

Autômatos Temporizados

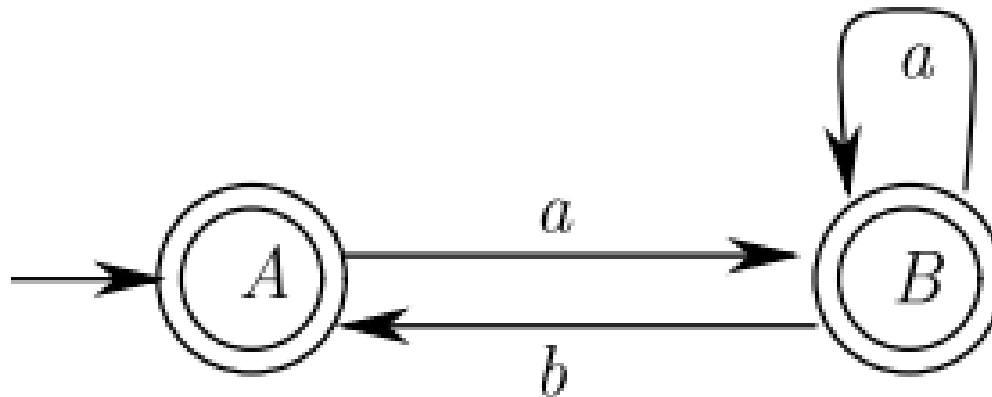
Autômatos Finitos

Modelagem de sistemas através da representação dos estados possíveis

Mudança de estado através da ocorrência de eventos

Modelos representativos:

Autômatos finitos determinísticos e não-determinísticos



Autômatos Finitos Não-Determinísticos

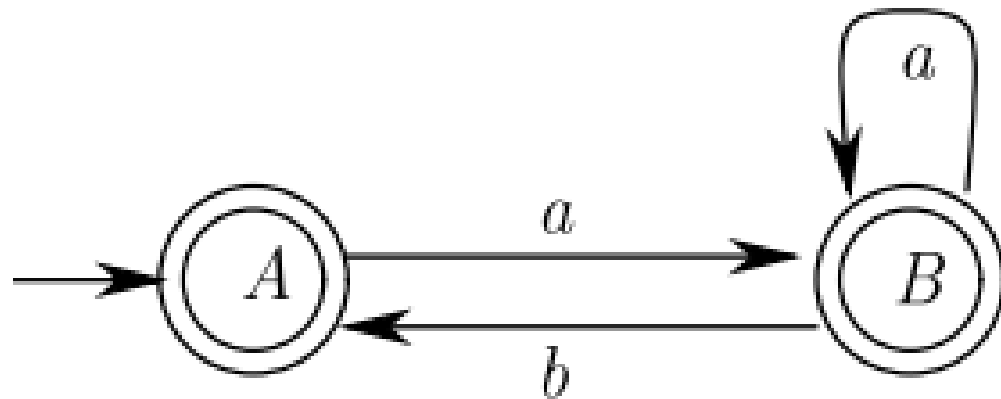
$$G_{nd} = (S, E \cup \{\epsilon\}, f, \Gamma, S_0, S_M)$$

- S - Conjunto de estados
- $S_0 \subseteq S$ - Conjunto de estados iniciais
- E - Conjunto de eventos (alfabeto)
- $F : S \times E \times S$ - Relação de próximos estados
- $\Gamma : S \rightarrow 2^E$ - Função de eventos factíveis
- $S_m \subseteq S$ - Conjunto de estados marcados

Autômatos Finitos Não-Determinísticos

$$G_{nd} = (S, E, f, \Gamma, s_0, S_M)$$

- $S = \{A, B\}$
- $S_0 = \{A\}$
- $E = \{a, b\}$
- $f(A, a) = B, f(B, a) = B, f(B, b) = A$
- $\Gamma(A) = \{a\}, \Gamma(B) = \{a, b\}$
- $S_m = \{A, B\}$



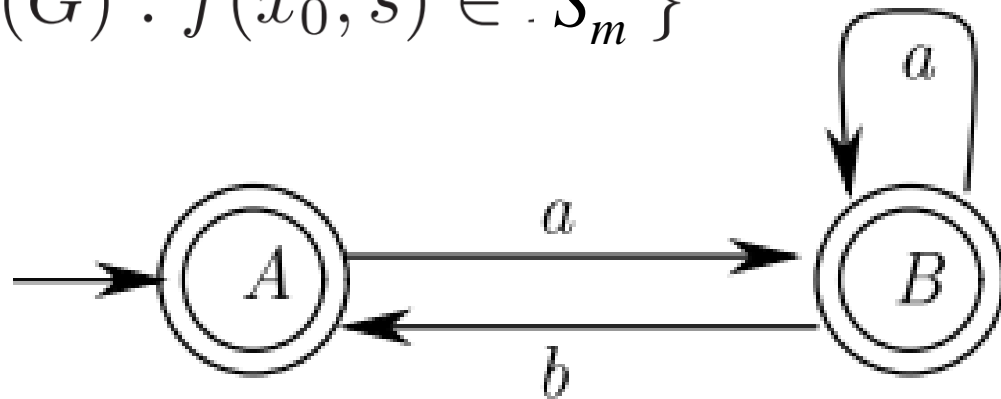
Autômatos Finitos Não-Determinísticos

Linguagem gerada

$$\mathcal{L}(G) := \{s \in E^* : f(x_0, s) \text{ is defined}\}$$

Linguagem Marcada

$$\mathcal{L}_m(G) := \{s \in \mathcal{L}(G) : f(x_0, s) \in \cdot S_m\}$$



Autômatos Temporizados

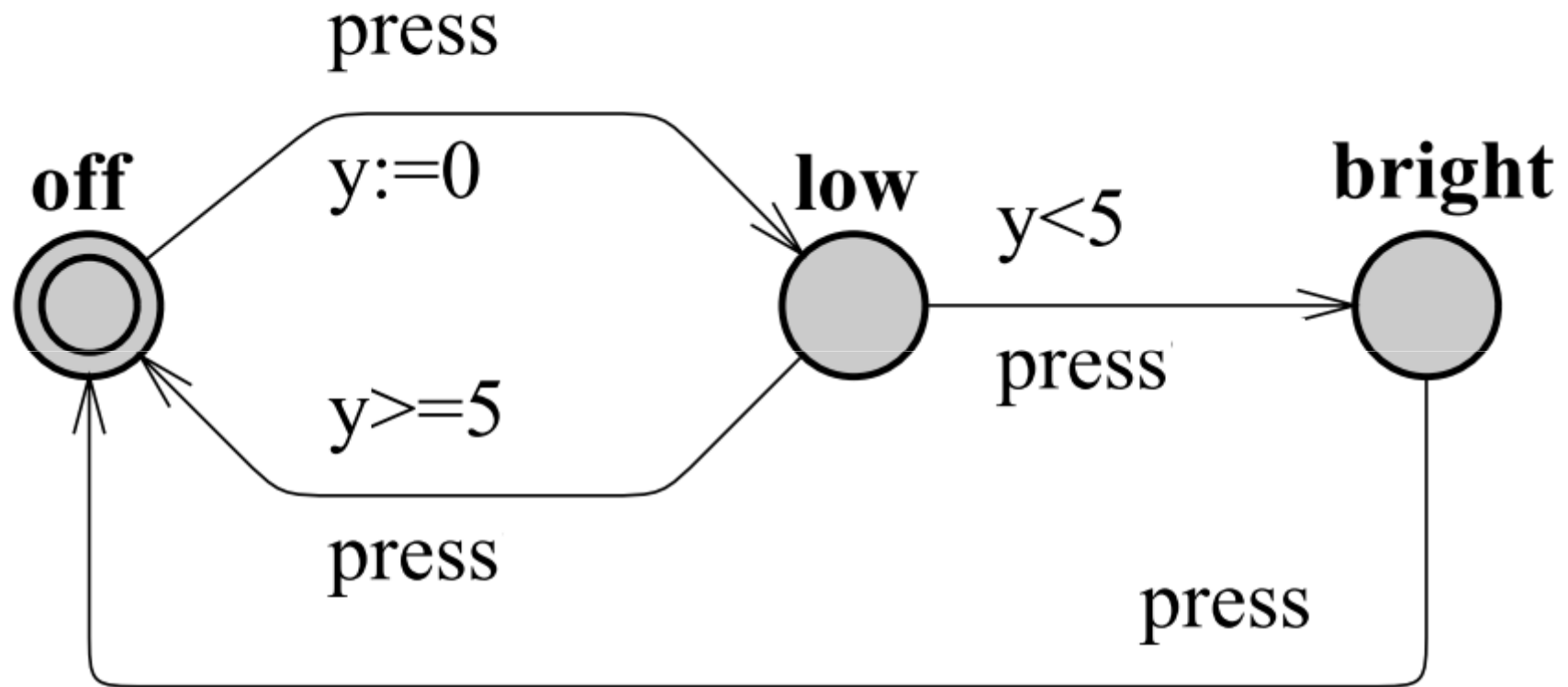
Autômatos finitos estendidos com *clocks/timers*
(relógios)

Assume-se um tempo contínuo - \mathfrak{R}^{+0}

(A mesma definição pode ser aplicada para
tempos discretos - \mathfrak{N})

Os clocks são incrementados de forma síncrona

Autômatos Temporizados



Autômatos Temporizados

$$G_t = (L, l_0, C, E, F, I)$$

- L - Conjunto de localizações
- C – Conjunto de *clocks*
- l_0 – Localização inicial
- E – Conjunto de eventos
- $F \subseteq L \times E \times CC_C \times \mathcal{P}(C) \times L$ – Conjunto de arestas entre localizações. $r \in \mathcal{P}(C)$ – os clocks a serem zerados
- $I : L \rightarrow CC_C$ - Mapeamento de invariantes a localizações

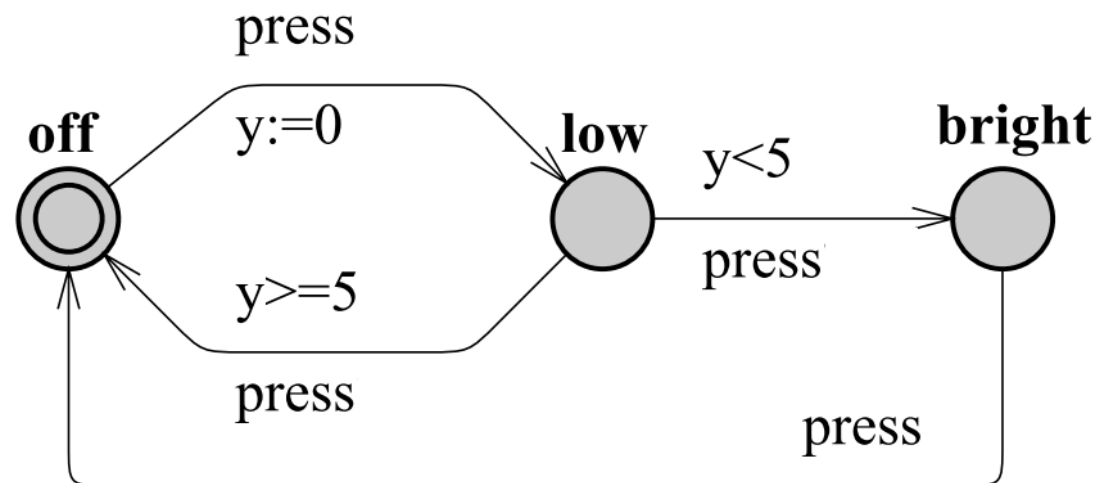
CC_C – Conjunto de restrições de *clock*

$$\phi \in CC_C, c \in \mathfrak{N}, x, y \in C$$

$$\phi ::= false \mid true \mid x \sim c \mid x - y \sim c \mid \phi \wedge \phi$$
$$\sim \in \{<, >, =, \leq, \geq\}$$

Autômatos Temporizados

- $L = \{\text{off}, \text{low}, \text{bright}\}$
- $C = \{y\}$
- $l_0 = \text{off}$
- $E = \{\text{press}\}$
- $F = \{(\text{off}, \text{press}, \text{true}, \{y\}, \text{low}), (\text{low}, \text{press}, y < 5, \{\}, \text{bright}), (\text{low}, \text{press}, y \geq 5, \{\}, \text{bright}), (\text{bright}, \text{press}, \text{true}, \{\}, \text{off})\}$
- $I = \{(\text{off}, \text{true}), (\text{low}, \text{true}), (\text{bright}, \text{true})\}$



Semântica

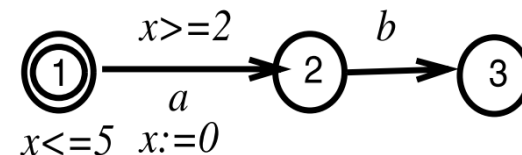
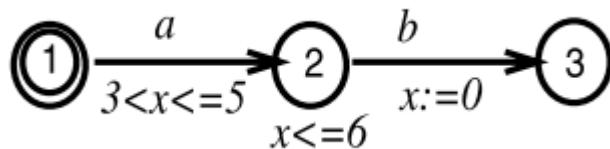
Invariantes definem o valor máximo que o tempo pode passar em uma determinada localização.

Uma localização não pode ser acessada caso o invariante seja falso (*strong invariant*)

Atingindo o valor máximo do *invariant*, a mudança torna-se urgente

Guarda na aresta indica quando a mesma estará habilitada

Não obriga a execução aresta (pode ficar permanentemente em uma determinada localização, considerando a violação da guarda)



Semântica

Adoção de um sistema transição temporizado (S, s_0, \rightarrow)

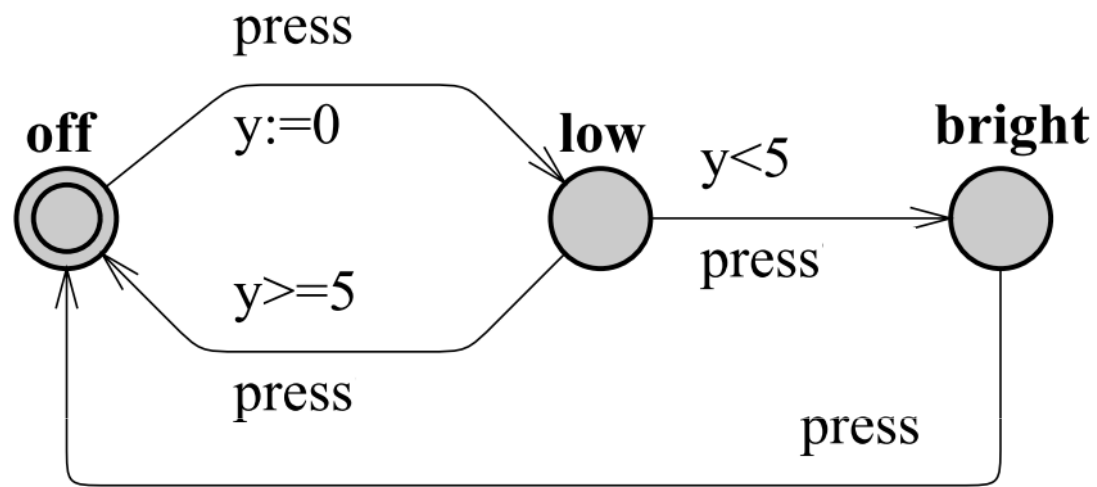
- $S \subseteq L \times \mathfrak{R}^C$ - Conjunto de estados
- $s_0 = (l_0, u_0)$ – Estado inicial
- $u : C \rightarrow \mathfrak{R}^{+0}; u_0(c) = 0, \forall c \in C; d \in \mathfrak{R}^{+0}$
 $[r \mapsto 0]u$, $\forall c \in r$, atribui o valor 0 e $\forall c \in C \setminus r$ não são afetados
- $\rightarrow \subseteq S \times \{\mathfrak{R}^{+0} \cup E\} \times S$, de tal forma que
 - $(l, u) \xrightarrow{d} (l, u + d)$ if $\forall d' : 0 \leq d' \leq d \implies u + d' \models I(l)$, and
 - $(l, u) \xrightarrow{a} (l', u')$ if there exists $e = (l, a, g, r, l') \in F$ s.t. $u \models g$,
 $u' = [r \mapsto 0]u$, and $u' \models I(l)$,

Strong Invariant

Execução – Sequência ρ . $s_i \in S, d_i \in \mathfrak{R}^{+0}, a \in E$

$$\rho \triangleq s_0 \xrightarrow{d_0} s_1 \xrightarrow{a_0} s_2 \xrightarrow{d_1} \dots$$

Semântica



$\rho = (\text{off}, u(y)=0) \xrightarrow{3} (\text{off}, u'(y)=3) \xrightarrow{\text{press}} (\text{low}, u''(y)=0) \xrightarrow{0.5} (\text{low}, u'''(y)=0.5) \dots$

Estados Simbólicos

O espaço de estados de um autômato temporizado é infinito. As ferramentas adotam estados simbólicos, tais como *Zones*. Um estado simbólico representa um nó no grafo de alcançabilidade

Estado Simbólico (l, Z)

- $l \in L$ – uma localização no autômato temporizado
- $Z \in CC_C$ – Zona, uma restrição de clock

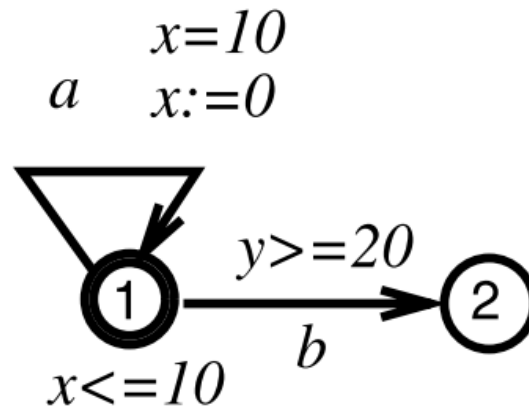
(l, Z) contempla o conjunto $\{[l, u] \mid u \models Z\}$.

Assuma a transição $t = l \xrightarrow{a, g, r} l'$ $Z_t \triangleq r(Z \wedge g) \wedge I(l')$

(l', Z_t) conjunto de todos os estados que podem ser alcançado por estados concreto em (l, Z) através da transição t

$$\forall s' \in (l', Z_t), \exists s \in (l, Z) . s \xrightarrow{a} s'$$

Estados Simbólicos

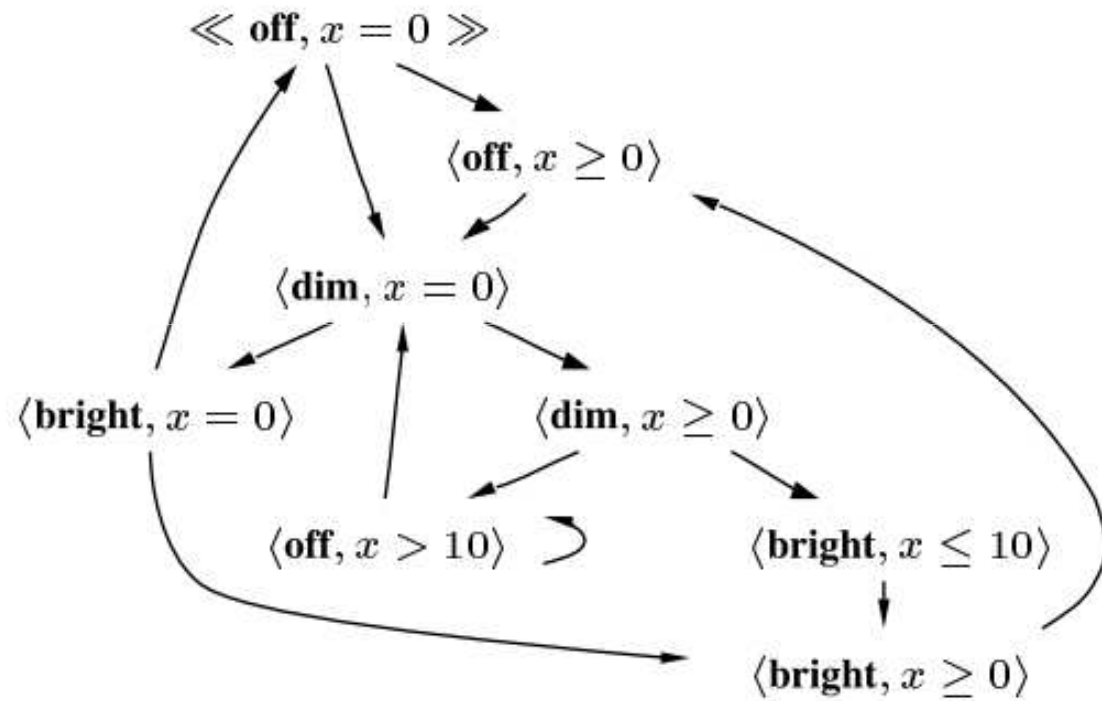
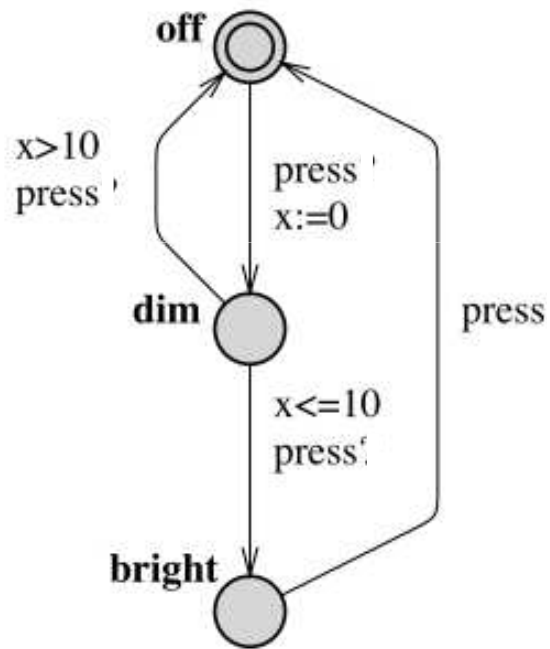


$$(1, Z_0) \xrightarrow{a} (1, Z_1)$$

$$Z_0 = x \leq 10 \wedge x = y$$

$$Z_1 = x \leq 10 \wedge y \leq 20 \wedge y - x = 10$$

Estados Simbólicos



Linguagem Temporizada

Considerando uma sequência ρ . $s_i \in S$, $d_i \in \mathfrak{R}^{+0}$, $a \in E$

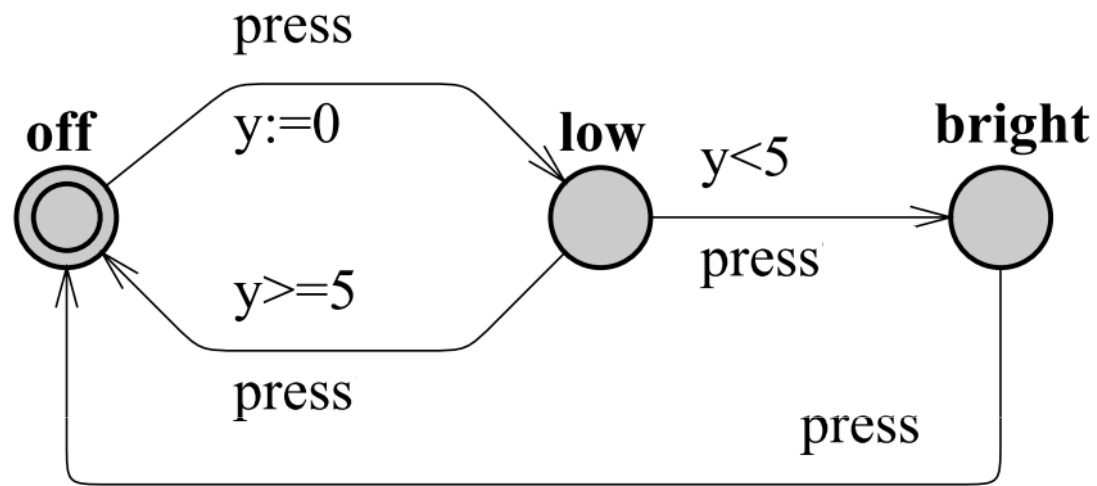
$$\rho \triangleq s_0 \xrightarrow{d_0} s_1 \xrightarrow{a_0} s_2 \xrightarrow{d_1} \dots$$

A respectiva palavra temporizada (*timed word*) é $(a_0, t_0) (a_1, t_1) (a_2, t_2) \dots$, no qual $t_0 = d_0$ $t_i = t_{i-1} + d_i$ $i \geq 1$

A linguagem temporizada $L_t(G_t)$ é o conjunto de todas palavras temporizadas.

Um autômato com localizações finais podem ser visto como aceitadores de linguagens temporizadas

Semântica



$\rho = (off, u(y)=0) \xrightarrow[3]{press} (off, u'(y)=3) \xrightarrow[0.5]{press} (low, u''(y)=0) \xrightarrow[press]{bright, u''''(y)=0.5} (bright, u''''(y)=0.5)$

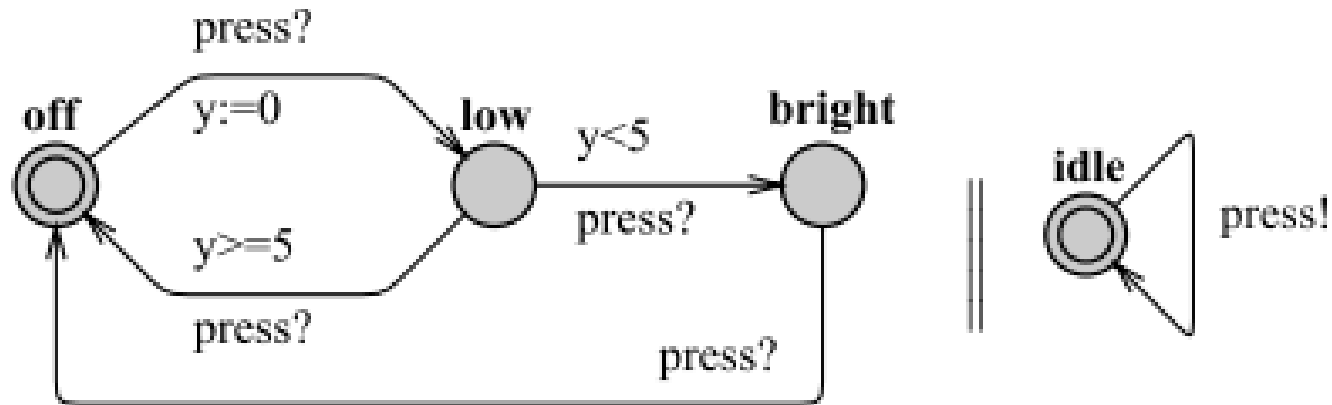
$w = (press, 3)(press, 3.5)$

Redes de Autômatos Temporizados

Uma rede de autômatos pode ser definida em termos de composição paralela

O operador é uma extensão do respectivos para autômatos não temporizados

Técnicas *On-the-fly* de geração do sistema de transição temporizado não necessitam da criação do autômato paralelo



(a) Lamp.

(b) User.

Composição Paralela

Composição paralela de rede de n autômatos $G_i = (L_i, l_i^0, C_i, E_i, F_i, I_i)$

$G = (L, l^0, C, E, F, I)$

- $l_0 = (l_1^0, \dots, l_n^0)$

- $L = \{l_0\} \cup \{\bar{l}' \mid \exists \bar{l} \in L, a, g, r . u \xrightarrow{a, g, r} \bar{l}'\}$

- $C = \bigcup_{i=1}^n C_i;$

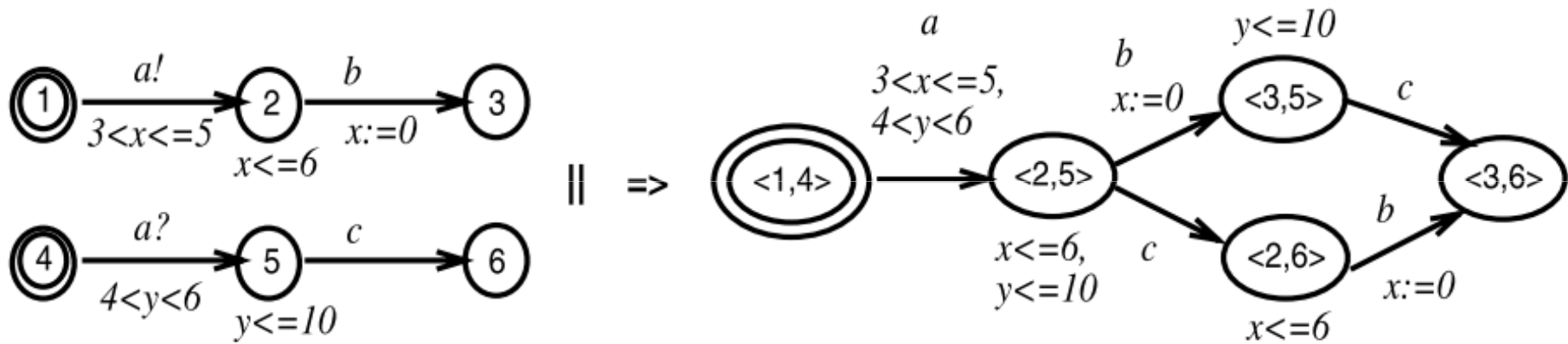
- $F : (1 \leq i \neq j \leq n)$

- $E = \bigcup_{i=1}^n E_i$

- $I(\bar{l}) = \bigwedge_i I_i(l_i)$

$$\left\{ \begin{array}{l} (P1) \frac{\bar{l}_i \xrightarrow{a?, g_i, r_i} l \quad \bar{l}_j \xrightarrow{a!, g_j, r_j} l'}{\bar{l} \xrightarrow{a, g_i \wedge g_j, r_i \cup r_j} \bar{l} [l \rightarrow i, l' \rightarrow j]} \\ (P2) \frac{\bar{l}_i \xrightarrow{a, g, r} l}{\bar{l} \xrightarrow{a, g, r} \bar{l} [l \rightarrow i]} \end{array} \right.$$

Redes de Autômatos Temporizados



Semântica

Rede de n autômatos $G_i = (L_i, l_i^0, C_i, E_i, F_i, I_i)$

Sistema transição temporizado (S, s_0, \rightarrow)

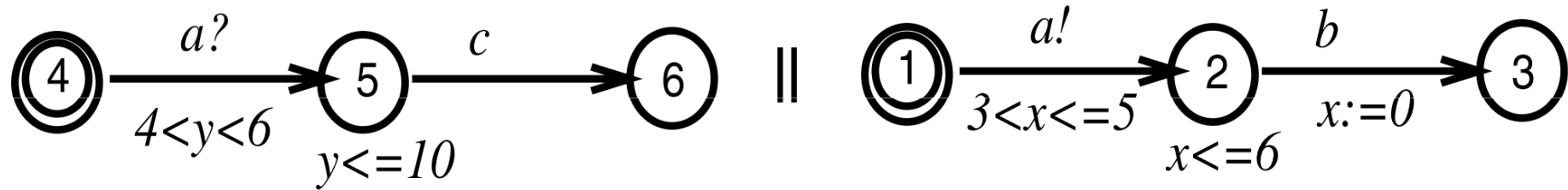
• $S \subseteq (L_1 \times \dots \times L_n) \times \mathfrak{R}^C$, $s_0 = (\bar{l}_0, u_0)$, $\bar{l}_0 = (l_1^0, \dots, l_n^0)$

$e \rightarrow \subseteq S \times S$, de tal forma que

- $(\bar{l}, u) \rightarrow (\bar{l}, u + d)$ if $\forall d' : 0 \leq d' \leq d \implies u + d' \models I(\bar{l})$.
- $(\bar{l}, u) \rightarrow (\bar{l}[l'_i/l_i], u')$ if there exists $l_i \xrightarrow{\tau g r} l'_i$ s.t. $u \models g$, $u' = [r \mapsto 0]u$ and $u' \models I(\bar{l})$.
- $(\bar{l}, u) \rightarrow (\bar{l}[l'_j/l_j, l'_i/l_i], u')$ if there exist $l_i \xrightarrow{c?g_i r_i} l'_i$ and $l_j \xrightarrow{c!g_j r_j} l'_j$ s.t. $u \models (g_i \wedge g_j)$, $u' = [r_i \cup r_j \mapsto 0]u$ and $u' \models I(\bar{l})$.

$u_0(c) = 0, \forall c \in C$

Semântica



Model-Checking

Adoção de lógica temporal para verificação de propriedades

CTL – Computation Tree Logic

- Modelo de tempo baseado em uma estrutura de árvore

Quantificadores de caminhos

- A : todos caminhos
- E: algum caminho

Operadores temporais

$$p \rightarrow q \equiv A[] (p \text{ imply } A\langle \rangle q)$$

- G ([]): todos estados
- F(⟨⟩): algum estado

UPPAAL

```
Prop ::= 'A[]' Expression | 'E⟨⟩' Expression | 'E[]'
        Expression | A⟨⟩ Expression | Expression -->
        Expression
```

Model-Checking

