Верификация параллельных программных и аппаратных систем

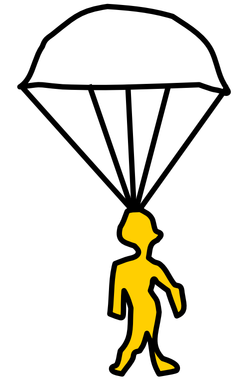
Курс лекций

Карпов Юрий Глебович
профессор, д.т.н., зав.кафедрой
“Распределенные вычисления и компьютерные сети”
Санкт-Петербургского политехнического университета
karpov@dcn.infos.ru

План курса

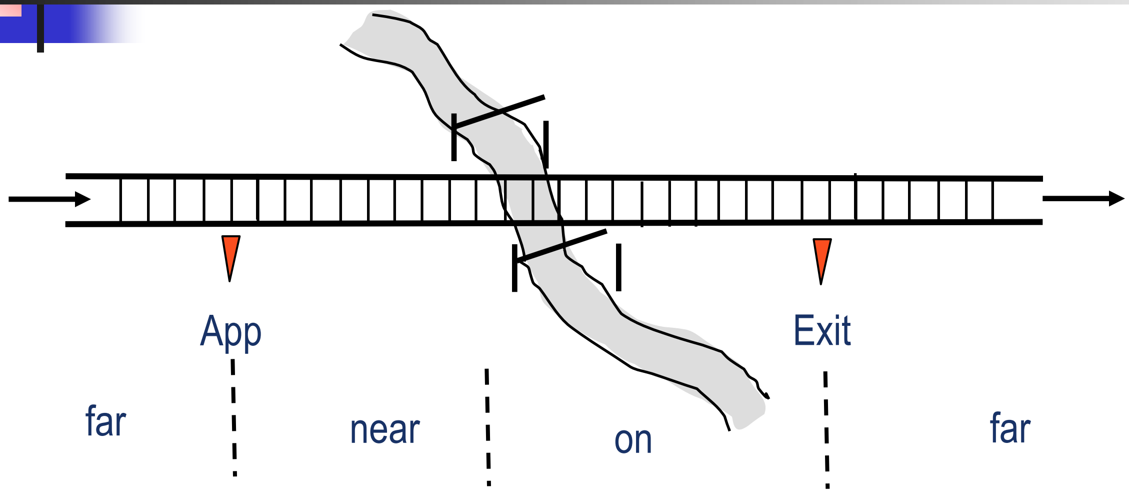
1. Введение
2. Метод Флойда-Хоара доказательства корректности программ
3. Исчисление взаимодействующих систем (CCS) Р.Милнера
4. Темпоральные логики
5. Алгоритм model checking для проверки формул CTL
6. Автоматный подход к проверке выполнения формул LTL
7. Структура Крипке как модель реагирующих систем
8. Темпоральные свойства систем
9. Система верификации Spin и язык Promela. Примеры верификации
10. Применения метода верификации model checking
11. BDD и их применение
12. Символьная проверка моделей
13. Количественный анализ дискретных систем
14. Верификация систем реального времени (I)
15. Верификация систем реального времени (II)
16. Консультации по курсовой работе

Общие положения



- Model Checking и структуры Крипке позволяют анализировать свойства систем без явного указания времени (Например, EFp – существует траектория поведения системы, на которой когда-нибудь в будущем выполнится свойство p. Для систем реального времени это утверждение бесполезно!)
- Системы реального времени – такие, в которых явные значения временных интервалов между событиями существенны, они влияют на свойства системы
  - Safety-critical systems – пропуск временной границы может привести к аварии
- Временные Автоматы и Временная Темпоральная логика – это расширения базовых моделей введением реального времени
- Впервые – в 1994 г. Alur, Dill. Сейчас проблема широко исследуется, существуют пакеты верификации систем реального времени, например, UPPAAL и KRONOS
- Поскольку эта область находится в стадии исследования, существует множество различных определений одних и тех же понятий, разных подходов к решению одних и тех же проблем

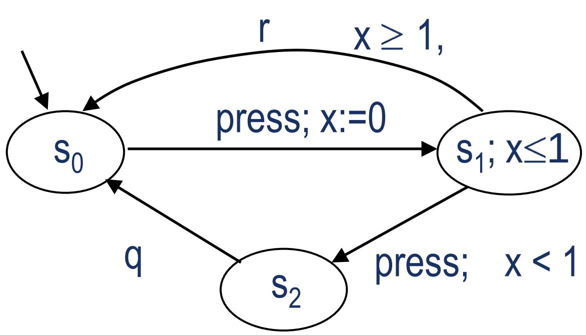
Пример: ж.д. переезд



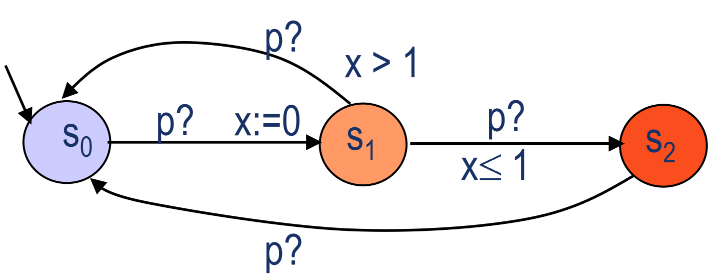
- Описание системы управления:
  - Ж.д. переезд оснащен шлагбаумом. Поезд извещает контроллер шлагбаума о своем приближении >= 2 мин до пересечения переезда, и уйдет не более, чем через 5 мин после пересечения.
  - Черз 1 мин после получения извещения контроллер начинает закрывать шлагбаум. На закрывание нужна 1 минута.
  - Не позже, чем через 1 минуту после того, как поезд прошел, контроллер начинает поднимать шлагбаум, на что требуется от 1 до 2 минут.
- Доказать:
  - Шлагбаум закрыт всегда, когда поезд проходит переезд
  - Шлагбаум закрыт всегда не более, чем на 10 минут

Временные автоматы (Timed Automata)

- Необходим формализм, позволяющий конечным способом задать поведение, зависящее от реального времени



Двойной (q) и одинарный (r) клик мышкой (press)



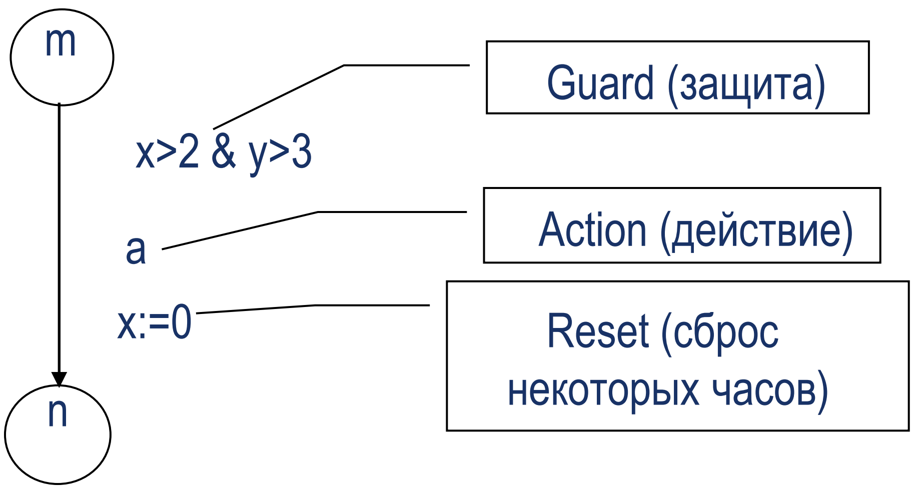
Двойное нажатие – яркий, однократное – бледный свет лампы, Следующее нажатие – погасить лампу

Временные автоматы - определение

TA (Timed Automaton) – это:

конечный автомат (помеченная система переходов)

+ конечное число синхронных real-time часов (таймеров) (здесь x – таймер)
+ (возможно) сброс некоторых часов на переходах
+ условия на переходах, зависящие от значений часов



m и n – не состояния, а локации

Состояния – это локация + значения всех параметров.
Во временном автомате часы также являются параметрами

Здесь состояния:
(m, x=0.1, y= 2.24)
(m, x=2.2, y= 4.44)
(n, x=0, y= 4.34939)
(n, x=0.21, y= 4.55939)

и т.д.

Формальное определение временного автомата

Обозначим $\Phi(X)$ – множество clock constraints. Это выражения вида:

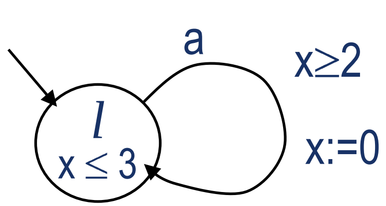
$\alpha ::= x<k \mid x>k \mid \neg\alpha \mid \alpha\wedge\alpha$, k – рациональное число (возможна любая точность)

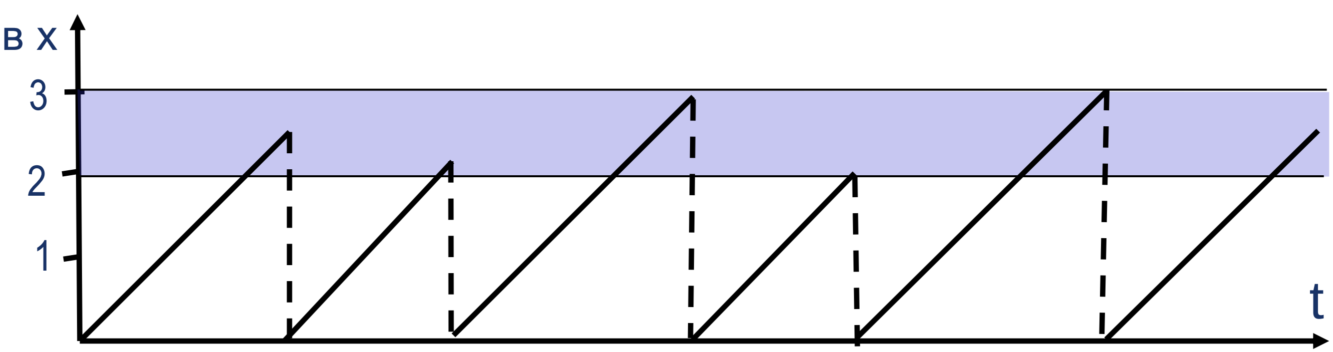
Конечное множество рациональных чисел можно заменить целыми, введя множитель

TA – это шестерка (L, l_0, Σ, X, inv, E), где:

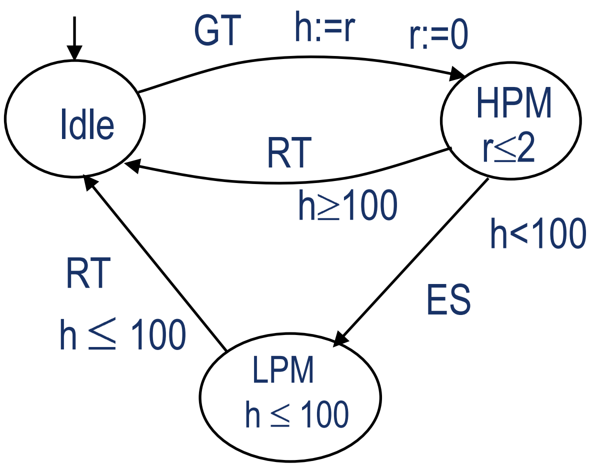
- L- конечное множество локаций, включающее начальную локацию l_0,
- Σ - множество пометок;
- X – конечное множество часов (clocks);
- inv: $L\rightarrow\Phi(X)$ – отображение локаций на clock constraints;
- $E \subseteq L\times \Sigma \times 2^X \times \Phi(X) \times L$ – переходы $\Phi(x)$ – защита перехода

Фактически, инварианты и ограничения определяют интервал





Пример: FDDI протокол (передатчик)



GT – get token

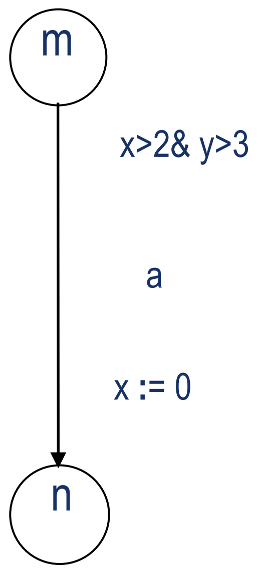
HPM – high priority messages

LPM – low priority messages

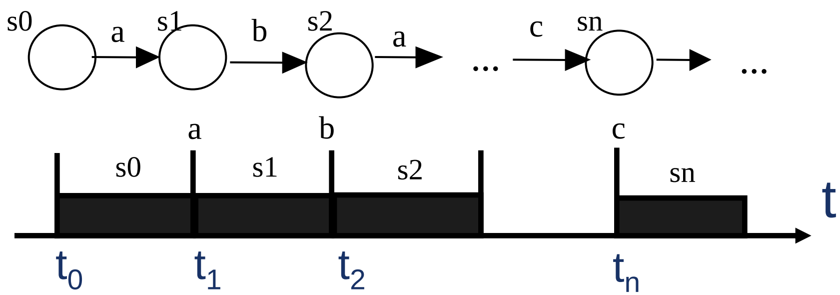
Описание взято из руководства по протоколу FDDI

- В локации Idle станция ожидает токен. GT – это действие get token – прибытие токена
- Таймер r считает время, прошедшее после последнего прибытия токена
- По приходе токена, h присваивается значение r и таймер r сбрасывается
- В локации HPM станция посылает высокоприоритетные сообщения. Это может длиться не более 2 е.в. Станция завершает передачу высокоприоритетных сообщений либо потому, что они кончились, либо потому, что истекли 2 е.в.
- Если прошло более 100 е.в. с момента принятия токена, станция возвращается в режим Idle. В противном случае она переходит в режим пересылки низкоприоритетных сообщений.
- В режиме LPM станция посылает низкоприоритетные сообщения до тех пор, пока они не кончатся, но не более, чем 100 е.в., после чего возвращается в режим Idle.

Как описать переходы временного автомата?

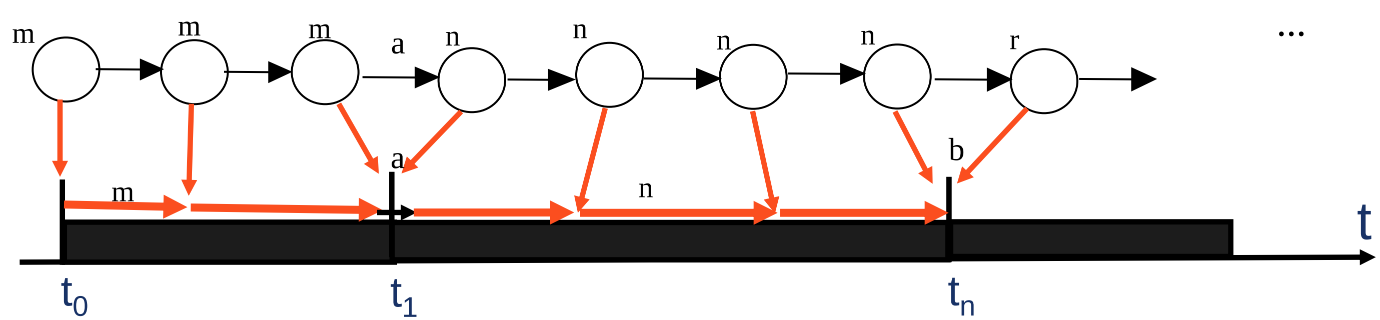


Поведение не временного автомата:



Переходы мгновенны

Поведение временного автомата:



(m, x=2.4, y=3.25) $\xrightarrow{a}$ (n, x=0, y=3.25)

Action transition - мгновенный переход

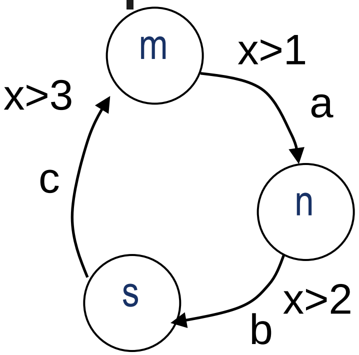
(m, x=1.0, y=1.85) $\xrightarrow{1.4}$ (m,x=2.4, y=3.25)

Timed transition – мгновенный переход

(m, x=2.4, y=3.25) $\xrightarrow{1.2}$ (m,x=3.6, y=4.45)

Недетерминизм выбора момента переключения локаций

Проблемы с переходами временного автомата

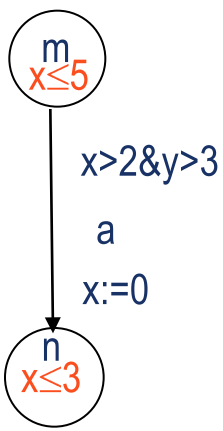


1 проблема: парадокс Зенона – бесконечное число действий в конечный промежуток времени

$(m, x=3.4) \xrightarrow{0.1} (m, x=3.5) \xrightarrow{a} (n, x=3.5) \xrightarrow{0.01} (n, x=3.51) \xrightarrow{b} (s, x=3.51)$

$\xrightarrow{0.001} (s, x=3.511) \xrightarrow{c} (m, x=3.511) \xrightarrow{0.0001} (m, x=3.5111) \xrightarrow{a} \ldots$

Решение: Non-Zeno автоматы -- только конечное число действий м.б. выполнено автоматом за конечный промежуток времени



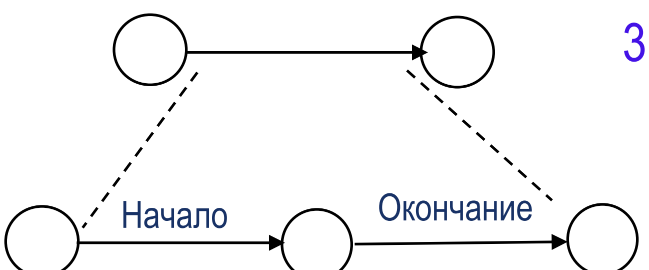
2 проблема: бесконечное нахождение в одной локации

$(m, x=2.4, y=3.25) \xrightarrow{1.2} (m, x=3.6, y=4.45) \xrightarrow{2} (m, x=5.6, y=4.65)$

$\xrightarrow{3} (m, x=8.6, y=7.65) \xrightarrow{10} (m, x=18.6, y=17.65) \xrightarrow{100} \ldots$

Решение: Timed Safety Automaton = TA + Инварианты

Инварианты гарантируют прогресс (если нужно!)

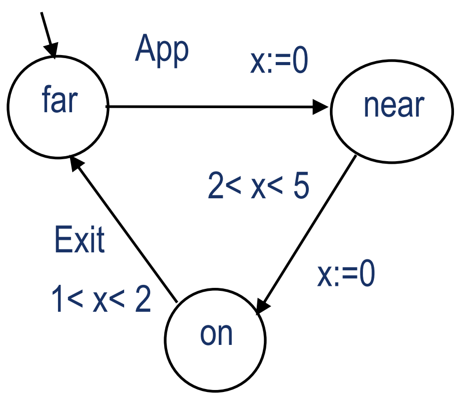


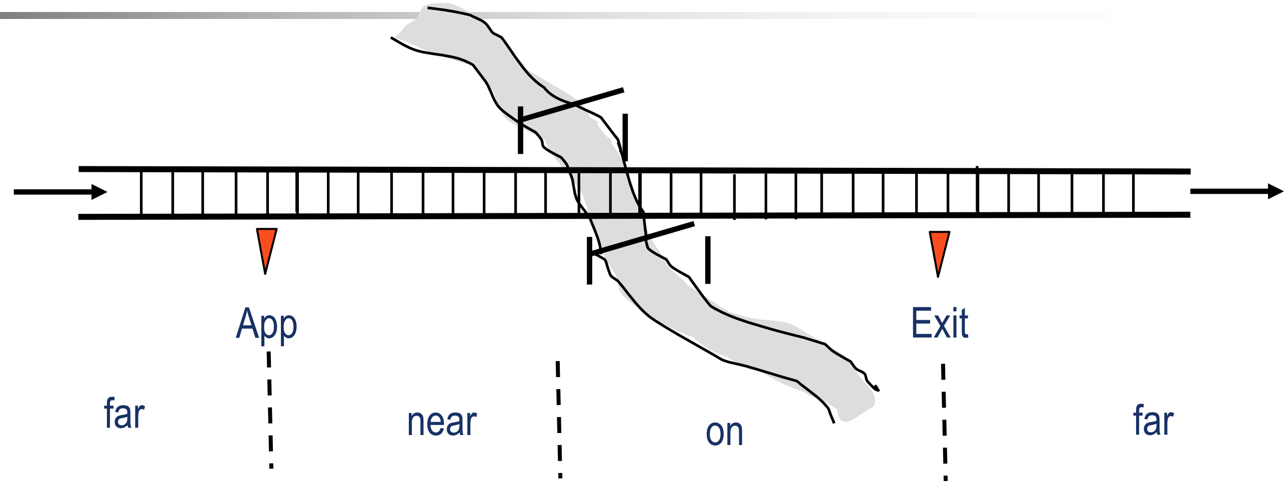
3 проблема:

Что делать, если само действие не мгновенно? Решение. Вводить мгновенные события начала и окончания действия и промежуточную локацию его выполнения

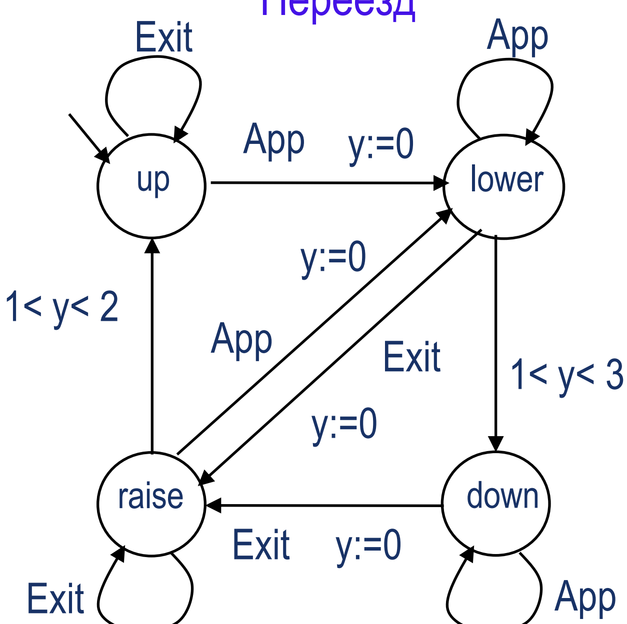
Сети временных автоматов

Поезда

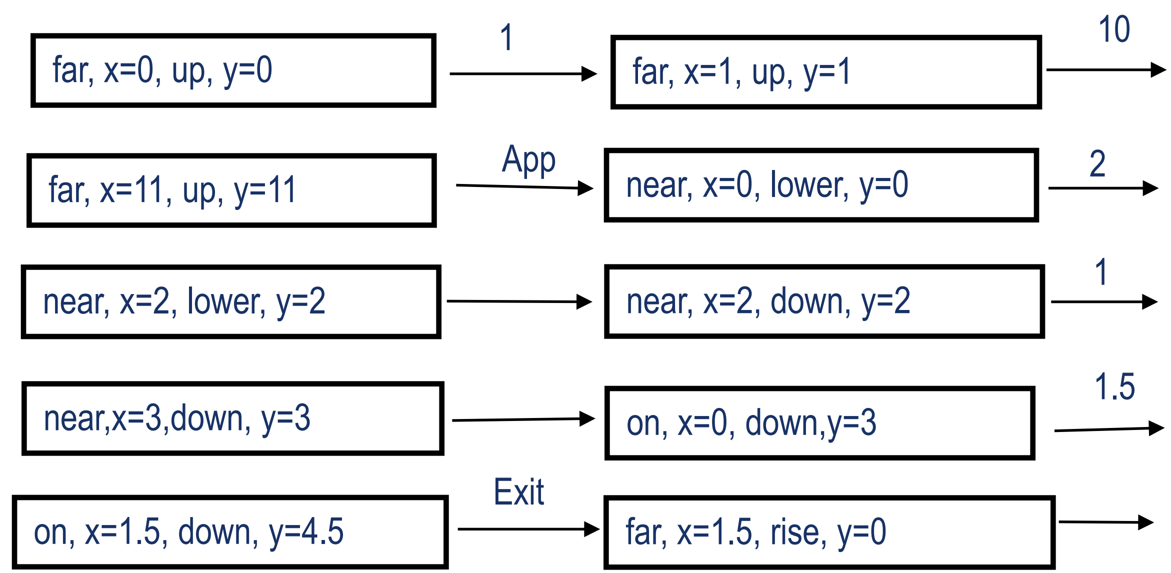




Переезд



Возможные переходы глобальных состояний



Параллельная композиция ТА

L- множество локаций

Σ- множество пометок

X – множество часов

E – множество переходов

$A_1 = (L_1, l_{01}, \Sigma_1, X_1, inv_1, E_1)$,

$A_2 = (L_2, l_{02}, \Sigma_2, X_2, inv_2, E_2)$ Пусть $X_1 \cap X_2 = \varnothing$ - часы свои

$A_1 \| A_2 = (L_1 \times L_2, (l_{01}, l_{02}), \Sigma_1 \cup \Sigma_2, X_1 \cup X_2, Inv, E)$,

где: $Inv(s_1,s_2) = inv_1(s_1) \wedge inv_2(s_2)$, а множество переходов $E \subseteq (L_1 \times L_2) \times \Sigma_1 \cup \Sigma_2 \times 2^X \times \Phi(X) \times L$ определяется следующими правилами

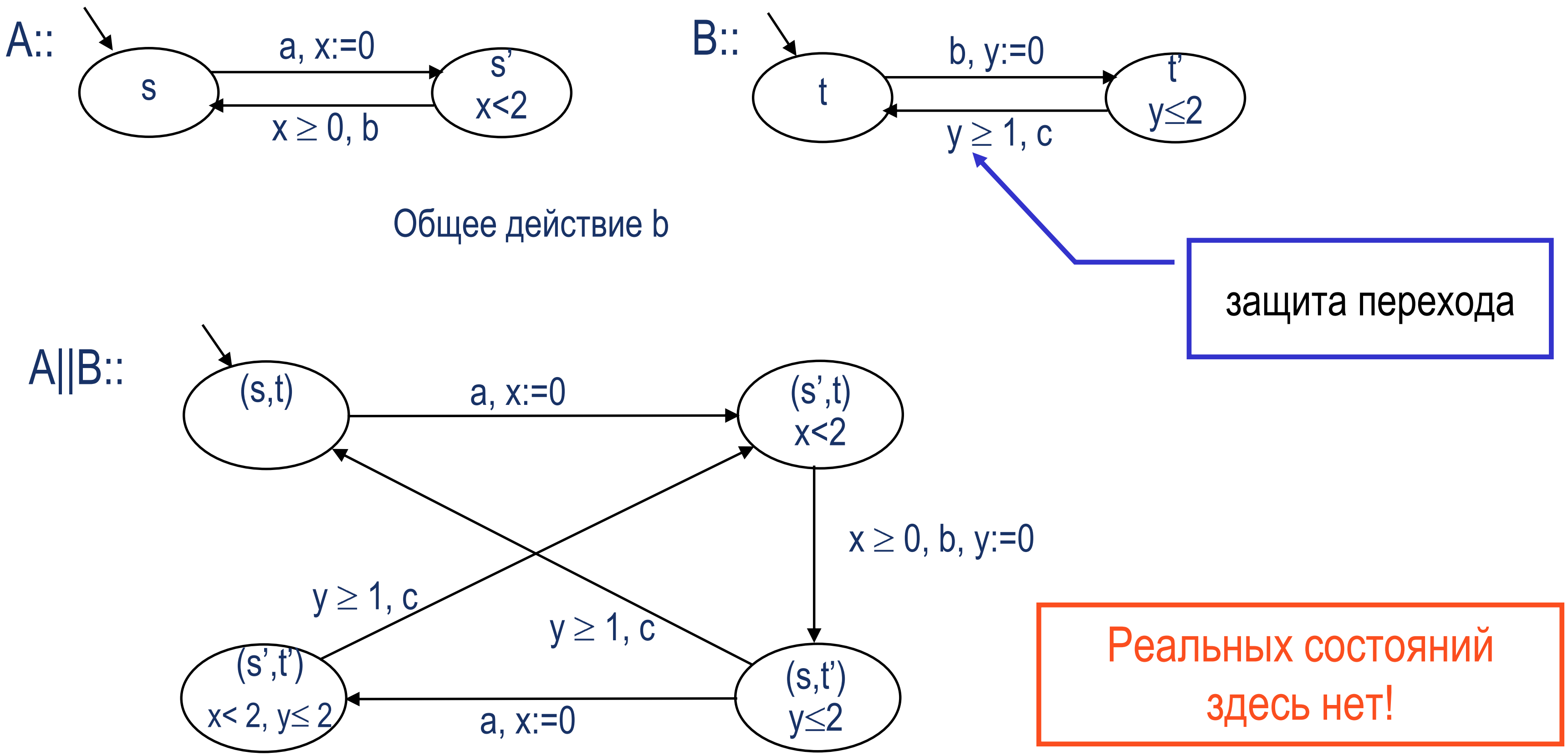
1. Если $a \in \Sigma_1 \cap \Sigma_2$, если $(l_1, a, \lambda_1, \varphi_1, l_1') \in E_1$ и $(l_2, a, \lambda_2, \varphi_2, l_2') \in E_2$ то E будет включать переход $((l_1, l_2), a, \lambda_1 \cup \lambda_2, \varphi_1 \wedge \varphi_2, (l_1', l_2'))$ (a – общее действие)

2. Если $a \in \Sigma_1 - \Sigma_2$, и $(l_1, a, \lambda, \varphi, l_1') \in E_1$, то для каждого $l_2 \in L_2$ E будет включать переход $((l_1, l_2), a, \lambda, \varphi, (l_1', l_2))$ (действия A1)

3. Если $a \in \Sigma_2 - \Sigma_1$, и $(l_2, a, \lambda, \varphi, l_2') \in E_2$, то для каждого l_1 E будет включать переход $((l_1, l_2), a, \lambda, \varphi, (l_1, l_2'))$ (действия A2)

Общие (взаимодополнительные) действия оба автомата выполняют одновременно (взаимодействие рандеву), и каждый из автоматов выполняет свои независимые действия независимо (интерливинг)

Пример параллельной композиции ТА



Общее (взаимодополнительное) действие b оба автомата выполняют одновременно (взаимодействие рандеву), и каждый из автоматов выполняет свои независимые действия a и c независимо (интерливинг, чередование)

Семантика временного автомата

Введем понятие “интерпретации часов v”: $X \rightarrow R^+$ - присваивание каждым часам неотрицательного значения; v+d означает увеличение всех часов x с v(x) до v(x)+d

Семантической моделью временного автомата A является бесконечный state transition graph $S(A)=(\Sigma, Q, Q_0, R)$, где:

Σ- множество действий,

Q – множество состояний, каждое состояние – пара (l,v), где l – локация, а v: $X \rightarrow R^+$ “интерпретации” часов ; Таких состояний бесконечное количество (континуум!)

R – множество переходов двух видов:

Задержка: $(l,v) \longrightarrow (l,v+d)$, $d \in R_{\geq 0}$; при условии выполнения инвариантов;

Действие: $(l,v) \longrightarrow (l',v')$, v’: показания сбрасываемых часов 0, остальных - то же

Верификация временных автоматов

Существует несколько задач и методов верификации временных автоматов:

- проверка выполнимости обычных формул темпоральной логики (в том числе, включающие условия на внутренние таймеры)

- использование контрольных автоматов (watchdogs);

- анализ достижимости;

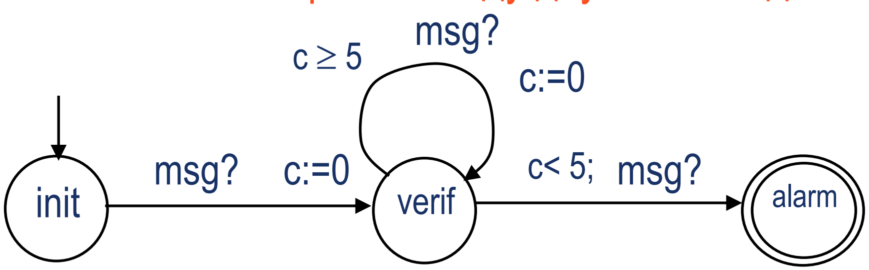
- достижимость за ограниченное время (темпоральное свойство $AF_{<5}\ p$)

- проверкой «*бисимуляционной эквивалентности*» заданного автомата и автомата, выражающего требуемые свойства

- применение темпоральной логики, расширенной ограничениями реального времени

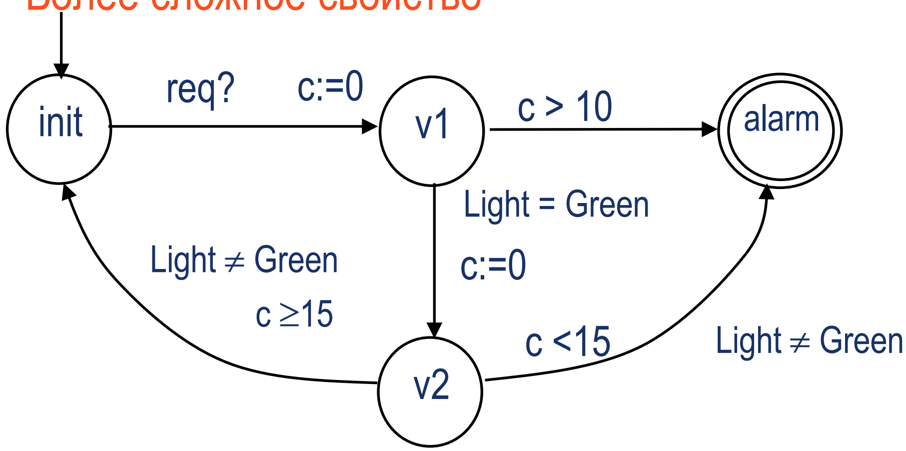
Примеры контрольных автоматов

Свойство: интервал между двумя последовательными сообщениями ≥ 5



Alarm, если интервал между последовательными сообщениями < 5

Более сложное свойство

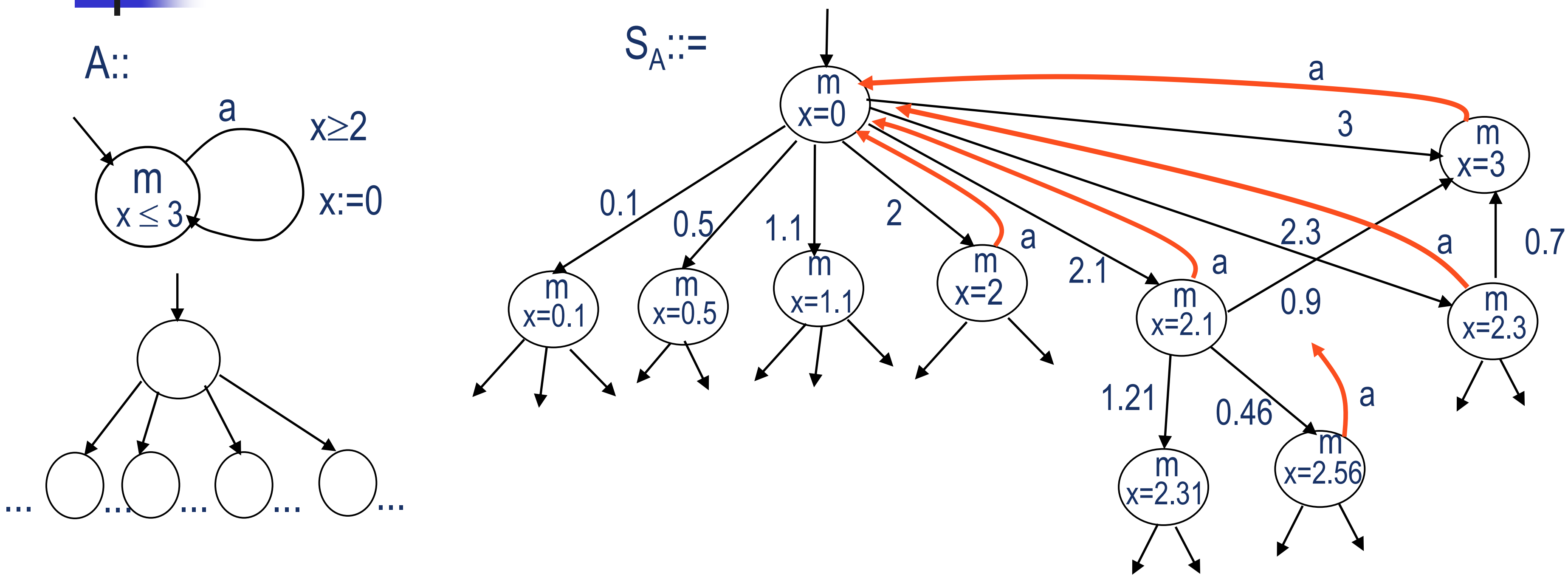


Если запрос req поступил в систему, то светофор загорится зеленым не позже, чем через 10 единиц времени, и будет гореть зеленым по меньшей мере 15 единиц времени

Получив параллельную композицию исследуемого и контрольного временных автоматов, можем решать проблему достижимости – достижимо ли глобальное состояние, в котором контрольная компонента находится в ошибочном состоянии?

Но как решать проблему достижимости для ТА?
И как проверять любые темпоральные свойства?

Временной граф переходов – семантическая модель



Семантическая модель временного автомата – временной граф переходов – имеет континуум состояний и переходов

- Важное свойство параллельной композиции ТА: системы переходов, представляющие семантику временных автоматов, $S_{A1}||S_{A2}$ и $S_{A1||A2}$, изоморфны
- Но как работать с такими автоматами??? У них БЕСКОНЕЧНОЕ число состояний

Конечное представление вычислений ТА

ТА может быть проверен на его семантической модели – его графе переходов

Но! Семантическое представление вычислений ТА содержит бесконечное число состояний. S_A-система переходов, представляющая вычисления А, бесконечна!!

Любая проверка свойств временного автомата, в том числе и достижимости состояний, требует конечного представления его вычислений!

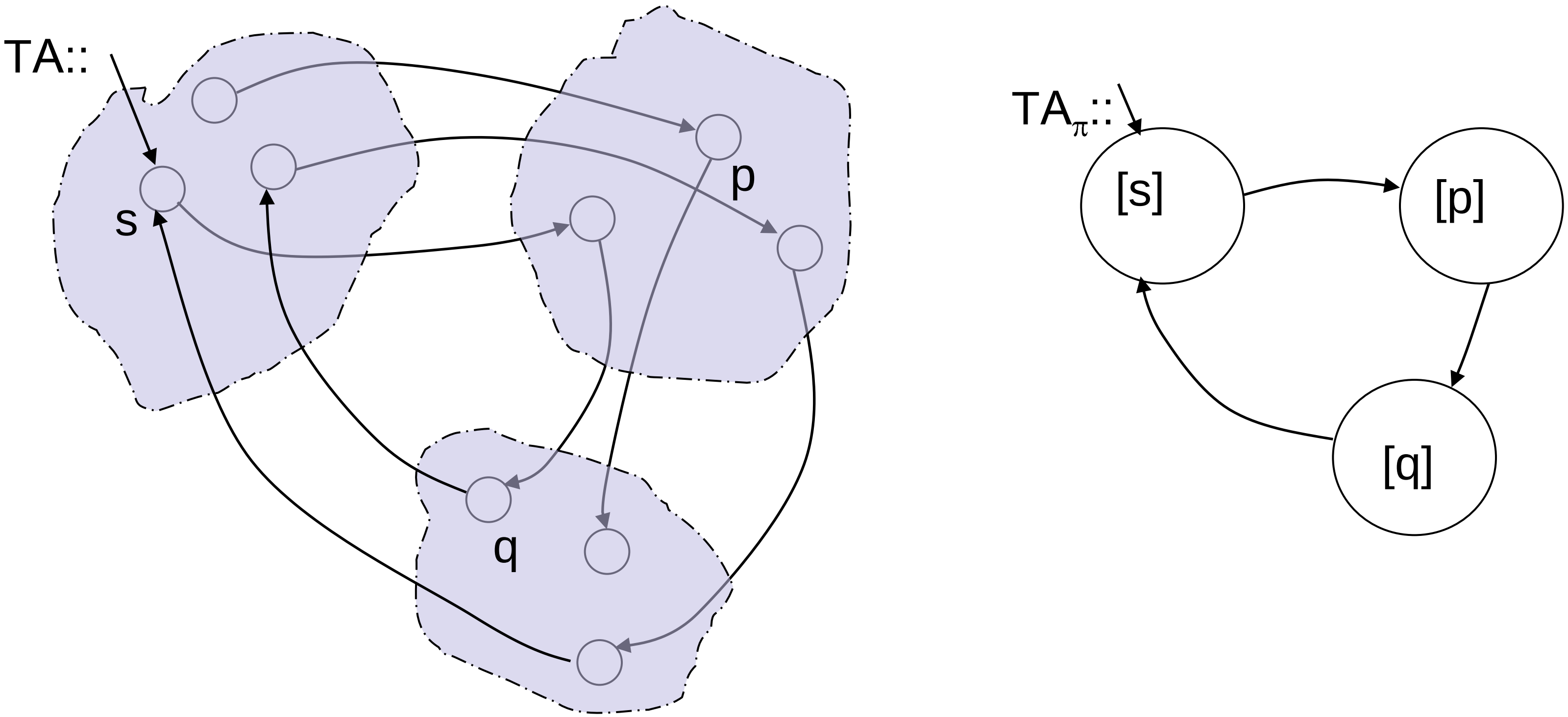
Бесконечность представления поведений имеет два источника: неограниченность показаний часов и непрерывность времени. Метод получения конечного представления – использование эквивалентных областей значений времени

Пусть (q, v) и (q,v ') – два состояния ТА с одними и теми же значениями локаций.

Например, (q, x=1.556, y=0.324) и (q, x=1.557, y=0.325). Можно ли сказать, что они эквивалентны относительно некоторой формулы TL, например $AG(a \Rightarrow AF_{<7}\, b)$?

Да, потому что мы остались между одной и той же парой целых, а все ограничения на часы – только относительно целых! **(Следствие того, что значения часов – рациональные числа)**

Фактор-автомат по эквивалентности π (TA_π)



Когда состояния (q, v), (q,v ') будут эквивалентны относительно всех свойств TA?

Когда для любой формулы TCTL формулы ϕ выполняется $\phi(q, v) \equiv \phi(q,v\,')$?

Эквивалентность на множестве состояний ТА

Вводим отношение эквивалентности $\cong$ - эквивалентность наборов значений таймеров.

Для двух интерпретаций часов v и v', $v \cong v'$ iff : ($\lfloor A \rfloor$ - целая часть A, fr(A) – дробная часть A)

1 $(\forall x \in X) \lfloor v(x) \rfloor = \lfloor v'(x) \rfloor \vee v(x) > c_x \,\&\, v'(x) > c_x$ (c_x – макс. константа в неравенствах)

(все значения часов, отличающиеся только дробной частью и любые $>c_x$ -- регион)

2 $(\forall x, y \in X)$: $v(x) \leq c_x \,\&\, v(y) \leq c_y \Rightarrow fr(v(x)) \leq fr(v(y)) \equiv fr(v'(x)) \leq fr(v'(y))$

(отношение между дробными частями разных часов не меняются, это – регион)

3 $(\forall x \in X)$ $v(x) \leq c_x \Rightarrow fr(v(x)) = 0 \equiv fr(v'(x)) = 0$

(целые значения составляют отдельные регионы)

Временной регион временного автомата A – это класс эквивалентности интерпретаций (наборов значений) часов, индуцированный отношением эквивалентности $\cong$

Обозначим $[v]_{\cong}$ класс эквивалентности, которому принадлежит интерпретация v

Например, пусть v(x)=0.3; Тогда $[v]_{\cong} = 0<x<1$

Для двух часов. Пусть v(x)=0.3, v(y)=0.7; Тогда $[v]_{\cong} = 0<x<y<1$

Свойства временных регионов

Чем удобны временные регионы?

Если все временные соотношения $c \sim k$ устанавливаются для конечного числа таймеров и рациональных k, то существует конечное множество регионов – классов эквивалентности интерпретаций

Пусть v_1 и v_2 – два набора значений таймеров (интерпретаций часов)

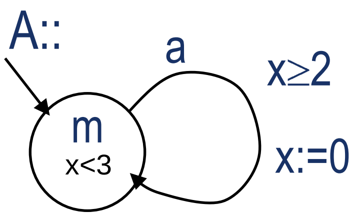
Если $[v_1]_{\cong} = [v_2]_{\cong}$ (т.е. эти наборы – в одном и том же регионе), то любое ограничение $f \in \Phi(X)$, $f(v_1) \equiv f(v_2)$ одновременно выполняется или не выполняется для этих двух наборов значений таймеров

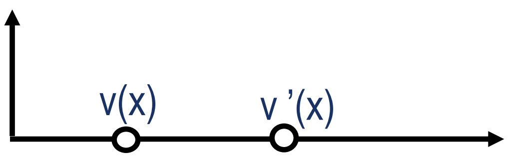
Теорема. Пусть ϕ - произвольная формула TCTL, а v_1 и v_2 - две интерпретации часов и s - произвольная локация временного автомата.

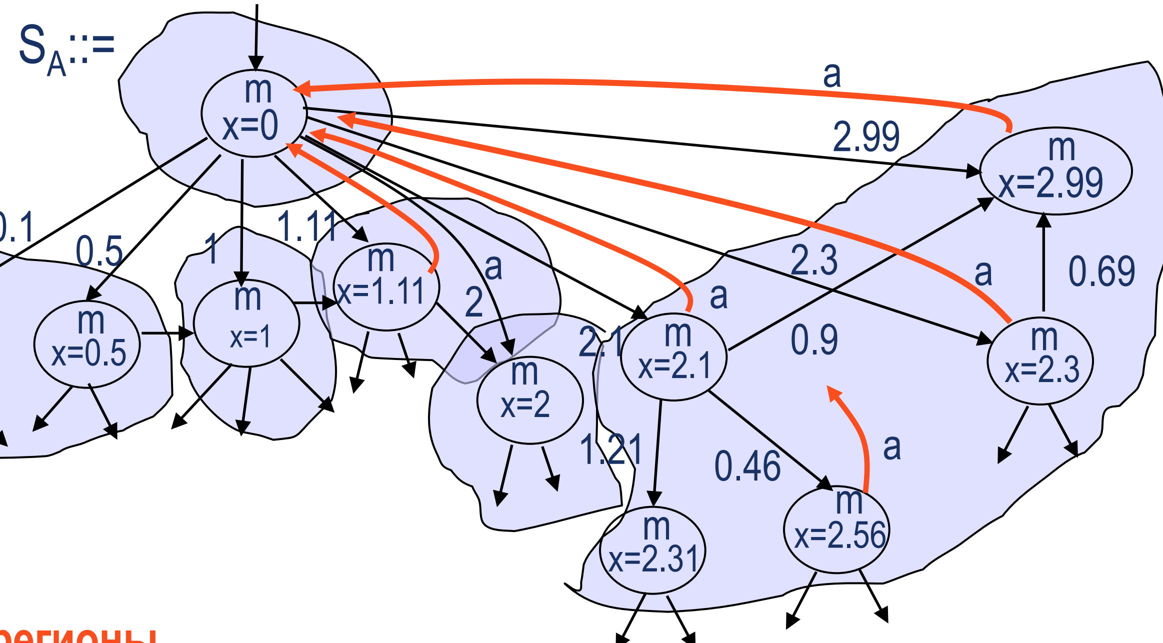
Тогда $v_1 \cong v_2 \Rightarrow (s, v_1) \text{ sat } \phi \equiv (s, v_2) \text{ sat } \phi$

Таким образом, нам нужно фиксировать состояния временного автомата с точностью до временных регионов

Один таймер: конечное представление ТА



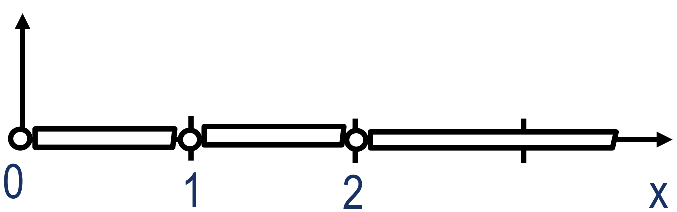


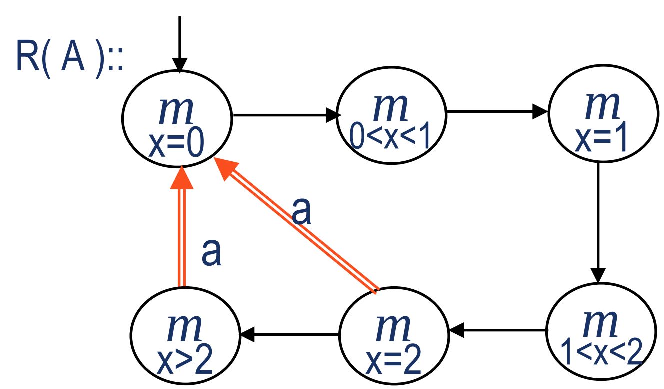


Когда v(x) ≅v'(x) ?

Введем отрезки временной оси – регионы

Регионы - это множества ***эквивалентных*** значений времени





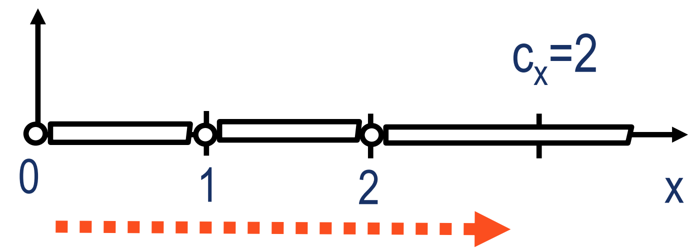
Любые две точки одного и того же временного региона эквивалентны относительно любого свойства, выраженного с помощью clock constraints

Граф регионов – один таймер

Фактор-автомат $A_{\cong}$ по эквивалентности $\cong$ называется графом регионов R(A)

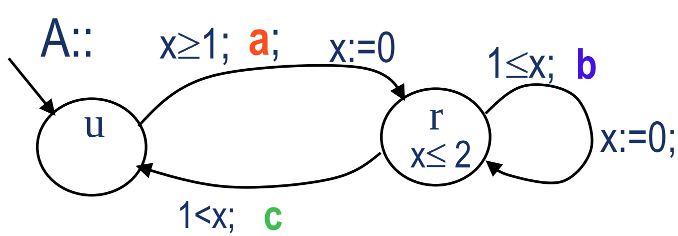
Пусть у нас один таймер x. Тогда $v(x) \cong v'(x)$ iff:

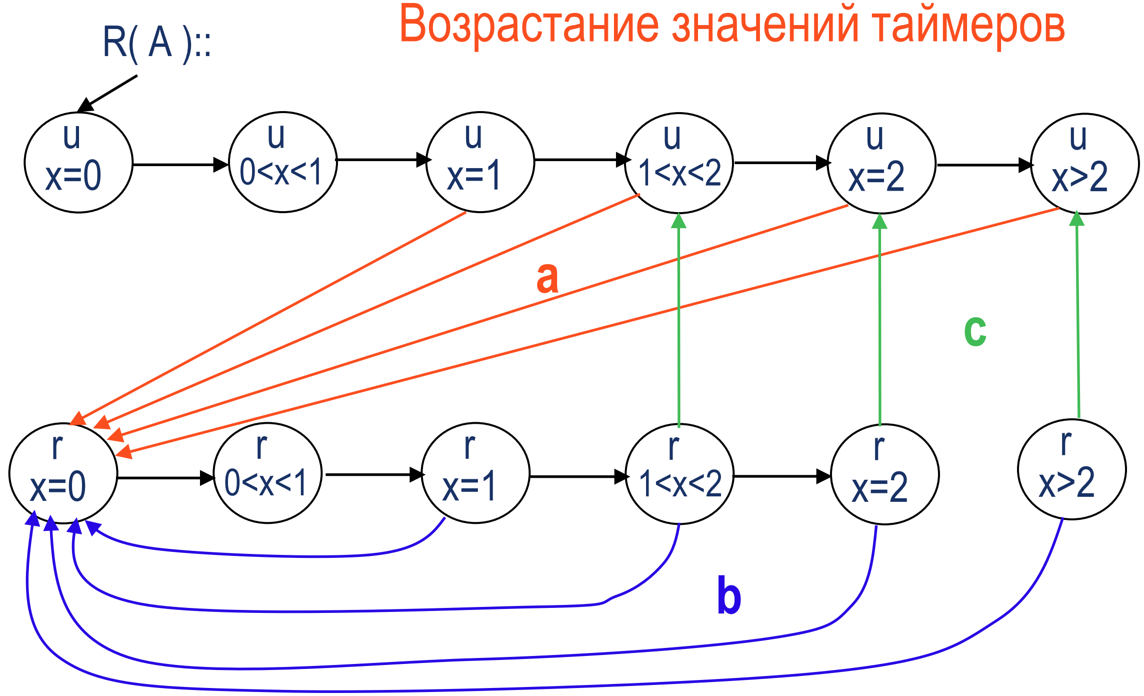
1. $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor \vee v(x)>c_x \;\&\; v'(x)>c_x$
2. $v(x) \leq c_x \Rightarrow fr(v(x)) = 0 \equiv fr(v'(x))= 0$



Возрастание значений таймеров

Пример:



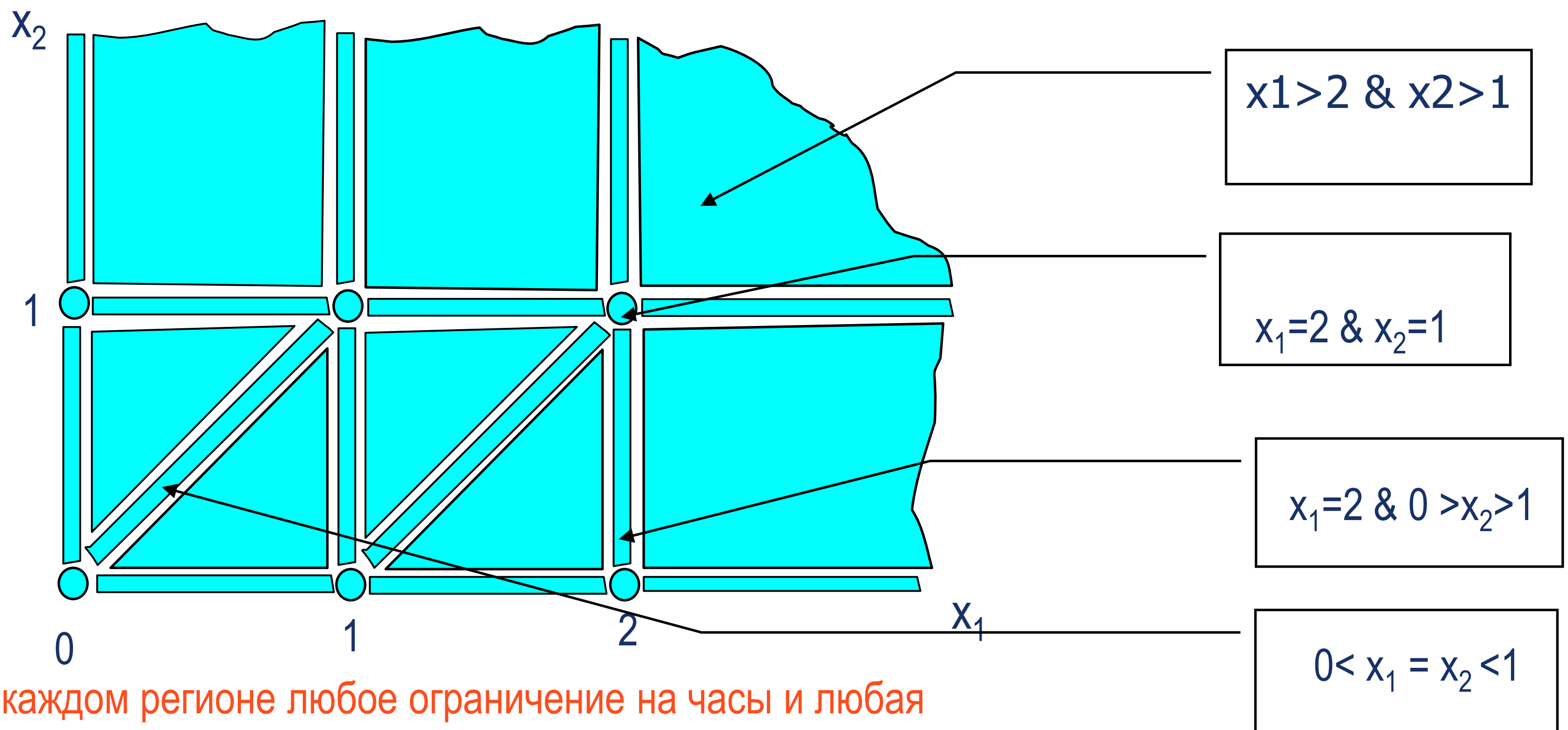


Состояние (r,x>2) недостижимо – его можно удалить

Для решения проблемы достижимости для автомата А, можно анализировать R(A)

Граф регионов – два таймера

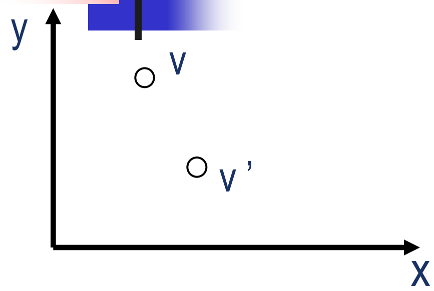
Пример. Пусть имеем два таймера, x_1 и x_2, с возможными временными соотношениями x_1~ k_1, x_2~ k_2, причем $k_1 \in \{0, 1, 2\}$, $k_2 \in \{0, 1\}$. Тогда имеется 28 регионов эквивалентных интерпретаций часов:



В каждом регионе любое ограничение на часы и любая темпоральная формула сохраняют свое значение

Такое представление конечно, но оно огромно: число областей = $|X|!\cdot 2^{|X|}\cdot \Pi_{x\in X}(2k_x+2)$ Оно экспоненциально растет от числа часов и верхней границы k

Граф регионов – два таймера - формально

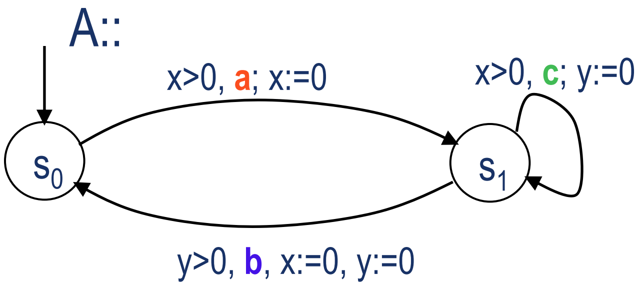


Две интерпретации часов v и v' эквивалентны ($v \cong v'$), iff:

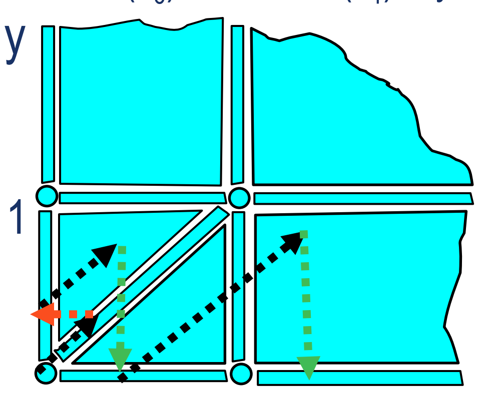
1. $(\forall x)$ $\lfloor v(x)\rfloor = \lfloor v'(x)\rfloor \vee v(x)>c_x \;\&\; v'(x)>c_x$

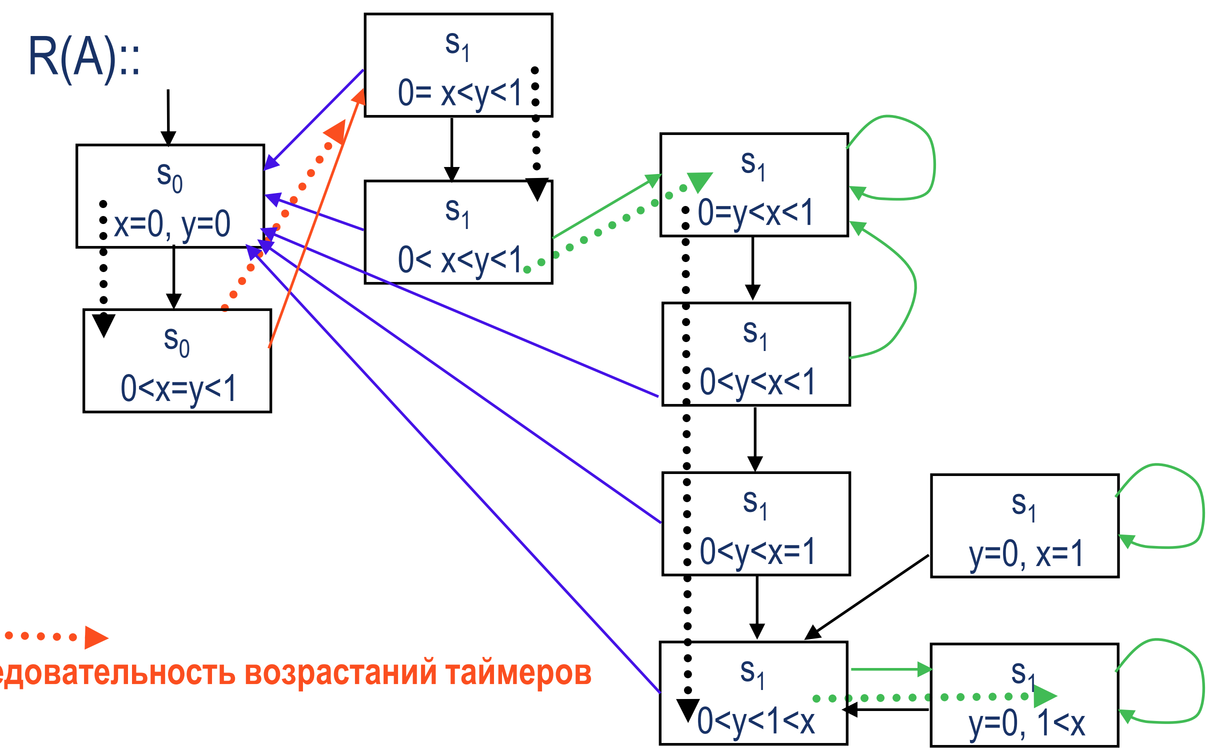
2. $(\forall x,y)$ $v(x)\le c_x \& \; v(y)\le c_y \Rightarrow fr(v(x))\le fr(v(y)) \equiv fr(v'(x))\le fr(v'(y))$

3. $(\forall x)$ $v(x) \le c_x \Rightarrow fr(v(x)) = 0 \equiv fr(v'(x)) = 0$



$Inv(s_0) = x<1$; $Inv(s_1) = y<1$





Последовательность возрастаний таймеров

Примеры свойств реального времени

1. [p $U_{<2}$ q] – p истинно непрерывно до тех пор, пока не станет истинно q, и истинность q наступит не позднее, чем через 2 единицы времени

2. AG(problem $\Rightarrow$ $AG_{\geq 5}$ alarm) – как только проблема возникла, сигнал alarm зазвучит сразу и будет звучать не менее 5

3. AG($\neg$far $\Rightarrow$ $AF_{< 7}$ far) – поезд покинет область контроля не позже, чем через 7

4. AG [send(m) $\Rightarrow$ $AF_{<5}$ receive(r_m)] – подтверждение приходит в пределах 5

5. EG [send(m) $\Rightarrow$ $AF_{> 4}$ receive(r_m)] - подтверждение может быть получено более, чем за 4

6. AG [$AF_{= 15}$ tick] – тики следуют периодически точно через 15 е.в. (но, кроме того, могут быть и в промежутках)

7. AG ($x \leq y$) - таймер x всегда не больше таймера y

8. A [off U $x \geq 3$] по любому пути из начального состояния если светофор выключен, то он будет выключен до тех пор, пока таймер x не будет иметь значение ≥ 3

Timed Temporal Logic – структура формул

TCTL – Timed CTL – естественное расширение операторов U, F, ... логики CTL количественной информацией.

Грамматика TCTL (= CTL + Time):

$\phi ::= p \mid \alpha \mid \neg\phi \mid \phi \wedge \phi \mid z \text{ in } \phi \mid E[\phi \, U \, \phi] \mid A[\phi \, U \, \phi]$

- p – атомарный предикат
- α - ограничение на таймеры и формульные часы
- z – формульные часы
- z in ϕ - введение новых часов в формулу ϕ
- $E[\phi \, U \, \phi]$, $A[\phi \, U \, \phi]$ – как в CTL

Обозначение: $E[\phi \, U_\alpha \psi] \equiv z \text{ in } E[(\phi \,\&\, \alpha) U \psi]$

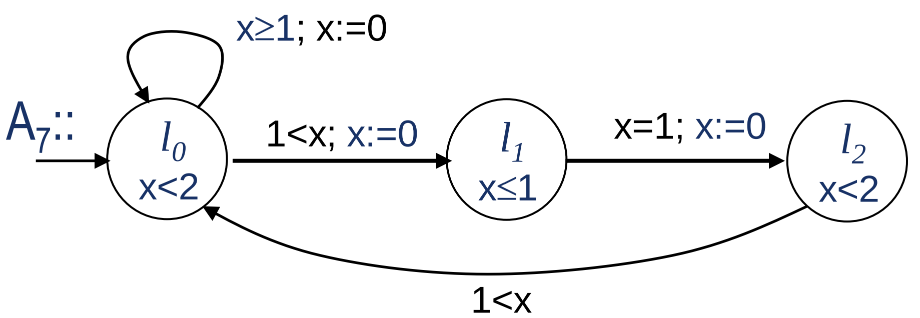
Выводимые операторы $EF_\alpha \phi \equiv E[\text{True } U_\alpha \phi]$ и т.д.

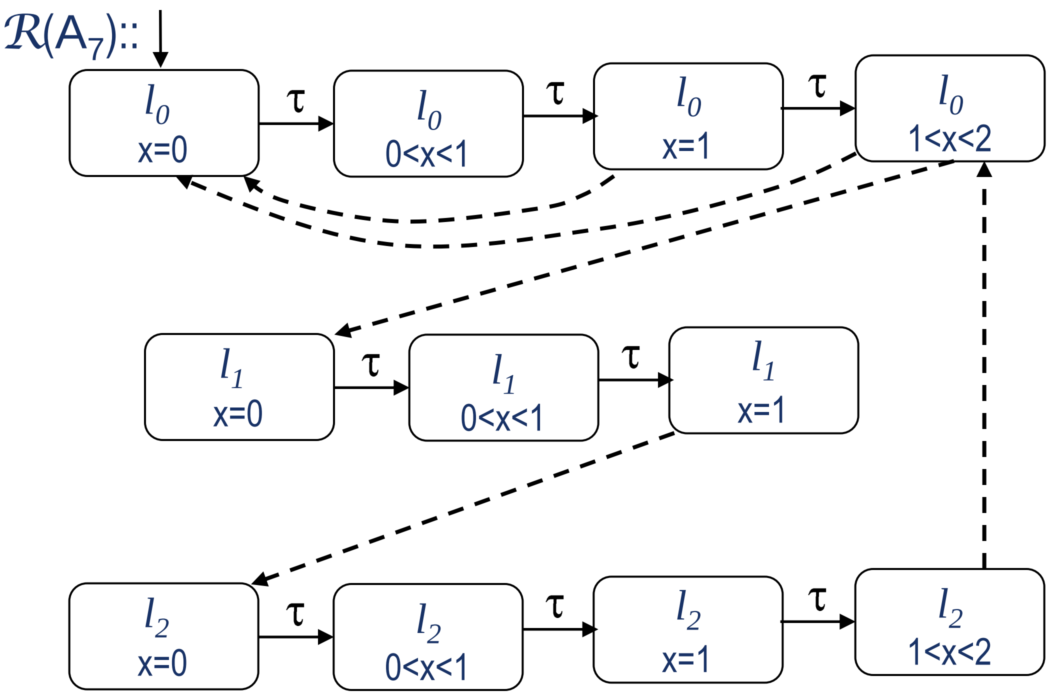
Формулы TCTL включают $E[\phi \, U_{\sim k} \psi]$, $A[\phi \, U_{\sim k} \psi]$, $EF_{\sim k}\phi$, $EG_{\sim k}\phi$, $AF_{\sim k}\phi$, $AG_{\sim k}\phi$
где ~ - любой символ из $\{<, \leq, =, \geq, >\}$ и k – рациональное число

Для верификации временного автомата A можно анализировать R(A)

Верификация Временных Автоматов

Сводится к верификации соответствующего графа регионов относительно CTL –формулы

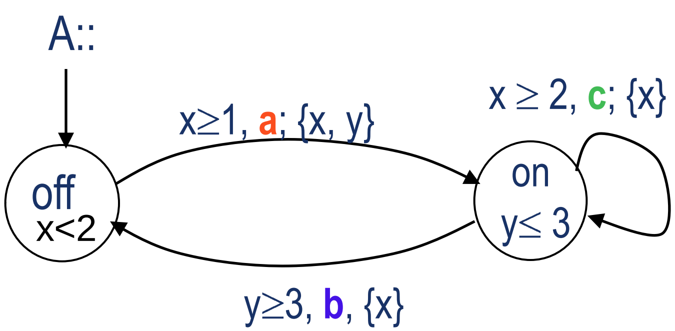


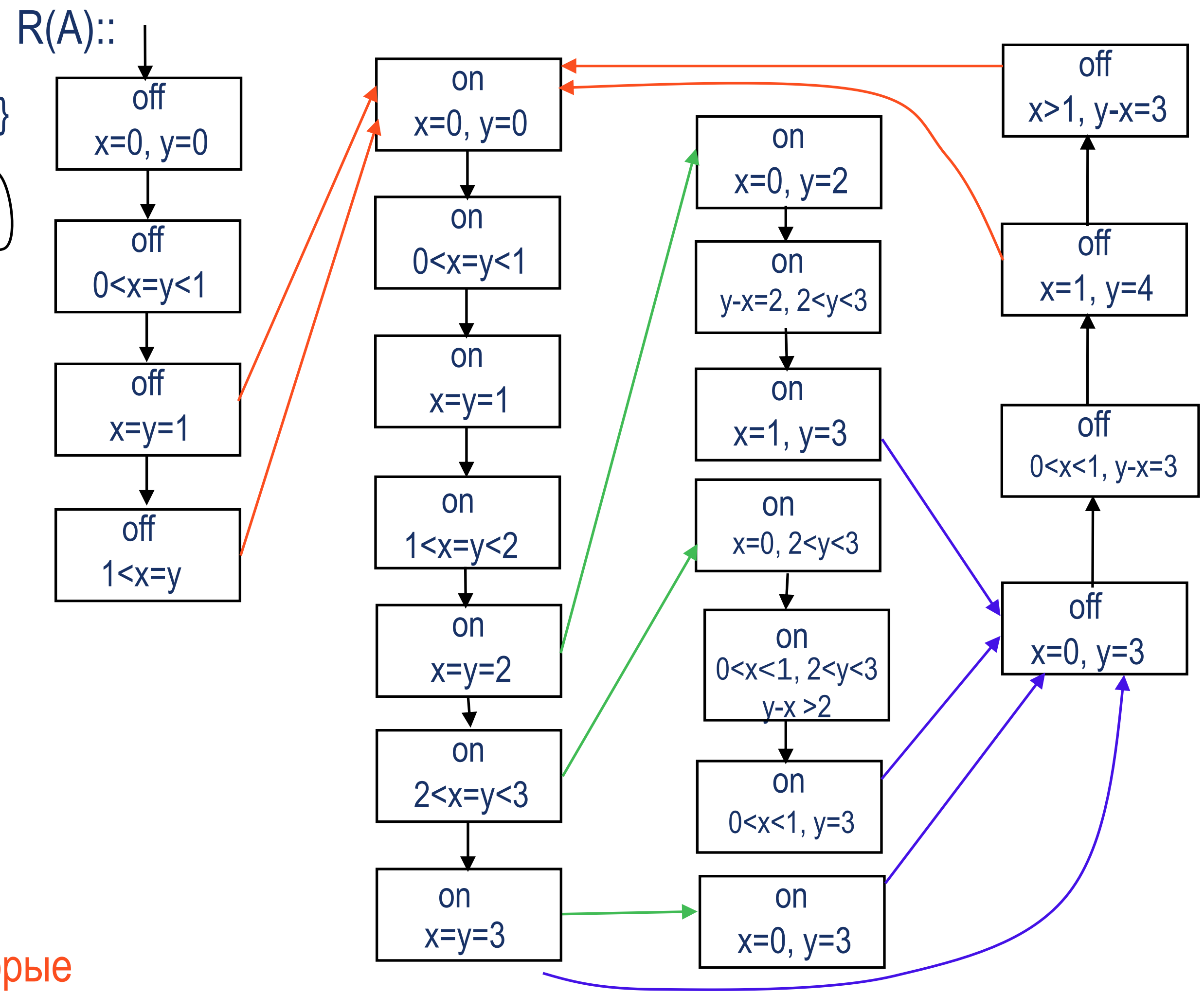


EG (x≤1) – выполняется

Верификация Временных Автоматов

Сводится к верификации соответствующего графа регионов относительно CTL –формулы





Свойства:

AG (x≤y) – выполняется

AG EF (off & x=y) –
не выполняется

AG(on ⇒ AF off) – выполняется

A [off U x≥ 3] – не выполняется

Все темпоральные свойства, которые выражаются формулами CTL и через таймеры – проверяются просто по R(A)

Верификация Временных Автоматов

Если TCTL-формула включает “формульные” ограничения, то проверка формулы проводится с некоторыми добавлениями – с введением “формульных часов”

$EG(on \Rightarrow EF_{\leq 2}\, off)$

Как доказать?

$EF_{\alpha}\phi \equiv E\,[\,True\ U_{\alpha}\ \phi\,]$

$AG(on \Rightarrow EF_{\leq 2}\, off) =$

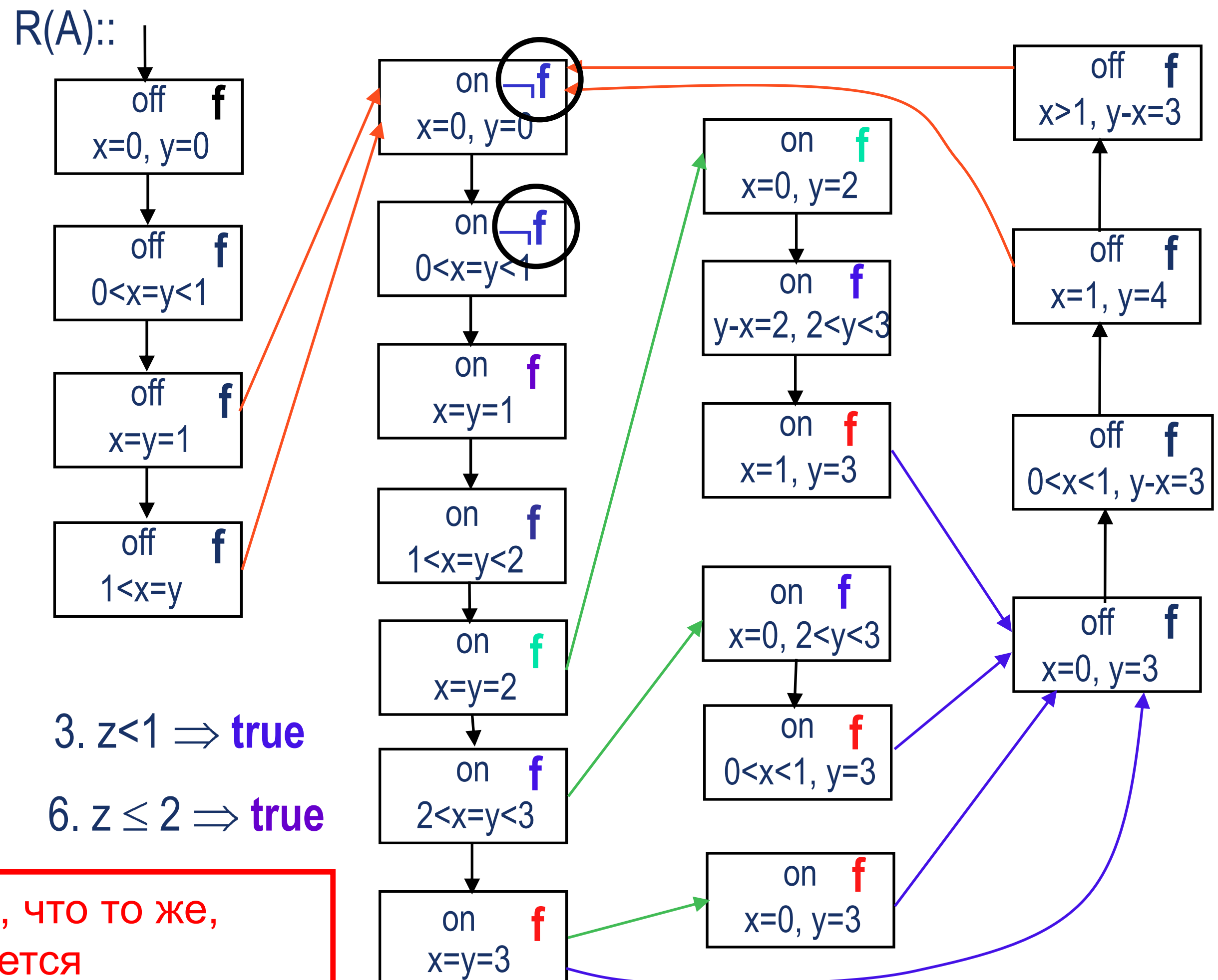
$AG(on \Rightarrow E\,[\,True\ U_{\leq 2}\ off\,]) =$

$AG(on \Rightarrow z\ in\ (\,E\,[\,(z \leq 2)\ U\ off\,]\,))$

Обозначим **f** = $E\,[\,(z \leq 2)\ U\ off\,]$

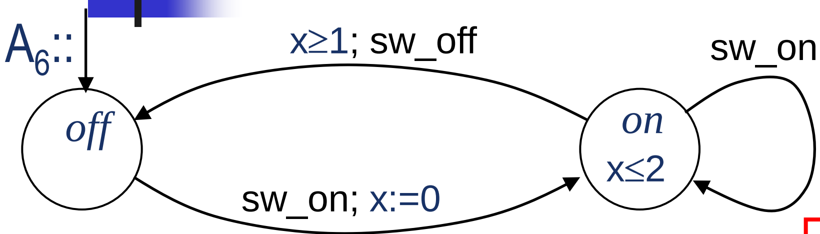
1. $off \Rightarrow$ **true** 2. $z=0 \Rightarrow$ **true** 3. $z<1 \Rightarrow$ **true**

4. $z \leq 1 \Rightarrow$ **true** 5. $z<2 \Rightarrow$ **true** 6. $z \leq 2 \Rightarrow$ **true**



Таким образом, $AG(on \Rightarrow f)$, или, что то же, $AG(on \Rightarrow EF_{\leq 2}\, off\,])$– не выполняется

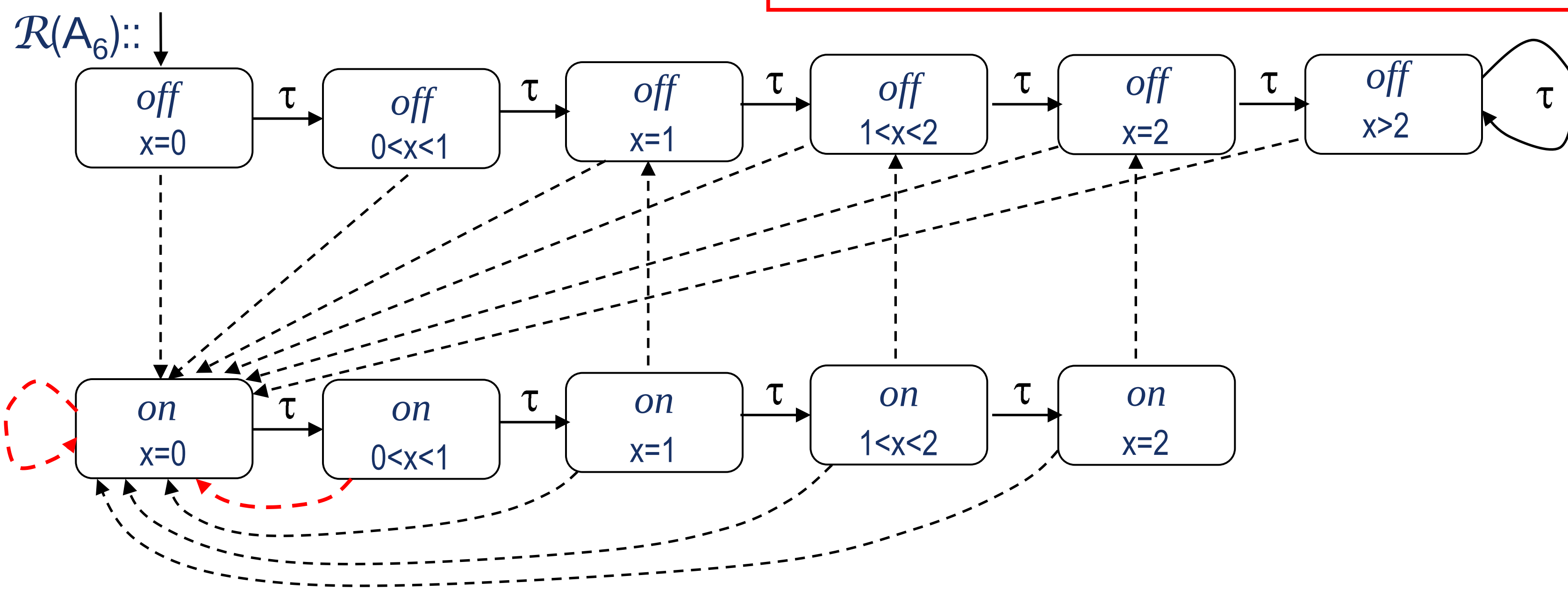
Верификация Временных Автоматов



это – конвергентные пути (Зенон)

⟨off, 0⟩ ⟨on, 0⟩ ⟨on, 0.1⟩ ⟨on, 0 ⟩ ⟨on, 0.01⟩...

Некоторые вычисления, в которых локация on никогда не покидается, являются конвергентными



$Sat(AF_{<1} off) = \{ \langle off, v(x) \rangle \mid x \geq 0 \}$

$Sat(EF_{<1} off) = \{ \langle off, v(x) \rangle \mid x) \geq 0 \} \cup \{ \langle on, v(x) \rangle \mid 0 < x \leq 2 \}$

$Sat(AF(on \wedge x=1)) = \{ \langle off, v(x) \rangle \mid x = 1 \}$

Символьная Верификация Временных Автоматов

Каждая временная область может быть представлена как логическая формула над линейными clock constraints (ограничениями таймеров) – их дизъюнкции и конъюнкции

Идея: нельзя ли анализировать ТА без явного априорного конструирования разбиений на регионы, а просто символически манипулируя всеми ограничениями на показания таймеров?

Такой метод предложили в 1994 г. T. Henzinger и др. – символьнкая верификация Временных Автоматов

Заключение

- Для систем реального времени требуется анализ количественных характеристик поведения систем, чего не может дать обычная темпоральная логика
- Временные автоматы удобны для спецификации широкого класса систем реального времени
- Можно представить все поведения временного автомата в замкнутом виде с помощью регионов, однако такое представление требует огромного объема информации
- Существует несколько методов проверки свойств систем реального времени, специфицированных моделью временного автомата. Три наиболее часто используемых – анализ достижимости, использование контрольного автомата, проверка формул TCTL
- Верификация Временного автомата относительно TCTL-формулы сводится к верификации соответствующего графа регионов относительно CTL –формулы (с некоторыми добавлениями)
- Исследования в этой области начались недавно ("*Timed automata are recent models*…" B.Berard, System and software verification, 2001). Исследования продолжаются, в основном, они направлены на поиск методов более компактного представления временных автоматов
- Существует разнообразие подходов, методов, даже трактовок одних и тех же терминов
- Инструменты автоматизированной верификации временных автоматов (KRONOS, UPPAAL, …) сами выполняют построение параллельной композиции временных автоматов, построение графа регионов и проверку формул TCTL для не очень больших систем

Спасибо за внимание!