Aula 5

Equivalências de Comportamento

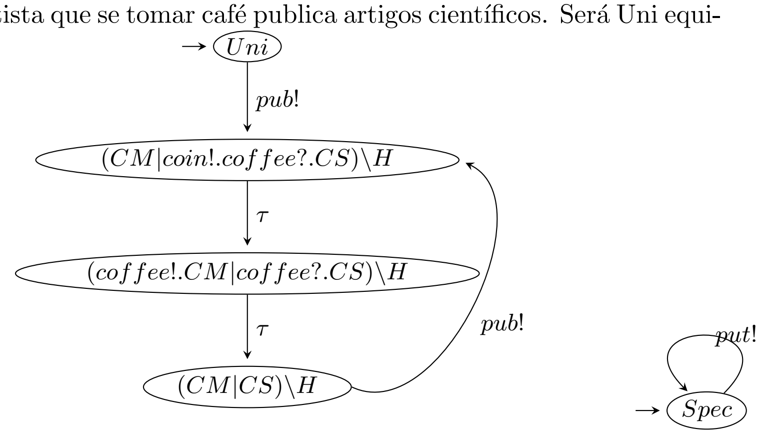
- O CCS tanto serve para a implementação de processos (SYS) com para a sua especifícação ($SPEC$).
- Dizer que SYS e $SPEC$ são equivalentes é dizer que têm o mesmo comportamento.
- Em particular pretende-se que SYS se comporte como $SPEC$, pois assim garante-se que a especificação é verificada pela implementação (*verificação*)

Exemplo

Seja $Spec := pub!.Spec$ e

$$\begin{array}{rcl} CM & := & coin?.coffee!.CM \\ CS & := & pub!.coin!.coffee?.CS \\ Uni & := & (CM|CS)\backslash\{coin, coffee\} \end{array}$$

Spec significa alguém que publica artigos científicos. CM máquina de café. CS significa um cientista que se tomar café publica artigos científicos. Será Uni equivalente a Spec?



O comportamento observável ($pub!$) é mesmo (ignorando os τ). Podemos dizer que Uni verifica *Spec*. Mas como definir esta relação?

Equivalência de Sistemas de Transição

Qual a relação entre estas expressões? quais os seus LTS?

$$
\begin{aligned}
0 + a.(b.0 + c.0) \\
a.(b.0 + \tau.c.0) \\
a.(b.0 + c.\tau.0) \\
a.(b.0 + c.(0 + 0)) \\
a.b.0 + a.c.0 \\
a.(b.0 + c.0) \\
a.(c.0 + b.0)
\end{aligned}
$$

Comportamento observável

Quais os aspectos do comportamento de um processo que são observáveis?

- acões de comunicação *Sim*
- ações internas *Não, mas para já sim...* τ é considerada uma ação
- sequência de estados por onde se passa *Não*
- sequência de ações (Traços) *Sim*
- nomes dos estados *Não*
- Dois processos podem ser iguais se não tiverem os mesmos traços *Não*
- Dois processos podem ser diferentes mesmo tendo os mesmos traços *Sim*

Traços

Dado um LTS $TS = (S, \longrightarrow, s_0)$ o *o conjunto de traços finitos* de TS é

$$
\begin{aligned}
Traces(TS) \quad = \quad & \{\alpha_1\alpha_2\cdots\alpha_n \in Act^* \mid n \geq 0 \wedge \\
& \exists s_1, \ldots s_n : s_{i-1} \xrightarrow{\alpha_i} s_i, \forall i\}
\end{aligned}
$$

isto é, um traço é uma sequência de acções que etiqueta um caminho em TS (i.e uma palavra em autómatos finitos).

- $\varepsilon \in Traces(TS)$ (palavra vazia)
- $s_0 \xrightarrow{\rho} s'$ se $\rho = \alpha_1\alpha_2\cdots\alpha_n$ é um traço de s_0 para $s_n = s'$
- Um traço de um processo P é um traço de $[\![P]\!]_\Gamma$ que começa em P.
- $Traces(P) = Traces([\![P]\!]_\Gamma)$.

Relação de equivalência

A igualdade de dois processos (expressões) do CCS deve ser uma relação de equivalência, i.e

- *reflexiva* $P \equiv P$ para todo o processo P
- *simétrica* se $P \equiv Q$ então $Q \equiv P$
- *transitiva* se $P \equiv Q$ e $Q \equiv R$ então $P \equiv R$
- e induz uma partição do CCS em classes de equivalência.

Algumas relações candidatas:

- $P, Q \in CCS$ são equivalentes sse são idênticos sintaticamente

$$Id(CCS) = \{(P, P) \mid P \in CCS\}$$

- $P, Q \in CCS$ são equivalentes sse são isomorfos

$$\sim_{iso} = \{(P, Q) \in CCS \mid [\![P]\!]_\Gamma \sim_{iso} [\![Q]\!]_\Gamma\}$$

- $P, Q \in CCS$ são equivalentes sse têm os mesmos traços

$$\sim_{tr} = \{(P, Q) \in CCS \mid Traces([\![P]\!]_\Gamma) = Traces([\![Q]\!]_\Gamma\})$$

- Todos os processos são iguais, $Univ(CCS) = CCS \times CCS$

Relações de equivalência mais finas ou mais grosseiras

- $Id(CCS)$ é a relação mais fina: cada classe de equivalência só tem um processo; há tantas classes quanto os processos
- $Univ(CCS)$ é a mais grosseira: todos os processos estão na mesma (única) classe
- E $\sim_{iso}$ e $\sim_{tr}$. São incomparáveis? Um é mais fino que o outro? Isto é, cada classe dum está contida numa classe do outro (mas podem lá estar mais)?

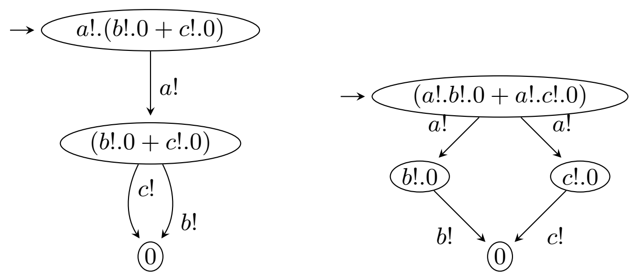
Isomorfismo e Traços

Temos que

$$a!.(b!.0 + c!.0) \sim_{tr} (a!.b!.0 + a!.c!.0)$$

mas

$$a!.(b!.0 + c!.0) \not\sim_{iso} (a!.b!.0 + a!.c!.0)$$



$$
\begin{array}{rcl}
Traces(a!.(b!.0 + c!.0)) & = & \{\varepsilon, a!, a!b!, a!c!\} \\
Traces(a!.b!.0 + a!c!.0) & = & \{\varepsilon, a!, a!b!, a!c!\}
\end{array}
$$

Mas $\sim_{iso} \subset \sim_{tr}$

Por contradição, suponhamos que $P \sim_{iso} Q$ e existe um traço $\rho \in Traces(\llbracket P \rrbracket_\Gamma)$ tal que $\rho \notin Traces(\llbracket Q \rrbracket_\Gamma)$. Seja k o comprimento de $\rho = \alpha_1 \cdots \alpha_k$.Como $P \sim_{iso} Q$ existe f bijeção entre os estados de $\llbracket P \rrbracket_\Gamma$ e $\llbracket Q \rrbracket_\Gamma$, mas então se

$$P = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \cdots \xrightarrow{\alpha_k} s_k$$

então

$$Q = f(s_0) \xrightarrow{\alpha_1} f(s_1) \xrightarrow{\alpha_2} f(s_2) \cdots \xrightarrow{\alpha_k} f(s_k)$$

Logo $\rho \in Traces(\llbracket Q \rrbracket_\Gamma)$ o que contradiz a hipótese. Logo $\sim_{iso} \subset \sim_{tr}$. E como vimos a inclusão é estrita.

Será que $\sim_{iso}$ ou $\sim_{tr}$ são boas noções de equivalência?

Seja

$$
\begin{array}{rcl}
P & := & a!.0 + b!.0 \\
Q & := & b!.0 + a!.0
\end{array}
$$

Será que

$$
\begin{array}{rcl}
P & \sim_{tr} & Q \\
a!.P & \sim_{tr} & a!.Q \\
a!.P + a!.P & \sim_{tr} & a!.Q + a!Q
\end{array}
$$

e o mesmo para $\sim_{iso}$?

Congruência

- Pretendemos que a relação seja também uma *congruência*
- Se $P \equiv Q$ e $C[\cdot]$ é um contexto então $C[P] \equiv C[Q]$.
- Um contexto $C[\cdot]$ é uma expressão com um buraco
- $C[\cdot] = 0 + a!b?.0|[\cdot]$ então $C[b!0] = 0 + a!b?.0|b!.0$
- ou $C[\cdot] = 0 + [\cdot]!b!.0$ então $C[a!b?.0] = 0 + a!b?.0|b!.0$
- no caso do CCS a $C[\cdot]$ chamámos um contexto CCS.

Relação de congruência no CCS

Uma relação de equivalência $\sim$ no CCS é uma relação de congruência, se para todas as expressões $P, Q \in CCS$ e para todos os contextos CCS $C[\cdot]$, $P \sim Q$ implica que $C[P] \sim C[Q]$.

Propriedades da igualdades no CCS

- relação de equivalência
- relação de congruência
- ter o mesmo conjunto de traços
- $Id(CCS)$, $\sim_{tr}$ e $Univ(CCS)$ são congruências
- $\sim_{iso}$ não é uma congruência: $(b!.0 + c!0) \sim_{iso} (c!.0 + b!0)$ mas $a!.(b!.0 + c!0) + a!.(b!.0 + c!0) \not\sim_{iso} a!.(b!.0 + c!0) + a!.(c!.0 + b!0)$
- Verifica!

$\sim_{tr}$ induz uma algebra

Temos as seguintes propriedades (comutatividade, associatividade e elemento neutro)

$$\begin{aligned} P + Q &\sim_{tr} Q + P \\ (P + Q) + R &\sim_{tr} P + (Q + R) \\ P + 0 &\sim_{tr} P \end{aligned}$$

Mas também $\alpha.(P + Q) \sim_{tr} \alpha.P + \alpha.Q$ e *não* queremos isso...

A equivalência por traços tem um problema!

Seja

$$\begin{aligned} CTM &:= coin?.(coffee!.CTM + tea!.CTM) \\ CTM' &:= coin?.coffee!.CTM' + coin?.tea!.CTM' \end{aligned}$$

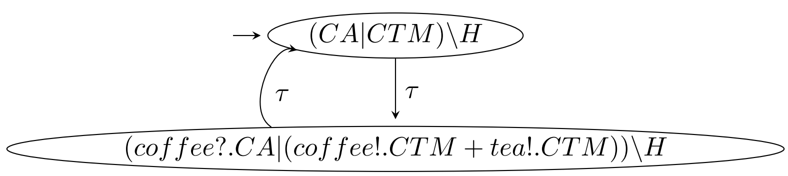
Temos que

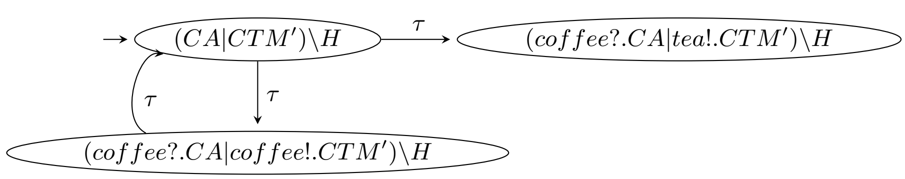
$$CTM \sim_{tr} CTM'$$

(Verifica!). Mas se só quisermos café o seu comportamento não é o mesmo.
Seja $CA := coin!.coffee?.CA$ e considere-se

$$(CA|CTM)\backslash H$$
$$(CA|CTM')\backslash H$$

com $H = \{coin, coffee, tea\}$.





Com CTM' o processo pode ficar bloqueado (*deadlock*) enquanto com CTM não fica.

- *Conclusão:* Duas expressões são equivalentes por traços mas têm comportamentos diferentes quando compostas em paralelo com outros processos, podendo um entrar em deadlock
- *Evitar* a regra $\alpha.(P + Q) \equiv \alpha.P + \alpha.Q$.

Deadlock

- Para um observador um deadlock corresponde a se ter chegado a um *estado terminal* s, i.e $Post(s, Act) = \emptyset$ e representa-se por $s \not\rightarrow$

- um traço terminal (ou completo) de um processo P é uma sequência de ações $\alpha_1 \cdots \alpha_k \in Act^*$ tal que

$$P = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \cdots \xrightarrow{\alpha_{k-1}} s_k \xrightarrow{\alpha_k} s_k \not\rightarrow$$

- Se P é terminal em $[\![P]\!]_\Gamma$ então P é um processo *em deadlock*.

- Os traços terminais de um processo P são todos os traços que acabam num estado terminal

$$TTraces(P) = \{\rho \in Traces([\![P]\!]_\Gamma) \mid \exists P' : P \overset{\rho}{\rightsquigarrow} P' \wedge P' \not\rightarrow \}$$

Exemplos

- $Traces(a!.b!.0 + a!.0) = \{\varepsilon, a!, a!b!\}$
- $Traces(a!.b!.0) = \{\varepsilon, a!, a!b!\}$
- $TTraces(a!.b!.0 + a!.0) = \{a!, a!b!\}$
- $TTraces(a!.b!.0) = \{a!b!\}$

Relações sensíveis a Deadlock

Uma relação $\equiv$ em CCS *é sensível a deadlocks* se

$$\forall P, Q \in CCS : \; P \equiv Q \Rightarrow TTraces(P) = TTraces(Q).$$

Pretendemos então uma relação de igualdade em CCS que:

- relação de equivalência
- relação de congruência
- ter o mesmo conjunto de traços
- seja sensível a deadlocks

- $\sim_{iso}$ é sensível a deadlocks mas não de congruência
- $\sim_{tr}$ não é sensível a deadlocks mas é de congruência.

Bisimulação

- Interessa a interação dos processo ao longo da execução e não apenas a sequência de ações

- Dois processos são equivalentes (iguais...) se podem mutuamente simular as ações um do outro

- isto é, se dois estados são equivalentes o seus sucessores também devem ser.

Formalmente

P e Q são equivalentes, $P \equiv Q$ se e só se para toda a acção $\alpha \in Act$

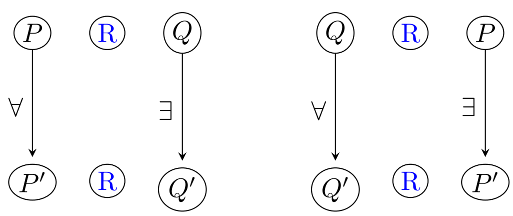
- Se $P \xrightarrow{\alpha} P'$ então existe Q' tal que $Q \xrightarrow{\alpha} Q'$ e $P' \equiv Q'$
- Se $Q \xrightarrow{\alpha} Q'$ então existe P' tal que $P \xrightarrow{\alpha} P'$ e $P' \equiv Q'$.

Mas esta formalização não é uma definição porque há muitas relações que a verificam (p.e $Id(CCS)$ e $Uni(CCS)$).

Relação de Bisimulação (Forte)

Seja $TS = (S, \longrightarrow, s_0)$. Uma relação $R \in S \times S$ é uma *bisimulação* se $(s,t) \in R$ (ou $s\ R\ t$) implica para todo $\alpha \in Act$:

- se $s \xrightarrow{\alpha} s'$ então existe t' tal que $t \xrightarrow{\alpha} t'$ e $(s',t') \in R$
- Se $t \xrightarrow{\alpha} t'$ então existe s' tal que $s \xrightarrow{\alpha} s'$ e $(s',t') \in R$.



Bisimilaridade

Dois estados s e t são *bisimilares* e escreve-se $s \sim t$, se existe uma bisimulação R tal que $(s,t) \in R$.

Isto é

$$\sim = \bigcup_{\text{R é bisimulação}} R$$

A relação $\sim$ chama-se *Bisimilaridade*

Teorema 5.1. *A relação $\sim$ é de equivalência.*

Notar que se R é uma bisimulação R^{-1} também é.

Teorema 5.2. *A relação $\sim$ é maior bisimulação.*

Mostrar que $\sim$ é uma bisimulação.

Teorema 5.3. *$s \sim t$ se e só se para cada $\alpha \in Act$*

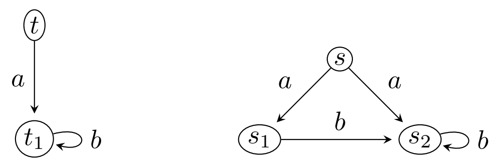
- *Se $s \xrightarrow{\alpha} s'$ então existe t' tal que $t \xrightarrow{\alpha} t'$ e $s' \sim t'$*
- *Se $t \xrightarrow{\alpha} t'$ então existe s' tal que $s \xrightarrow{\alpha} s'$ e $s' \sim t'$.*

$\Rightarrow$: porque $\sim$ é bisimulação $\Leftarrow$: temos que ter uma bisimulação $\mathcal{R}$ tal que $(s,t) \in \mathcal{R}$. Então podemos temos também que ter $(s',t') \in \mathcal{R}$ o que podemos fazer se adicionarmos a $\mathcal{R}$ todos os pares de $\sim$. Tomando $\mathcal{R} = \{s,t\} \cup \sim$ temos o que queremos

Exercício 5.1. *Termina cautelosamente todas as provas anteriores.* $\diamond$

Exemplo 1

Seja $TS = (S, Act, \longrightarrow)$ tal que $S = \{s, s_1, t, t_1\}$, $Act = \{a, b\}$ e $\longrightarrow$ definida pelos seguintes diagramas:



Mostrar que $s \sim t$. Temos de encontrar $R \in S \times S$ tal que

1. R (forte) bisimulação
2. $(s,t) \in R$.

Se $(s,t) \in R$ então para s temos $s \xrightarrow{a} s_1$ podemos selecionar t_1 com $t \xrightarrow{a} t_1$ e portanto também terá de ser $(s_1,t_1) \in R$; $s \xrightarrow{a} s_2$ também podemos selecionar t_1 com $t \xrightarrow{a} t_1$ e portanto também terá de ser $(s_2,t_1) \in R$; Se $(s,t) \in R$ então para t temos $t \xrightarrow{a} t_1$ podemos selecionar s_2 com $s \xrightarrow{a} s_2$ e portanto também terá de ser $(s_2,t_1) \in R$;e não há mais transições. Se $(s_1,t_1) \in R$ então para s_1 temos que $s_1 \xrightarrow{b} s_2$ e podemos selecionar t_1 com $t_1 \xrightarrow{b} t_1$ e $(s_1,t_1) \in R$. Se $(s_1,t_1) \in R$ então para t_1 temos que $t_1 \xrightarrow{b} t_1$ e podemos selecionar s_2 com $s_1 \xrightarrow{b} s_2$ e $(s_2,t_1) \in R$. Se $(s_2,t_1) \in R$ então para s_2 temos que $s_2 \xrightarrow{b} s_2$ e podemos selecionar t_1 com $t_1 \xrightarrow{b} t_1$ e $(s_2,t_1) \in R$. Se $(s_2,t_1) \in R$ então para t_1 temos que $t_1 \xrightarrow{b} t_1$ e podemos selecionar s_2 com $s_2 \xrightarrow{b} s_2$ e $(s_2,t_1) \in R$.
Concluímos que é uma bisimulação a relação

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

e portanto $s \sim t$. Mostra também que $s_1 \sim s_2$ porque $R_1 = \{(s_1,s_2),(s_2,s_2)\}$ é uma bisimulação.

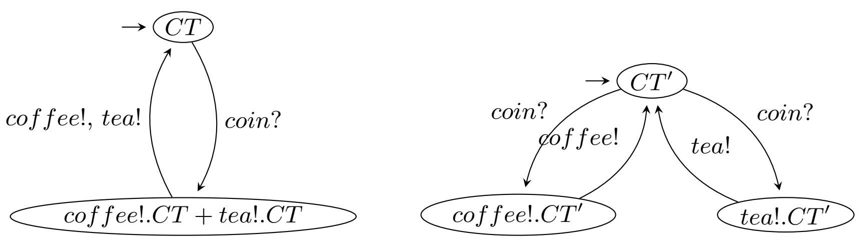
Exemplo 2

Seja $TS = (S, Act, \longrightarrow)$ tal que $S = \{s_i \mid i \geq 1\} \cup \{t\}$, $Act = \{a\}$ e $\xrightarrow{a} = \{(s_i, s_{i+1} \mid i \geq 1\} \cup \{(t,t)\}$.
Mostrar que $s_1 \sim t$ provando que $R = \{(s_i, t) \mid i \geq 1\}$ é uma bisimulação.

Como mostrar que dois processos não são bisimilares?

Mostrar que CT e CT' não são bisimilares, onde

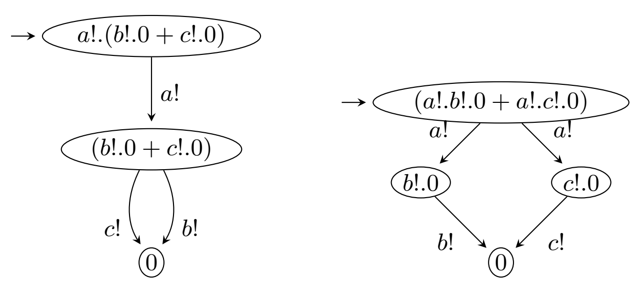
$$\begin{array}{rcl} CT & := & coin?.(coffee!.CT + tea!.CT) \\ CT' & := & coin?.coffee!.CT' + coin?.tea!.CT' \end{array}$$



Seja R uma bisimulaçao com $(CT, CT') \in R$.

Como $CT' \xrightarrow{coin?} tea!.CT'$, tem de existir P' tal que $CT \xrightarrow{coin?} P'$ e $(tea!.CT', P') \in R$. Só pode ser $P' = coffee!.CT + tea!.CT$. Mas então $P' \xrightarrow{coffee!} CT$ e $tea!.CT'$ não tem transições por $coffee!$. Logo $CT \not\sim CT'$.

Jogo da bisimulação (forte)



Podemos mostrar que $a!.(b!.0 + c!.0) \not\sim a!.b!.0 + a!.c!.0$ jogando um jogo de dois jogadores! O atacante quer mostrar que dois estados (s, t) não são bisimilares e o defesa quer mostrar que são.

- O *atacante* escolhe um dos estados (s ou t) e uma ação α duma transição do estado escolhido
- O *defesa* escolhe uma transição pela mesma ação do outro estado

- Se o defesa não poder jogar então os estados *não são bisimilares*
- A: $a!.(b!.0 + c!.0) \xrightarrow{a!} (b!.0 + c!.0)$
- D: $a!.b!.0 + a!.c!.0) \xrightarrow{a!} b!.0$
- A: $b!.0 + c!.0 \xrightarrow{c!} 0$
- D não pode jogar, logo $a!.(b!.0 + c!.0) \not\sim a!.b!.0 + a!.c!.0$.

A Bisimilaridade é uma congruência

Teorema 5.4. *Para todo $P, Q \in CCS$ e todos os contextos CCS $C[\cdot]$, $P \sim Q$ implica que $C[P] \sim C[Q]$.*

Dem: Por indução na estrutura dos possíveis $C[\cdot]$.
Em particular se $P \sim Q$ e $\alpha \in Act$, $R \in CCS$ e $H \subseteq Com$,

$$\begin{array}{rcl} \alpha.P & \sim & \alpha.Q \\ P+R & \sim & Q+R \\ R+P & \sim & R+Q \\ P|R & \sim & Q|R \\ R|P & \sim & R|Q \\ P\backslash H & \sim & Q\backslash H \end{array}$$

Se $P \sim Q$ **então** $(P|R) \sim (Q|R)$

Seja

$$\mathcal{R} \;=\; \{(P'|R', Q'|R') \mid P', Q', R' \in CCS \wedge P' \sim Q'\}$$

- É fácil ver que $(P|R, Q|R) \in \mathcal{R}$.
- Temos que mostrar que $\mathcal{R}$ é uma bisimulação.
- Usando a simetria basta ver que se $(P'|R', Q'|R') \in \mathcal{R}$ e $P'|R' \xrightarrow{\alpha} S$ então existe S' tal que $Q'|R' \xrightarrow{\alpha} S'$ e $(S, S') \in \mathcal{R}$.
- A prova segue por análise de casos às possíveis maneiras de obter a transição $P'|R' \xrightarrow{\alpha} S$ (i.e última regra aplicada).

A última regra aplicada pode ser:

- $ParE$: então $P'|R' \xrightarrow{\alpha} S$ resulta de $P' \xrightarrow{\alpha} P''$ e $S = P''|R'$. Neste caso, sei que existe Q'' tal que $Q' \xrightarrow{\alpha} Q''$ e $P'' \sim Q''$. Pela regra $ParE$, $Q'|R' \xrightarrow{\alpha} Q''|R'$ e por definição $(P''|R', Q''|R') \in \mathcal{R}$, logo basta tomar $S'' = Q''|R'$.

- $ParD$: análogo ao anterior (verifica!)
- Syn: então $P'|R' \xrightarrow{\tau} S$ resulta de $P' \xrightarrow{a} P''$ e $R' \xrightarrow{\overline{a}} R''$, sendo $S = P''|R''$. Como $P \sim Q$ existe Q'' com $Q' \xrightarrow{a} Q''$ tal que $P'' \sim Q''$. Então também, $Q'|R' \xrightarrow{\tau} Q''|R''$ e $(P''|R'', Q''|R'') \in \mathcal{R}$, logo basta tomar $S'' = Q''|R''$.

A Bisimilaridade é sensível a deadlocks

Sempre que $P \sim Q$ então $TTraces(P) = TTraces(Q)$.

Dem: Temos que

$$TTraces(P) = \{\rho \in Traces([\![P]\!]_\Gamma) \mid \exists P' : P \overset{\rho}{\rightsquigarrow} P' \wedge P' \not\rightarrow\}$$

.

- Seja $P \sim Q$, $\sigma \in TTraces(P)$ e $\sigma \notin TTraces(Q)$.
- Seja $\sigma = \alpha_1\alpha_2 \cdots \alpha_k$. Do estado inicial $s_0 = P$ temos

$$s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \ldots \xrightarrow{\alpha_k} s_k \not\rightarrow$$

- Como $P \sim Q$ existe uma bisimulação R com $(P, Q) \in R$.
- E então existem $t_0, \ldots, t_k$ com $t_0 = Q$ e $(s_i, t_i) \in R$ e

$$t_0 \xrightarrow{\alpha_1} t_1 \xrightarrow{\alpha_2} t_2 \ldots \xrightarrow{\alpha_k} t_k$$

- mas então $t_k \not\rightarrow$ senão R não era uma bisimulação dado que existia t' e α tal que $t_k \xrightarrow{\alpha} t'$ e não existia transição por α de s_k.

Mais Propriedades de $\sim$

$$\begin{aligned}
P + Q &\sim Q + P \\
P|Q &\sim Q|P \\
P + 0 &\sim P \\
P|0 &\sim P \\
(P + Q) + R &\sim P + (Q + R) \\
(P|Q)|R &\sim P|(Q|R)
\end{aligned}$$

Mas $\alpha.(P + Q) \not\sim \alpha.P + \alpha.Q$.