Aula 7

Equivalências de Comportamento

- O CCS tanto serve para a implementação de processos (SYS) como 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*)

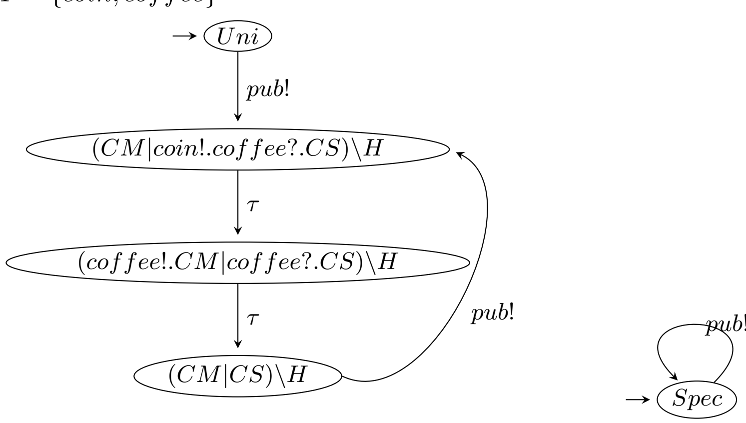
Exemplo

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 $Spec$?

$H = \{coin, coffee\}$

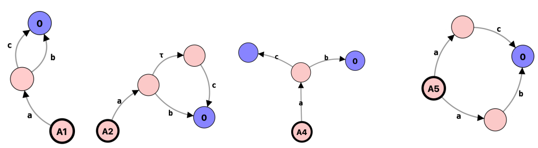


O comportamento observável ($pub!$) é o mesmo (ignorando os τ). Podemos dizer que Uni verifica $Spec$. Mas como definir esta relação?

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 $TS = (S, \longrightarrow, s_0)$ e Act conjunto de ações *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 de Sistemas de Transição

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

$$f : \mathit{Reach}(TS) \to \mathit{Reach}(TS')$$

com

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

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

- Dois LTSs isomorfos são indistinguíveis para um observador.

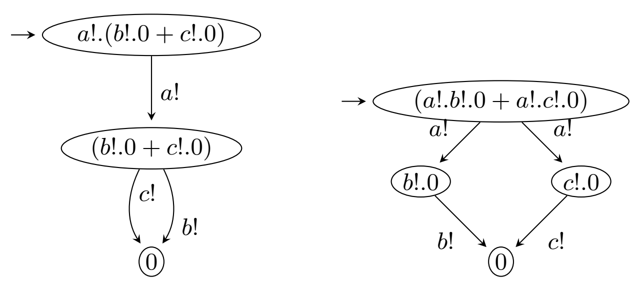
Isomorfismo e Traços

Temos que

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

mas

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



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

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

Por contradição, suponhamos que $P \sim_{iso} Q$ e existe um traço $\rho \in Traces([\![P]\!]_\Gamma)$ tal que $\rho \notin Traces([\![Q]\!]_\Gamma)$. Seja k o comprimento de $\rho = \alpha_1 \cdots \alpha_k$.Como $P \sim_{iso} Q$ existe f bijeção entre os estados de $[\![P]\!]_\Gamma$ e $[\![Q]\!]_\Gamma$, 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 $\rho \in Traces([\![Q]\!]_\Gamma)$ 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 $\sim_{iso}$? Isto é: será que se P e Q forem equivalentes ($\sim_{iso}$ ou $\sim_{tr}$) onde usarmos um podemos substituir pelo outro? Isto é, serão *serem congruentes*?

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 $C[P] \sim C[Q]$. Isto é, onde usar um posso substituir pelo outro.

Propriedades da igualdade 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)

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

Mas também $\alpha.(P + Q) \sim_{tr} \alpha.P + \alpha.Q$ e neste caso *não* têm o memso comportamento "observável" pelo que não é desejável.

A equivalência por traços tem um problema!

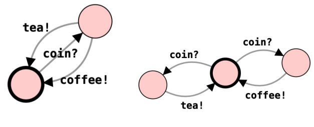
Seja

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

Temos que

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

(Verifica!).

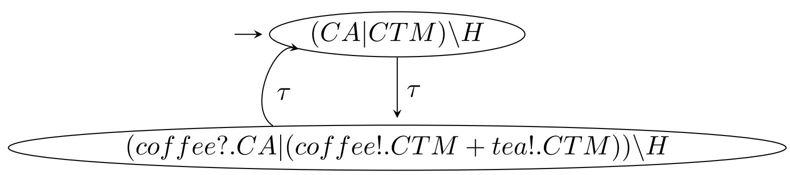


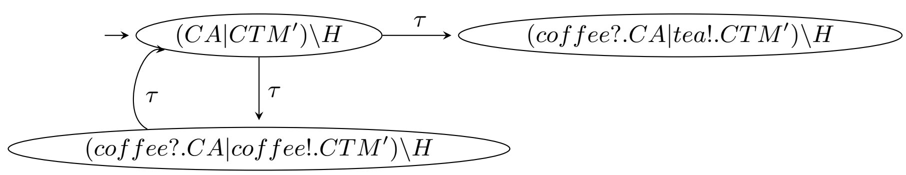
Mas se só quisermos café o seu comportamento não é o mesmo.

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 seja:

- relação de equivalência
- relação de congruência
- tenha 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.