

Álgebras de Processos Temporizadas

Álgebra de processos

Modelagem de sistemas através do uso de ações

Comportamento está associado às ações que um sistema pode executar (incluindo ordem e outros aspectos – tempo)

Ações são unidades atômicas de comportamento

Usualmente, assume-se que ações ocorrem separadamente no tempo.

Processo é a unidade básica de modularidade

- Representa um comportamento de um sistema

pbLOTOS

Language of Temporal Ordering Specification

Definido pela ISO (International Organization for Standardization) para especificação formal de sistema distribuídos

Composto por duas partes

- Álgebra de processo baseada em CSP e CCS
- Tipo de dados

pbLOTOS

Primitive Basic LOTOS

$S ::= B \mid B \text{ where } D$

$D ::= (P[x_1, \dots, x_n] := B) \mid (P[x_1, \dots, x_n] := B) D$

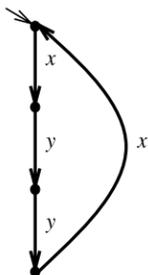
$B ::= \text{stop} \mid \text{exit} \mid a; B \mid B_1 \parallel B_2 \mid B_1 \llbracket x_1, \dots, x_n \rrbracket B_2 \mid B_1 \gg B_2 \mid \text{hide } x_1, \dots, x_n \text{ in } B \mid B[y_1/x_1, \dots, y_n/x_n] \mid P[x_1, \dots, x_n]$

$a \in \text{Act} \cup \{i\}; x_p, y_i \in \text{Act}; P \in \text{Pid} \text{ (conjunto de identificadores de processos)}$

pbLOTOS

$P [z, w] := z ; w ; P [w, z]$

$P [x, y]$



Operadores

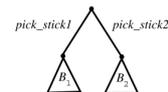
Prefixo

$a ; B$



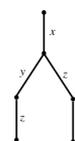
Escolha

$\text{pick stick1} ; B_1 \quad [] \quad \text{pick stick2} ; B_2$



Concorrência (Paralelismo)

$(x ; y ; \text{stop}) \quad || \quad [x] \quad (x ; z ; \text{stop})$



Extensões Temporais

Diversas extensões temporais propostas para LOTOS

Algumas extensões realizam o mapeamento para outros modelos

- Autômatos Temporizados
- Redes de Petri com Tempo (Time Petri Nets)

Apresentação de conceitos utilizando tLOTOS de Bowman e Gomez

tLOTOS

Adoção de tempo contínuo (\mathbb{R}^{+0})

Intervalos de tempo associados à ações observáveis (não urgentes) e internas (urgentes – limite superior)

- Ex: $a[u, u'] ; B$

Não persistente

- Passagem do tempo pode afetar uma ação de ser retraída

Operador de atraso

- $\text{wait}[t] B$

tLOTOS

$S ::= B \mid B \text{ where } D$

$D ::= (P := B) \mid (P := B) D$

$B ::= \text{stop} \mid \text{exit } I \mid \text{wait } [t] B \mid a I; B \mid B_1 \parallel B_2 \mid B_1 \parallel [G] B_2 \mid B_1 >> B_2 \mid \text{hide } G \text{ in } B \mid B[y_1/x_1, \dots, y_n/x_n] \mid P$

$a \in \text{Act} \cup \{\}; x_i, y_i \in \text{Act}; P \in \text{Pid} \text{ (conjunto de identificadores de processos)}$

- $a(t); B \triangleq a[t, \infty); B;$
- $x; B \triangleq x(0); B$ for $x \in \text{Act};$
- $a[t]; B \triangleq a[t, t]; B;$
- $i; B \triangleq i[0]; B;$
- $\text{exit}(t) \triangleq \text{exit}[t, \infty);$
- $\text{exit} \triangleq \text{exit}[0, \infty);$
- $\text{exit}[t] \triangleq \text{exit}[t, t];$ and
- $B \triangleright; B' \triangleq B \parallel (\text{wait}[t] i; B')$

Exemplos

$P := i[2, 10]; B$

$Q := (x[5, 10]; \text{stop}) \parallel [x] \parallel (x[11, 15]; \text{stop})$

$R := (x[5, 10]; \text{stop}) \parallel [x] \parallel (x[8, 15]; \text{stop})$

$S := i[5]; \text{stop} \parallel i[2, 4]; \text{stop}$

$T := (x[2]; B1) \parallel (y[4]; B2)$

Semântica Operacional

Adoção de um sistema de transição temporizado (S, TL, T, s_0)

- S – Conjunto não vazio de estados
- $TL \subseteq A \cup \mathfrak{R}^+$ – Conjunto de rótulos de transição
- $T \subseteq S \times \{v\} \times S, v \in TL$ – Conjunto de relações de transição
- s_0 – Estado inicial

Operações em Intervalos

$$I \oplus J = \{t_1 + t_2 \mid t_1 \in I \wedge t_2 \in J\}$$

$$I \ominus J = \{(t_1 - t_2) \in \mathbb{R}^{+0} \mid t_1 \in I \wedge t_2 \in J\}$$

$$I \oplus d = I \oplus [d, d]; I \ominus d = I \ominus [d, d]$$

$$I = [t, t']$$

- $\downarrow I = t; \uparrow I = t'$; Se $t' < t$, o intervalo é vazio \emptyset . $\downarrow \emptyset = \infty; \uparrow \emptyset = 0$
- $d > \uparrow I$, então $I \ominus d = \emptyset$

Semântica Operacional

Prefixo (Ações Observáveis)

$$(tAP.i) \frac{}{xI; B \xrightarrow{x} B} \quad (0 \in I) \quad (tAP.ii) \frac{}{xI; B \xrightarrow{x} x(I \ominus d); B}$$

Prefixo (Ações Internas)

$$(tIAP.i) \frac{}{iI; B \xrightarrow{i} B} \quad (0 \in I)$$

$$(tIAP.ii) \frac{}{iI; B \xrightarrow{i} i(I \ominus d); B} \quad (d \leq \uparrow I)$$

Operador de Atraso

$$(tD.i) \frac{B \xrightarrow{a} B'}{\text{wait}[0] B \xrightarrow{a} B'}$$

$$(tD.ii) \frac{}{\text{wait}[t] B \xrightarrow{a} \text{wait}[t-d] B} \quad (d \leq t)$$

$$(tD.iii) \frac{B \xrightarrow{a} B'}{\text{wait}[t] B \xrightarrow{a} B'}$$

Semântica Operacional

Composição Paralela

$$(tPA.i) \frac{B_1 \xrightarrow{a} B'_1}{B_1 \parallel [G] B_2 \xrightarrow{a} B'_1 \parallel [G] B_2} \quad (a \notin G \cup \{\delta\})$$

$$(tPA.ii) \frac{B_1 \xrightarrow{x} B'_1 \quad B_2 \xrightarrow{x} B'_2}{B_1 \parallel [G] B_2 \xrightarrow{x} B'_1 \parallel [G] B'_2} \quad (x \in G \cup \{\delta\})$$

$$(tPA.iii) \frac{B_1 \xrightarrow{d} B'_1 \quad B_2 \xrightarrow{d} B'_2}{B_1 \parallel [G] B_2 \xrightarrow{d} B'_1 \parallel [G] B'_2}$$

Tick

Adoção de um evento tick para representar passagem de tempo

Permite adoção da semântica não temporizada

```
SENDER := tick; preparePkg; tick;
tick; send; SENDER
```

RT-LOTOS

Adoção de tempo contínuo (\Re^{+0})

Operadores temporais:

delay(d) – atraso determinístico

latency(l) – atraso não-determinístico

a{t} – limita o tempo, no qual uma ação *a* observável é oferecida para o ambiente

delay(dmin,dmax) significa delay(dmin)latency(dmax-dmin)

RT-LOTOS

$$(2.a) \quad g\{u\}; P \xrightarrow{g} P \quad (g \in GV)$$

$$(2.b) \quad g\{u+t\}; P \xrightarrow{t} g\{u\}; P \quad (g \in GV)$$

$$(2.c) \quad g\{0\}; P \xrightarrow{t} stop \quad (g \in GV)$$

$$(10.a) \quad \frac{P \xrightarrow{g} P'}{\Delta^0 P \xrightarrow{g} P'} \quad (g \in GV^{i,\delta})$$

$$(10.b) \quad \Delta^{u+t} P \xrightarrow{t} \Delta^u P$$

$$(10.c) \quad \frac{P \xrightarrow{t} P'}{\Delta^0 P \xrightarrow{t} P'}$$

$$(11.a) \quad \frac{P \xrightarrow{g} P'}{\Omega^u P \xrightarrow{g} P'} \quad (g \in GV^\delta)$$

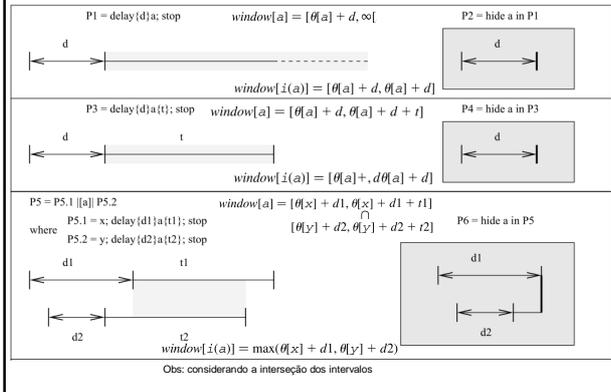
$$(11.d) \quad \frac{P \xrightarrow{t} P'}{\Omega^{u+t} P \xrightarrow{t} \Omega^u P}$$

$$(11.b) \quad \frac{P \xrightarrow{i} P'}{\Omega^u P \xrightarrow{i} P'}$$

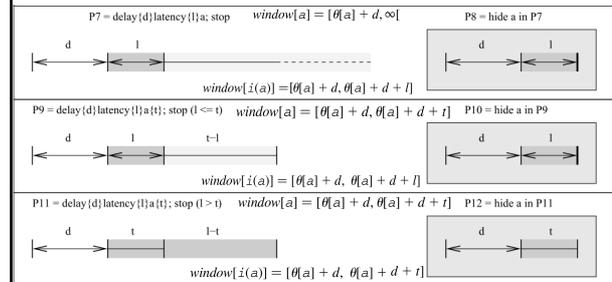
$$(11.e) \quad \frac{P \xrightarrow{t} P'}{\Omega^0 P \xrightarrow{t} P'}$$

$$(11.c) \quad \frac{P \xrightarrow{g} P'}{\Omega^0 P \xrightarrow{g} P'} \quad (g \in GV^{i,\delta})$$

RT-LOTOS



RT-LOTOS



RT-LOTOS

Modelos são avaliados através da tradução para autômatos temporizados ou redes de Petri com Tempo

- Autômatos temporizados
 - Uso de Autômatos temporizados dinâmicos
 - Associação de um *clock* para cada componente paralelo da especificação

RT-LOTOS

Autômato temporizado dinâmico (S, NClock, E, s_0)

S - conjunto finito de estados de controle

NClock : $S \rightarrow \mathbb{N}$ - mapeia a quantidade de clocks em cada estado de controle

E - conjunto de transições ($s, s', K, U, a, C, \theta$)

• s, s' - Estados de controle de origem e destino

• K, U - Condições temporizadas. Conjunção de desigualdades da forma $m \sim c_i \sim M$, $\sim \in \{<, \leq\}$ e $c_i \in C_{\text{set}}$

• a - o rótulo de uma ação

• $C \subseteq \{1, \dots, \text{NClock}(s')\}$ - conjunto de clocks a serem zerados quando a transição ocorrer

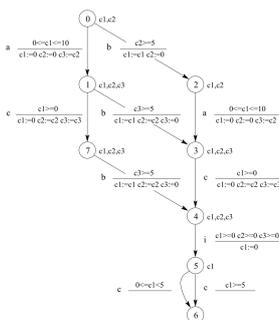
• $\theta : \mathbb{N} \rightarrow \mathbb{N}$ - função parcial de ajuste de relógio. $\text{Dom } \theta \subseteq \{1, \dots, \text{NClock}(s)\}$ and $\text{Im } \theta = \{1, \dots, \text{NClock}(s')\} - C$. $|\text{Dom } \theta| = |\text{Im } \theta|$, $|\text{Dom } \theta| \leq \text{NClock}(s)$, $|\text{Im } \theta| + |C| = \text{NClock}(s')$.

s_0 - Estado de controle inicial

RT-LOTOS

((a{10}; (c;exit)|||exit)) ||| (delay(5)b;exit)) >> latency(5)c{10};stop

$s_0 = (\frac{(a\{10\};(c;exit) \parallel | \parallel exit)}{c_1} \parallel \frac{delay(5)b;exit}{c_2}) \gg latency(10)d\{8\};stop$
 $s_1 = (\frac{(c;exit \parallel | \parallel exit)}{c_1} \parallel \frac{delay(5)b;exit}{c_2}) \gg latency(10)d\{8\};stop$
 $s_2 = (\frac{(a\{10\};(c;exit) \parallel | \parallel exit)}{c_1}) \gg latency(10)d\{8\};stop$
 $s_3 = (\frac{(c;exit \parallel | \parallel exit)}{c_1} \parallel \frac{exit}{c_3}) \gg latency(10)d\{8\};stop$
 $s_4 = (\frac{exit \parallel | \parallel exit}{c_1} \parallel \frac{exit}{c_2}) \gg latency(10)d\{8\};stop$
 $s_5 = latency(10)d\{8\};stop$
 $s_6 = stop$
 $s_7 = (\frac{exit \parallel | \parallel exit}{c_1} \parallel \frac{delay(5)b;exit}{c_2}) \gg latency(10)d\{8\};stop$



RT-LOTOS

Tradução RT-LOTOS para redes de Petri com tempo (time Petri net)

Adoção de blocos básicos denominados componentes

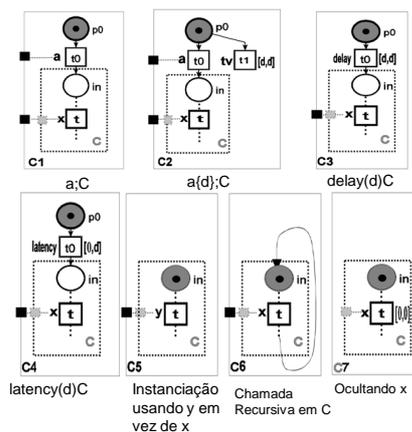
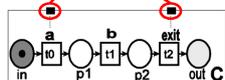
Bloco básico é uma TPN que descreve um comportamento

Disparo de uma transição corresponde a ocorrência de uma ação

Comunicação com o ambiente através de pontos de interação

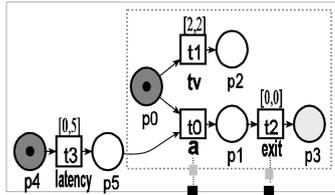
RT-LOTOS

Pontos de interação

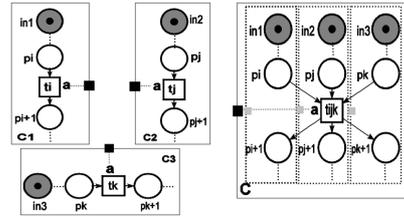


RT-LOTOS

Process P[a]:exit :=
latency(5) a{2}; exit
endproc

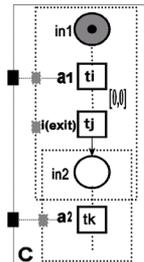
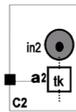
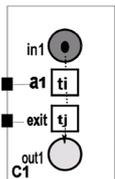


RT-LOTOS



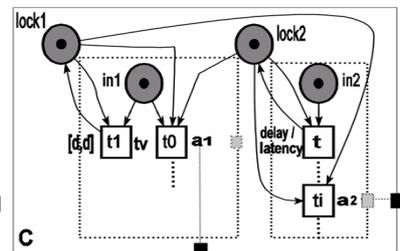
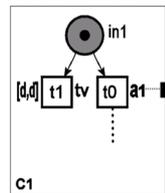
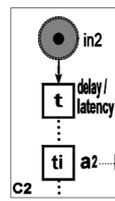
c := C1[a] C2[a] C3

RT-LOTOS



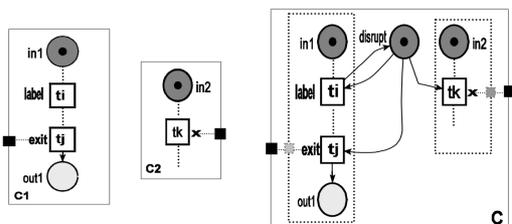
c := C1 >> C2

RT-LOTOS



C := C1 [] C2

RT-LOTOS



$C := C1 \triangleright C2$

Exemplo

Sistema de controle de trilhos
 -Trilho
 -Sensores controlam uma barreira

Sensor 1: quando acionado, fecha a barreira. Tempo: 8s a 16s

Sensor 2: quando acionado, levanta a barreira. Tempo: 10s a 20s

Trem: Após passar pelo sensor1, demora entre 15s e 20s para percorrer o trecho com cruzamento (enter)

Entre 10s e 15s, alcança o sensor 2

Em seguida, entre 100s e 150s, alcança novamente o sensor1

Specification Railway_Control_System : noexit

behaviour

```
hide sensor1, sensor2, enter, closed, open in
  Train[sensor1, enter, sensor2]
  Barrier[sensor1, closed, sensor2, open]
```

where

```
process Train[sensor1, enter, sensor2] : noexit :=
  sensor1; delay(15,20) enter;
  delay(10,15) sensor2;
  delay(100,150) Train[sensor1, enter, sensor2]
```

endproc

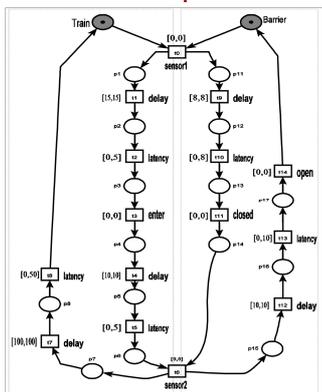
```
process Barrier[sensor1, closed, sensor2, open] : noexit :=
  sensor1; delay(8,16) closed;
  sensor2; delay(10,20) open;
  Barrier[sensor1, closed, sensor2, open]
```

endproc

endspec

$\text{delay}(d_{\min}, d_{\max}) \equiv$
 $\text{delay}(d_{\min}) \text{ latency}(d_{\max} - d_{\min})$

Exemplo



Exemplo

Specification Railway_Control_System : noexit

behaviour

```
hide sensor1, sensor2, enter, closed, open in
  Train[sensor1, enter, sensor2]
  Barrier[sensor1, closed, sensor2, open]
```

where

```
process Train[sensor1, enter, sensor2] : noexit :=
  sensor1; delay(15,20) enter;
  delay(10,15) sensor2;
  delay(100,150) Train[sensor1, enter, sensor2]
```

endproc

```
process Barrier[sensor1, closed, sensor2, open] : noexit :=
  sensor1; delay(8,16) closed;
  sensor2; delay(10,20) open;
  Barrier[sensor1, closed, sensor2, open]
```

endproc

endspec

