

НАДЁЖНОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

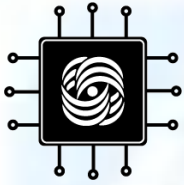
Лекция 4: *Временные автоматы*

ВМиК МГУ им. М.В. Ломоносова,
Кафедра АСВК, Лаборатория Вычислительных Комплексов
Ассистент Волканов Д.Ю.



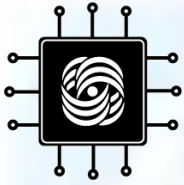
План лекции

- Временной автомат:
 - Общее описание
 - Пример
 - Синтаксис
 - Семантика
- Сеть временных автоматов:
 - Описание
 - Пример
- Временные свойства (логика TCTL)



Временной автомат

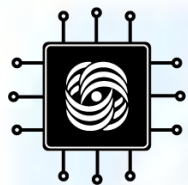
- Описание, пригодное для моделирования и анализа (асинхронных) систем реального времени
- Содержит конечное множество состояний управления
- Дополнительно содержит таймеры, принимающие действительные значения
- В структуре автомата отражено влияние таймеров на поведение системы



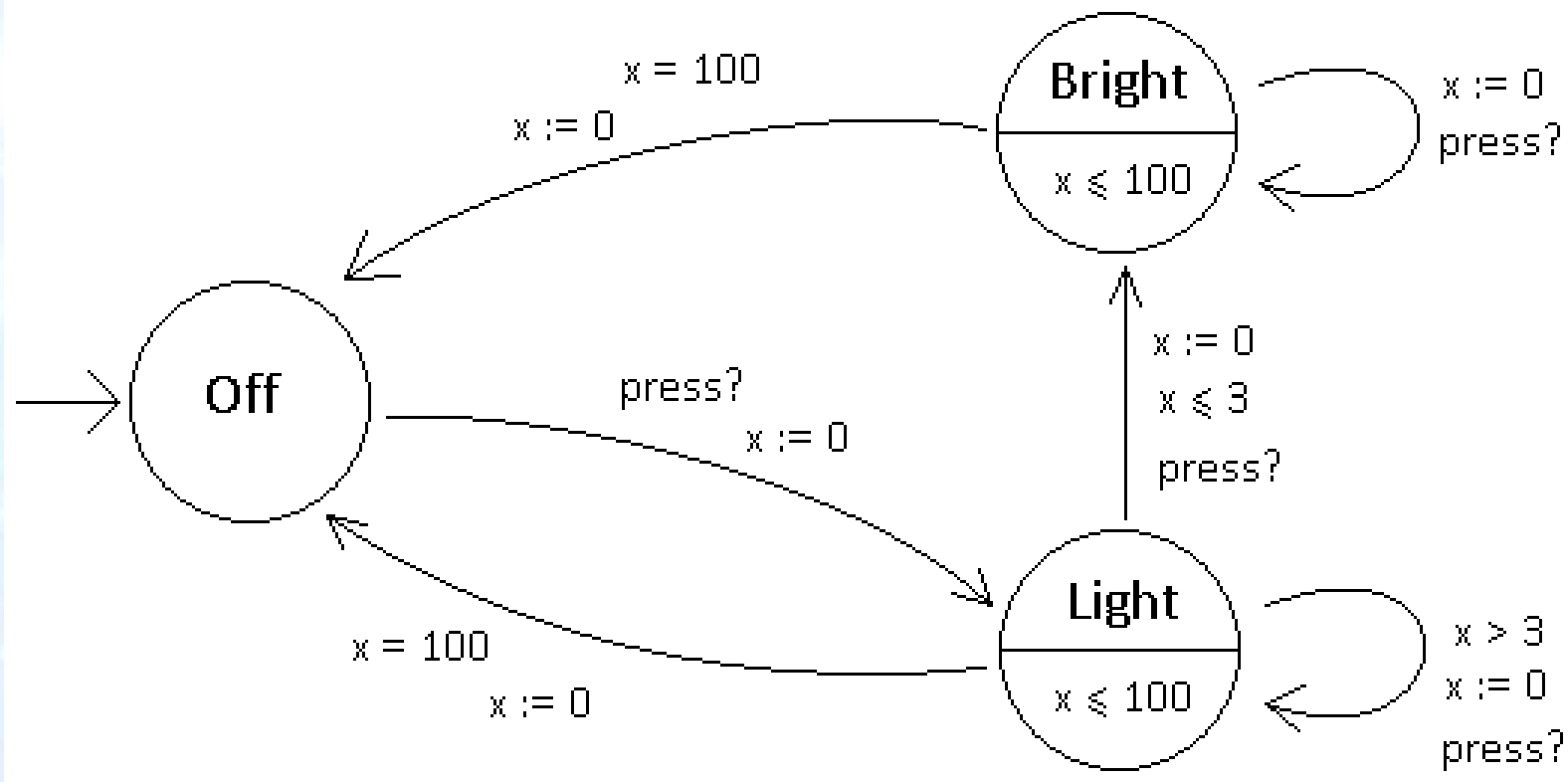
Пример временного автомата

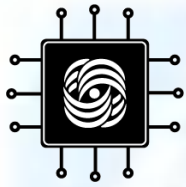
Модель переключателя освещения

- Содержит 3 состояния управления:
 - Off (выключено)
 - Light (обычное освещение)
 - Bright (яркое освещение)
- Свет выключается при 100 секундах неактивности
- Свет включается при нажатии на кнопку (press)
- Яркий свет включается при двойном нажатии (двух нажатиях, отстоящих не более чем на 3 секунды)

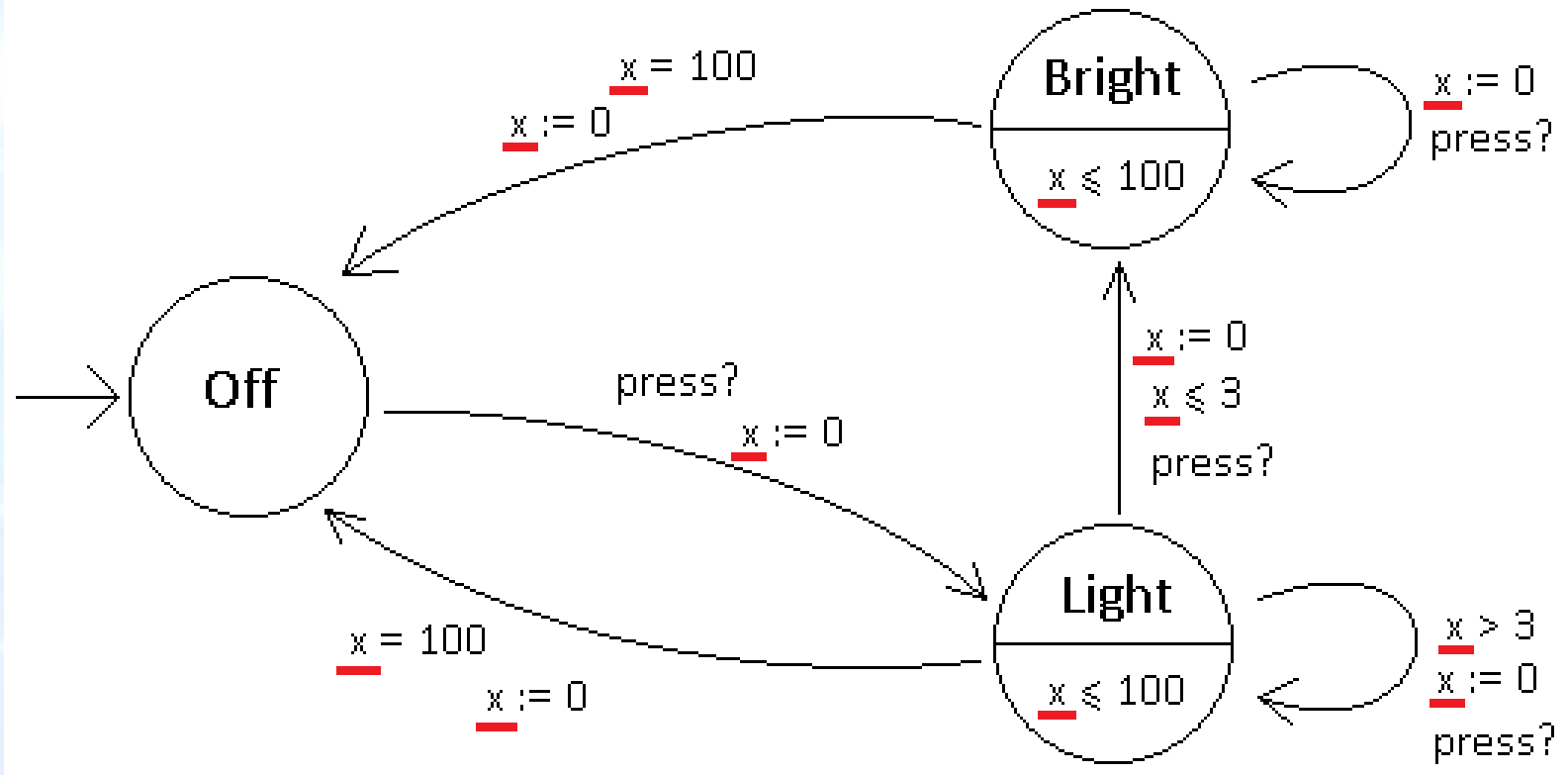


Пример временного автомата

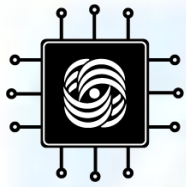




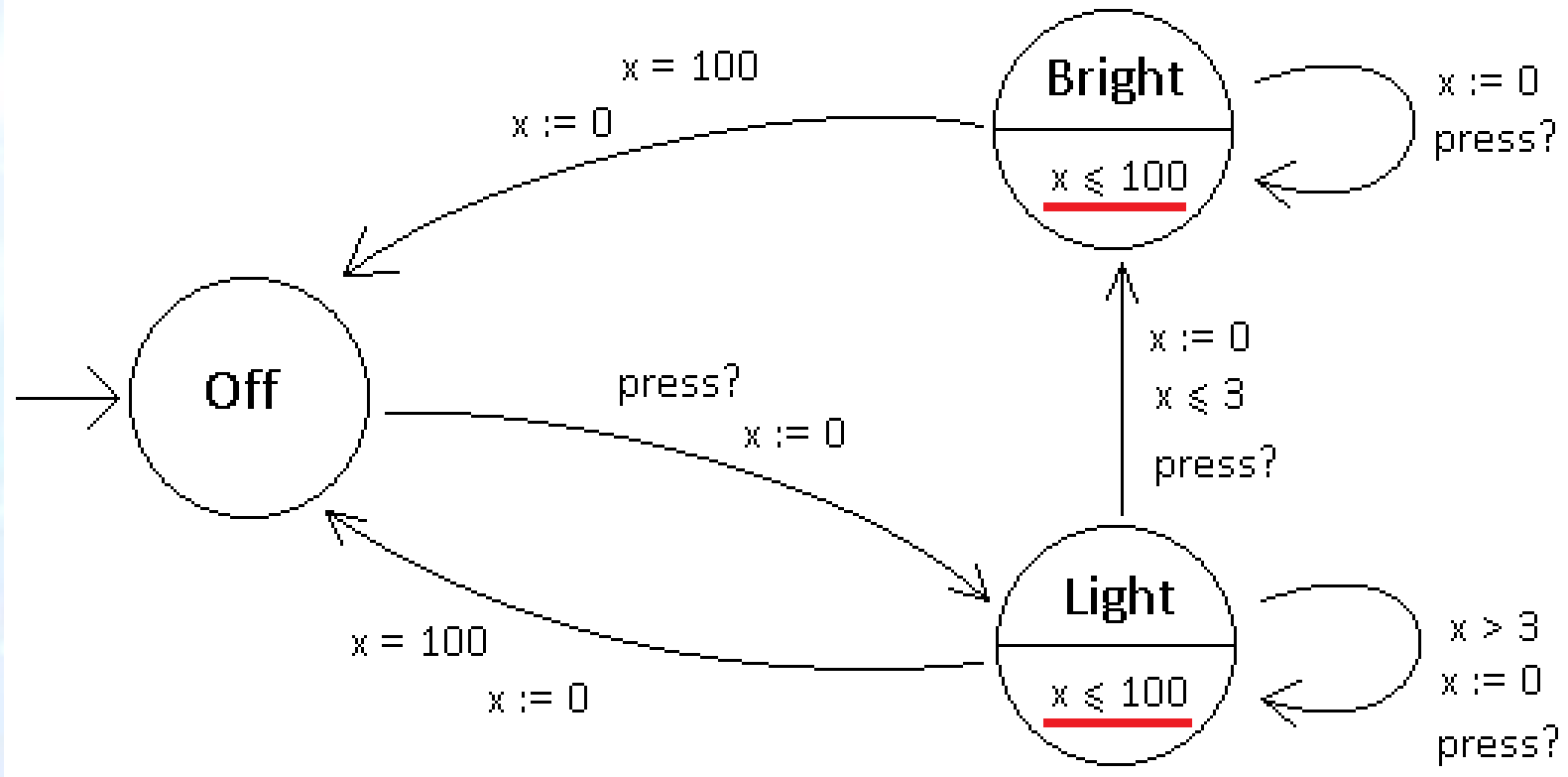
Таймеры



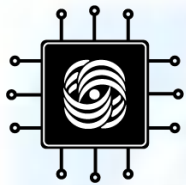
- x – таймер, принимающий действительные неотрицательные значения



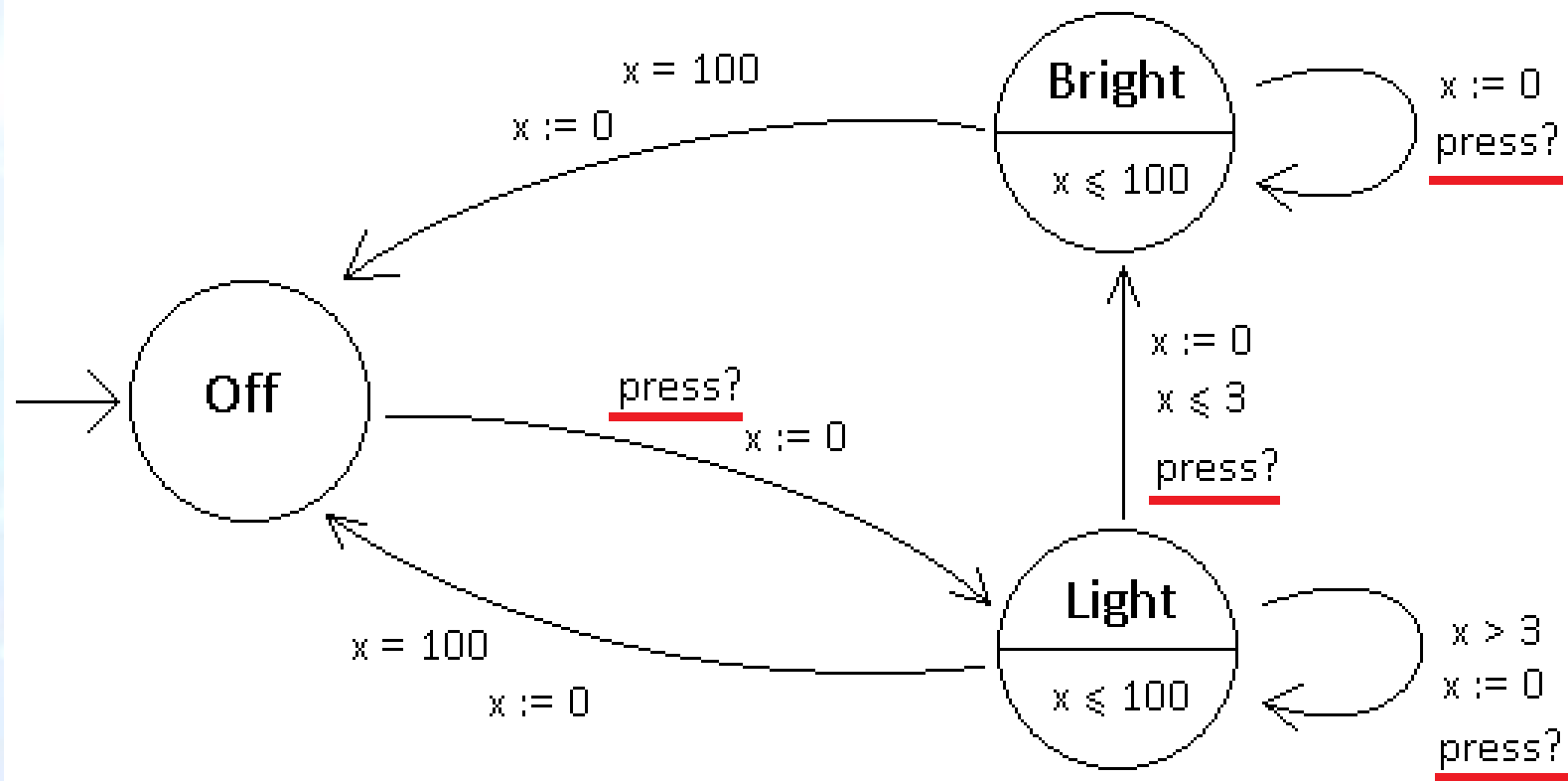
Инварианты



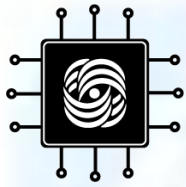
- Инвариант – это необходимое условие активности состояния



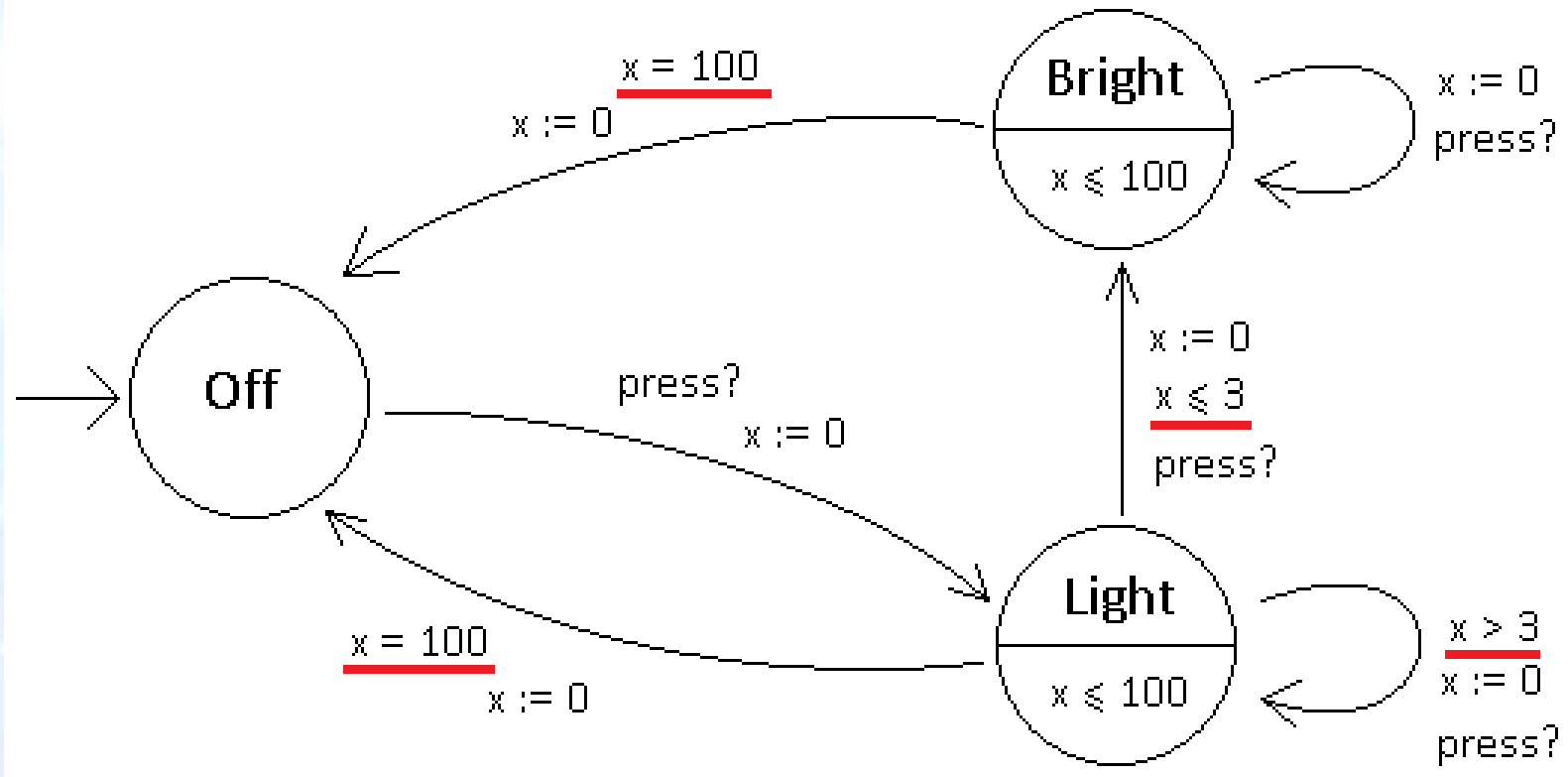
Каналы связи



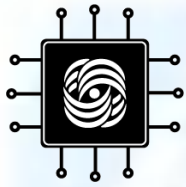
- **press?** – внешнее событие нажатия кнопки включения



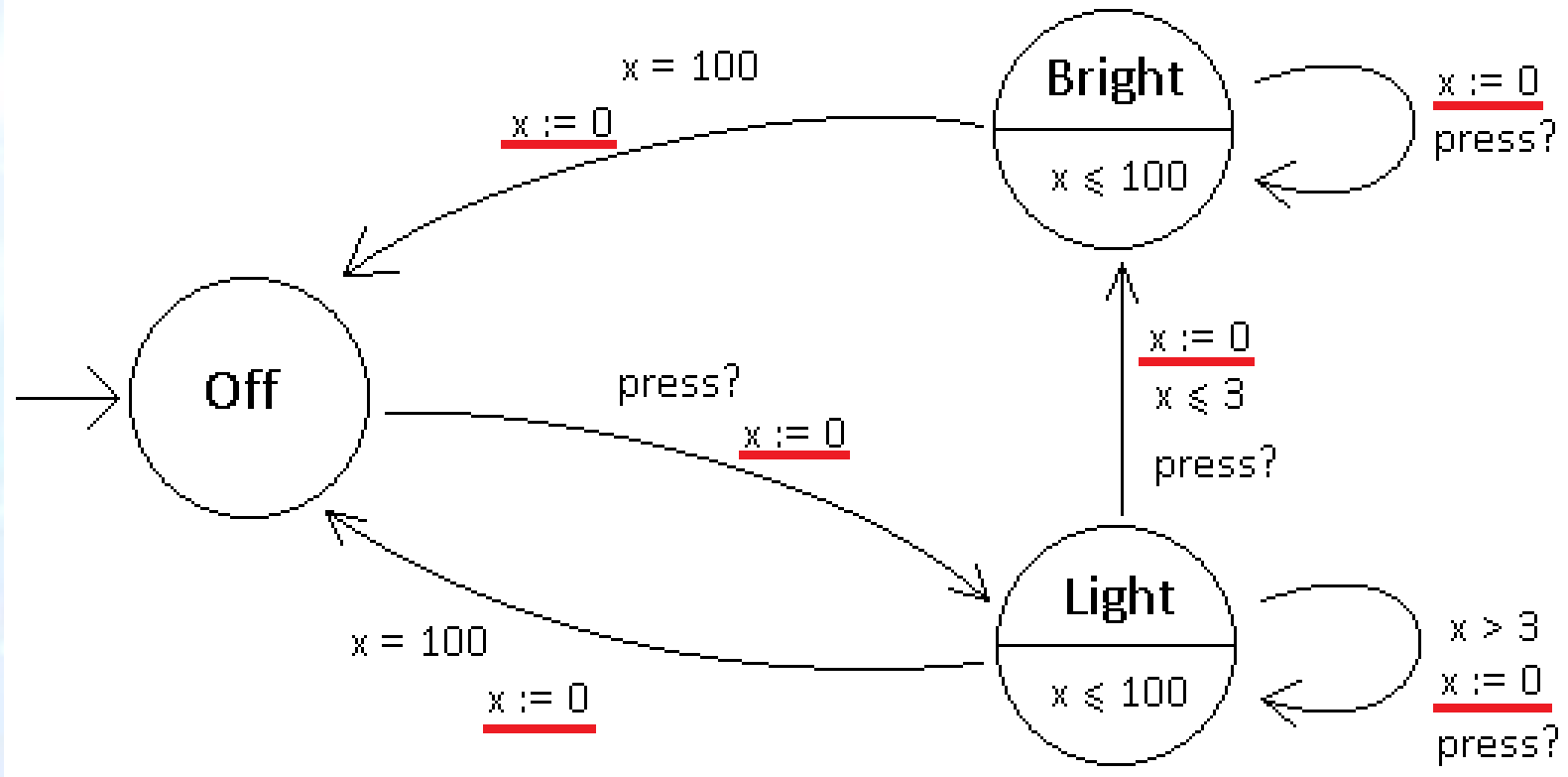
Предусловия



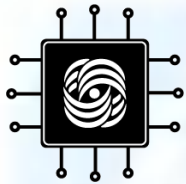
- Предусловие – это необходимое условие выполнения перехода



Сброс таймеров



- При выполнении перехода таймер может быть сброшен



Временные ограничения

- Элементарное временное ограничение:

$\text{ETC} ::= x \text{ op } n \mid x - y \text{ op } n$

$\text{op} ::= < \mid \leq \mid = \mid \geq \mid >$

$(x, y \in \mathbb{C}, n \in \mathbb{Z})$

- Временное ограничение:

$\text{TC} ::= \text{ETC} \mid \text{TC} \wedge \text{TC} \mid \text{TC} \vee \text{TC} \mid \neg \text{TC} \mid (\text{TC})$

- Элементарное инвариантное ограничение:

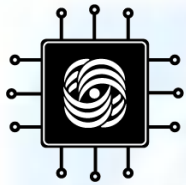
$\text{ETC}\Downarrow ::= x \text{ op}\Downarrow n$

$\text{op}\Downarrow ::= < \mid \leq$

$(x \in \mathbb{C}, n \in \mathbb{Z})$

- Инвариантное ограничение:

$\text{TC}\Downarrow ::= \text{ETC}\Downarrow \mid \text{TC}\Downarrow \wedge \text{TC}\Downarrow$

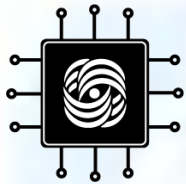


Синтаксис временного автомата

Временной автомат – это система

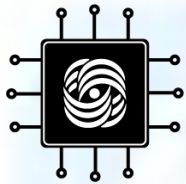
(C, A, L, I_0, P, E) , где

- C – множество таймеров
- A – множество действий
- L – множество состояний управления
- I_0 – начальное состояние
- $P : L \rightarrow TC \Downarrow$ – назначение инвариантов
- $E \subseteq L \times TC \times A \times 2C \times L$ – множество переходов



Конфигурация автомата

- Конфигурация автомата: (l, v)
 - l – состояние управления
 - $v : C \rightarrow R^+$ - оценка таймеров
- Начальная конфигурация: (l_0, v_0)
 - l_0 – начальное состояние автомата
 - $v_0(x) = 0$ для всех таймеров x
- Обозначения:
 - $v+d, d \in R^+$: $v+d(x) = v(x) + d$
 - $v[C'], C' \subseteq C$: $v[C'](x) = 0$ при $x \in C'$; $v[C'](x) = v(x)$ иначе
 - $v \models g$: формула g истинна при значениях таймеров, определяемых оценкой v

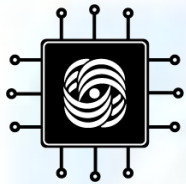


Вычисление автомата

Шаг вычисления автомата состоит в

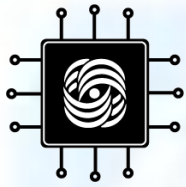
- Продвижении времени:
 - состояние управления не изменяется,
 - все таймеры продвигаются на константу d ;
- Выполнении перехода:
 - выбирается переход из I в I' ,
 - состояние управления меняется на I' ,
 - таймеры, которыми помечен переход, сбрасываются;

После применения шага инварианты всех состояний должны быть выполнены



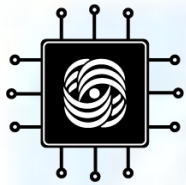
Вычисление автомата

- Продвижение времени: $(l, v) \rightarrow_d (l, v+d)$, если
 - $\forall l \in L (v+d \models P(l))$
- Выполнение перехода: $(l, v) \rightarrow_a (l', v')$, если
 - $(l, g, a, C', l') \in E$
 - $v \models g$
 - $v' = v[C']$
 - $v' \models P(l')$
- $\rightarrow = \rightarrow_d \cup \rightarrow_a$
- Вычисление:
$$c1 \rightarrow c2 \rightarrow \dots \rightarrow c_n \rightarrow \dots$$
- Инициальное вычисление:
 - $c1$ – начальная конфигурация автомата



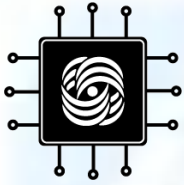
Синхронизация автоматов

- Выделяются действия синхронизации:
 - Активные: $a!$
 - Пассивные: $a?$
- Они соответствуют посылке и приему сигнала по каналу a
- При выполнении перехода с синхронизацией выбирается второй переход с парной синхронизацией, и эти два перехода в различных автоматах выполняются одновременно



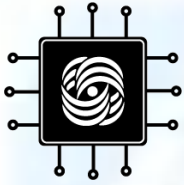
Сеть временных автоматов

- C, A – множество таймеров и множество действий сети
- Сеть – это совокупность автоматов
$$N = (A_1, A_2, \dots, A_n), \text{ где}$$
- $A_i = (C, A, L_i, I_{O_i}, P_i, E_i)$ – временные автоматы
- Сеть – это параллельная композиция автоматов с синхронизацией типа handshake



Семантика сети

- Конфигурация: (I, v)
 - $I = (I_1, I_2, \dots, I_n)$ – вектор управляющих состояний автоматов сети
 - $v : C \rightarrow R^+$ – оценка таймеров
- Начальная конфигурация: (I_0, v_0)
 - I_0 – вектор начальных состояний автоматов
 - $v_0(x) = 0$ для всех таймеров x
- Обозначение:
 - $I[i \leftarrow I']$ – вектор, полученный из I заменой i -й компоненты на I'

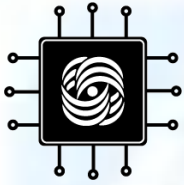


Семантика сети

- Правила продвижения времени и выполнения перехода обобщаются естественным образом (отношение $\Rightarrow$)
- Правило выполнения переходов с синхронизацией:

$$(I, v) \Rightarrow (I', v'), \text{ где}$$

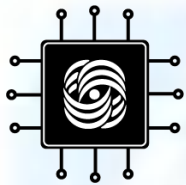
- $I' = I[i \leftarrow I', j \leftarrow I'']$
- $i \neq j$
- $(li, !a, g', C', I') \in Li, (lj, ?a, g'', C'', I'') \in Lj$
- $v' = v[C' \cup C'']$
- $v \models g' \wedge g''$
- $\forall I \in I' (v' \models P(I))$
- Вычисление сети:
$$c1 \Rightarrow c2 \Rightarrow \dots \Rightarrow cn \Rightarrow \dots$$
- Инициальное вычисление сети:
 - $c1$ – начальная конфигурация сети



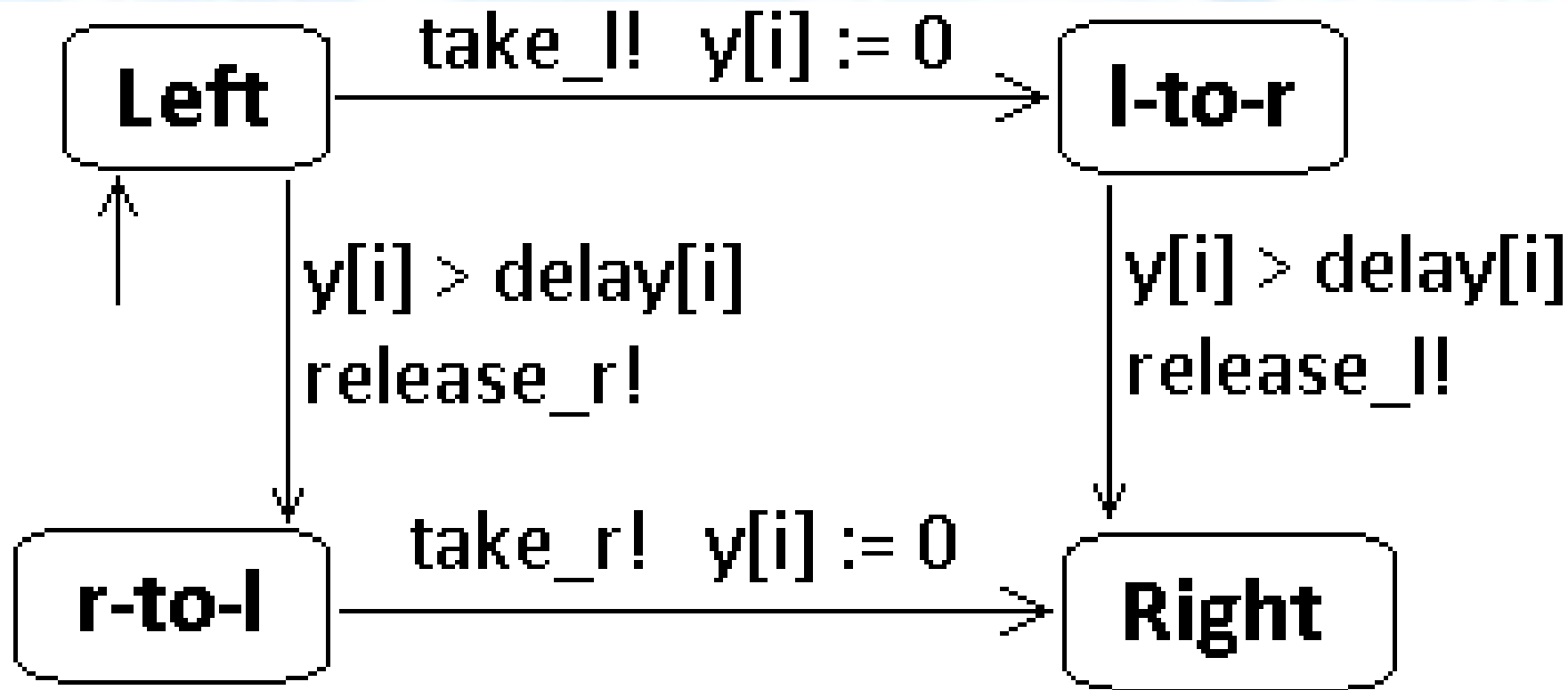
Пример сети

Задача о викингах:

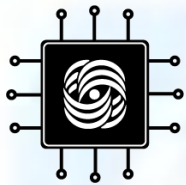
- N викингов хотят перейти через мост
- У каждого викинга своя максимальная скорость
- Имеется один факел
- Идти через мост можно только при факеле одному или вдвоем
- Требуется перевести викингов через мост как можно быстрее



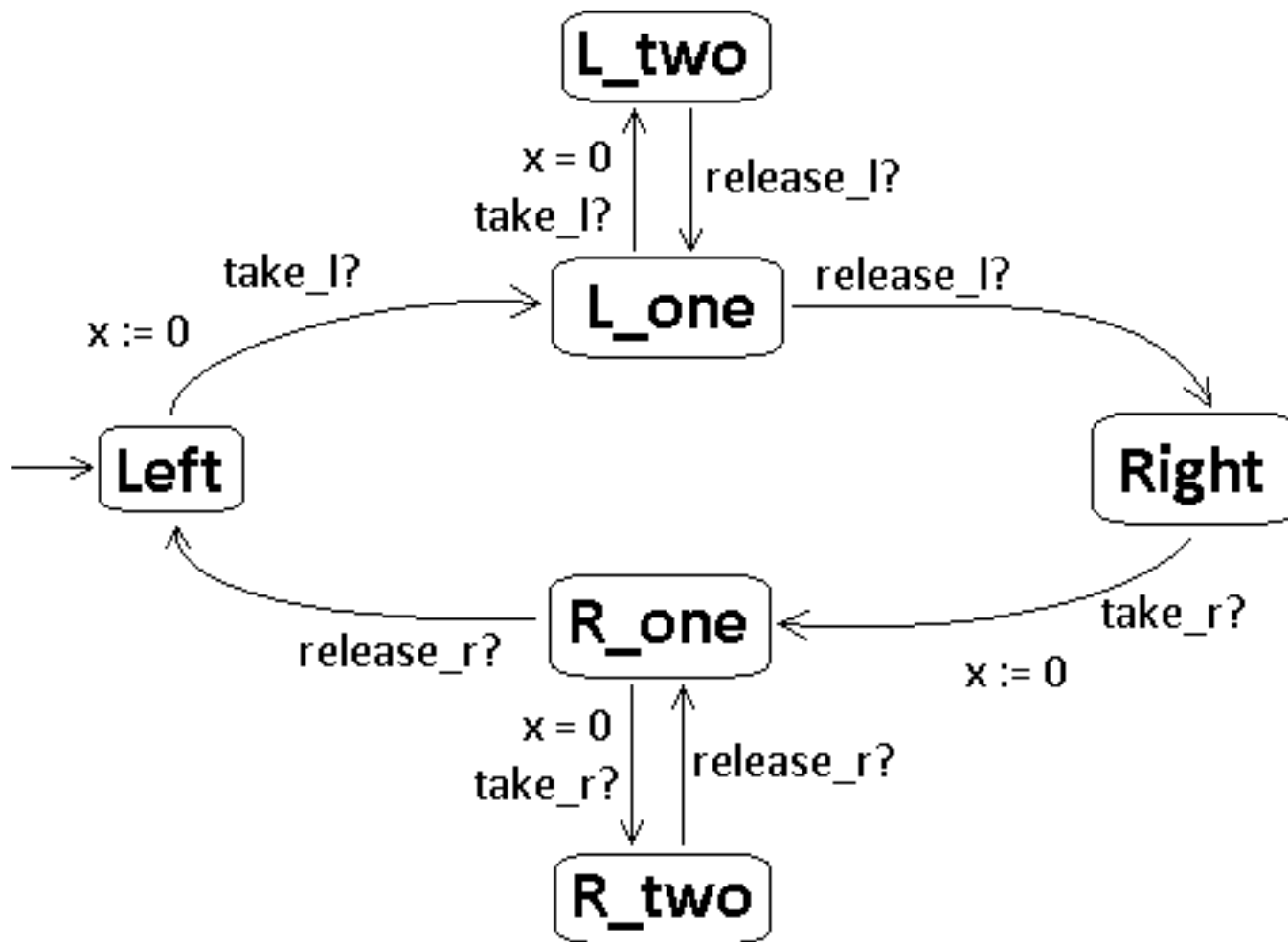
Пример сети



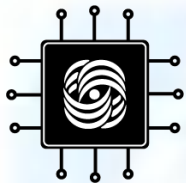
- Автомат для i -го викинга
- $y[i]$ – уникальный для i -го викинга таймер
- $\text{delay}[i]$ заменяется на конкретную константу – минимальное время прохода по мосту



Пример сети



- Временной автомат для факела



Логика TCTL

- Временное ограничение:

$CC ::= x \text{ op } n \mid x - y \text{ op } n$

$\text{op} ::= < \mid \leq \mid = \mid \geq \mid >$

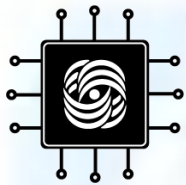
$(x, y \in C; n \in \mathbb{Z})$

- Формула:

$\Phi ::= P \mid CC \mid z.\Phi \mid \Phi \vee \Phi \mid \Phi \wedge \Phi \mid \neg \Phi$

$\mid AG \Phi \mid AF \Phi \mid EG \Phi \mid EF \Phi$

$\mid A(\Phi \cup \Phi) \mid E(\Phi \cup \Phi)$



Логика TCTL (упрощенный вариант)

- Элементарный предикат:

$EP ::= Ai.l \mid x \text{ op } n \mid x - y \text{ op } n$

$op ::= < \mid \leq \mid = \mid \geq \mid >$

$(l \in Li; x, y \in C; n \in Z)$

- Формула:

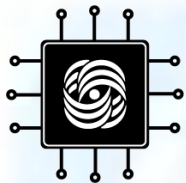
$\Phi ::= EP \mid \Phi \vee \Phi \mid \Phi \wedge \Phi \mid \neg \Phi$

$\mid AG \Phi \mid AF \Phi \mid EG \Phi \mid EF \Phi$

$\mid A(\Phi \cup \Phi) \mid E(\Phi \cup \Phi)$

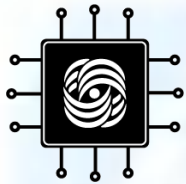
- Обозначение:

- $N, s \models \Phi$: формула Φ выполняется в конфигурации с сети N



Логика TCTL (упрощенный вариант)

- $N, ((l_1, l_2, \dots, l_n), v) \models A_i.l \Leftrightarrow l_i = l$
- $N, (l, v) \models x \text{ op } n \Leftrightarrow v \models x \text{ op } n$
- $N, (l, v) \models x - y \text{ op } n \Leftrightarrow v \models x - y \text{ op } n$
- $N, s \models AG \Phi \Leftrightarrow$ в каждой конфигурации каждого вычисления, начинающегося в s , выполнена формула Φ
- $N, s \models AF \Phi \Leftrightarrow$ хотя бы в одной конфигурации каждого вычисления, начинающегося в s , выполнена формула Φ
- $N, s \models EG \Phi \Leftrightarrow N, s \models \neg AF \neg \Phi$
- $N, s \models EF \Phi \Leftrightarrow N, s \models \neg AG \neg \Phi$
- Соотношения для булевых связок определяются естественным образом



Примеры временных свойств

- Можно перевести всех викингков через мост за время T (здесь `time` – таймер сети)

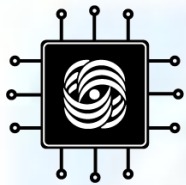
$EF(Viking[1].Right \wedge \dots \wedge Viking[n].Right \wedge time \leq T)$

- Нельзя перевести викингга за время, меньшее времени его перехода через мост

$AG \neg (Viking[i].Right \text{ and } time < delay[i])$

- Если факел был захвачен, рано или поздно он будет освобожден (не выполняется)

$AG((Torch.Left \vee Torch.Right) \vee AF(Torch.Left \vee Torch.Right))$



Спасибо за внимание