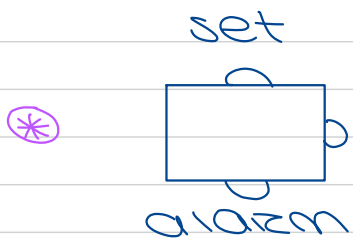
AULAS TEÓRICAS

IC

SISTEMAS REATIVOS



(*) → pontos de interação com o ambiente

Álgebras de processos

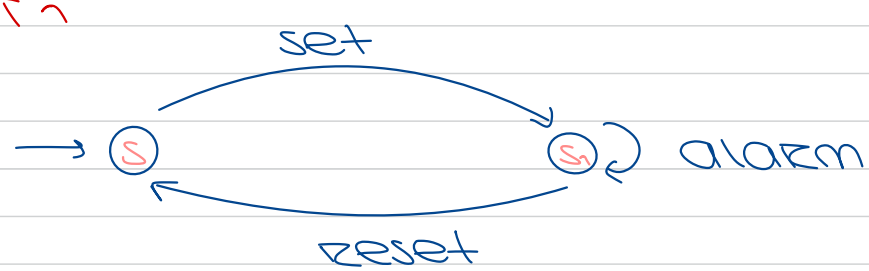
CCS → para representar processos
CTS → autómatos
HML → lógica

Labelled transition system (LTS)

$A = (S, Act, \rightarrow, s_i, T)$

S - set of states
Act - set of actions
→ - transition relation $\rightarrow \subseteq S \times Act \times C$
s_i - estado inicial
T - estado final ou de terminação

Autómato que representa o sistema anterior: (*)
LST 1



Não há sobreposição de ações!

$S = \{s, s_1\}$

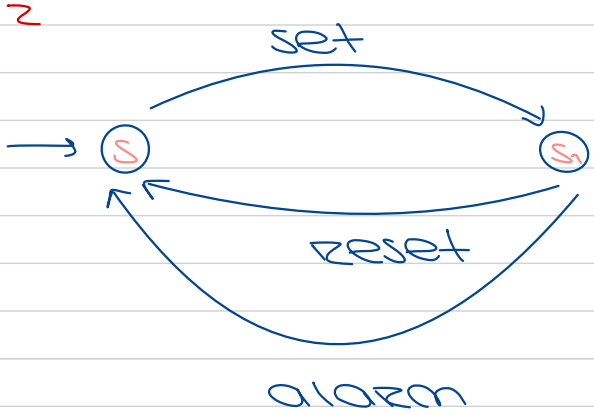
$Act = \{set, reset, alarm\}$

$\rightarrow_1 = \{(s, set, s_1), (s_1, alarm, s_1), (s_1, reset, s)\}$

$s_i = s$

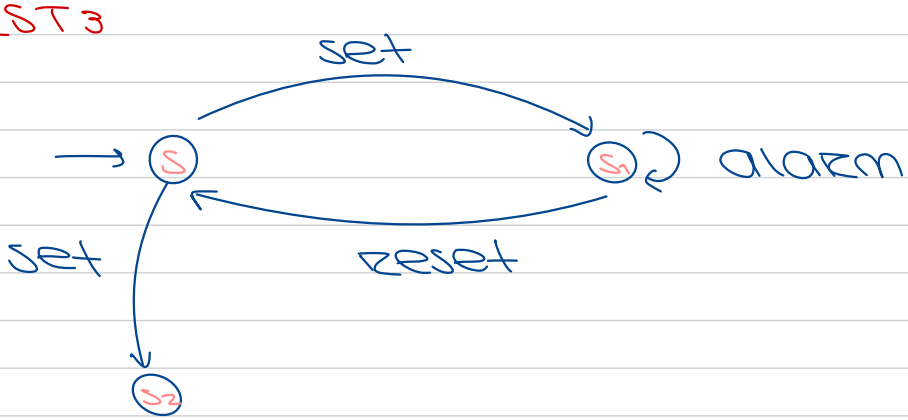
$\tau = \{\} \rightarrow$ não há

LST2



$\rightarrow_2 = \{(s, set, s_1), (s_1, reset, s), (s_1, alarm, s)\}$

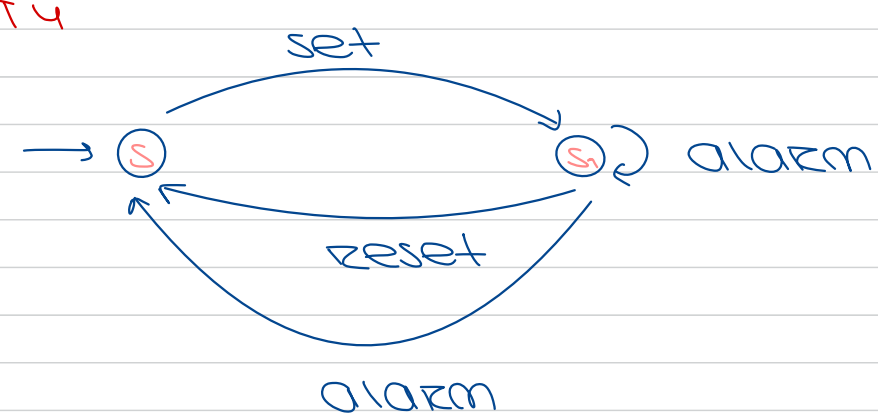
LST3



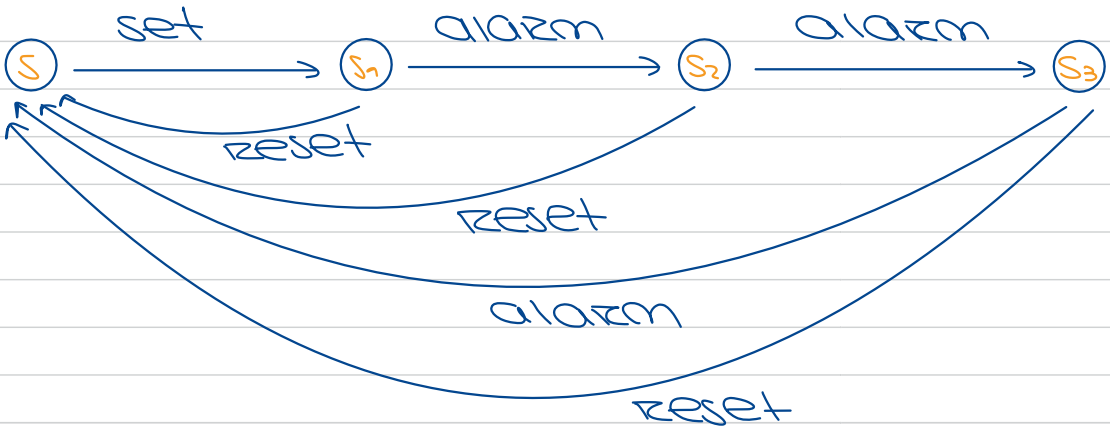
Não determinista (deadlock)

$\rightarrow_3 = \rightarrow_1 \cup \{(s, set, s_2)\}$

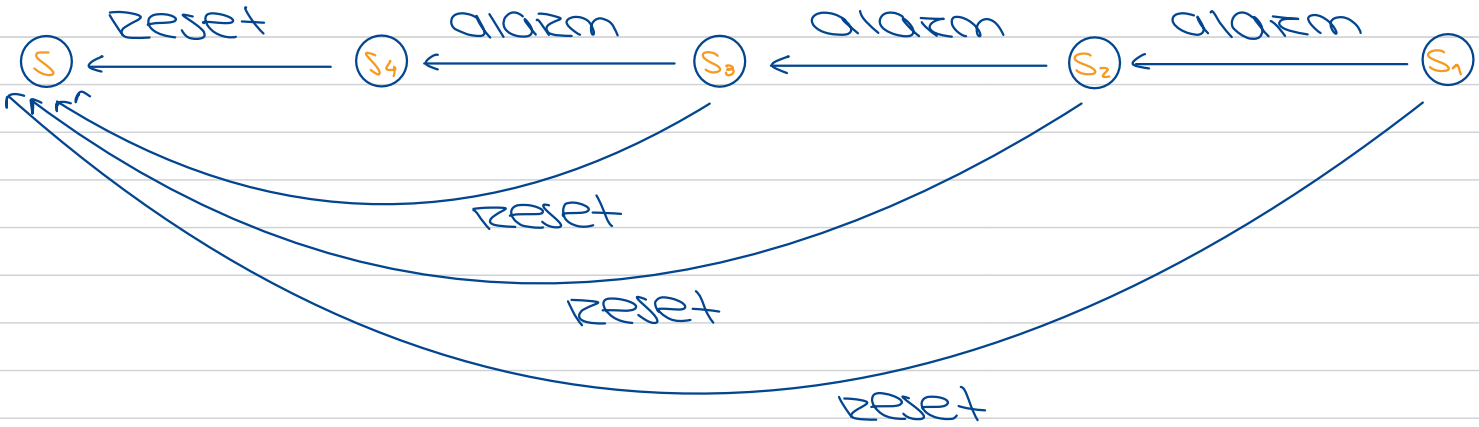
LST4

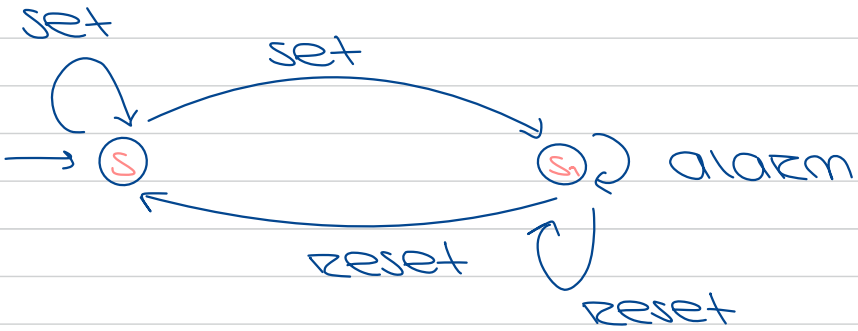


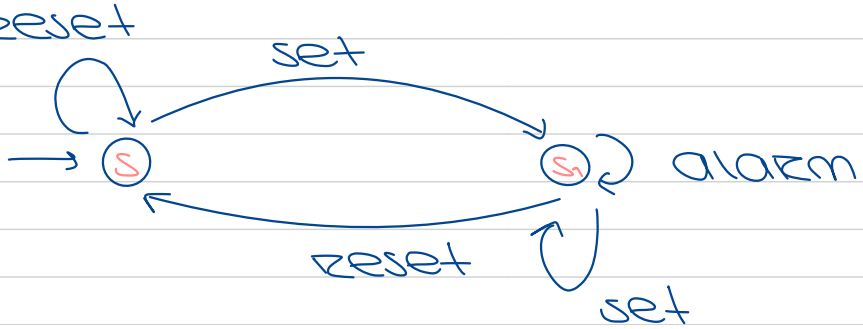
Mostrar se os sistemas são diferentes ou não

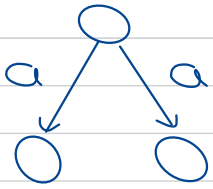


(verificar se são equivalentes)









Conceito de Trace:

Dado um LST, $A = (S, Act, \rightarrow, s_i, s_i, \tau)$, chamemos Trace ao conj. de caminhos que partam de t.

traces(t).

- $\varepsilon \in traces(t)$
- se $s \xrightarrow{a} s'$
  - $a \in traces(s')$
  - $a\sigma \in traces(s)$

traces(t) = traces(s)

↳ quando os caminhos são iguais a partir do estado inicial

Olhando para os exemplos anteriores:

LST1: $traces(s) = \{\varepsilon, set, set \cdot alarm, set \cdot reset, set \cdot alarm \cdot alarm, \dots\}$

LST2: $traces(s) = \{\varepsilon, set, set \cdot reset, set \cdot alarm, reset \cdot set, \dots\}$

LST1 equiv LST2?

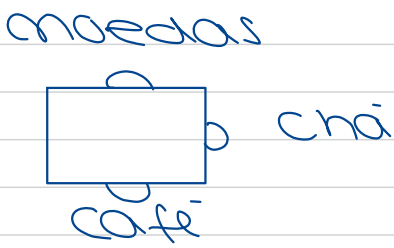
Não, há caminhos possíveis em LST2 que não são possíveis em LST1:

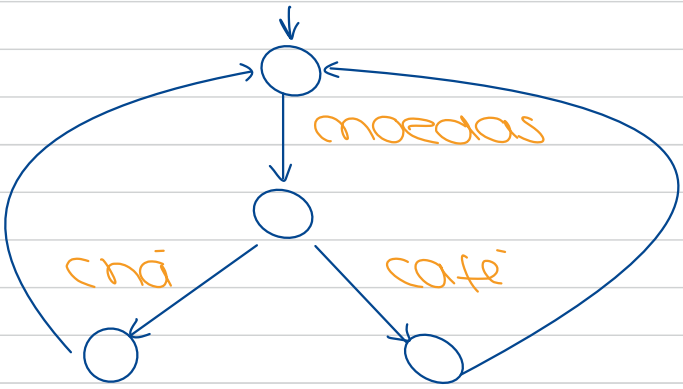
set · alarm · reset (LST2)

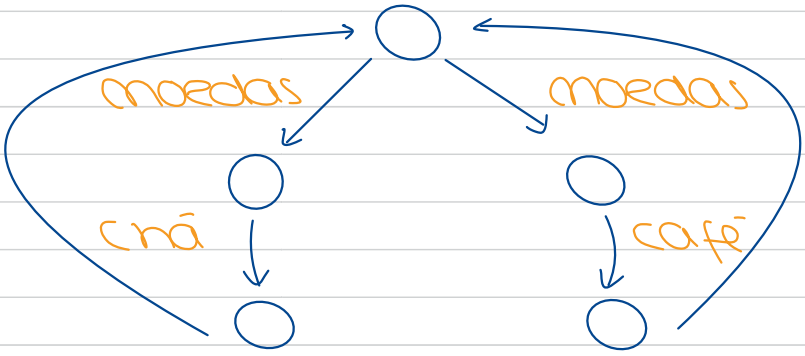
set · alarm · set (LST1)

obrigatoriamente

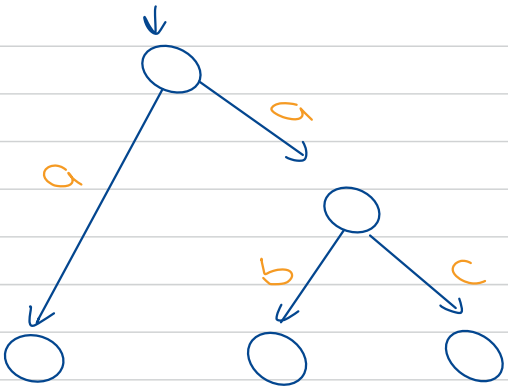
Exemplo máquina café

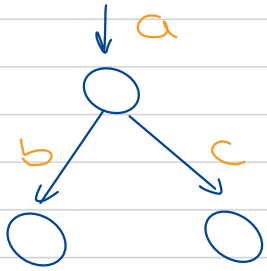






Outro tipo de sistemas





Distinguir sistemas como estes:

$A = (S, Act, \rightarrow, s_i, T)$

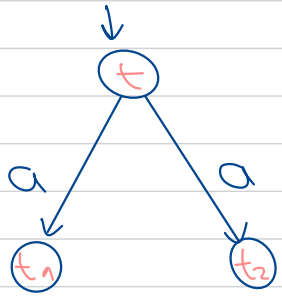
$R \subseteq S \times S$ is strong bissimulation (equivalência)

sse $\forall s,t \in S$, if sRt then $(s,t) \in R$

- if $s \xrightarrow{a} s'$ então existe $t' \in S$ tal que $t \xrightarrow{a} t'$ com $(s',t') \in R$

$t \xrightarrow{a} t'$, $s \xrightarrow{a} s'$, $(s',t') \in R$

Outro sistema:



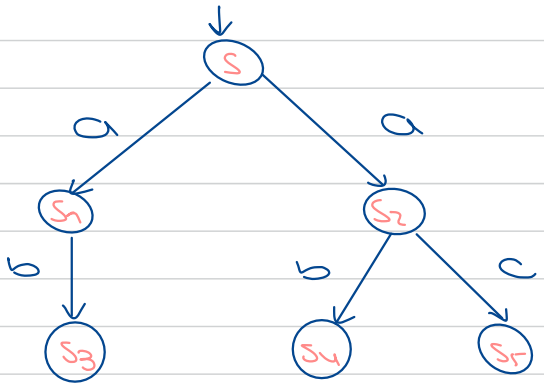
equiv.

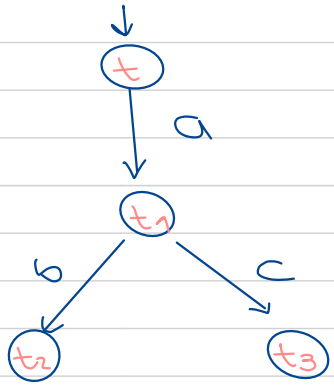
$(s,t) \in R$?

$s \xrightarrow{a} s_1$	$t \xrightarrow{a} t_1$	$(s_1, t_1) \in R$
$t \xrightarrow{a} t_1$	$s \xrightarrow{a} s_1$	$(s_1, t_1) \in R$
$t \xrightarrow{a} t_2$	$s \xrightarrow{a} s_1$	$(s_1, t_2) \in R$

$R = \{(s,t), (s_1,t_1), (s_1,t_2)\}$

Exemplo:





NÃO equiv.

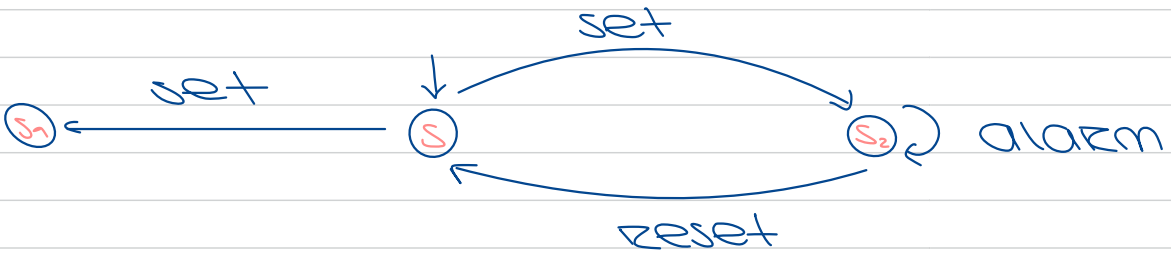
$(s,t) \in R$?

$s \xrightarrow{a} s_1$	$t \xrightarrow{a} t_1$	$(s_1, t_1) \overset{?}{\in} R$
$s \xrightarrow{a} s_2$	$t \xrightarrow{a} t_1$	$(s_1, t_1) \overset{?}{\in} R$
$t \xrightarrow{a} t_1$	$s \xrightarrow{a} s_1$	$(s_1, t_1) \overset{?}{\in} R$

$(s_1, t_1) \notin R$

$t_1 \xrightarrow{c} t_3$ $\quad$ $s_1 \xrightarrow{c} \times$

Outro sistema:

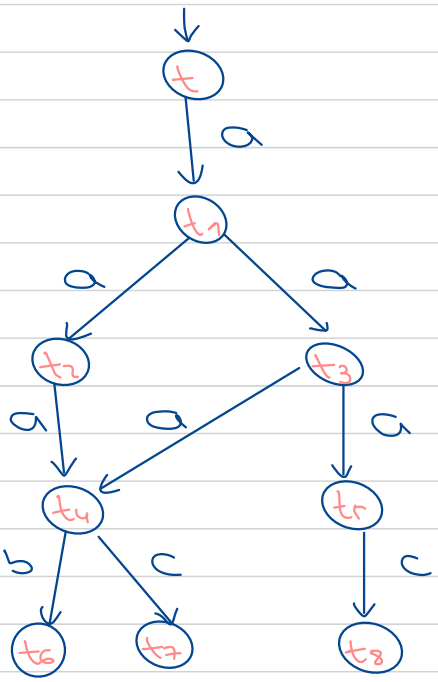


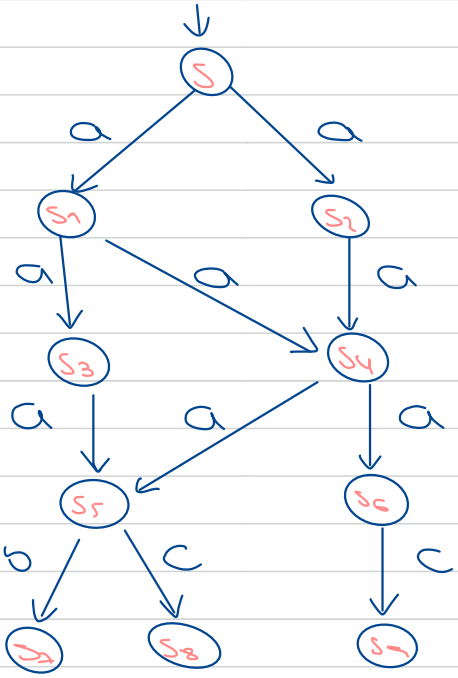
$s \xrightarrow{set} s_1$

$t \xrightarrow{set} t_1$

$\Big\} \; (s_1, t_1) \in R$

Exercício:





Trace equivalent? Sim, traces

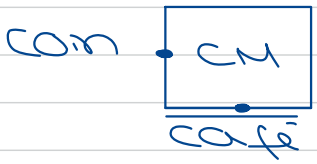
Bissimilar? (Atribuir estados e verificar se $(s,t) \in R$??

traces = {ε, a, aa, aaa, acab, aaac}

Não será aaab?

Processos

Até agora os estados eram abstratos. Vamos substituir por processos.

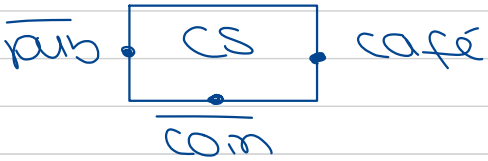


$CM = com \cdot \overline{café} \cdot CM$

ação complementar

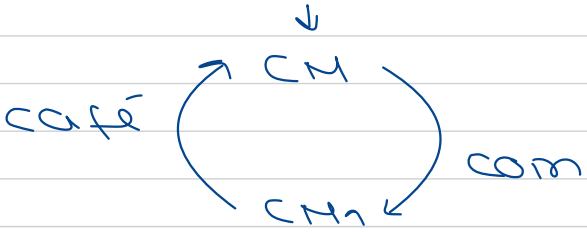
máquina de café

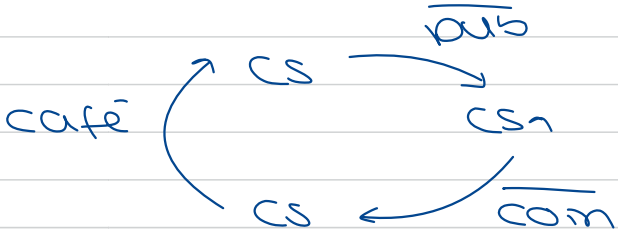
Para publicar precisa de um café



$CS = \overline{pub} \cdot \overline{com} \cdot café \cdot CS$

$CM = com \cdot CM_1$
$CM_1 = \overline{café} \cdot CM$





$LTS = \{P, A, S_i, \rightarrow\}$

$LTS_2 = \ldots$

$P = \{CM, CM_1\}$

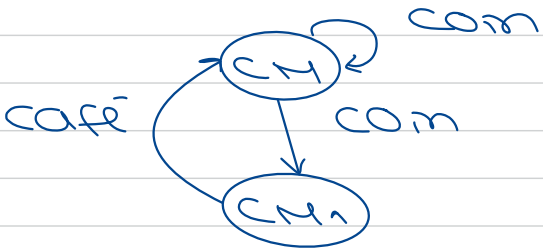
$A = \{com, \overline{café}\}$

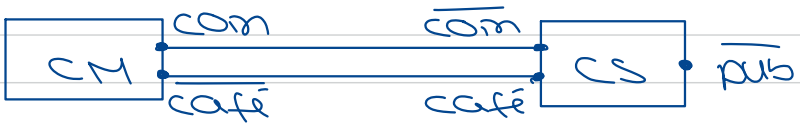
$S_i = CM$

$\xrightarrow{com} = \{(CM, CM_1)\}$

$\xrightarrow{\overline{café}} = \{(CM_1, CM)\}$

$CM = com \cdot (café \cdot CM + CM)$





$S = (CM \mid CS) \setminus com \setminus café$ → ações que não são visíveis a partir do exterior

$CM = com \cdot \overline{café} \cdot CM$ $\quad CM = VM[café/item]$

$TM = com \cdot \overline{tea} \cdot TM$ $\quad TM = VM[tea/item]$

$VM = com \cdot \overline{item} \cdot VM$

A - ações

$\overline{A} = \{\overline{a} : a \in A\}$

$K \leftarrow$ Processo comes

$Act = A \cup \overline{A} \cup \{\tau\}$

$P, Q ::= K \mid \alpha.P \mid P \mid Q \mid P \setminus L \mid P[f] \mid \sum_{i \in I} P_i$ ①

$L \subseteq A \cup \overline{A}$

$f(\tau) = \tau$
$f(\overline{a}) = \overline{f(a)}$

$\alpha.P \xrightarrow{\alpha} P$ $\quad CM = com \cdot CM_1$

$CM \xrightarrow{com} CM_1$

Regras Estruturais:

$$\frac{P_j \xrightarrow{a} P'_j}{\sum_{i \in I} P_i \xrightarrow{a} P'_j}$$

ver

Não determinístico

Ações em paralelo

$$\frac{P \Downarrow v \quad P' \Downarrow v'}{P \mid Q \Downarrow v \mid v'}$$

$$\frac{Q \Downarrow v \quad Q' \Downarrow v'}{P \mid Q \to P \mid Q'}$$

$$\frac{P \Downarrow v \quad P' \Downarrow v' \quad Q \Downarrow \overline{a} \quad Q' \Downarrow a}{P \mid Q \to P' \mid Q'}$$

Ações que não são internas

$$\frac{P \Downarrow v \quad P' \Downarrow v'}{P + Q \Downarrow v \quad P' \Downarrow v'} \quad a \neq \tau$$

$$\frac{P \Downarrow v \quad P' \Downarrow v'}{P[f] \Downarrow f(v) \quad P'[f]}$$

$$\frac{P \Downarrow v \quad P' \Downarrow v'}{X \Downarrow v} \quad X \stackrel{def}{=} P$$

Definição CCS

Programa
→ é uma série de definições de processos.

ver Definition CCS

Processos → maiúsculas
Ações → minúsculas

$a.(b.A)+B$ → dá um processo (válido)
↳ prefixo do processo A

$(a.Nil + \bar{a}.A) \setminus \{a, b\}$ (válido)
↳ ações não visíveis

$(a.Nil + \bar{a}.A) \setminus \{a, \tau\}$ (não válido)
↳ não pode aparecer aqui
↳ ação interna (ocorrem mas não são visíveis do exterior)

$a.B + [a/b]$ (não válido)

$(\tau.(\tau.B)) + Nil$ (válido)
↳ prefixo de um processo
↳ prefixo de um processo

$(a.B + b.B)[a/b, b/a]$ (válido)

verificar se deste lado temos um processo

$(a.B + b.B)[a/\tau, b/a]$ (não válido)
não posso substituir τ por cá outra vez.

verificar se deste lado temos um processo

$(a.b.A + \bar{a}.Nil) \setminus B$ (válido)

processo processo

$(a.b.A + \bar{a}.Nil).B$ (não válido)

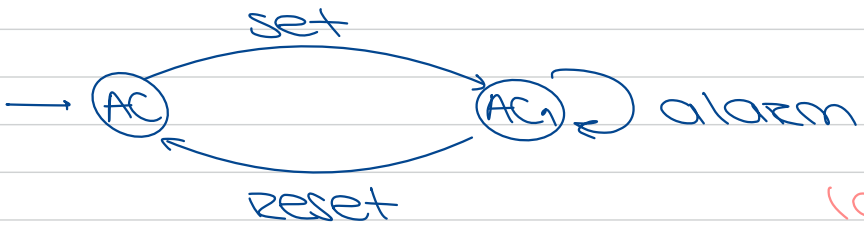
aqui temos que ter ações

$(Nil \mid Nil) + Nil$ (válido)
↳ a.P

Semântica CCS

→ ações são axiomas

Exemplo aplicação das regras



(a) $AC \xrightarrow{set} AC_1$

(b) $AC_1 \xrightarrow{alarm} AC_1$

(c) $AC_1 \xrightarrow{reset} AC$

LST

$$\begin{cases} AC \overset{def}{=} set \cdot AC_1 \\ AC_1 \overset{def}{=} alarm \cdot AC_1 + reset \cdot AC \end{cases}$$

(a)

$$\frac{\dfrac{}{set \cdot AC_1 \xrightarrow{set} AC_1}\ Act}{AC \xrightarrow{set} AC_1}\ CON \quad AC \overset{def}{=} set \cdot AC_1$$

(b)

$$\frac{\dfrac{}{alarm \cdot AC_1 \xrightarrow{alarm} AC}\ Act}{(alarm \cdot AC_1 + reset \cdot AC) \xrightarrow{alarm} AC_1}\ SUM_j$$

$$\frac{}{AC_1 \xrightarrow{alarm} AC_1}\ CON \quad AC_1 \overset{def}{=} alarm \cdot AC_1 + reset \cdot AC$$

(c)

$$\frac{\dfrac{}{alarm \cdot AC_1 \xrightarrow{reset} AC}\ Act}{(alarm \cdot AC_1 + reset \cdot AC) \xrightarrow{reset} AC}\ SUM_j$$

$$\frac{}{AC_1 \xrightarrow{reset} AC}\ CON \quad AC_1 \overset{def}{=} alarm \cdot AC_1 + reset \cdot AC$$

$$\dfrac{\dfrac{}{a.Nil \xrightarrow{a} Nil}\ ACT \qquad \dfrac{}{\overline{a}.Nil \xrightarrow{\overline{a}} Nil}\ ACT}{(a.Nil \mid \overline{a}.Nil) \xrightarrow{\tau} (Nil \mid Nil)}\ COM_3$$

Exercício 1:

$A \overset{def}{=} a\,A$ → PROCESSO

$$\dfrac{}{a.A \xrightarrow{a} A}\ ACT$$

$$\dfrac{a.A \xrightarrow{a} A}{A \xrightarrow{a} A}\ CON\ A \overset{def}{=} a.A$$

$$\dfrac{A \xrightarrow{a} A}{(A \mid \overline{a}.Nil) \xrightarrow{a} (A \mid \overline{a}.Nil)}\ COM_1$$

$$\dfrac{(A \mid \overline{a}.Nil) \xrightarrow{a} (A \mid \overline{a}.Nil)}{((A \mid \overline{a}.Nil) \mid b.Nil) \xrightarrow{a} ((A \mid \overline{a}.Nil) \mid b.Nil)}\ COM_1$$

$$\dfrac{((A \mid \overline{a}.Nil) \mid b.Nil) \xrightarrow{a} ((A \mid \overline{a}.Nil) \mid b.Nil)}{((A \mid \overline{a}.Nil) \mid b.Nil)[c/a] \xrightarrow{c} ((A \mid \overline{a}.Nil) \mid b.Nil)[c/a]}\ REL$$

Exercício 2: $A \overset{def.}{=} b.a.B$

$$\dfrac{}{\overline{b}.(a\,B) \xrightarrow{\overline{b}} (a\,B)}\ ACT$$

$$\dfrac{\overline{b}.(a\,B) \xrightarrow{\overline{b}} (a\,B)}{A \mid \overline{b}.a.B \xrightarrow{\overline{b}} (A \mid a.B)}\ COM_2$$

$$\dfrac{A \mid \overline{b}.a.B \xrightarrow{\overline{b}} (A \mid a.B)}{(A \mid \overline{b}.a.B) + (\overline{b}.A)[a/b] \xrightarrow{\overline{b}} (A \mid a.B)}\ SUM_1$$

Exercício 3:

$$\frac{}{b \cdot (a \cdot B) \xrightarrow{b} a \cdot B}\ ACT$$

$$\frac{}{A \xrightarrow{b} a \cdot B}\ CON_{(1)} \qquad \frac{}{\overline{b} \cdot Nil \xrightarrow{\overline{b}} Nil}\ ACT$$

$$\frac{}{(A \mid \overline{b} \cdot Nil) \xrightarrow{\tau} (a \cdot B \mid Nil)}\ COM_3$$

$$\frac{}{(A \mid \overline{b} \cdot Nil) \setminus \{b\} \xrightarrow{\tau} (a \cdot B \mid Nil) \setminus \{b\}}\ RES$$

(1) $A \overset{def}{=} b \cdot a \cdot B$

Exercício 4: $A \overset{def}{=} b \cdot a \cdot B$

$$\frac{}{\overline{a} \cdot A[a/b] \xrightarrow{\overline{a}} A[a/b]}\ ACT$$

$$\frac{}{(\overline{b} \cdot A)[a/b] \xrightarrow{\overline{a}} A[a/b]}\ REL$$

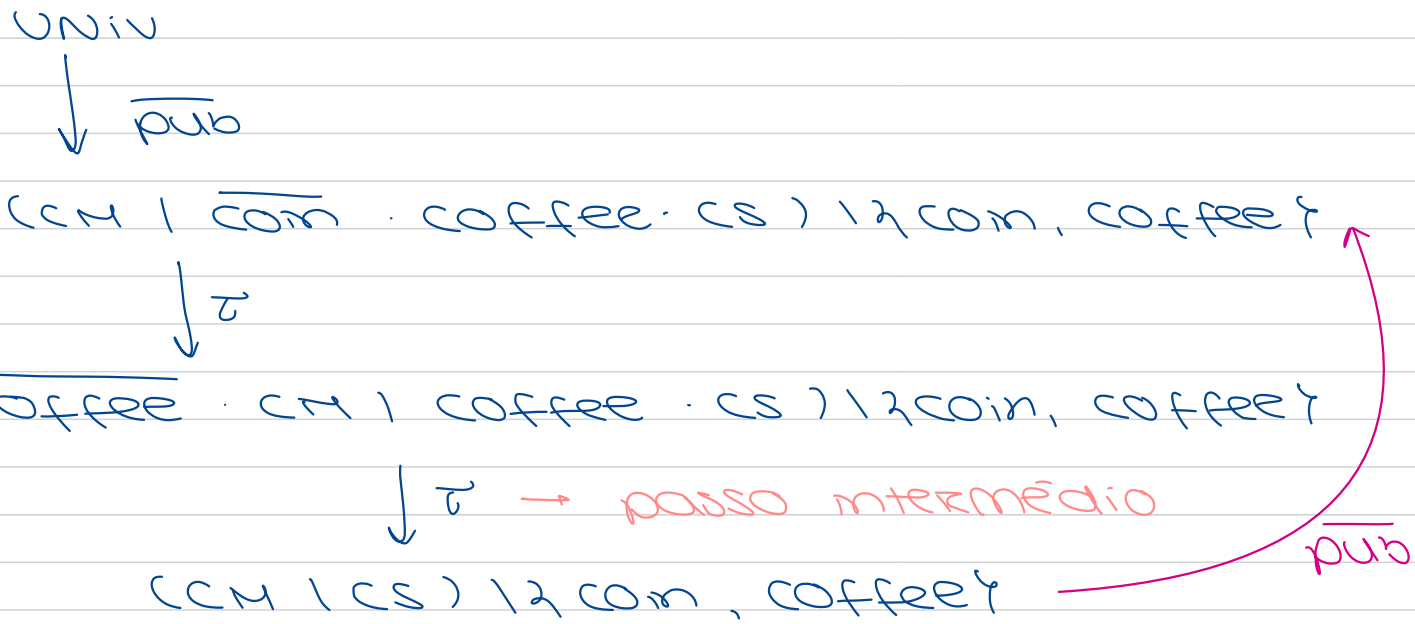
$$\frac{}{(A \mid (\overline{b} \cdot a \cdot B)) + (\overline{b} \cdot A)[a/b] \xrightarrow{\overline{a}} A[a/b]}\ SUM_2$$

Exercício 5:

$CM \overset{def}{=} \overline{com} \cdot \overline{coffee} \cdot CM$ → volta a estar operacional

$CS \overset{def}{=} \overline{pub} \cdot \overline{com} \cdot coffee \cdot CS$

$Univ \overset{def}{=} (CM \mid CS) \setminus \{com, coffee\}$



ACT

$$\overline{pub} \cdot (\overline{com} \cdot coffee \cdot cs) \xrightarrow{\overline{pub}} (com \cdot coffee \cdot cs)$$

COM2

$$(cm \mid \overline{pub} \cdot \overline{com} \cdot coffee \cdot cs) \setminus \{\ldots\} \xrightarrow{\overline{pub}} (cm \mid \overline{com} \cdot coffee \cdot cs) \setminus \{\ldots\}$$

CON (2)

$$(cm \mid cs) \setminus \{\ldots\} \xrightarrow{\overline{pub}} (cm \mid \overline{com} \cdot coffee \cdot cs) \setminus \{\ldots\}$$

CON (1)

$$univ \xrightarrow{\overline{pub}} (cm \mid \overline{com} \cdot coffee \cdot cs) \setminus \{\ldots\}$$

(1) $univ \stackrel{def}{=} (cm \mid cs)$

(2) $cs \stackrel{def}{=} \ldots$

(REVER)

Exemplo value passing CCS

$Bank(Total) = save(x).\ Bank(Total + x)$

$\overline{pay(6)}.\ Nil \mid pay(x).\ \overline{save(x+2)}.\ Nil \mid Bank(100)$

$\downarrow \tau$

$Nil \mid \overline{save(3)}.\ Nil \mid Bank(100)$

$\downarrow \tau$

$Nil \mid Nil \mid Bank(103)$

Relação de Transição Fraca

$$\stackrel{a}{\Rightarrow} = \begin{cases} (\xrightarrow{\tau})^* \circ \xrightarrow{a} \circ (\xrightarrow{\tau})^* & \text{if } a \neq \tau \\ (\xrightarrow{\tau})^* & \text{if } a = \tau \end{cases}$$

existe pelo menos uma transição por a.

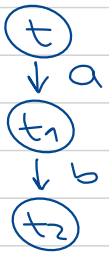
Bissimulação Fraca

se $s \xrightarrow{a} s'$ então $t \stackrel{a}{\Rightarrow} t'$ $\quad (s', t') \in R$

se $t \xrightarrow{a} t'$ então $s \stackrel{a}{\Rightarrow} s'$ $\quad (s', t') \in R$

⇓ transição fraca

Exercício:



são fracamente bissimilares

$s \mathcal{R} t$

$s \approx t$

$R = \{(s,t), (s_1,t_1), (s_2,t_1)\}$

$s \xrightarrow{a} s_1$ $\quad t \stackrel{a}{\Rightarrow} t_1$ $\quad (s_1,t_1) \in R$

$t \xrightarrow{a} t_1$ $\quad s \stackrel{a}{\Rightarrow} s_1$

(s_1, t_1)

$s_1 \xrightarrow{\tau} s_2$ $\quad t_1 \stackrel{a}{\Rightarrow} t_1$ $\quad (s_2,t_1) \in R$

(s_2, t_1)

$s_2 \xrightarrow{b} s_3$ $\quad t_1 \stackrel{b}{\Rightarrow} t_2$ $\quad (s_3,t_2) \in R$

≈ $\quad t \xrightarrow{a} t_1$

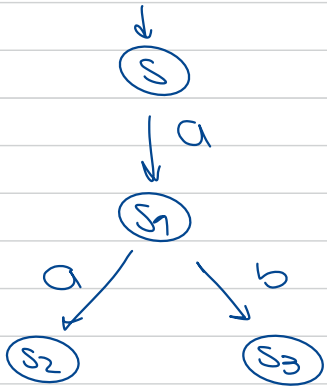
$s \xrightarrow{\tau} s \quad t \xrightarrow{\tau} t \quad (s,t)$

$s \xrightarrow{a} s_1 \quad t \xrightarrow{a} t_1 \quad (s_1,t_1)$

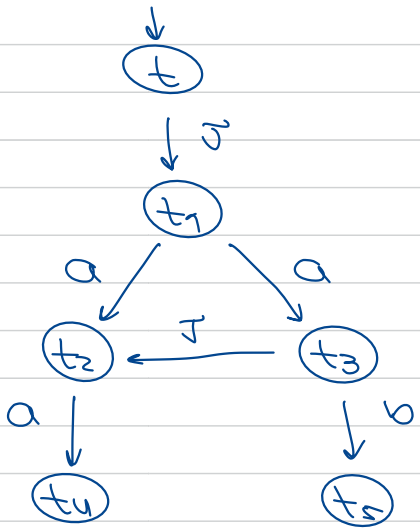
são fracamente bissimilares

Ver definição de game

Exercício



≉



A: $s \xrightarrow{a} s_1$
D: $t \xrightarrow{a} t_3$ $\quad (s_1, t_3)$

A: $t_3 \xrightarrow{\tau} t_2$ $\quad s_1 \xrightarrow{\tau} s_1 \quad (s_1, t_2)$
D: $s_1 \xrightarrow{b} s_3$ $\quad t_2 \not\xrightarrow{b} X$

Equivalence checking: $\sim$, $\simeq$, ...

Imp $\models$ property

↑ lógica?

Sintaxe:

$\phi := p \mid true \mid false \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \rightarrow \phi_2 \mid \Box\phi \mid \Diamond\phi$

$p \in Prop$

$\Box$ - necessidade ; $\Diamond$ - possibilidade

Semantics

Model: $\langle F, v \rangle$

↙ Kripke Frame ; ↘ valuation

$F = \langle W, R \rangle$

↙ World points ; ↘ $R \subseteq W \times W$

$v : Prop \longrightarrow 2^W$ ($\mathcal{P}(W)$)

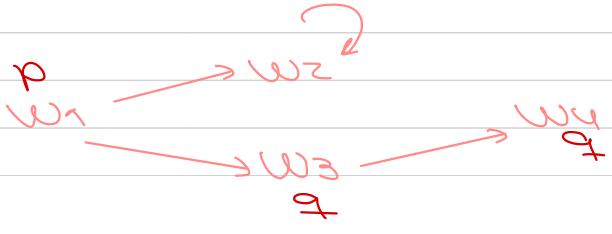
$v(p) = \{ \ldots \}$

$M = \langle F, v \rangle$

$F = \langle \{w_1, w_2, w_3, w_4\}, \{(w_1, w_2), (w_2, w_3), (w_3, w_4), (w_2, w_2)\} \rangle$

$v(p) = \{w_1\}$

$v(w) = \{w_3, w_1\}$



Satisfaction relation for a model and a world

$M, w \models true$
$M, w \not\models false$
$M, w \models p \quad w \in V(p)$
$M, w \models \neg\phi$ iff $M, w \not\models \phi$
$M, w \models \phi_1 \wedge \phi_2$ iff $M, w \models \phi_1$ and $M, w \models \phi_2$
$M, w \models \phi_1 \rightarrow \phi_2$ iff $M, w \not\models \phi_1$ or $M, w \models \phi_2$
$M, w \models \Box\phi$ iff for all $v \in w$ such that $(w, v) \in R$ e $M, v \models \phi$
$M, w \models \Diamond\phi$ iff exists $v \cdot (w, v) \in R$ and $M, v \models \phi$
$\exists v \in w \cdot (a, v) \in R$ and $M, v \models \phi$

Uma fórmula é satisfazível num modelo

se é satisfeita por uma condição de M

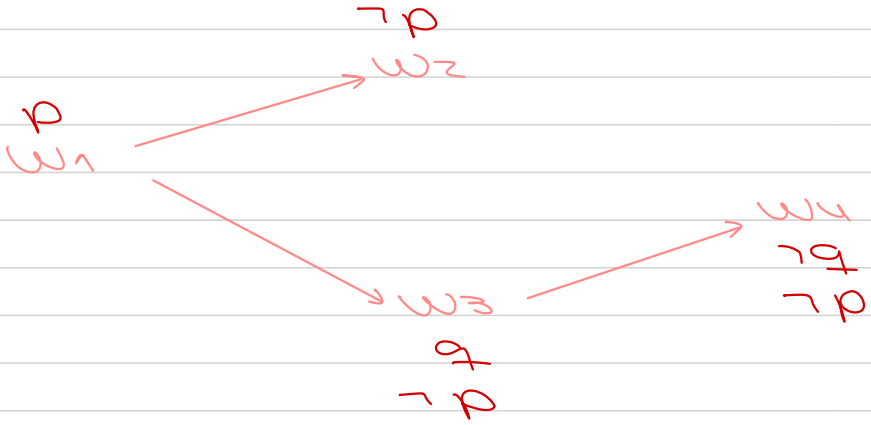
$$\exists w \in w \cdot M, w \models \phi$$

Uma fórmula é globalmente satisfeita em M

↳ $M \models \phi$

Validade

$\models \phi$ se é globalmente satisfeita em todos os modelos.



$M, w_2 \models \neg p$ é o mesmo que $M, w \not\models p$, $w_2 \notin V(p)$

$M, w_2 \models \neg p$ é o mesmo que $M, w \not\models p$, $w_2 \notin V(p)$

$M, w_2 \models \Diamond \neg q$ — $M, w_2 \models \neg q_2$, $M, w_2 \not\models q$, $w_2 \notin V(q)$ ✓

$M, w_3 \models \Diamond \neg q$ — $M, w_4 \models \neg q$, $M, w_4 \not\models q$, $w_4 \notin V(q)$ ✓

$M, w_1 \models \Box \neg p \rightarrow \neg p$

↓

vejamos se a implicação é válida

$M, w_1 \models \Box p$ ✓
$M, w_2 \models \Box p$ ×
$M, w_4 \models \Diamond \neg p$

$M, w_1 \not\models \Box \neg p$ como $M, w_1 \models \Box \neg p$ então a implicação não é válida

HML

$Prop = \{\}$ → não há proposições atómicas

ω = processos (termos / constantes para representar processos (CCS))

$K \in Act$

Modality corresponds to labelled by an element of K.

Model frame

$T = \langle Proc, \{\xrightarrow{a} \mid a \in Act\} \rangle$ $\xrightarrow{a} \subseteq Proc \times Proc$

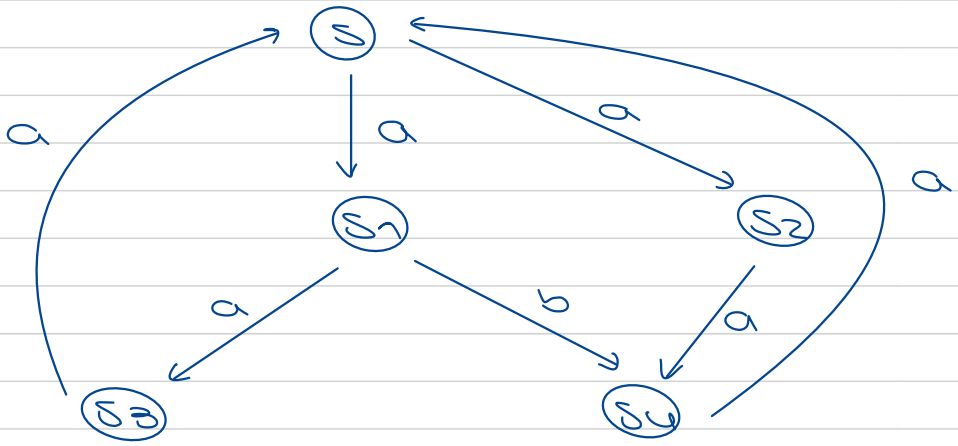
$p \models \langle a \rangle \phi$ iff $\exists q \in \{p' \mid p \xrightarrow{a} p'\} \cdot q \models \phi$

$p \models [a] \phi$ iff $\forall q \in \{p' \mid p \xrightarrow{a} p'\} \cdot q \models \phi$

↓ processo

Análise no contexto dos processos

Exemplo:



FAZER

$PROC = \{s, s_1, s_2, s_3, s_4\}$

$\xrightarrow{a} = \{(s, s_1), (s, s_2), (s_1, s_3), (s_3, s), (s_4, s), (s_2, s_4)\}$

$\xrightarrow{b} = \{(s_1, s_4)\}$

$F = \langle PROC, \{\xrightarrow{a}, \xrightarrow{b}\} \rangle$

① $s \models \langle a \rangle$ true
$(s, s_1) \in \xrightarrow{a}$, $s_1 \models$ true

② $s \models [a]$ true
$s \xrightarrow{a} s_1$ $s_1 \models$ true
$s \xrightarrow{a} s_2$ $s_2 \models$ true

③ $s \models [a]$ false
$s \xrightarrow{a} s_1$ $s_1 \models$ false
$s \xrightarrow{a} s_2$ $s_2 \models$ false

④ $s \models [b]$ false
$s \not\xrightarrow{}$

⑤ $s \not\models \langle b \rangle$ false
$s \not\xrightarrow{}$

⑥ $s \not\models \langle b \rangle$ true
$s \not\xrightarrow{}$

$s \models \langle a \rangle \langle b \rangle$ true
$s \xrightarrow{a} s_1$
$s_1 \models \langle b \rangle$ true
$s_1 \xrightarrow{b} s_4$
$s_4 \models$ true

$s \not\models [a] \langle b \rangle$ true
$s \xrightarrow{a} s_1$ $s_1 \models \langle b \rangle$ true
$s \xrightarrow{a} s_2$ $s_2 \not\models \langle b \rangle$ true

- $s \models [a]\langle a\rangle[a]\langle b\rangle true$

$s \xrightarrow{a} s_1 \quad s_1 \models \langle a\rangle[a]\langle b\rangle true$

$s \xrightarrow{a} s_2 \quad s_2 \models \langle a\rangle[a]\langle b\rangle true$

$s_2 \xrightarrow{a} s_4$

$s_4 \models [a]\langle b\rangle true$

(...)

$s_1 \xrightarrow{a} s_3$

$s_3 \models [a]\langle b\rangle true$

$s_3 \xrightarrow{a} s$

$s \models \langle b\rangle true$ X

$s_4 \xrightarrow{a} s$

$s_4 \models \langle b\rangle true$ X

$s \models [a]\langle a\rangle[a][b] true$ ✓

$s \models [b] true$ ✓

- $s \models \langle a\rangle(\langle a\rangle true \wedge \langle b\rangle true)$

$s \xrightarrow{a} s_1 \quad s_1 \models \langle a\rangle true \wedge \langle b\rangle true$

$s_1 \models \langle a\rangle true \wedge s_1 \models \langle b\rangle true$

$s_1 \xrightarrow{a} s_3 \quad s_3 \models true \vee s_1 \xrightarrow{b} s_4$

$s_4 \models true$

① $s \models \langle a\rangle([b][a] false \wedge \langle b\rangle true)$

② $s \models \langle a\rangle([a](\langle a\rangle true \wedge [b] false))$

③ $s \models \langle a\rangle[a](\langle a\rangle true \wedge [b] false \wedge \langle b\rangle false)$

Prova ②

$s \xrightarrow{a} s_2 \quad s_2 \models [a](\langle a\rangle true \wedge [b] false)$

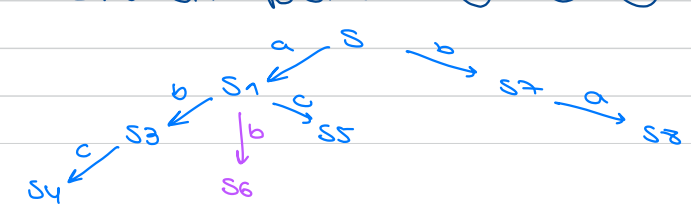
$s_2 \xrightarrow{a} s_4$

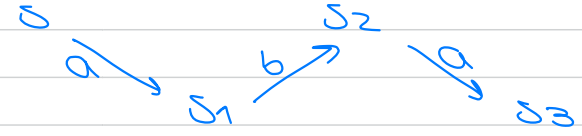
$s_4 \models \langle a\rangle true \wedge [b] false$

$s_4 \xrightarrow{a} s \quad s \models true$ ✓

$s_4 \not\xrightarrow{b}$ ✓

Trivial para ① e ②





FAZER

(a) $\langle a \rangle (\langle b \rangle \langle c \rangle true \wedge \langle c \rangle true)$ (*)

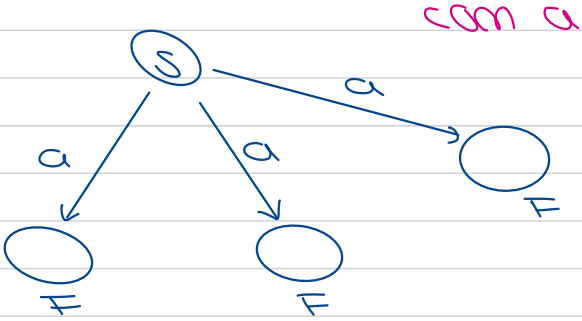
(b) $\langle a \rangle \langle b \rangle ([a] false \wedge [b] false \wedge [c] false)$

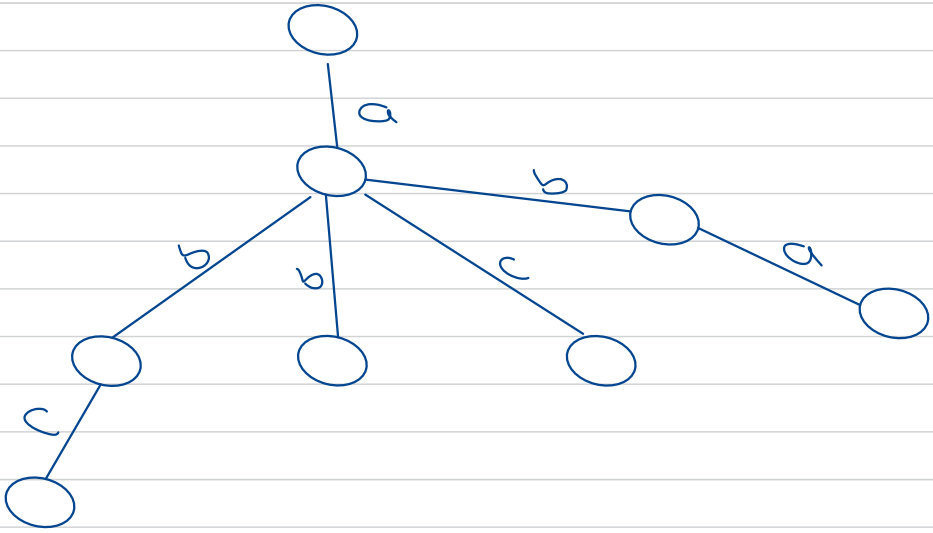
(c) $[a] \langle b \rangle ([c] false \wedge \langle a \rangle true)$

não há nenhuma trans com a,b, c

$S \models \langle a \rangle F$

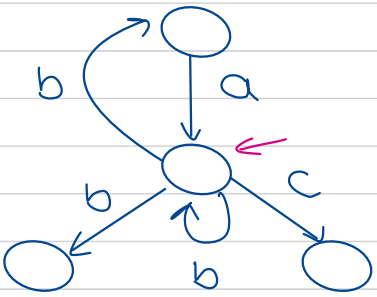
$S \models [a] F$



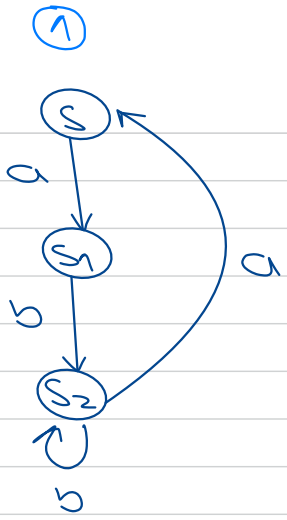


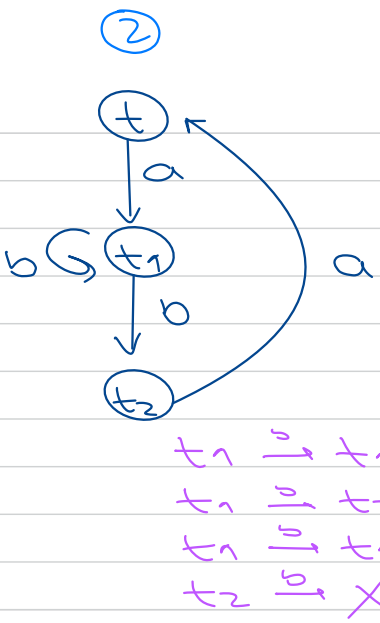
→ construção de (a) + (b) + (c)

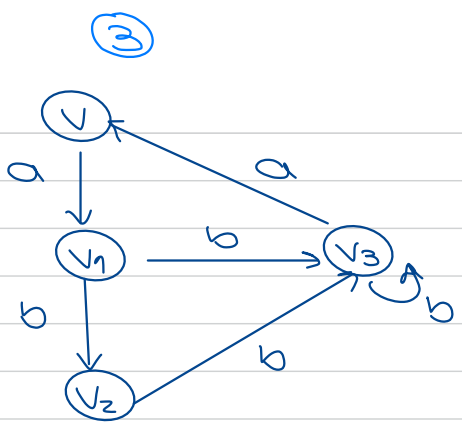
Forma mais compacta



começamos a validar aparte daqui (*)







Exercício:

$F = \langle a \rangle [b] \langle b \rangle \text{true}$ — analisar todas as trans.

Em qual dos sistemas a fórmula é válida

$s \models F$? ✓ $\quad t \models F$? ✗ $\quad v \models F$? ✓

$s \models \langle a \rangle [b] \langle b \rangle \text{true}$

$\exists s' : s \xrightarrow{a} s' \wedge (s' \models [b] \langle b \rangle \text{true})$

$s' \models [b] \langle b \rangle \text{true}$

$\forall s'' : s' \xrightarrow{b} s'' \wedge (s'' \models \langle b \rangle \text{true})$

Mt imp.

Queremos encontrar uma fórmula que permita distinguir ① e ③.

$s \models \langle a \rangle [b] \langle a \rangle$

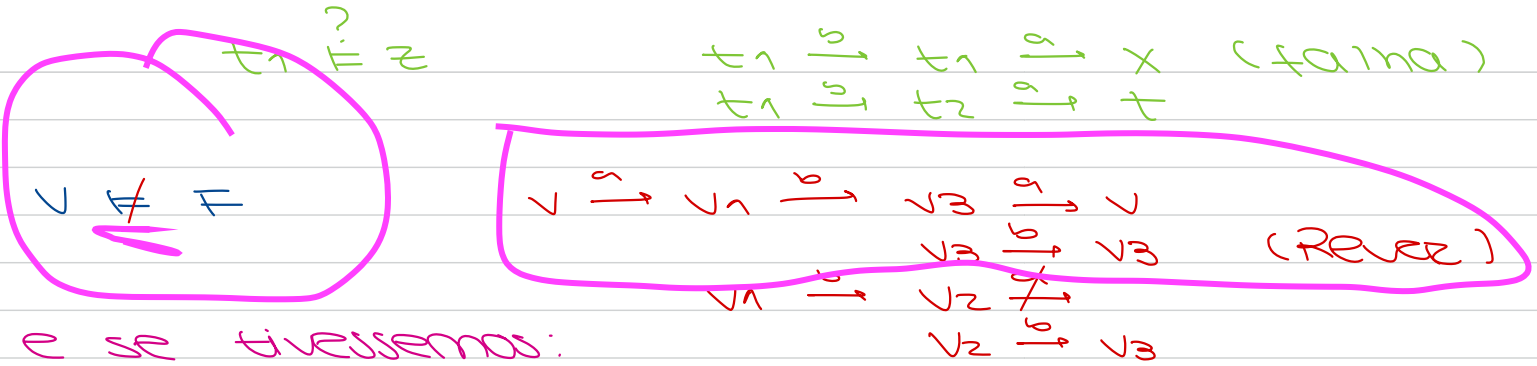
$v \not\models \langle a \rangle [b] \langle a \rangle$

Verificar se a expressão dada se verifica em ①, ② e ③:

$\langle a \rangle \langle b \rangle (\langle a \rangle \text{true} \wedge \langle b \rangle \text{true}) = F$

$s \models F$

$t \not\models F$ pq quando tenho caminhos por a não tem por b e vice-versa.



e se tivessemos:

$\langle a \rangle [b] (\langle a \rangle true \wedge \langle b \rangle true) = F$

S ≠ F
T ≠ F
V ≠ F

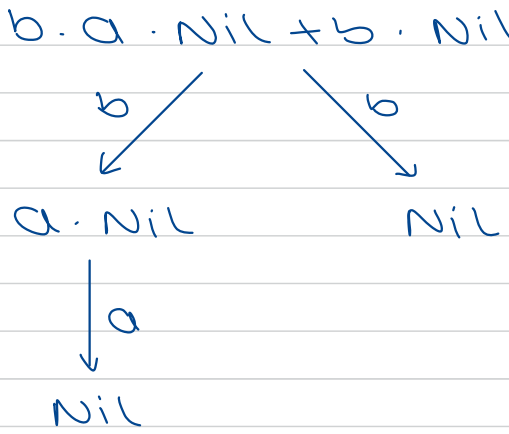
Exemplo em CCS.

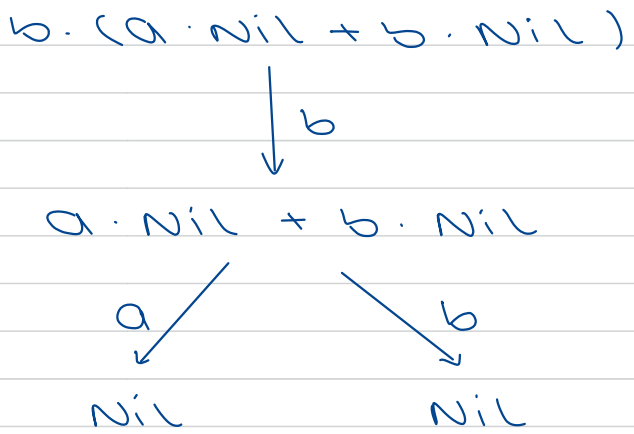
FAZER

Descobrir uma fórmula que os distinga:

① $b.a.Nil + b.Nil$

② $b.(a.Nil + b.Nil)$





$F = \langle b \rangle [a] false$

① ⊨ F

② ⊭ F

$F = \langle b \rangle (\langle a \rangle true \wedge \langle b \rangle true)$

① ⊭ F

② ⊨ F

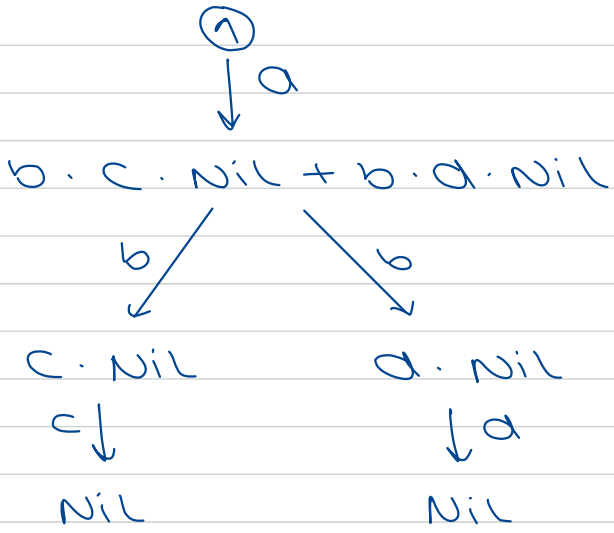
Possibilidade $\langle a \rangle$

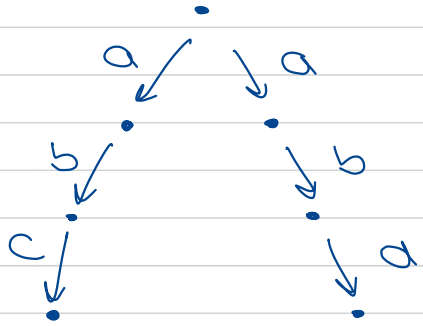
Necessidade $[a]$

Exemplo 2:

① $a \cdot (b \cdot c \cdot Nil + b \cdot a \cdot Nil)$

② $a \cdot b \cdot c \cdot Nil + a \cdot b \cdot a \cdot Nil$





$F = [a] \langle b \rangle \langle c \rangle true \wedge \langle b \rangle \langle a \rangle true$

① F F

② $\not\models$ F

$F_2 = [a]([b][c] false \wedge \langle b \rangle \langle a \rangle false)$

① F F

② $\not\models$ F

Exercício 1:

$(a \cdot Nil \mid b \cdot Nil) + c \cdot a\ Nil$

$a \cdot Nil \mid (b \cdot Nil + c \cdot Nil)$

Exercício 2:

$a \cdot Nil \mid b \cdot Nil \qquad a \cdot b \cdot Nil +_3 a \cdot Nil$

Hennessy-Milner Logic - Denotational semantics

$[\![\text{false}]\!] = \lambda\, \{\}$

$[\![\text{true}]\!] = \text{PROC}$

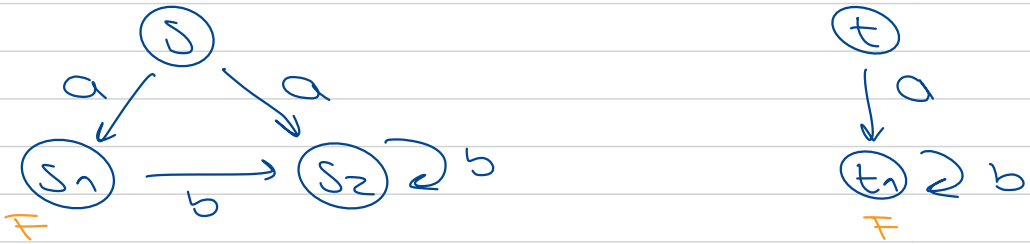
$[\![F \wedge G]\!] = [\![F]\!] \cap [\![G]\!]$

$[\![F \vee G]\!] = [\![F]\!] \cup [\![G]\!]$

$[\![\langle a \rangle F]\!] = \langle\cdot a \cdot\rangle\, [\![F]\!]$

$[\![[a] F]\!] = [\cdot a \cdot]\, [\![F]\!]$

Exemplo:



$\langle\cdot a \cdot\rangle \{s_1, t_1\} = \{s, t\}$

é possível chegar a $\underline{\underline{s_1}}$ e $\underline{\underline{t_1}}$ por $\underline{\underline{a}}$?

Sim, por $\underline{\underline{s}}$ e $\underline{\underline{t}}$

$[\cdot a \cdot]\, \{s_1, t_1\} = \{s_1, s_2, t, t_1\}$

$\langle\cdot b \cdot\rangle \{s_1, t_1\} = \{t_1\}$

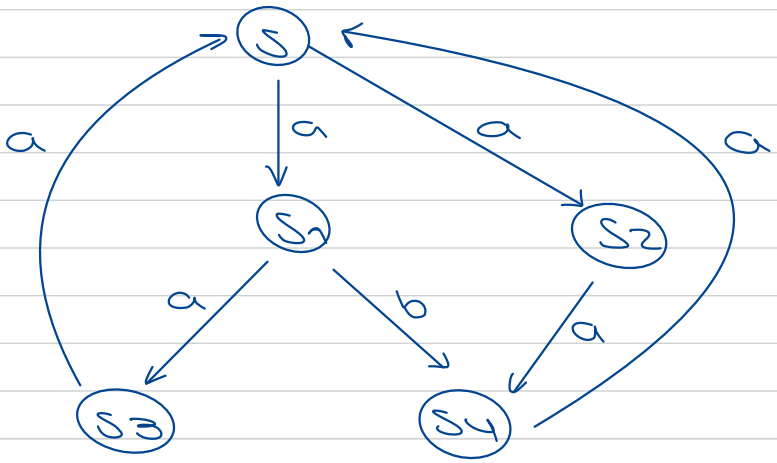
$[\cdot b \cdot]\, \{s_1, t_1\} = \{s, t, \cancel{s_1}, t_1\}$

↳ $s_1 \xrightarrow{b} s_2$ mas $s_2 \notin \{\dots\}$

↓

todos os estados de onde não parte nenhum b ⊕ trans. por b que estão no conjunto $\{\dots\}$

Exercício:



(a) $[\![[a][b] ff]\!]$ → $\{\}$

(b) $[\![\langle a \rangle (\langle a \rangle true \wedge \langle b \rangle true)]\!]$

(c) $[\![[a][a][b] false]\!]$

(d) $[\![[a](\langle a \rangle true \vee \langle b \rangle true)]\!]$

(a) $[\cdot a \cdot]([\![[b] ff]\!])$

$= [\cdot a \cdot]([\cdot b \cdot][\![ff]\!])$

$= [\cdot a \cdot]([\cdot b \cdot]\ \phi)$

$= [\cdot a \cdot]\ \{s, s_2, s_3, s_4\}$

todos os estados que não têm trans. por b.

$= \{s_2, s_3, s_4, s_1\}$

↳ pq $s_1 \xrightarrow{a} s_3$ e $s_3 \in \{\ldots\}$
$s \xrightarrow{a} s_1$ mas $s_1 \notin \{\ldots\}$

(b) $[\![\langle a \rangle (\langle a \rangle true \wedge \langle b \rangle true)]\!]$

$= \langle \cdot a \cdot \rangle (\langle \cdot a \cdot \rangle \{s, s_1, s_2, s_3, s_4\} \cap \langle \cdot b \cdot \rangle true)$

$= \langle \cdot a \cdot \rangle (\{s, s_1, s_2, s_3, s_4\} \cap \{s_1\})$

$= \langle \cdot a \cdot \rangle \{s_1\}$

$= \{s_4\}$

(c) $[\![[a][a][b] \text{false}]\!]$

$= [\cdot a \cdot]([\cdot a \cdot]([\cdot b \cdot]\ \phi))$

$= [\cdot a \cdot]([\cdot a \cdot]\{s, s_2, s_3, s_4\})$

$= [\cdot a \cdot]\{s_1, s_2, s_3, s_4\}$

$= \{s_1, s_2, s_4\}$

⚠

Todas as trans:

✗ $s \xrightarrow{a} s_1$ mas $s_1 \notin$

$s \xrightarrow{a} s_2$

✓ $s_1 \xrightarrow{a} s_3$

✓ $s_2 \xrightarrow{a} s_4$

✓ $s_3 \xrightarrow{a} s$

$s_4 \xrightarrow{a}$

Notas:

impossibilidade de a: [a] false

$\langle K \rangle \phi = \langle a_1 \rangle \phi \vee \langle a_2 \rangle \phi \vee \dots \vee \langle a_n \rangle \phi$

$[K]\phi = [a_1]\phi \wedge [a_2]\phi \dots \wedge [a_n]\phi$

com $K = \{a_1, a_2, \dots, a_n\}$

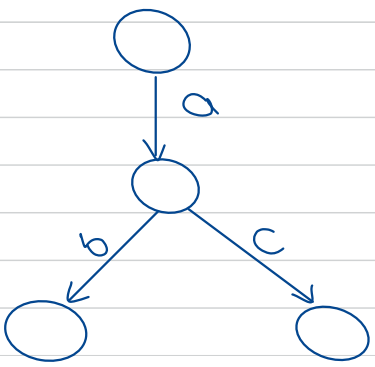
Representação do deadlock: [_] false

Inevitabilidade de a: <_> true

Exemplo

$\underbrace{\langle a \rangle}\ \underbrace{(\langle b \rangle \text{true} \wedge \langle c \rangle \text{true})}$

profundidade 1 prof. 2

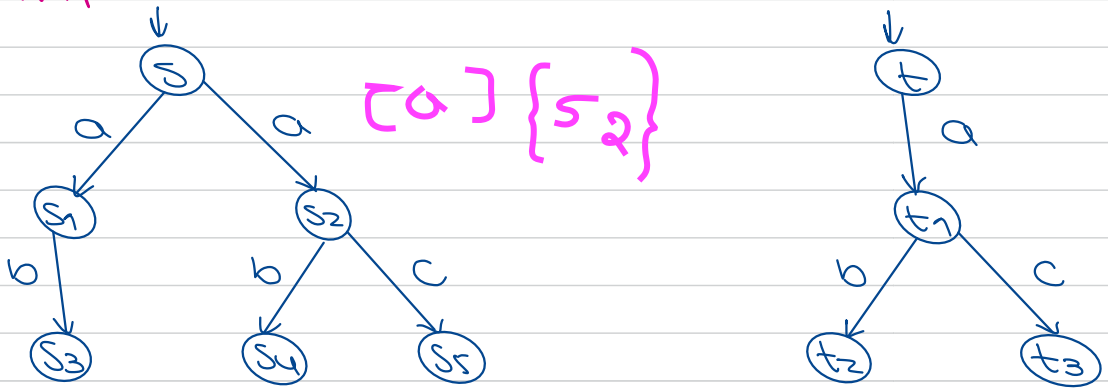


profundidade = 2

Notas:

- $\langle a \rangle$ false = false (conj. vazio, não há transições)
- $[\![[a] \text{ true}]\!] = [\![\text{true}]\!]$
- $[a] A = \neg \langle a \rangle \neg A$ e $\langle a \rangle A = \neg [a] \neg A$
- $\langle s \rangle \Phi = \bigvee_{a \in s} \langle a \rangle \Phi$ e $[s] \Phi = \bigwedge_{a \in s} [a] \Phi$
- $\frac{\langle Act \rangle \text{true}}{[Act]\, \Phi} = \bigvee_{a \in Act} \langle a \rangle \text{true}$
- $\langle a \rangle (\langle b \rangle \text{true} \vee \langle c \rangle \text{false})$ e $\neg [a] ([b] \text{false} \wedge [a] \text{false})$ } não são equivalentes
- $\langle a \rangle (\langle b \rangle \text{true} \vee \langle c \rangle \text{false})$ e $[a] ([b] \text{false} \wedge [c] \text{true})$ } são equivalentes

Exemplo:



$[a]\{s_2\}$

$\langle a \rangle [c]$ false → valida em s e inválida em t.

$[a](\langle b \rangle \text{true} \wedge \langle c \rangle \text{true})$ → válida (⊨) em s e t

$[\![a]\!]([\![\langle b \rangle]\!] \text{PROC} \wedge [\![\langle c \rangle]\!] \text{PROC})$ todos os nós de s e t.

$= [\![a]\!](\{s_1, s_2, t_1\} \wedge \{s_2, t_1\})$

$= [\![a]\!]\{s_2, t_1\} = \{t, s_1, s_2, t_1, t_2, t_3\}$

↳ s não faz parte pq $s_1 \notin \{s_2, t_1\}$

Proc - 2a4 ≡ [-a] false ← só é possível transitar por a.

<-a> ← só a não é possível

Invariante:

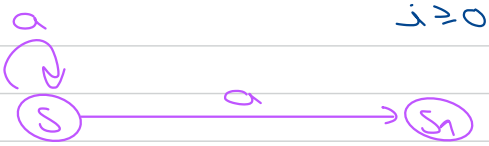
Inv(<a> true) = <a> true ∧ [a]<true> ∧ [a][a]<a> true ∧ ...

$$= \bigwedge_{i \geq 0} [a]^i \langle a \rangle \text{true}$$

Possibilidade:

Pos([a] false) = [a] false ∨ <a>[a] false ∨ <a><a>[a] false ∨ ...

$$= \bigvee_{i \geq 0} \langle a \rangle^i [a] \text{false}$$



Exercícios:

Inv(<a> true)

ϕ_x

max

$X = \langle a \rangle \text{true} \wedge [a] X$

$[\![X]\!] = [\![\langle a \rangle \text{true} \wedge [a] X]\!]$

Pos([a] false)

β

min

$X = [a] \text{false} \vee \langle a \rangle X$

↳ menor dos pontos fixos

Verificar se a expressão é válida

$X = \phi_X \rightarrow [\![X]\!] = \{\}$

$[\![\phi_X]\!] = \langle \cdot a \cdot \rangle Proc \cap [\cdot a \cdot] \{\}$

$\{p, q\} \cap \{a\} = \{\}$

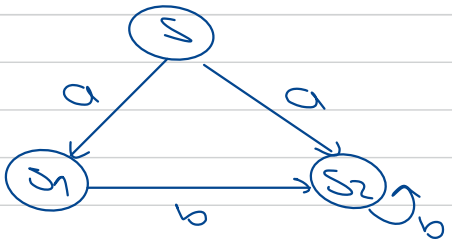
↳ $[\![X]\!] = \{\}$ ∧ ↳ $p \notin \{\}$

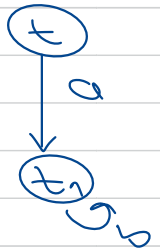
$[\![X]\!] = \lambda p.\top$

$[\![\phi x]\!] = \langle\cdot a\cdot\rangle \text{Proc} \cap [\cdot a\cdot]\, \lambda p.\top$
$= \lambda p,q.\top \cap \lambda p.\top$
$= \lambda p.\top$

HML

Exercícios:





Maior e menor dos pontos fixos

$X \overset{max}{=} \langle b\rangle \text{true} \wedge [b] X$ ① → é sempre possível uma transição por b

$Y \overset{min}{=} \langle b\rangle \text{true} \vee \langle\lambda a,b.\top\rangle Y\ (\langle a\rangle Y \vee \langle b\rangle Y)$ ②

A

① $[\![X]\!]$ Proc $[\![\langle b\rangle \text{true} \wedge [b] X]\!]$

Nível 1:

$[A] = \langle\cdot b\cdot\rangle \text{Proc} \cap [\cdot b\cdot]\, \text{Proc}$
$= \lambda s_1, s_2, t_1.\top \cap \lambda s_1, s_2, t_1, s, t.\top$
$= \lambda s_1, s_2, t_1.\top$ (ainda não é ponto fixo)

Nível 2:

$[\![X]\!] = \lambda s_1, s_2, t_1.\top$

$[A] = \langle\cdot b\cdot\rangle \text{Proc} \cap [\cdot b\cdot]\, \lambda s_1, s_2, t_1.\top$
$= \lambda s_1, s_2, t_1.\top \cap \lambda s_1, s_2, t_1, s, t.\top$
$= \lambda s_1, s_2, t_1.\top$

A fórmula é sempre válida para qq estado $s \in \lambda s_1, s_2, t_1.\top$

② $[\![y]\!] = \lambda\, 4$

Nível 1:

$\beta = \langle b \rangle \text{true} \vee (\langle a \rangle y \vee \langle b \rangle y)$

$[\![\beta]\!] = \langle\cdot b \cdot\rangle \text{Proc} \vee (\langle\cdot a\cdot\rangle \lambda\, 4 \vee \langle\cdot b\cdot\rangle \lambda\, 4)$

$= \lambda s_1, s_2, t_1 4 \vee \lambda\, 4$

$= \lambda s_1, s_2, t_1 4$ (ainda não é ponto fixo)

Nível 2:

$[\![y]\!] = \lambda s_1, s_2, t_1 4$

$[\![\beta]\!] = \langle\cdot b\cdot\rangle \text{Proc} \vee (\langle\cdot a\cdot\rangle \lambda s_1, s_2, t_1 4 \vee \langle\cdot b\cdot\rangle \lambda s_1, s_2, t_1 4$

$= \lambda s_1, s_2, t_1 4 \vee \lambda s, t 4 \vee \lambda s_1, s_2, t_1 4$

$= \text{Proc}$ (ainda não é ponto fixo)

Nível 3:

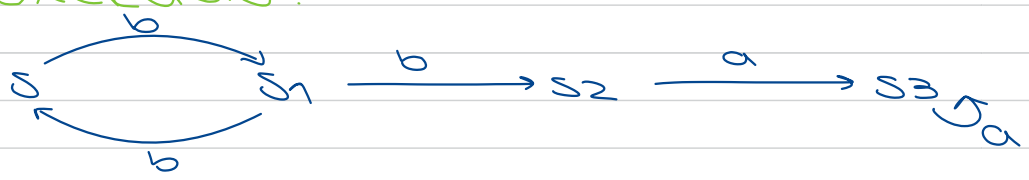
$[\![y]\!] = \text{Proc}$

$[\![\beta]\!] = \lambda s_1, s_2, t_1 4 \vee \langle\cdot a\cdot\rangle \text{Proc} \vee \langle\cdot b\cdot\rangle \text{Proc}$

$= \lambda s_1, s_2, t_1 4 \vee \lambda s, t 4 \vee \lambda s_1, s_2, t_1 4$

$= \text{Proc}$

Exercício:



$S \models \langle a \rangle \text{true} \vee \langle b \rangle X$

a)

$[\![X]\!] = \lambda\, 4$

$[\![\langle a \rangle \text{true} \vee \langle b \rangle X]\!]$

$\langle\cdot a\cdot\rangle \text{Proc} \vee \langle\cdot b\cdot\rangle \lambda\, 4$

$= \lambda s_2, s_3 4 \vee \lambda\, 4 = \lambda s_2, s_3 4$

2)

$[\![X]\!] = \lambda s_2, s_3 s_4$

$\langle a \rangle PROC \cup \langle b \rangle \lambda s_2, s_3 s_4$

$= \lambda s_2 s_3 s_4 \cup \lambda s_3 s_4$

$= \lambda s_2, s_3 s_4$

3)

$[\![X]\!] = \lambda s_2 s_3, s_4$

$\langle a \rangle PROC \cup \langle b \rangle \lambda s_2, s_3, s_4$

$= \lambda s_2, s_3 s_4 \cup \lambda s_4, s_4 = PROC$

4)

$[\![X]\!] = PROC$

$[\![\langle a \rangle PROC \cup \langle b \rangle PROC]\!]$

$= \lambda s_2, s_3 s_4 \cup \lambda s, s_4 = PROC$

Tendo: $X \overset{mm}{=} \langle a \rangle true \vee ([b] X \wedge \langle b \rangle true)$

Verificar se $s_1 \overset{?}{\models}$ (é valido em s_1)

1)

$[\![X]\!] = \lambda s_4$

$\langle a \rangle PROC \vee ([b] \lambda s_4 \wedge \langle b \rangle PROC)$

$= \lambda s_2, s_3 s_4 \cup (\lambda s_2, s_3 s_4 \cap \lambda s, s_4)$

$= \lambda s_2, s_3 s_4$

2)

$[\![X]\!] = \lambda s_2, s_3 s_4$

$\langle s_2, s_3 \rangle \cup ([b] \langle s_2, s_3 \rangle \cap \langle s, s_1 \rangle)$

$= \langle s_2, s_3 \rangle \cup (\langle s_1, s_2, s_3 \rangle \cap \langle s, s_1 \rangle)$

$= \langle s_1, s_2, s_3 \rangle$

3)

$[\![X]\!] = \langle s_1, s_2, s_3 \rangle$

$\langle s_2, s_3 \rangle \cup ([b] \langle s_1, s_2, s_3 \rangle \cap \langle s, s_1 \rangle)$

$\langle s_2, s_3, s_1, s \rangle = PROC$

4)

$[\![X]\!] = PROC$

$\langle a \rangle PROC \cup ([b] PROC \cap \langle b \rangle true)$

$= \langle s_2, s_3 \rangle \cup (\langle s, s_1, s_2 s_3 \rangle \cap \langle s, s_1 \rangle)$

$= PROC$

Regras do jogo:

① $(s, F_1 \wedge F_2)$ ou $(s, [a]F) \rightarrow$ Atacante
② $(s, F_1 \vee F_2)$ ou $(s, \langle a \rangle F) \rightarrow$ Defensor
③ $(s, X) \rightarrow (s, FX)$
④ (s, tt) ou (s, ff)
⑤ Atacante $(s, [a]f)$ $s \not\rightarrow$
⑥ Defensor $(s, \langle a \rangle f)$ $s \not\rightarrow$
⑦ jogo infinito

} jogo termina

Winner:

① A: O jogo termina $(s, false)$ ou D bloqueado
② D: O jogo termina $(s, true)$ ou A bloqueado
③ A: jogo infinito $X \overset{min}{=} FX$
④ D: jogo infinito $X \overset{max}{=} FX$

Exercícios:

$s \models [b](\langle b \rangle [b] false \wedge \langle b \rangle [a] false)$

$(s, Y) \xrightarrow{A} (s_1, \langle b \rangle [b] false \wedge \langle b \rangle [a] false) \xrightarrow{D}$

① $(s_1, \langle b \rangle [b] false) \xrightarrow{D} (s_2, [b] false) \xrightarrow{A}$
$s_2 \not\rightarrow$ A bloqueado

② $(s_1, \langle b \rangle [b] false) \xrightarrow{D} (s, [b] false) \xrightarrow{A}$
$(s_1, false)$ → jogo terminou

$X \overset{min}{=} \langle a \rangle true \vee \langle b \rangle X$

$(s, X) \rightarrow (s, \langle a \rangle true \vee \langle b \rangle X) \xrightarrow{D} (s, \langle b \rangle X) \xrightarrow{D}$
$(s_1, X) \rightarrow (s_1, \langle a \rangle true \vee \langle b \rangle X) \xrightarrow{D}$
$(s_1, \langle b \rangle X) \xrightarrow{D} (s_2, X) \rightarrow (s_2, \langle a \rangle true \vee \langle b \rangle X) \xrightarrow{D} (s_2, \langle a \rangle true) \xrightarrow{D} (s_3, true)$

↳ Defensor ganhou

$X \overset{mm}{=} \langle a \rangle true \vee ([b]X \wedge \langle b \rangle true)$

$(s_1, X) \rightarrow (s_1, \langle a \rangle true \vee ([b]X \wedge \langle b \rangle true)$

① $\rightarrow (s_1, \langle a \rangle true) \xrightarrow{a} s_1 \not\rightarrow$ bloqueado

② $\rightarrow (s_1, [b]X \wedge \langle b \rangle true)$

① $(s_1, [b]X)$

1.1 $\rightarrow (s_1, [b]X) \xrightarrow{A} (s_2, X)$

1.2 $\rightarrow (s_1, [b]X) \xrightarrow{A} (s, X)$

$\rightarrow (s, \langle a \rangle true \vee ([b]X \wedge \langle b \rangle true)$
$\xrightarrow{a} (s, [b]X \wedge \langle b \rangle true) \xrightarrow{A} (s, [b]X) \xrightarrow{A} (s_1, X)$

② $\rightarrow (s_1, \langle b \rangle true)$

1.1 $(s_1, \langle b \rangle true) \xrightarrow{b} (s_2, true)$

1.2 $(s_1, \langle b \rangle true) \xrightarrow{b} (s, true)$

Descobrir ponto fixo:

$X \overset{mm}{=} \langle a \rangle true \vee ([b]X \vee \langle b \rangle true)$ (A)

$[\![X]\!] = \lambda Y$

$[\![A]\!] = \langle \cdot a \cdot \rangle Proc \cup ([\cdot b \cdot] \lambda Y \cap \langle \cdot b \cdot \rangle Proc)$
$= \lambda\{s_2, s_3\} \cup (\lambda\{s_2, s_3\} \cap \lambda\{s, s_1\})$
$= \lambda\{s_2, s_3\}$

$[\![X]\!] = \lambda\{s_2, s_3\}$

$[\![A]\!] = \langle \cdot a \cdot \rangle Proc \cup ([\cdot b \cdot] \lambda\{s_2, s_3\} \cap \langle \cdot b \cdot \rangle Proc)$
$= \lambda\{s_2, s_3\} \cup (\lambda\{s_2, s_3\} \cap \lambda\{s, s_1\})$
$= \lambda\{s_2, s_3\}$

⚠ → Acho que chegamos ao ponto fixo.

No mesmo sistema:

$X \overset{max}{=} \langle b \rangle true \wedge [b] X$ (3)

$[\![X]\!] = [\![\langle b \rangle true \wedge [b] X]\!]$

$[\![X]\!] = Proc$

$[\![3]\!] = \langle \cdot b \cdot \rangle true \cap [\cdot b \cdot] Proc$
$= \lambda s, s_1 \rangle \cap \lambda s, s_1 \rangle$
$= \lambda s, s_1 \rangle$

$[\![X]\!] = \lambda s, s_1 \rangle$

$[\![B]\!] = \langle \cdot b \cdot \rangle true \cap [\cdot b \cdot] \lambda s, s_1 \rangle$
$= \lambda s, s_1 \rangle \cap \lambda s_2, s_3, s \rangle$
$= \lambda s \rangle$

↳ $s_1 \notin \lambda \ldots \rangle$ pq
$s_1 \xrightarrow{b} s$
$s_1 \xrightarrow{b} s_2$ mas $s_2 \notin \lambda \ldots \rangle$

$[\![X]\!] = \lambda s \rangle$

$[\![B]\!] = \langle \cdot b \cdot \rangle true \cap [\cdot b \cdot] \lambda s \rangle$
$= \lambda s, s_1 \rangle \cap \lambda s_2, s_3 \rangle$ → acho que é assim
$= \lambda \rangle$

$[\![X]\!] = \lambda \rangle$

$[\![B]\!] = \langle \cdot b \cdot \rangle true \cap [\cdot b \cdot] \lambda \rangle$
$= \lambda s, s_1 \rangle \cap \lambda s_2, s_3 \rangle$
$= \lambda \rangle$

A → Atacante $s \not\models F$

D → Defensor $s \models F$

Configurações (s, F)

Próximas configurações:

$(s, true)$, $(s, false)$ sem conf. seguinte

$(s, F_1 \wedge F_2) \rightarrow (s, F_1)$ e (s, F_2)

$(s, F_1 \vee F_2) \rightarrow (s, F_1)$ e (s, F_2)

$(s, [a]F)$ ou $(s, \langle a \rangle F)$ (s', F) todos os $s \xrightarrow{a} s'$

$(s, X) \rightarrow (s, FX)$ $X = FX$

Leis algébricas

≈ – Bissimulação forte

Choice Operator (processos – letra maiúscula)

$p + (q + r) \approx (p + q) + r$ (associatividade)
$p + q \approx q + p$ (comutatividade)
$p + Nil \approx p$ (identidade)
$p + p \approx p$ (idempotência)
$a \cdot (p + q) \approx a \cdot p + a \cdot q$

Parallel composition

$p \mid (q \mid r) \approx (p \mid q) \mid r$ (associatividade)
$p \mid q \approx q \mid p$ (comutatividade)
$p \mid Nil \approx p$ (identidade)
$p \mid r \not\approx p$

$A \overset{def}{=} a \cdot A$ $\quad \cdot \xrightarrow{a} \cdot \circlearrowright_a$

Expansion Law

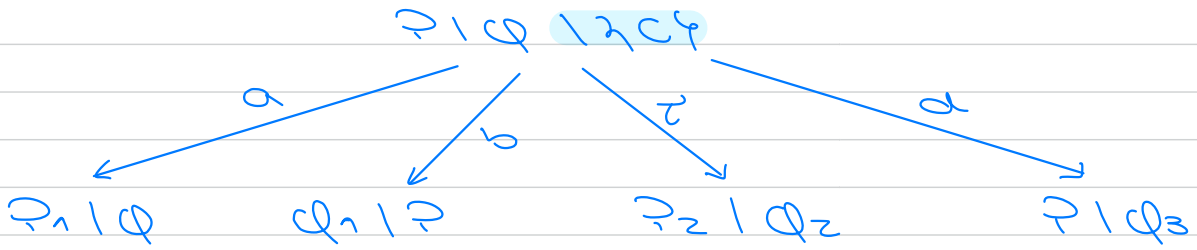
Let $p = \sum_{i=1}^{n} a_i \cdot p_i$ AND $q = \sum_{i=1}^{n} b_i \cdot q_i$

THEN

$$p \mid q = \sum_{i=1}^{n} a_i (p_i \mid q) + \sum_{j=1}^{n} b_i (p \mid q_i) + \sum_{a_i = b_i} \tau (p_i \mid q_i)$$

$a_i (p_i \mid q)$: avança o processo p
$b_i (p \mid q_i)$: avança o processo q
$\tau (p_i \mid q_i)$: avançam os dois processos

$P = a.P_1 + c.P_2$ $\quad Q = b.Q_1 + \bar{c}.Q_2 + a.Q_3$



Exercício:

$(P + Q) \mid R \overset{?}{\sim} (P \mid R) + (Q \mid R)$

É escolhido um dos processos (choice)

são bissimilares?

(fazer)

jump:

Restrição

$Nil \setminus s \overset{?}{\sim} Nil$

$(P \setminus s_1) \setminus s_2 \overset{?}{\sim} (P \setminus s_2) \setminus s_1$

$(P \setminus s) \setminus s \overset{?}{\sim} P \setminus s$

$(a.P) \setminus s \overset{?}{\sim} a.(P \setminus s) \quad a \notin s$

substituição sintática

$Nil\,[b/a] = Nil$

$(a.P)\,[b/a] = b.(P\,[b/a])$

$(\bar{a}.P)\,[b/a] = \bar{b}.(P\,[b/a])$

$(c.P)\,[b/a] = c.(P\,[b/a]) \quad c \neq a, \bar{a}$

$(P + Q)[b/a] = P[b/a] + Q[b/a]$
$(P \mid Q)[b/a] = P[b/a] \mid Q[b/a]$
$(P \setminus S)[b/a] = (P[b/a]) \setminus S \quad b, a \notin S$
$(P \setminus \{a\})[b/a] = P \setminus \{a\}$
$(P \setminus \{b\})[b/a] = P \setminus \{b\}$
$= ((P[c/b])[b/a]) \setminus \{c\}$
se a ocorre em P

Exercício:
$A = a \cdot A + b \cdot B \qquad A[c/a]$
$B = c \cdot a \cdot A + a \cdot B$

1º renomear c em B (ficamos com B')

$B' = B[e/c] = e \cdot a \cdot A + a \cdot B'$
$A' = a \cdot A' + b \cdot B'$

já podemos fazer a substituição $A'[c/a]$

$A' = c \cdot A' + b \cdot B'$
$B' = e \cdot a \cdot A' + c \cdot B'$

Regras importantes:

$P \setminus \{a\} \sim P$ se a não ocorre em P
$(P \setminus \{a\}) \mid Q \sim (P \mid Q) \setminus \{a\}$ se a não ocorre em Q
$P \mid (Q \setminus \{a\}) \sim (P \mid Q) \setminus \{a\}$ se a não ocorre em P
$P \setminus \{a\} \sim (P[b/a]) \setminus \{b\}$ se b não ocorre em P

2 - congruence

$P \sim Q$

- $a \cdot P \sim a \cdot Q$
- $P + R \sim Q + R$
- $P \mid R \sim Q \mid R$
- $P \setminus \{a\} \sim Q \setminus \{a\}$

≈ (bissimilação fraca) é uma congruência?

NÃO

P ≈ Q

- $a.P \approx a.Q$
- $P + R \not\approx Q + R$
- $P \mid R \approx Q \mid R$
- $P \backslash \{a\} \approx Q \backslash \{a\}$

justificação:

$\tau.a.Nil \approx a.Nil$

$(\tau.a.Nil + b\,Nil) \approx a.Nil + b.Nil$

$\downarrow \tau$

$a.Nil \not\approx a.Nil + b.Nil$

$\downarrow b$

$(\ldots)$

Rules of equational theories

Reflexividade $\dfrac{}{t = t}$ (1)

Simetria $\dfrac{t_1 = t_2}{t_2 = t_1}$ (2)

Transitividade $\dfrac{t_1 = t_2 \quad t_2 = t_3}{t_1 = t_3}$ (3)

Substituição $\dfrac{t_i = t_i'}{f(t_1, \ldots, t_i, \ldots, t_n) = f(t_1, \ldots, t_i', \ldots, t_n)}$ (4)

Instanciação $\dfrac{t_1 = t_2}{t_1[\rho] = t_2[\rho]}$ (5)

Axiomas

$$\frac{}{t_1 = t_2}$$ (6)

Exercício:

A1 $x + (y + z) = (x + y) + z$
A2 $x + y = y + x$
A3 $x + Nil = x$
A4 $x + x = x$

$\{A_1, A_2, A_3, A_4\} \vdash a \cdot Nil + (b \cdot P + a \cdot Nil) = a \cdot Nil + b \cdot P$

(Não sai nada deste género)

Teste:

Problemas → definir CCS → respetivo LTS
→ analisar processos