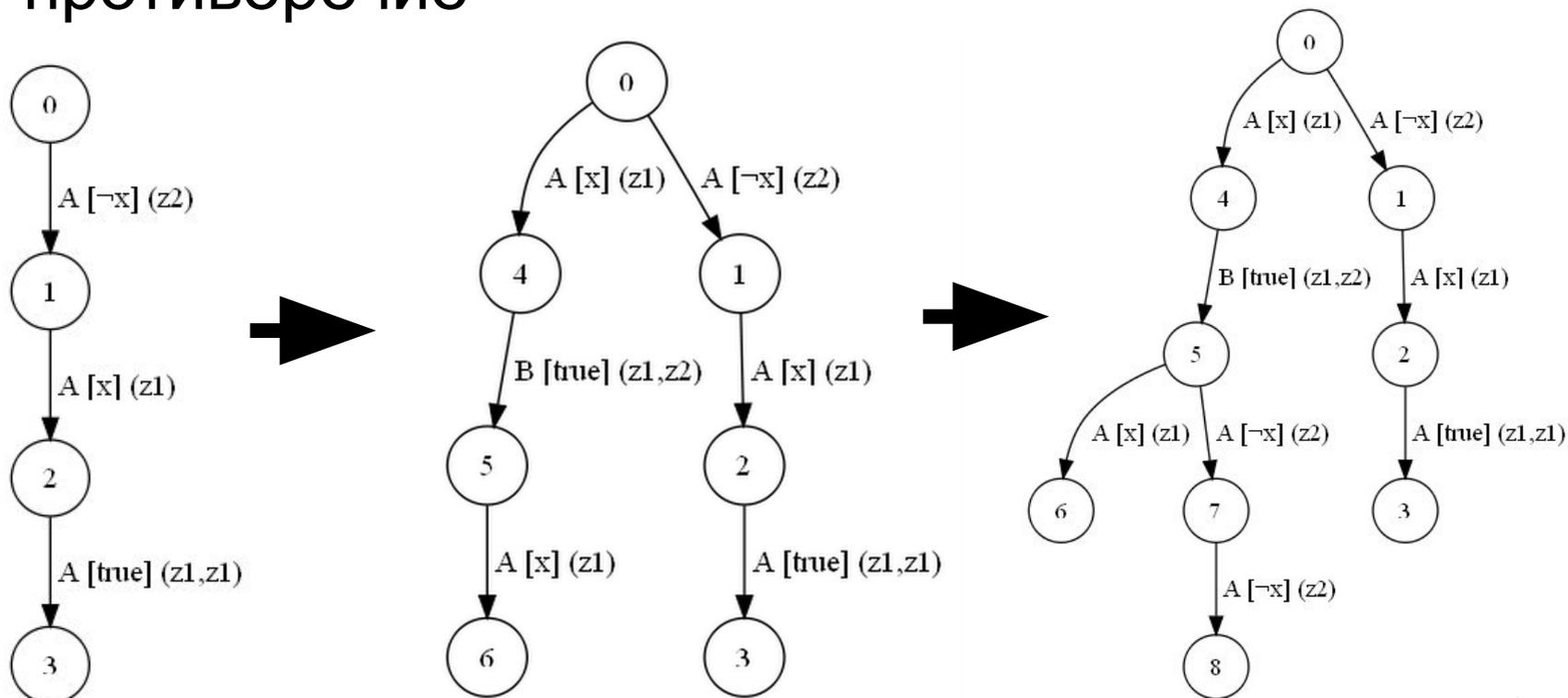
- Построение дерева сценариев
- Построение графа совместимости
- **Построение булевой КНФ-формулы**
 - Набор ограничений на целочисленные переменные
- Запуск сторонней программы, находящей решение
- Построение искомого управляющего автомата

Предварительные вычисления

- Для каждой пары охранных условий, встречающихся в заданных сценариях:
 - Равны ли как булевы формулы
 - Имеют ли общую выполняющую подстановку
- Асимптотическая оценка:
 - $O(n^2 2^{2m})$ где n – общий размер сценариев, m – наибольшее число переменных, встречающихся в охранном условии (на практике m не превышает пяти)

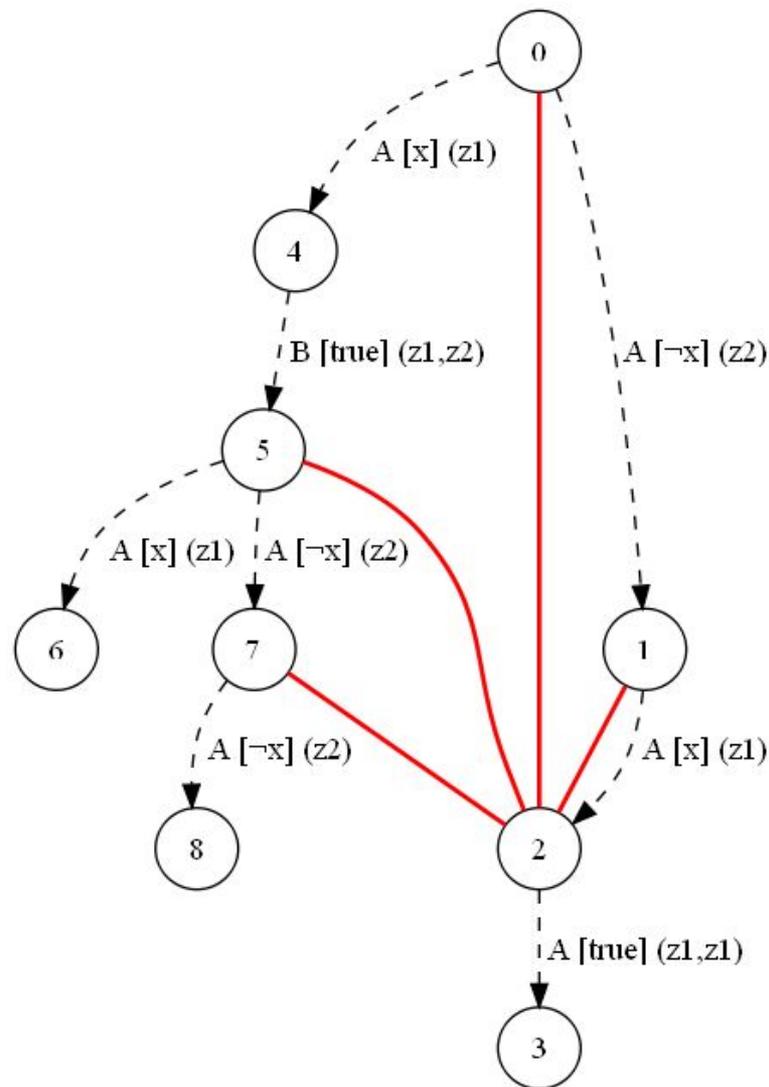
1. Построение дерева сценариев

- Аналогично построению бора
- Алгоритм прерывается, если найдено противоречие



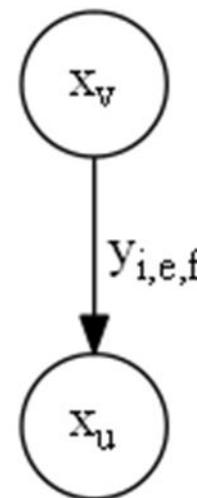
2. Построение графа СОВМЕСТИМОСТИ

- Вершины совпадают с вершинами дерева
- Вершины соединены ребром, если существует последовательность, различающая их
- Для каждой вершины, начиная с листьев, строится множество несовместимых с ней вершин
- Используется динамическое программирование



3. Используемые переменные

- Целочисленные переменные:
 - x_v – цвет, в который покрашена вершина v ($1 \leq x_v \leq C$)
 - $y_{i,e,f}$ – номер состояния, в которое ведет переход из состояния i , помеченный событием e и охранным условием f ($1 \leq y_{i,e,f} \leq C$)



3. Построение ограничений для требования непротиворечивости

- Типы ограничений:
 - $x_v \neq x_u$ – цвета несовместимых вершин должны быть различны
 - $(x_v = i) \Rightarrow (y_{i,e,f} = x_u)$ – цвета вершин дерева не должны противоречить переходам автомата

3. Построение ограничений для

требования полноты

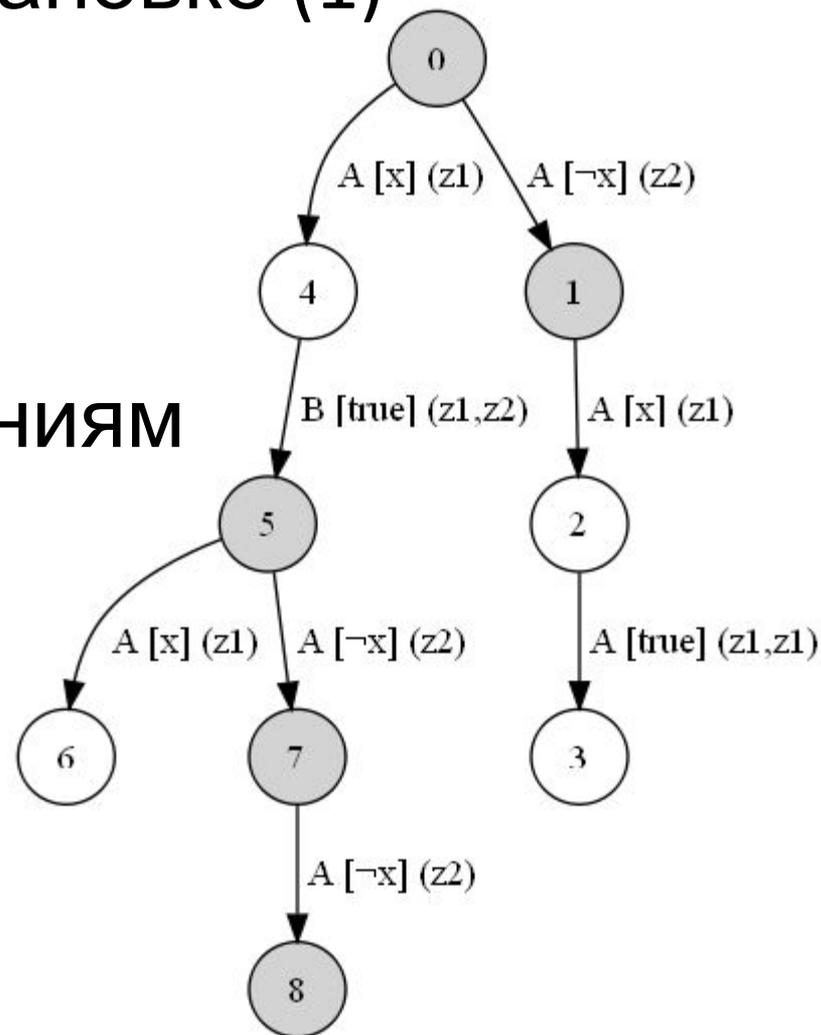
- При условии непротиворечивости можно выразить условие полноты
- Сумма выполняющих подстановок S равна 2^m , где m – число переменных, используемых в переходах данного состояния и события

4. Использование программного средства, решающего задачу удовлетворения ограничениям

- Использовался пакет *choco* для нахождения выполняющей подстановки
- Пакет для языка *Java*, предназначенный, в том числе, для нахождения оптимальных расписаний

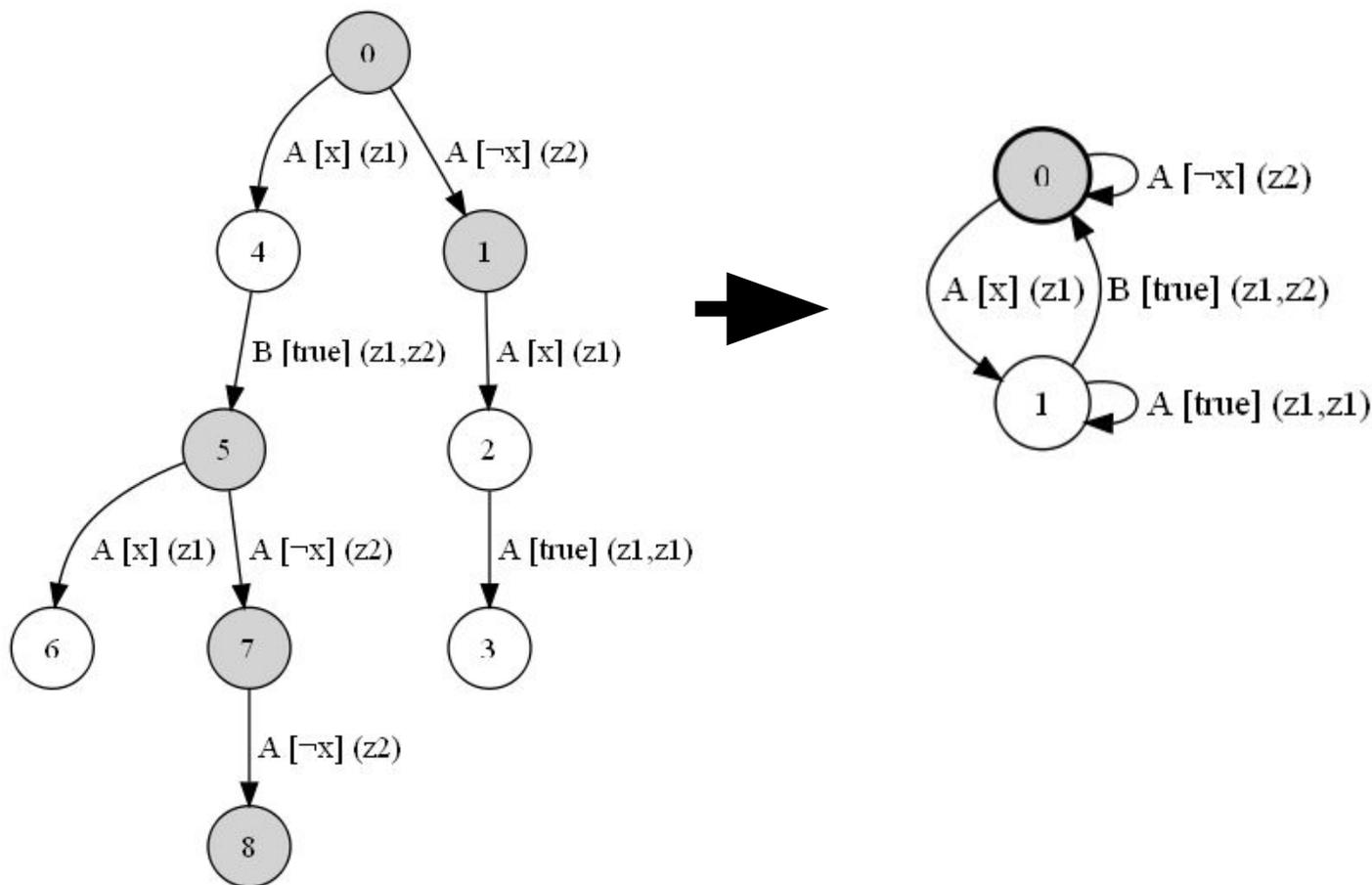
5. Построение управляющего автомата по выполняющей подстановке (1)

- Раскраска дерева сценариев
- Цвет вершины соответствует значениям X_v



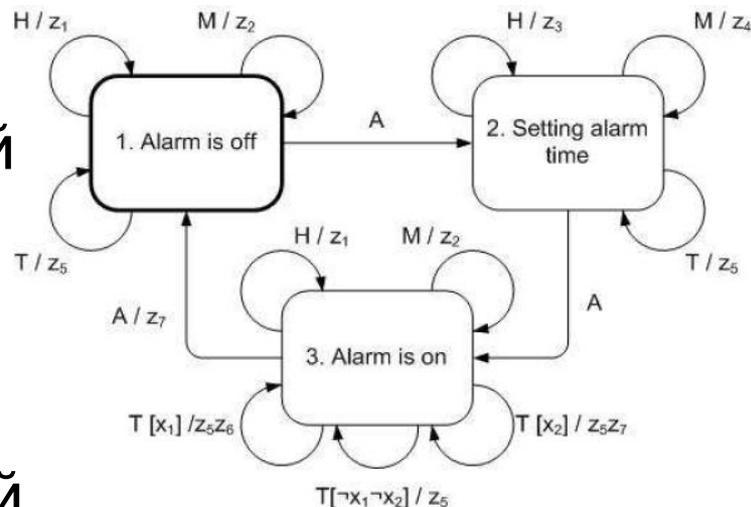
5. Построение управляющего автомата по выполняющей подстановке (2)

- Вершины одинаковых цветов сливаются



Экспериментальные исследования

- Управляющий автомат для часов с будильником:
 - 38 сценариев работы суммарной длины 242 + один сценарий
- Наборы случайных сценариев работы:
 - Генерируется случайный полный по переменным автомат
 - Сценарии – случайные пути в автомате
 - Найдены наборы сценариев, на которых проявляется преимущество разработанного метода



Результаты

- Разработан и реализован метод построения управляющих автоматов, основанный на сведении задачи к задаче удовлетворения ограничениям
- Экспериментальные исследования продемонстрировали существование задач, для которых метод находит качественно лучшее решение

Спасибо за внимание!

ulyantsev@rain.ifmo.ru