

УНИВЕРСИТЕТ ИТМО

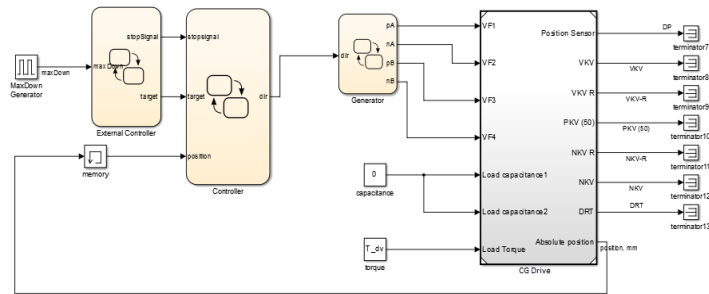
Разработка методов формальной
верификации промышленных кибер-
физических систем в замкнутом цикле

Овсянникова П.А. гр. М4236с

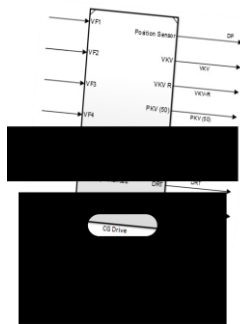
Научный руководитель: Ульянов В.И., к.т.н., доцент каф. КТ

Санкт-Петербург, 2018 г.

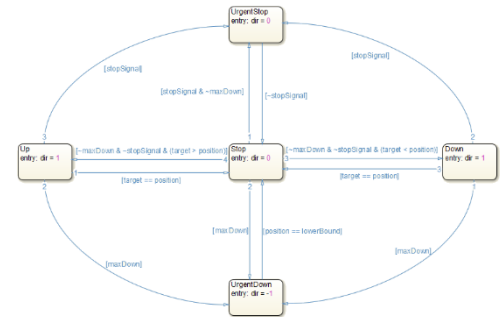
Кибер-физическая система (КФС)



КФС

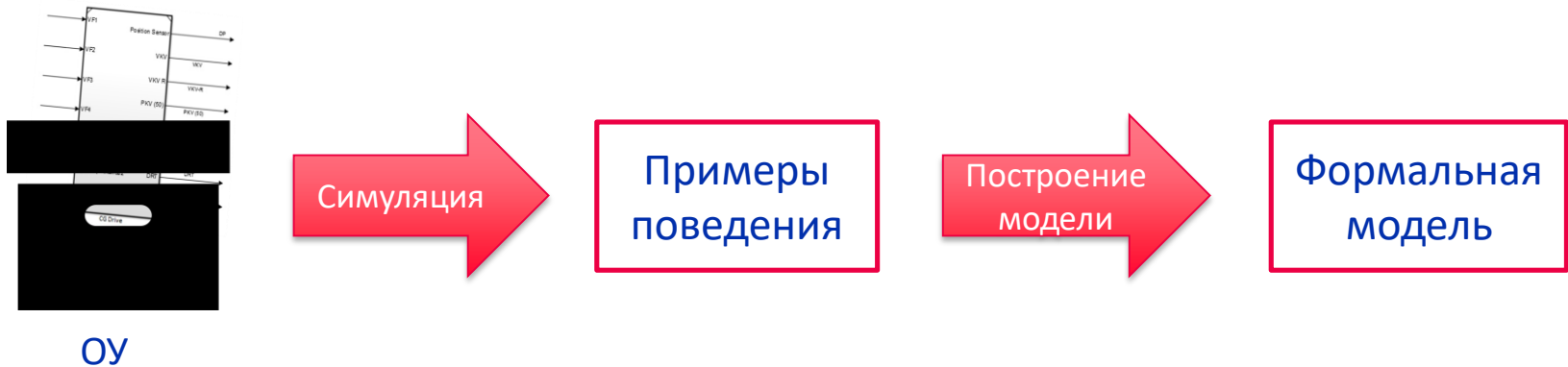


объект управления (ОУ)
черный ящик



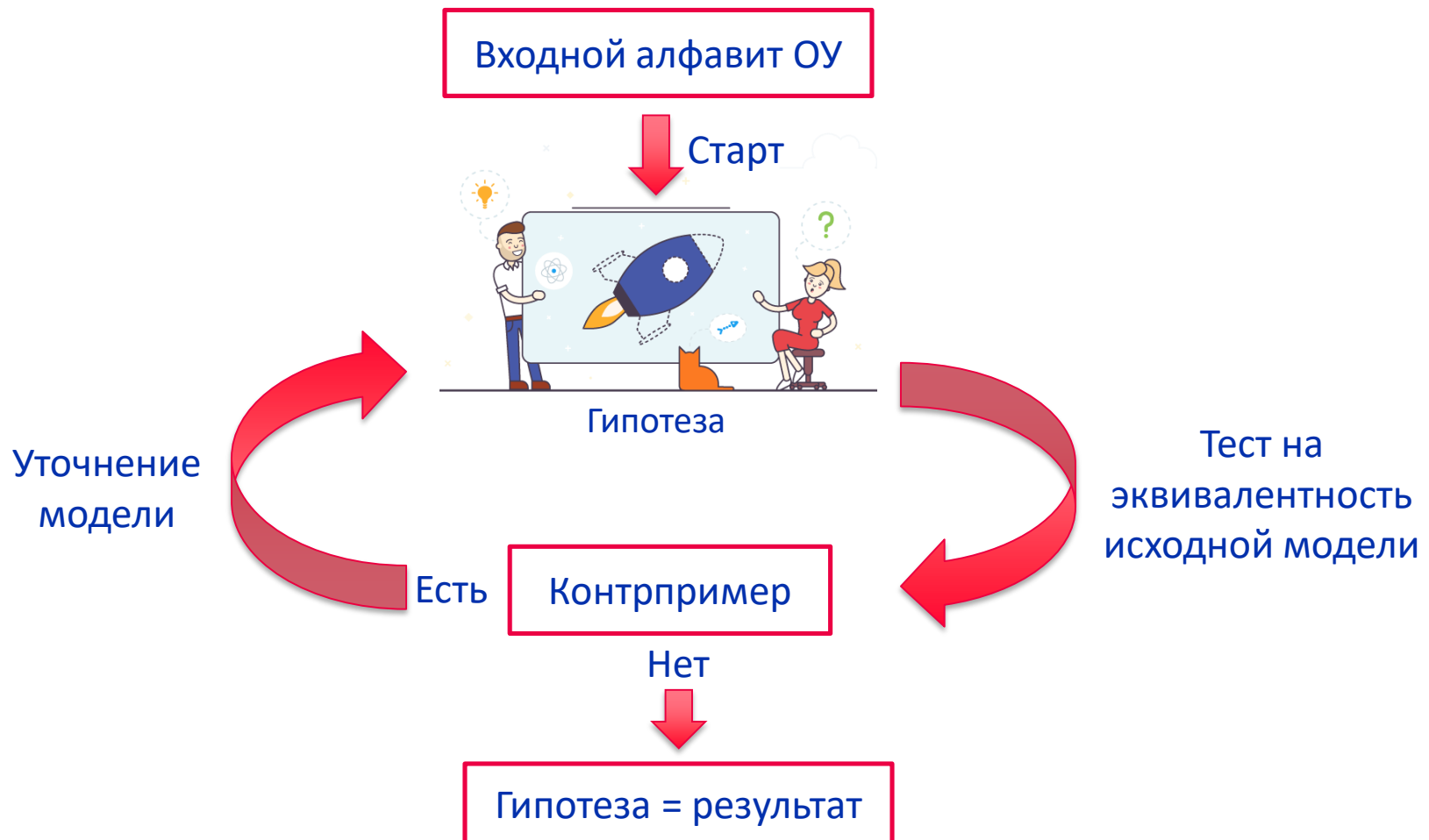
контроллер (автомат)

Построение формальных моделей: пассивное обучение



- A. Maier, “Online passive learning of timed automata for cyber-physical production systems” // INDIN 2014
- G. Giantamidis, S. Tripakis, “Learning moore machines from input output traces” // International Symposium on Formal Methods, 2016
- I. Buzhinsky, V. Vyatkin, “Automatic inference of finite-state plant models from traces and temporal properties” // IEEE Transactions on Industrial Informatics, 2017

Построение формальных моделей: активное обучение (алгоритм L^*)



Актуальность исследования

Пассивное обучение

- Полностью зависит от предоставленных обучающих данных

Активное обучение L^*

- Предназначено для систем **с контекстом**
- Требуется наличия оракула
- Неочевидно как учитывать вещественные переменные

Однако

- ✓ ОУ в идеальном случае не имеет контекста
- ✓ ОУ, как правило, содержит вещественные переменные

Разработка методов генерации формальных моделей представленных в виде **черного ящика дискретных бесконтекстных ОУ** с поддержкой вещественных переменных является **актуальной**.

Цель

- Разработка метода формальной верификации промышленных кибер-физических систем в замкнутом цикле на основе автоматического синтеза формальной модели объекта управления с помощью активного обучения

Задачи

- Разработать алгоритм активного обучения бесконтекстных детерминированных ОУ
- Обеспечить поддержку вещественных переменных
- Реализовать и протестировать разработанный алгоритм на симуляционной модели КФС

Основные понятия

➤ Входной символ

$I_1..I_n$ – входные переменные, $i_1..i_n$ – значения

$D_{I_1}..D_{I_n}$ – дискретные множества значений

$$\left. \begin{array}{l} i_1 \in D_{I_1} \\ \vdots \\ i_n \in D_{I_n} \end{array} \right\} i_m \in D_{I_m}$$

Входной символ $I_k = (i_{1k}, \dots, i_{nk})$

➤ Выходной символ:

$O_1..O_n$ – выходные переменные, $o_1..o_n$ – значения

$D_{O_1}..D_{O_n}$ – дискретные множества значений

$$\left. \begin{array}{l} o_1 \in D_{O_1} \\ \vdots \\ o_n \in D_{O_n} \end{array} \right\} o_m \in D_{O_m}$$

Выходной символ $O_k = (o_{1k}, \dots, o_{nk})$



Применение обхода в ширину для построения модели дискретного ОУ

Входные переменные:

F (forward) = $\{0, 1\}$

B (backward) = $\{0, 1\}$

Входной алфавит:

$\Sigma = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$

Выходные переменные:

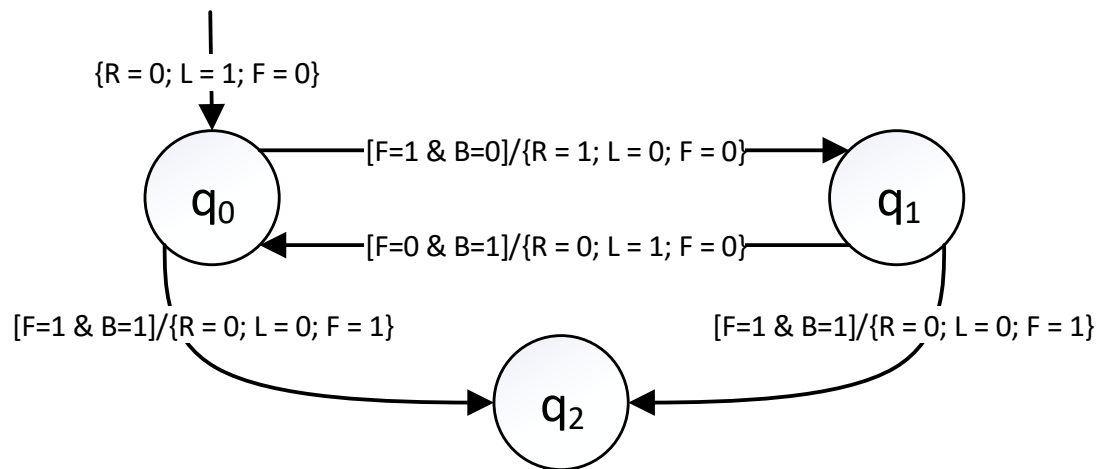
R (right) = $\{0, 1\}$

L (left) = $\{0, 1\}$

F (failure) = $\{0, 1\}$

Выходной алфавит:

$O = \{(0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1)\}$

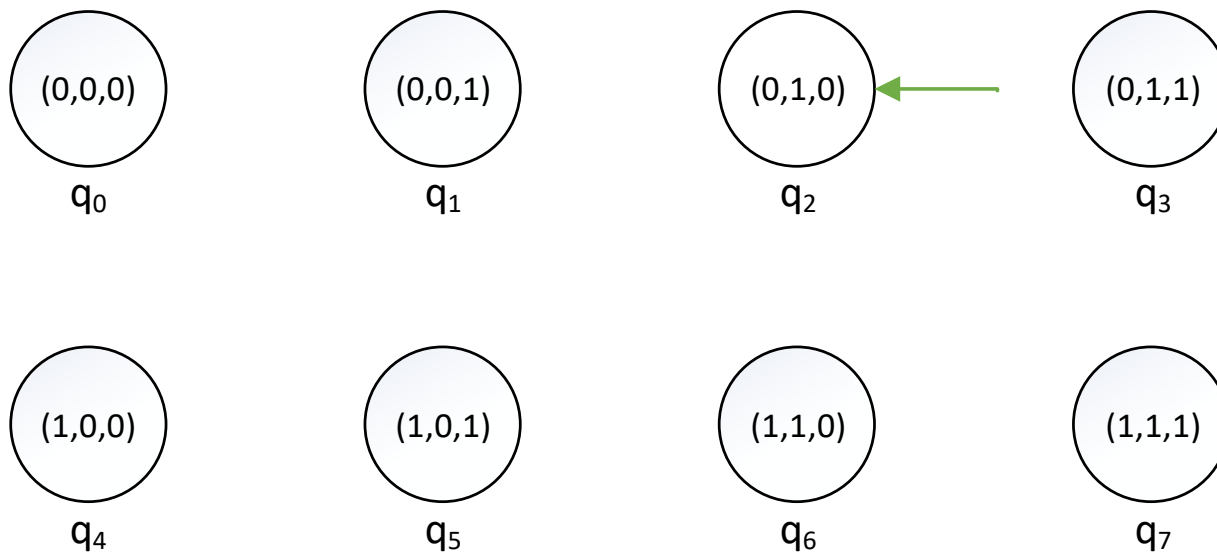


Пример

Начальное состояние – q_2

nextStates: q_2

processedStates: empty

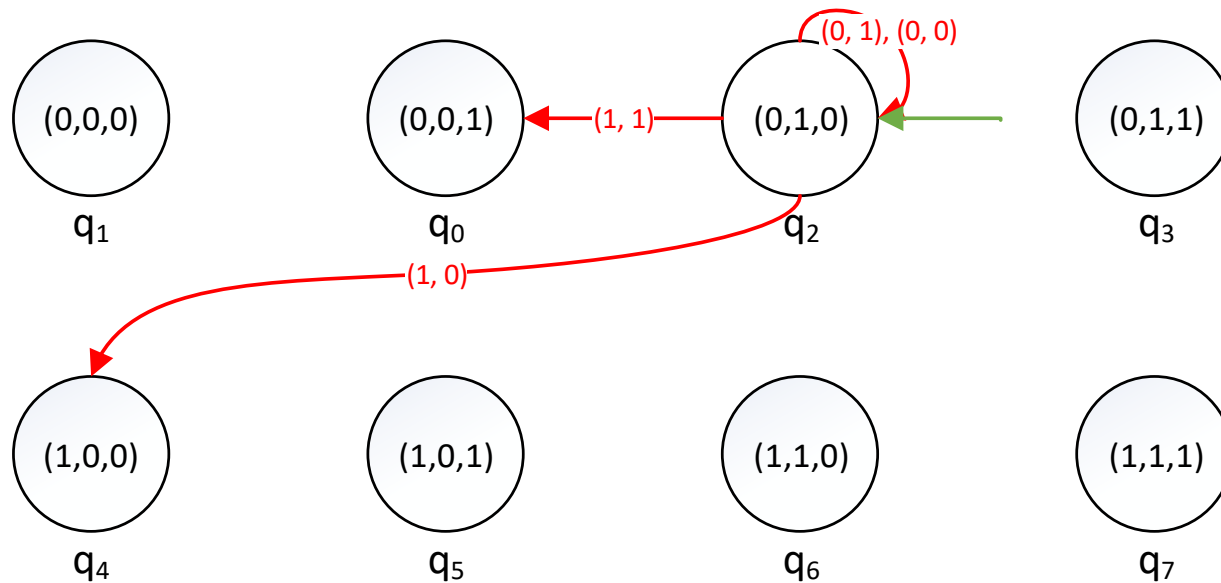


Пример

Начальное состояние – q_2

nextStates: q_0, q_2, q_4

processedStates: q_2

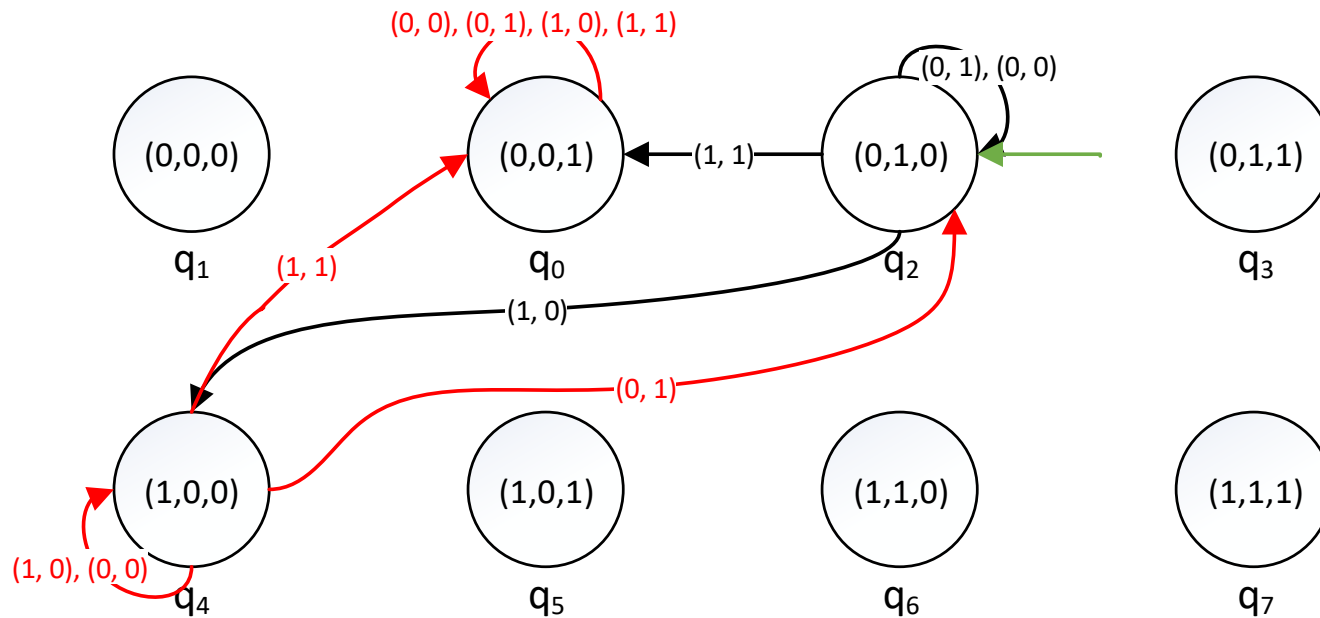


Пример

Начальное состояние – q_2

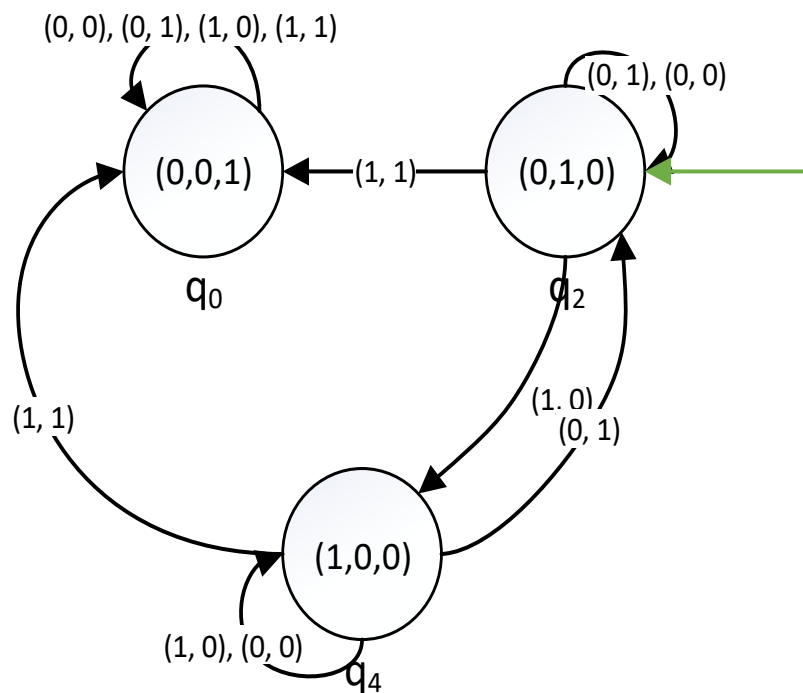
nextStates: q_0, q_2, q_4

processedStates: q_2, q_4, q_0



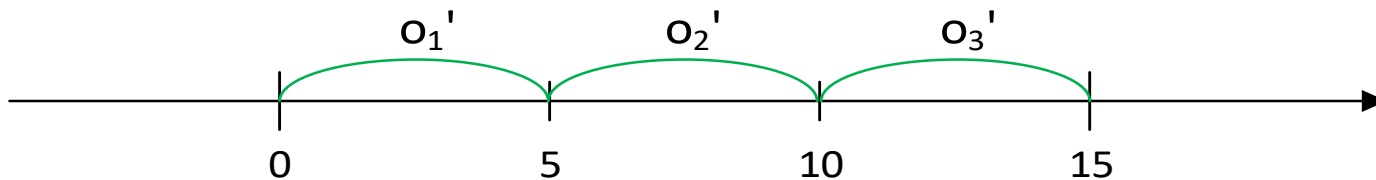
Пример

Готово!

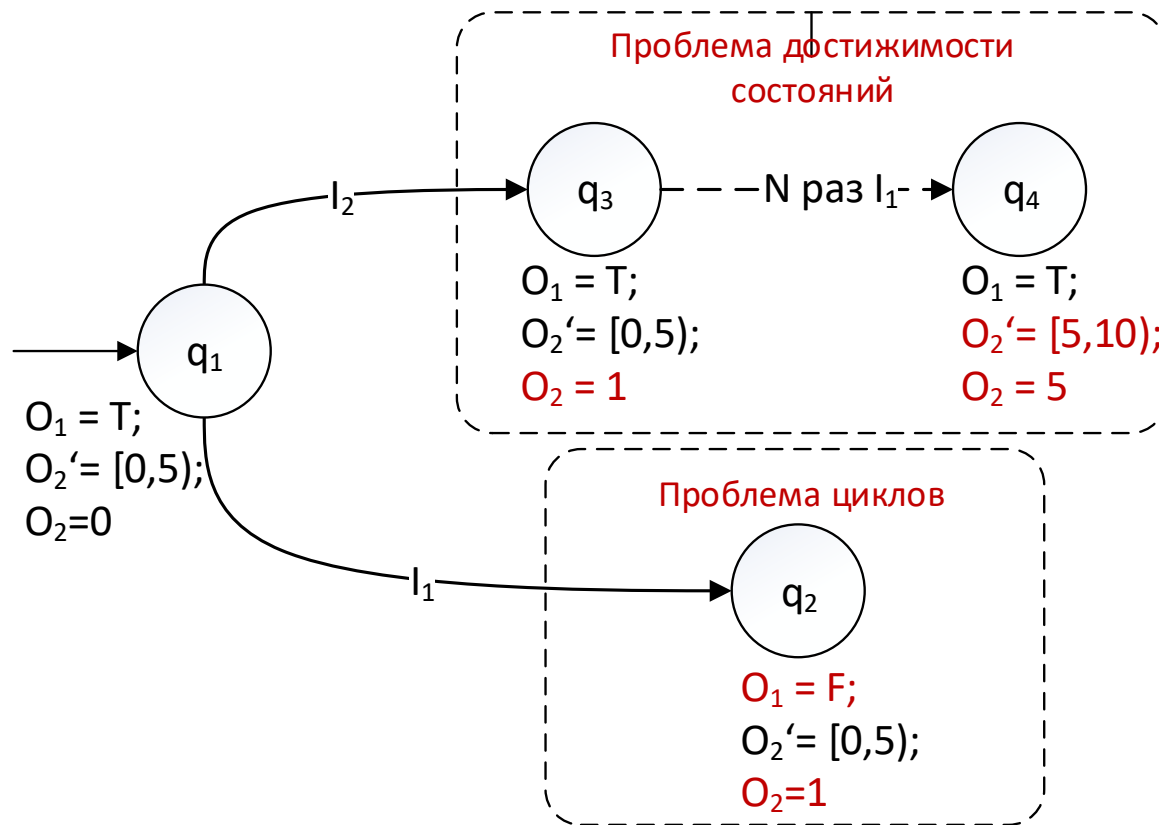


Дискретизация вещественных переменных

- Вещественная переменная $O \in [0, 15]$
- Дискретизация: $O' \in \{[0,5), [5,10), [10,15]\}$
- $O = v_1, O = v_2,$
 $v_1 = v_2, \text{ если } \exists o_i' \in O', (v_1 \in o_i') \wedge (v_2 \in o_i')$



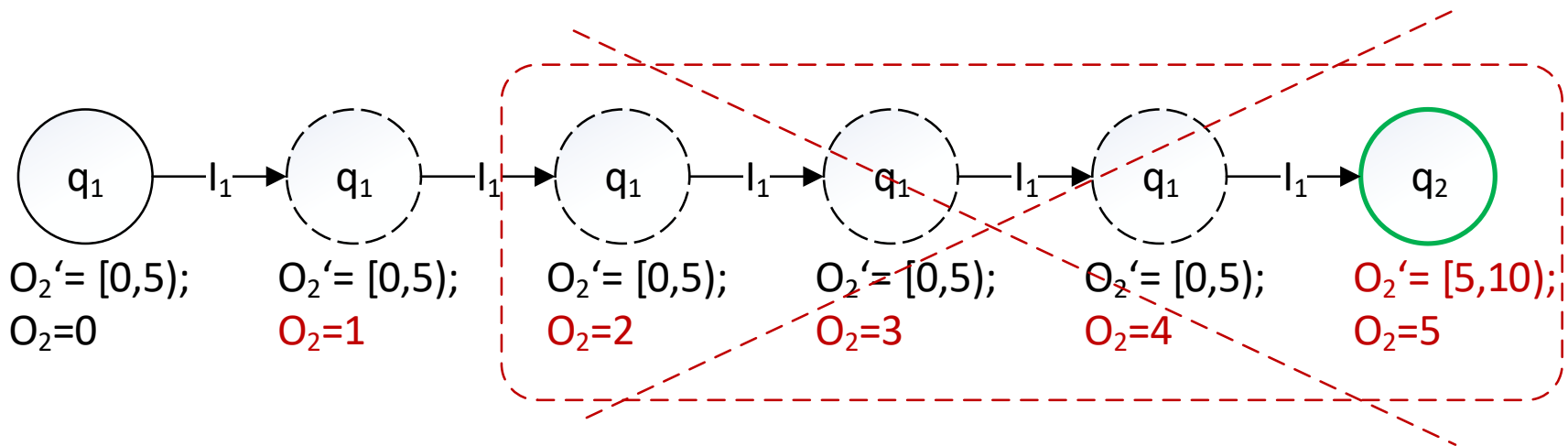
Проблемы, возникающие при обработке вещественных переменных



Проблема достижимости состояний

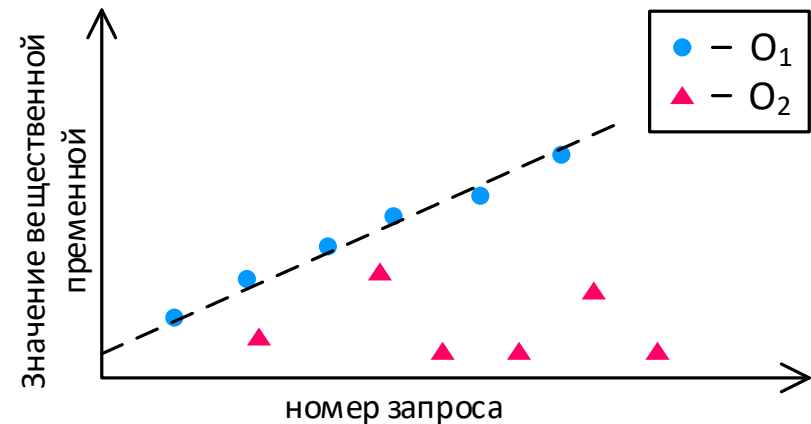
Дискретизация:

$$O_2 \in [0, 10], O_2' \in \{[0, 5), [5, 10]\}$$

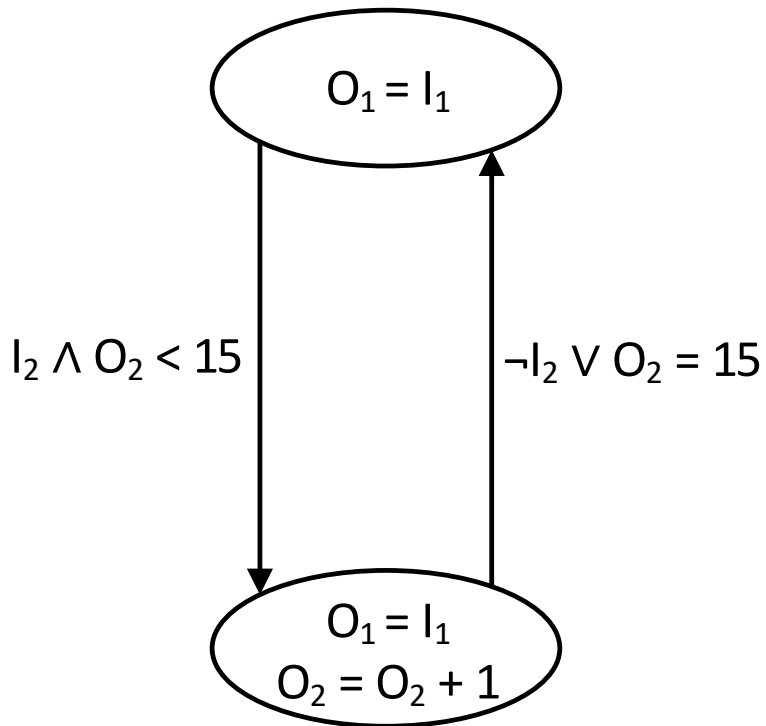


Предлагаемая стратегия обработки вещественных переменных

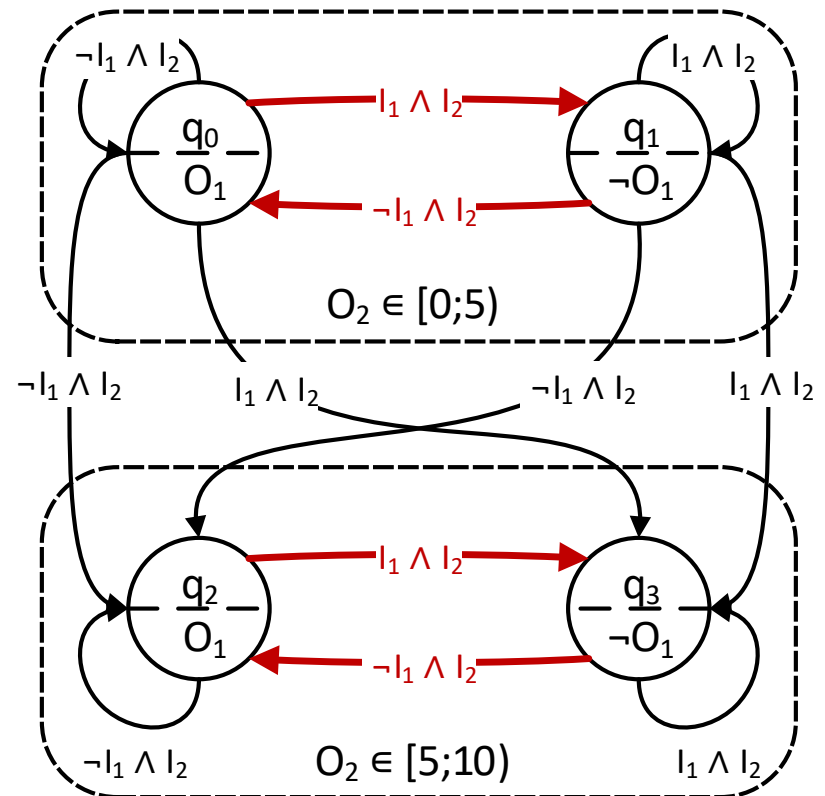
1. Повторить запрос N раз
2. Рассчитать **коэффициент корреляции Пирсона (r)** между значением вещественной переменной и номером итерации
3. Если $|r| \geq d$, значение переменной **монотонно убывает или возрастает**, продолжать отправлять запрос до перехода в следующее состояние
4. Если $|r| < d$, обнаружена **обратная петля**, прекратить повторение запроса



Проблема циклов



Моделируемая система



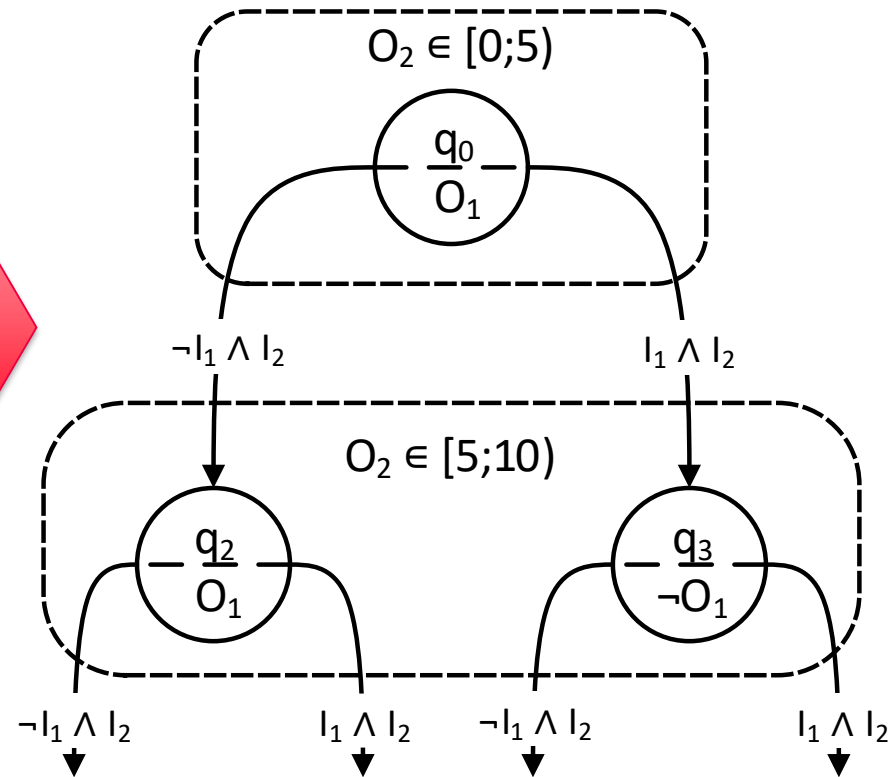
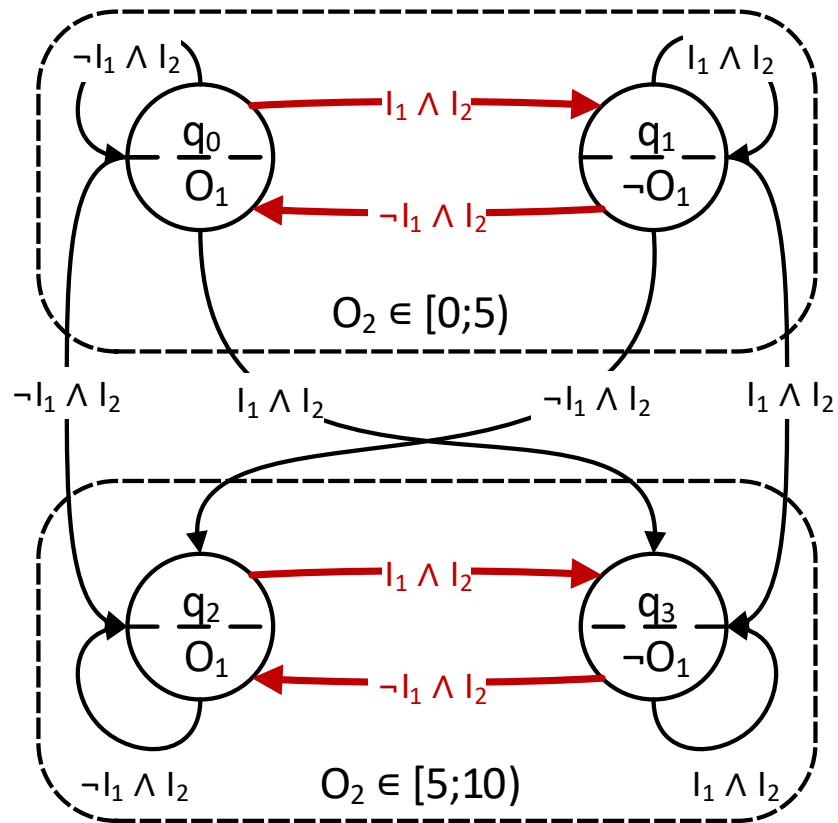
Часть полученного автомата

Решение проблемы циклов

Если при переходе от q_n к q_k по R меняется значение вещественной переменной, то

1. Проверить **наличие обратной петли** в новом состоянии по R и, если она присутствует, то продолжить отправлять запрос R до перехода в следующий интервал, иначе
2. q_k – новое состояние, которое отмечается, как то, в котором вещественные переменные **сравниваются по конкретным значениям**
3. Если состояние q_l получено **после перехода из отмеченного состояния q_k** и находится **в том же интервале**, что и q_k , оно так же отмечается

Циклы: решение



Предлагаемый метод

- **Базовый алгоритм** обнаружения новых состояний – алгоритм поиска в ширину
- Для **решения проблемы достижимости состояний** – повторение одного запроса до тех пор, пока вещественная переменная не перейдет в следующий интервал
- Для **решения проблемы циклов**
 1. Проверить наличие обратной петли в новом состоянии по тому же запросу и, если она присутствует, то решать, как проблему достижимости состояний, иначе
 2. Сравнивать вещественные переменные в этом состоянии и всех, получающихся из него по конкретному значению до достижения следующего интервала вещественной переменной

Применение: пример верификации промышленной КФС в замкнутом цикле

Выходные переменные:

- Boolean *carAtFloor0..2*, *doorClosed0..2*
- Real *carPos* $\in \{[30; 30.5), [30.5; 224.5), [224.5; 225.5), [225.5; 418.5), [418.5; 419)\}$

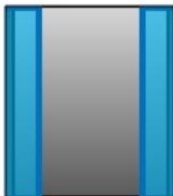


Входные переменные:

- Boolean *motorUp*, *motorDown*, *doorOpen0..2*

Сравнение с алгоритмами:

- Метод, основанный на ограничениях [2]
- Метод, основанный на явных состояниях [2]
- Адаптированный L^*

Была проведена верификация тех же LTL свойств, что и в работе [1].

	Этаж	CarPos
	2	4
		3
	1	2
		1
	0	0

[1] D. Avdyukhin et al. "Plant trace generation for formal plant model inference: Methods and case study" // INDIN 2017

[2] I. Buzhinsky et al. "Automatic inference of finite-state plant models from traces and temporal properties" // IEEE Transactions on Industrial Informatics, 2017

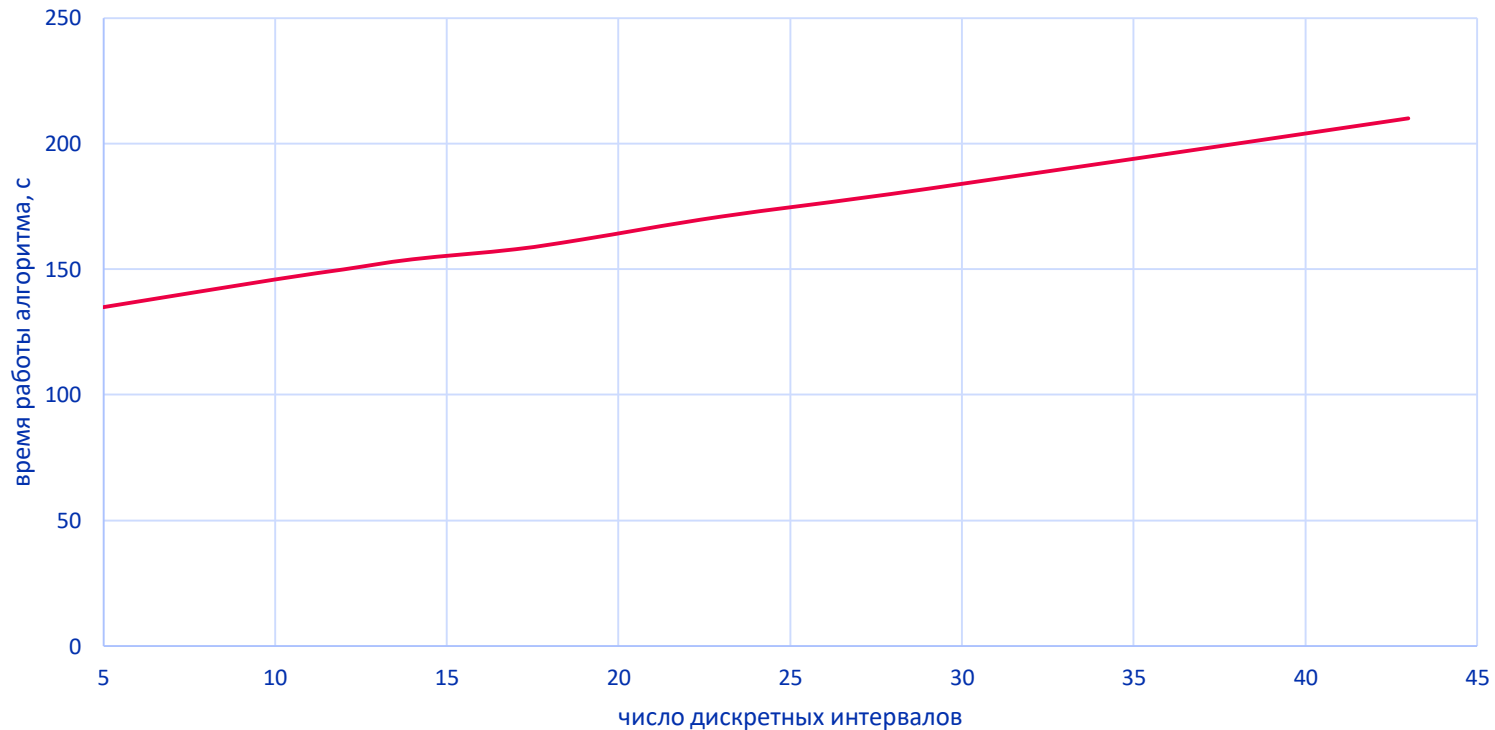
Сравнение метрик алгоритмов

Алгоритм	Время построения	Число состояний
Пассивные методы		
Метод, основанный на ограничениях	3 ч сбор трассировок + 120 с построение модели	220 ограничений
Метод явных состояний	3 ч сбор трассировок + 367 с построение модели	20
Активные методы		
Адаптированный L*	3 ч	40
Предлагаемый метод	135 с	40

Примеры LTL свойств и результаты верификации в замкнутом цикле

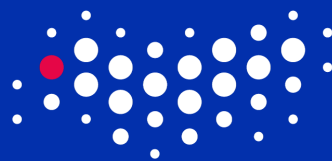
Свойство	Корректное значение	Пассивные		Активные	
		Ограничения	Явные состояния	L*	Предлагаемый
$G \ F \neg \text{motorDown}$	-	-	-	-	-
$G (G (\neg \text{motorUp}) \wedge G (\text{motorDown} \vee \text{carAtFloor0}) \rightarrow F \text{carAtFloor0})$	+	-	-	+	+
$G (\text{carPos} = 0 \wedge \neg \text{motorDown} \wedge \text{motorUp} \rightarrow X (\text{carPos} = 1))$	+	-	+	+	+
$G (\text{buttonPressed1} \wedge (\text{не застрял на этаже})) \rightarrow F \text{carAtFloor1}$	+	-	-	+	+
$G (\text{carPos} \in \{1,3\} \rightarrow \text{door0Closed} \wedge \text{door1Closed} \wedge \text{door2Closed})$	+	-	+	+	+

Зависимость времени работы алгоритма от числа дискретных интервалов непрерывной переменной



Заключение

- Разработан метод формальной верификации промышленных КФС в замкнутом цикле, основанный на алгоритме автоматической генерации моделей
- Разработанный метод был протестирован на симуляционной модели лифта
- Принята статья «Active learning of formal plant models for cyber-physical systems» на INDIN 2018
- Статья «Closed-loop verification of a compensating group drive model using synthesized formal plant model» и выступление на ETFA 2017
- Доклад «Разработка метода автоматической генерации формальных моделей кибер-физических систем на основе активного обучения» на КМУ 2018
- Часть работы выполнялась в Lulea University of Technology в рамках программы Erasmus+
- Исследования выполнены в рамках проекта ФЦП «Разработка методов, средств и технологий проектирования, верификации и тестирования ответственных кибер-физических систем», руководитель Шалыто А.А.



УНИВЕРСИТЕТ ИТМО

Спасибо за внимание!

Альтернативное решение проблемы циклов

