

3. CCS: Calculus of Communicating Systems (sequential)

José Proença

(slides mainly from Nelma Moreira)

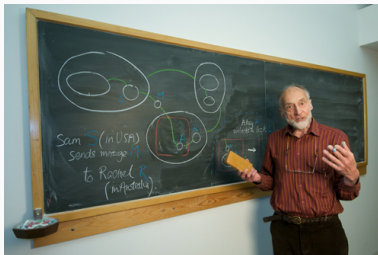
Concurrent programming (CC3040) 2025/2026

CISTER – U.Porto, Porto, Portugal

<https://fm-dcc.github.io/cp2526>



Avô Robin Milner: https://en.wikipedia.org/wiki/Robin_Milner (1934-2010)



1989

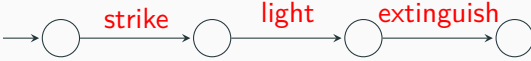
- 1970s** Edinburgh LCF (Logic of Computable Functions) um dos primeiros demonstradores interactivos de teoremas
- 1970s** ML (MetaLanguage): linguagem de programação funcional tipificada (antecessor do OCAML) com inferência de tipos polimórficos (sistema de tipos Hindley-Damas-Milner)
- 1980s** Sistemas concorrentes CCS, π -calculus
- 1991** Prémio ACM Turing
- 2000s** Bigrafos para computação ubiqua

- o processo mais simples é o que não executa nenhuma ação (deadlock): 0

- o processo mais simples é o que não executa nenhuma ação (deadlock): 0
- Se P é um processo e a uma ação, $a.P$ é um processo que executa a e depois comporta-se como P :



- o processo mais simples é o que não executa nenhuma ação (deadlock): 0
- Se P é um processo e a uma ação, $a.P$ é um processo que executa a e depois comporta-se como P :

- Um fósforo: $strike.light.extinguish.0$ 

- o processo mais simples é o que não executa nenhuma ação (deadlock): 0
- Se P é um processo e a uma ação, $a.P$ é um processo que executa a e depois comporta-se como P :
- Um fósforo: $strike.light.extinguish.0$
- Uma máquina de vender café: $coin.coffee.0$

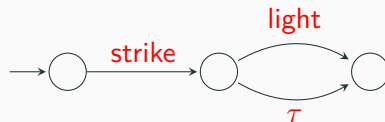


- o processo mais simples é o que não executa nenhuma ação (deadlock): 0
- Se P é um processo e a uma ação, $a.P$ é um processo que executa a e depois comporta-se como P :
- Um fósforo: *strike.light.extinguish.0*
- Uma máquina de vender café: *coin.coffee.0*
- Ações internas: τ

- o processo mais simples é o que não executa nenhuma ação (deadlock): 0
- Se P é um processo e a uma ação, $a.P$ é um processo que executa a e depois comporta-se como P :
- Um fósforo: $strike.light.extinguish.0$
- Uma máquina de vender café: $coin.coffee.0$
- Ações internas: τ
- Um fósforo que não acende: $strike.\tau.0$

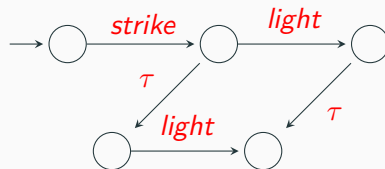


- o processo mais simples é o que não executa nenhuma ação (deadlock): 0
- Se P é um processo e a uma ação, $a.P$ é um processo que executa a e depois comporta-se como P :
- Um fósforo: $strike.light.extinguish.0$
- Uma máquina de vender café: $coin.coffee.0$
- Ações internas: τ
- Um fósforo que não acende: $strike.\tau.0$



- Se houver uma escolha: $strike.(light.0 + \tau.0)$

- o processo mais simples é o que não executa nenhuma ação (deadlock): 0
- Se P é um processo e a uma ação, $a.P$ é um processo que executa a e depois comporta-se como P :
- Um fósforo: $strike.light.extinguish.0$
- Uma máquina de vender café: $coin.coffee.0$
- Ações internas: τ
- Um fósforo que não acende: $strike.\tau.0$
- Se houver uma escolha: $strike.(light.0 + \tau.0)$



- Dois fósforos em paralelo: $strike.(light.0|\tau.0)$

- "." **Prefixo** a execução de $\alpha.P$ começa com a execução da ação $\alpha \in Act$ e depois comporta-se como P .
- "+" **Escolha** O processo $P + Q$ comporta-se como o processo P ou o processo Q . É a escolha não determinística.
- "|" **Composição Paralela** O processo $P|Q$ representa a execução concorrente de P e Q (que progridem independentemente um do outro).

Seja Act um conjunto de ações. O conjunto de expressões do CCS₀ são dadas pela seguinte gramática

$$P ::= 0 \mid P + P \mid \alpha.P$$

onde supomos as seguintes regras de prioridades

- $P + Q + R$ é $(P + Q) + R$

Seja *Act* um conjunto de ações. O conjunto de expressões do CCS₀ são dadas pela seguinte gramática

$$P ::= 0 \mid P + P \mid \alpha.P$$

onde supomos as seguintes regras de prioridades

- $P + Q + R$ é $(P + Q) + R$
- $\alpha.\beta.P$ é $\alpha.(\beta.P)$

Seja *Act* um conjunto de ações. O conjunto de expressões do CCS₀ são dadas pela seguinte gramática

$$P ::= 0 \mid P + P \mid \alpha.P$$

onde supomos as seguintes regras de prioridades

- $P + Q + R$ é $(P + Q) + R$
- $\alpha.\beta.P$ é $\alpha.(\beta.P)$
- $\alpha.P + Q$ é $(\alpha.P) + Q$

Seja *Act* um conjunto de ações. O conjunto de expressões do CCS₀ são dadas pela seguinte gramática

$$P ::= 0 \mid P + P \mid \alpha.P$$

onde supomos as seguintes regras de prioridades

- $P + Q + R$ é $(P + Q) + R$
- $\alpha.\beta.P$ é $\alpha.(\beta.P)$
- $\alpha.P + Q$ é $(\alpha.P) + Q$
- e por vezes omitimos o 0: $\alpha.\beta$ em vez de $\alpha.\beta.0$

Seja *Act* um conjunto de ações. O conjunto de expressões do CCS₀ são dadas pela seguinte gramática

$$P ::= 0 \mid P + P \mid \alpha.P$$

onde supomos as seguintes regras de prioridades

- $P + Q + R$ é $(P + Q) + R$
- $\alpha.\beta.P$ é $\alpha.(\beta.P)$
- $\alpha.P + Q$ é $(\alpha.P) + Q$
- e por vezes omitimos o 0: $\alpha.\beta$ em vez de $\alpha.\beta.0$
- Ex: $\text{strike}.(light.0 + \tau.0) \in \text{CCS}_0$

Exemplo

Quais dos seguintes termos são expressões válidas do CCS₀:

1. 0
2. $a.b.0$
3. $(a.b).0$
4. $(a.(b.0))$
5. $a.b$
6. $0.a.b$
7. $a.(b.0) + 0$
8. $0 + 0$
9. $a.(a.0 + b.c.0)$
10. $(a.0 + b.c.0).a.0$

<https://book.pseuco.com/#> Capítulo CCS_0 .

A semântica duma expressão P é um LTS $\llbracket P \rrbracket = (S, \longrightarrow, s_0)$. onde:

- $S = \{Q \mid Q \in CCS_0\}$, i.e., as expressões válidas do CCS_0

A semântica duma expressão P é um LTS $\llbracket P \rrbracket = (S, \longrightarrow, s_0)$. onde:

- $S = \{Q \mid Q \in CCS_0\}$, i.e., as expressões válidas do CCS_0
- $s_0 = P$

A semântica duma expressão P é um LTS $\llbracket P \rrbracket = (S, \longrightarrow, s_0)$. onde:

- $S = \{Q \mid Q \in CCS_0\}$, i.e., as expressões válidas do CCS_0
- $s_0 = P$
- $\longrightarrow \subseteq (CCS_0 \times Act \times CCS_0)$

A semântica duma expressão P é um LTS $\llbracket P \rrbracket = (S, \longrightarrow, s_0)$. onde:

- $S = \{Q \mid Q \in CCS_0\}$, i.e., as expressões válidas do CCS_0
- $s_0 = P$
- $\longrightarrow \subseteq (CCS_0 \times Act \times CCS_0)$

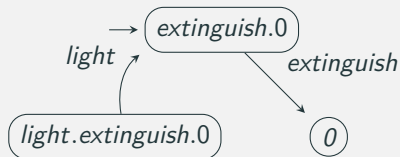
A semântica dum expressão P é um LTS $\llbracket P \rrbracket = (S, \longrightarrow, s_0)$. onde:

- $S = \{Q \mid Q \in CCS_0\}$, i.e., as expressões válidas do CCS_0
- $s_0 = P$
- $\longrightarrow \subseteq (CCS_0 \times Act \times CCS_0)$

Exemplo (Fragmento dum LTS para o processo $extinguish.0$)

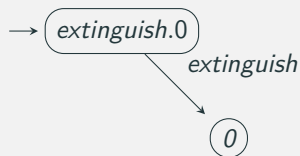
$Act = \{strike, light, extinguish, \tau\}$
para $\llbracket extinguish.0 \rrbracket$

<https://lmf.di.uminho.pt/ccs-caos/?light.extinguish.0>



Exemplo (Fragmento do LTS acessível para *extinguish.0*)

$Act = \{strike, light, extinguish, \tau\}$ para $\llbracket extinguish.0 \rrbracket$. Mas, pretendemos apenas o fragmento acessível, i.e., $Reach(extinguish.0)$:



- Permitem definir um LTS cujos estados são expressões e existe uma transição $P \xrightarrow{\alpha} Q$ se esta puder ser demonstrada a partir das regras.

- Permitem definir um LTS cujos estados são expressões e existe uma transição $P \xrightarrow{\alpha} Q$ se esta puder ser demonstrada a partir das regras.
- Uma regra é da forma

$$\frac{P_1 \cdots P_n}{P}$$

- Permitem definir um LTS cujos estados são expressões e existe uma transição $P \xrightarrow{\alpha} Q$ se esta puder ser demonstrada a partir das regras.

- Uma regra é da forma

$$\frac{P_1 \cdots P_n}{P}$$

- $P_1, \dots, P_n$ são as **Premissas** ou Hipóteses

- Permitem definir um LTS cujos estados são expressões e existe uma transição $P \xrightarrow{\alpha} Q$ se esta puder ser demonstrada a partir das regras.

- Uma regra é da forma

$$\frac{P_1 \cdots P_n}{P}$$

- $P_1, \dots, P_n$ são as **Premissas** ou Hipóteses
- P é a **Conclusão**

- Permitem definir um LTS cujos estados são expressões e existe uma transição $P \xrightarrow{\alpha} Q$ se esta puder ser demonstrada a partir das regras.

- Uma regra é da forma

$$\frac{P_1 \cdots P_n}{P}$$

- $P_1, \dots, P_n$ são as **Premissas** ou Hipóteses
- P é a **Conclusão**
- Significa: se $P_1, \dots, P_n$ se verificarem então P também se verifica (pode ser inferido), i.e., os P_i implicam P .

- Permitem definir um LTS cujos estados são expressões e existe uma transição $P \xrightarrow{\alpha} Q$ se esta puder ser demonstrada a partir das regras.

- Uma regra é da forma

$$\frac{P_1 \cdots P_n}{P}$$

- $P_1, \dots, P_n$ são as **Premissas** ou Hipóteses
- P é a **Conclusão**
- Significa: se $P_1, \dots, P_n$ se verificarem então P também se verifica (pode ser inferido), i.e., os P_i implicam P .
- Se $n = 0$ então a P é um **Axioma**

Regras do CCS_0

$$\text{Prefixo} \frac{}{\alpha.P \xrightarrow{\alpha} P}$$

$$\text{EscolhaE} \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'}$$

$$\text{EscolhaD} \frac{Q \xrightarrow{\alpha} Q'}{P + Q \xrightarrow{\alpha} Q'}$$

$\llbracket \text{coin.coffee.0} \rrbracket$

Regras: Prefixo $\frac{}{\alpha.P \xrightarrow{\alpha} P}$ EscolhaE $\frac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'}$ EscolhaD $\frac{Q \xrightarrow{\alpha} Q'}{P+Q \xrightarrow{\alpha} Q'}$

$$\llbracket \text{coin.coffee.0} \rrbracket$$

$$\frac{}{\text{coin.coffee.0} \xrightarrow{\text{coin}} \text{coffee.0}} \text{Prefixo}$$

Regras: Prefixo $\frac{}{\alpha.P \xrightarrow{\alpha} P}$ EscolhaE $\frac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'}$ EscolhaD $\frac{Q \xrightarrow{\alpha} Q'}{P+Q \xrightarrow{\alpha} Q'}$

$$\llbracket \text{coin.coffee.0} \rrbracket$$

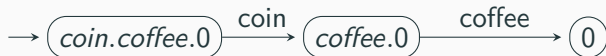
$$\frac{}{\text{coin.coffee.0} \xrightarrow{\text{coin}} \text{coffee.0}} \text{Prefixo}$$

$$\frac{}{\text{coffee.0} \xrightarrow{\text{coffee}} 0} \text{Prefixo}$$

Regras: Prefixo $\frac{}{\alpha.P \xrightarrow{\alpha} P}$ EscolhaE $\frac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'}$ EscolhaD $\frac{Q \xrightarrow{\alpha} Q'}{P+Q \xrightarrow{\alpha} Q'}$

$$\llbracket \text{coin.coffee.0} \rrbracket$$

$$\frac{}{\text{coin.coffee.0} \xrightarrow{\text{coin}} \text{coffee.0}} \text{Prefixo}$$

$$\frac{}{\text{coffee.0} \xrightarrow{\text{coffee}} 0} \text{Prefixo}$$


Regras:

Prefixo

$$\frac{}{\alpha.P \xrightarrow{\alpha} P}$$

EscolhaE

$$\frac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'}$$

EscolhaD

$$\frac{Q \xrightarrow{\alpha} Q'}{P+Q \xrightarrow{\alpha} Q'}$$

$\llbracket \text{coin.coffee.0} \rrbracket$

$$\frac{}{\text{coin.coffee.0} \xrightarrow{\text{coin}} \text{coffee.0}} \text{Prefixo}$$

$$\frac{}{\text{coffee.0} \xrightarrow{\text{coffee}} 0} \text{Prefixo}$$


Regras: Prefixo $\frac{}{\alpha.P \xrightarrow{\alpha} P}$ EscolhaE $\frac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'}$ EscolhaD $\frac{Q \xrightarrow{\alpha} Q'}{P+Q \xrightarrow{\alpha} Q'}$

$$\llbracket (a.0 + 0) + b.(0 + 0) \rrbracket$$

Regras: Prefixo $\frac{}{\alpha.P \xrightarrow{\alpha} P}$ EscolhaE $\frac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'}$ EscolhaD $\frac{Q \xrightarrow{\alpha} Q'}{P+Q \xrightarrow{\alpha} Q'}$

$$\llbracket (a.0 + 0) + b.(0 + 0) \rrbracket$$

$$\frac{\frac{\frac{}{a.0 \xrightarrow{a} 0} \text{Prefixo}}{a.0 + 0 \xrightarrow{a} 0} \text{EscolhaE}}{(a.0 + 0) + b.(0 + 0) \xrightarrow{a} 0} \text{EscolhaE}$$

Regras: Prefixo $\frac{}{\alpha.P \xrightarrow{\alpha} P}$ EscolhaE $\frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'}$ EscolhaD $\frac{Q \xrightarrow{\alpha} Q'}{P + Q \xrightarrow{\alpha} Q'}$

$$\llbracket (a.0 + 0) + b.(0 + 0) \rrbracket$$

$$\frac{\frac{\frac{}{a.0 \xrightarrow{a} 0} \text{Prefixo}}{a.0 + 0 \xrightarrow{a} 0} \text{EscolhaE}}{(a.0 + 0) + b.(0 + 0) \xrightarrow{a} 0} \text{EscolhaE}$$

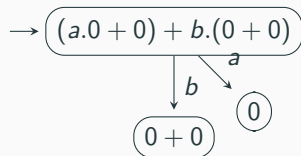
$$\frac{b.(0 + 0) \xrightarrow{b} 0 + 0}{(a.0 + 0) + b.(0 + 0) \xrightarrow{b} 0 + 0} \text{EscolhaD}$$

Regras: Prefixo $\frac{}{\alpha.P \xrightarrow{\alpha} P}$ EscolhaE $\frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'}$ EscolhaD $\frac{Q \xrightarrow{\alpha} Q'}{P + Q \xrightarrow{\alpha} Q'}$

$$\llbracket (a.0 + 0) + b.(0 + 0) \rrbracket$$

$$\frac{\frac{\frac{}{a.0 \xrightarrow{a} 0} \text{Prefixo}}{a.0 + 0 \xrightarrow{a} 0} \text{EscolhaE}}{(a.0 + 0) + b.(0 + 0) \xrightarrow{a} 0} \text{EscolhaE}$$

$$\frac{b.(0 + 0) \xrightarrow{b} 0 + 0}{(a.0 + 0) + b.(0 + 0) \xrightarrow{b} 0 + 0} \text{EscolhaD}$$



A relação $\longrightarrow$ é a menor tal que, para todo $\alpha \in Act$ e $P, Q \in CCS_0$,

- $(\alpha.P, \alpha, P) \in \longrightarrow$
- $(P + Q, \alpha, P') \in \longrightarrow$ se $(P, \alpha, P') \in \longrightarrow$
- $(P + Q, \alpha, Q') \in \longrightarrow$ se $(Q, \alpha, Q') \in \longrightarrow$
- nada mais está em $\longrightarrow$

Regras:

$$\text{Prefixo} \quad \frac{}{\alpha.P \xrightarrow{\alpha} P}$$

$$\text{EscolhaE} \quad \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'}$$

$$\text{EscolhaD} \quad \frac{Q \xrightarrow{\alpha} Q'}{P + Q \xrightarrow{\alpha} Q'}$$

O conjunto de todos os LTSs sobre expressões do CCS_0 é

$$LTS_0 = \{(CCS_0, T, P) \mid T \subseteq (CCS_0 \times Act \times CCS_0), P \in CCS_0\}$$

A semântica das expressões do CCS_0 é então

$$\llbracket _ \rrbracket : CCS_0 \rightarrow LTS_0$$

tal que

$$\llbracket P \rrbracket = (CCS_0, \longrightarrow, P)$$

com $\longrightarrow$ definida anteriormente.

- Contudo apenas interessa um fragmento de $\llbracket P \rrbracket$
- Só interessam estados atingíveis de P
- Só interessam as acções dos estados atingíveis
- Não interessam os nomes dos estados atingíveis
- Por exemplo $\alpha.0 + 0$ e $0 + \alpha.0$ não têm o mesmo LTS (verifica) mas são **isomorfos**.

Dois LTS $TS = (S, \longrightarrow, s_0)$ and $TS' = (S', \longrightarrow', s'_0)$ são isomorfos, $TS \sim TS'$, se existe uma bijeção f com

$$f : Reach(TS) \rightarrow Reach(TS')$$

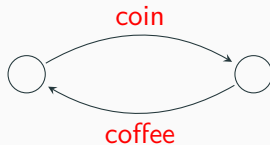
com

- $f(s_0) = s'_0$
- para todos os $s_1, s_2 \in Reach(TS)$ e para toda $\alpha \in Act$

$$s_1 \xrightarrow{\alpha} s_2 \quad sse \quad f(s_1) \xrightarrow{\alpha} f(s_2)$$

- Dois LTSs isomorfos são indistinguíveis para um observador.

- Como representar o seguinte LTS (máquina de café):

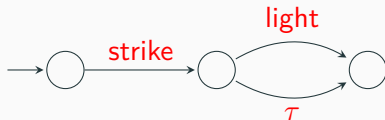


- Como representar o seguinte LTS (máquina de café):
- Usando **nomes** para sub-expressões e que podem ser usadas em expressões.

- Como representar o seguinte LTS (máquina de café):
- Usando **nomes** para sub-expressões e que podem ser usadas em expressões.
- Em vez de $strike.(light.0 + \tau.0)$

$Match \quad := \quad strike.MatchOneZeroLight$

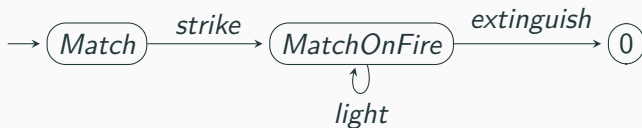
$MatchOneZeroLight \quad := \quad light.0 + \tau.0$



- Como representar o seguinte LTS (máquina de café):
- Usando **nomes** para sub-expressões e que podem ser usadas em expressões.
- Em vez de $strike.(light.0 + \tau.0)$

$$Match := strike.MatchOneZeroLight$$
$$MatchOneZeroLight := light.0 + \tau.0$$

- sendo que agora podemos representar repetição (iteração)

$$Match := strike.MatchOnFire$$
$$MatchOnFire := light.MatchOnFire + extinguish.0$$


- Como representar o seguinte LTS (máquina de café):
- Usando **nomes** para sub-expressões e que podem ser usadas em expressões.
- Em vez de $\text{strike}.(light.0 + \tau.0)$

$$\text{Match} := \text{strike.MatchOneZeroLight}$$
$$\text{MatchOneZeroLight} := light.0 + \tau.0$$

- sendo que agora podemos representar repetição (iteração)

$$\text{Match} := \text{strike.MatchOnFire}$$
$$\text{MatchOnFire} := light.MatchOnFire + \text{extinguish}.0$$

- **Convenção:** nomes (de sub-expressões) começam por uma letra maiúscula e ações por minúscula. Também são chamadas de *constantes* ou *variáveis*.

Seja Act um conjunto de ações e Var um conjunto de nomes (variáveis). As expressões do CCS_0^ω são

$$P ::= 0 \mid X \mid P + P \mid \alpha.P$$

onde

- $\alpha \in Act$,
- $X \in Var$ e
- uma função parcial de $Var \rightarrow CCS_0^\omega$, Γ , que pode ser representada por um conjunto de equações da forma $X := P$ ou representada por pares (X, P)

$$\Gamma = \{(X_1, P_1), \dots, (X_n, P_n)\}$$

onde $\Gamma(X_i) = P_i$ ou apenas $X_i := P_i, \quad 1 \leq i \leq n.$

A função Γ representa um *ambiente*.

Exemplo

1. $X := a.b.X$ então temos $\Gamma = \{(X, a.b.X)\}$

Exemplo

1. $X := a.b.X$ então temos $\Gamma = \{(X, a.b.X)\}$
2. Se

$$X := a.b.Y$$

$$Y := b.Z + a.Y$$

$$Z := a.Y$$

temos $\Gamma = \{(X, a.b.Y), (Y, b.Z + a.Y), (Z, a.Y)\}$.

$$\text{Rec} \frac{P \xrightarrow{\alpha} P' \quad \Gamma(X) = P}{X \xrightarrow{\alpha} P'}$$

$$\text{Rec} \frac{P \xrightarrow{\alpha} P' \quad \Gamma(X) = P}{X \xrightarrow{\alpha} P'}$$

Exemplo

Determinar $\llbracket X \rrbracket_{\Gamma}$ com $\Gamma = \{(X, a.b.X)\}$, i.e., temos a equação $X := a.b.X$

$$\text{Prefixo} \frac{}{a.b.X \xrightarrow{a} b.X} \quad \Gamma(X) = a.b.X$$

$$\text{Rec} \frac{}{X \xrightarrow{a} b.X}$$

$$\text{Prefixo} \frac{}{b.X \xrightarrow{b} X}$$

$$\text{Rec} \frac{P \xrightarrow{\alpha} P' \quad \Gamma(X) = P}{X \xrightarrow{\alpha} P'}$$

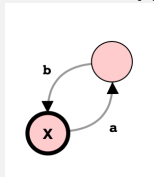
Exemplo

Determinar $\llbracket X \rrbracket_{\Gamma}$ com $\Gamma = \{(X, a.b.X)\}$, i.e., temos a equação $X := a.b.X$

$$\text{Prefixo} \frac{}{a.b.X \xrightarrow{a} b.X} \quad \Gamma(X) = a.b.X$$

$$\text{Rec} \frac{}{X \xrightarrow{a} b.X}$$

$$\text{Prefixo} \frac{}{b.X \xrightarrow{b} X}$$



O segundo estado é etiquetado por $b.X$

Regras do CCS_0^ω

$$\text{Prefixo} \frac{}{\alpha.P \xrightarrow{\alpha} P}$$

$$\text{EscolhaE} \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'}$$

$$\text{EscolhaD} \frac{Q \xrightarrow{\alpha} Q'}{P + Q \xrightarrow{\alpha} Q'}$$

$$\text{Rec} \frac{P \xrightarrow{\alpha} P' \quad \Gamma(X) = P}{X \xrightarrow{\alpha} P'}$$

A relação $\longrightarrow_\Gamma$ é a menor tal que, para todo $\alpha \in Act$ e $P, Q \in CCS_0^\omega$,

- $(\alpha.P, \alpha, P) \in \longrightarrow_\Gamma$
- $(P + Q, \alpha, P') \in \longrightarrow_\Gamma$ se $(P, \alpha, P') \in \longrightarrow_\Gamma$
- $(P + Q, \alpha, Q') \in \longrightarrow_\Gamma$ se $(Q, \alpha, Q') \in \longrightarrow_\Gamma$
- $(X, \alpha, P') \in \longrightarrow_\Gamma$ se $\Gamma(X) = P$ e $(P, \alpha, P') \in \longrightarrow_\Gamma$
- nada mais está em $\longrightarrow_\Gamma$

A relação $\longrightarrow_\Gamma$ é a menor tal que, para todo $\alpha \in Act$ e $P, Q \in CCS_0^\omega$,

- $(\alpha.P, \alpha, P) \in \longrightarrow_\Gamma$
- $(P + Q, \alpha, P') \in \longrightarrow_\Gamma$ se $(P, \alpha, P') \in \longrightarrow_\Gamma$
- $(P + Q, \alpha, Q') \in \longrightarrow_\Gamma$ se $(Q, \alpha, Q') \in \longrightarrow_\Gamma$
- $(X, \alpha, P') \in \longrightarrow_\Gamma$ se $\Gamma(X) = P$ e $(P, \alpha, P') \in \longrightarrow_\Gamma$
- nada mais está em $\longrightarrow_\Gamma$

Regra prática

Tentar aplicar as regras a um P até encontrar uma nova transição $P \xrightarrow{\alpha} P'$. E repetir para P' caso não fosse conhecido ele ou o seu comportamento.

Exemplo

Determinar $\llbracket M \rrbracket_\Gamma$ com $\Gamma = \{(M, \text{coin.coffee.M})\}$ ou seja

$$M := \text{coin.coffee.M}$$

Exemplo

Determinar $\llbracket M \rrbracket_{\Gamma}$ com $\Gamma = \{(M, \text{coin.coffee.M})\}$ ou seja

$$M := \text{coin.coffee.M}$$

$$\begin{array}{c} \text{Prefixo} \frac{}{} \\ \text{Rec} \frac{\text{coin.coffee.M} \xrightarrow{\text{coin}} \text{coffee.M} \quad \Gamma(M) = \text{coin.coffee.M}}{M \xrightarrow{\text{coin}} \text{coffee.M}} \end{array}$$

Exemplo

Determinar $\llbracket M \rrbracket_\Gamma$ com $\Gamma = \{(M, \text{coin.coffee.M})\}$ ou seja

$$M := \text{coin.coffee.M}$$

$$\text{Prefixo} \frac{}{\text{coin.coffee.M} \xrightarrow{\text{coin}} \text{coffee.M}} \quad \Gamma(M) = \text{coin.coffee.M}$$
$$\text{Rec} \frac{}{M \xrightarrow{\text{coin}} \text{coffee.M}}$$

$$\text{Prefixo} \frac{}{\text{coffee.M} \xrightarrow{\text{coffee}} M}$$

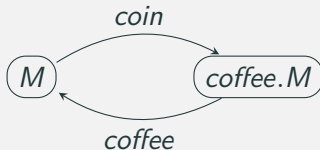
Exemplo

Determinar $\llbracket M \rrbracket_\Gamma$ com $\Gamma = \{(M, \text{coin.coffee.M})\}$ ou seja

$$M := \text{coin.coffee.M}$$

$$\text{Prefixo} \frac{}{\text{coin.coffee.M} \xrightarrow{\text{coin}} \text{coffee.M}} \quad \Gamma(M) = \text{coin.coffee.M}$$
$$\text{Rec} \frac{}{M \xrightarrow{\text{coin}} \text{coffee.M}}$$

$$\text{Prefixo} \frac{}{\text{coffee.M} \xrightarrow{\text{coffee}} M}$$



Como calcular:

- $\llbracket X \rrbracket_{\Gamma} \text{ e } \Gamma = \{(X, X)\}$
- $\llbracket X \rrbracket_{\Gamma} \text{ e } \Gamma = \{(X, X + a.0)\}$
- $\llbracket X_0 \rrbracket_{\Gamma} \text{ e } \Gamma = \{(X_i, a.X_{i+1}) \mid i \geq 1\}$

Como calcular:

- $\llbracket X \rrbracket_{\Gamma} \text{ e } \Gamma = \{(X, X)\}$
- $\llbracket X \rrbracket_{\Gamma} \text{ e } \Gamma = \{(X, X + a.0)\}$
- $\llbracket X_0 \rrbracket_{\Gamma} \text{ e } \Gamma = \{(X_i, a.X_{i+1}) \mid i \geq 1\}$

Exercício

Tenta calcular $\llbracket X \rrbracket$ com $\Gamma = (X, X + a.X)$ $\diamond$

Se se usar a regra EscolhaE podemos ter árvores de dedução infinitas que não levam a novas transições. Assim iremos evitar esse tipo de expressão (embora sintaticamente correctas).

- Uma variável X é **guardada** na expressão P se cada ocorrência de X em P ocorre numa expressão $\alpha.Q$

- Uma variável X é **guardada** na expressão P se cada ocorrência de X em P ocorre numa expressão $\alpha.Q$
- caso contrário **não é guardada**

- Uma variável X é **guardada** na expressão P se cada ocorrência de X em P ocorre numa expressão $\alpha.Q$
- caso contrário **não é guardada**
- Uma expressão é guardada se todas as suas variáveis são guardadas

- Uma variável X é **guardada** na expressão P se cada ocorrência de X em P ocorre numa expressão $\alpha.Q$
- caso contrário **não é guardada**
- Uma expressão é guardada se todas as suas variáveis são guardadas
- caso contrário **não é guardada**

- Uma variável X é **guardada** na expressão P se cada ocorrência de X em P ocorre numa expressão $\alpha.Q$
- caso contrário **não é guardada**
- Uma expressão é guardada se todas as suas variáveis são guardadas
- caso contrário **não é guardada**
- Ex. não guardadas: X , $a.0 + X$, $\tau.X + Y$, $a.X + Y$

- Uma variável X é **guardada** na expressão P se cada ocorrência de X em P ocorre numa expressão $\alpha.Q$
- caso contrário **não é guardada**
- Uma expressão é guardada se todas as suas variáveis são guardadas
- caso contrário **não é guardada**
- Ex. não guardadas: X , $a.0 + X$, $\tau.X + Y$, $a.X + Y$
- Ex. guardadas: $a.X$, $a.(X + Y)$, $\tau.X + a.Y$, $a.(X + b.Y)$

- Uma variável X é **guardada** na expressão P se cada ocorrência de X em P ocorre numa expressão $\alpha.Q$
- caso contrário **não é guardada**
- Uma expressão é guardada se todas as suas variáveis são guardadas
- caso contrário **não é guardada**
- Ex. não guardadas: X , $a.0 + X$, $\tau.X + Y$, $a.X + Y$
- Ex. guardadas: $a.X$, $a.(X + Y)$, $\tau.X + a.Y$, $a.(X + b.Y)$
- Se todas as imagens de um ambiente Γ (i.e. $\Gamma(X)$) são guardadas dizemos que Γ é *guardado*.

- Uma variável X é **guardada** na expressão P se cada ocorrência de X em P ocorre numa expressão $\alpha.Q$
- caso contrário **não é guardada**
- Uma expressão é guardada se todas as suas variáveis são guardadas
- caso contrário **não é guardada**
- Ex. não guardadas: X , $a.0 + X$, $\tau.X + Y$, $a.X + Y$
- Ex. guardadas: $a.X$, $a.(X + Y)$, $\tau.X + a.Y$, $a.(X + b.Y)$
- Se todas as imagens de um ambiente Γ (i.e. $\Gamma(X)$) são guardadas dizemos que Γ é *guardado*.
- Se $P \in CCS_0^\omega$ é guardada e os valores de Γ são guardados, é possível calcular $\llbracket P \rrbracket_\Gamma$ pela regra prática, isto é o processo termina.

- Uma variável X é **guardada** na expressão P se cada ocorrência de X em P ocorre numa expressão $\alpha.Q$
- caso contrário **não é guardada**
- Uma expressão é guardada se todas as suas variáveis são guardadas
- caso contrário **não é guardada**
- Ex. não guardadas: X , $a.0 + X$, $\tau.X + Y$, $a.X + Y$
- Ex. guardadas: $a.X$, $a.(X + Y)$, $\tau.X + a.Y$, $a.(X + b.Y)$
- Se todas as imagens de um ambiente Γ (i.e. $\Gamma(X)$) são guardadas dizemos que Γ é *guardado*.
- Se $P \in CCS_0^\omega$ é guardada e os valores de Γ são guardados, é possível calcular $\llbracket P \rrbracket_\Gamma$ pela regra prática, isto é o processo termina.
- No pseuco.com só há expressões guardadas

Exemplo

Exemplo

Exemplos de 1 – Buffer (de uma posição)

Considera as equações de Γ seguintes.

1. Calcula $\llbracket \text{Buffer} \rrbracket_{\Gamma}$.

$$\text{Buffer} := \text{put?}.\text{get?}.\text{Buffer}$$

2. Calcula $\llbracket \text{put?}.\text{BufferM} \rrbracket_{\Gamma}$.

$$\text{BufferM} := \text{put?}.\text{get?}.\text{BufferM} + \text{get?}.\text{put?}.\text{BufferM}$$

3. Calcula $\llbracket \text{Buffer0} \rrbracket_{\Gamma}$.

$$\text{Buffer0} := \text{put?}.\text{Buffer1}$$

$$\text{Buffer1} := \text{get?}.\text{Buffer2} + \text{get?}.\text{Buffer0}$$

$$\text{Buffer2} := \text{get?}.\text{Buffer1}$$

`https://pseuco.com/`

`https://lmf.di.uminho.pt/ccs-caos/`