

Aula 9

Bissimulação fraca e Congruência Observável

2 – *Buffer*

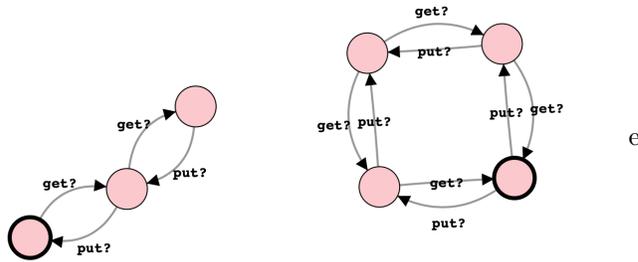
$$Buffer := put?.get?.Buffer$$

$$Buffer0 := put?.Buffer1$$

$$Buffer1 := put?.Buffer2 + get?.Buffer0$$

$$Buffer2 := get?.Buffer1$$

Será que $Buffer0 \sim (Buffer|Buffer)?$

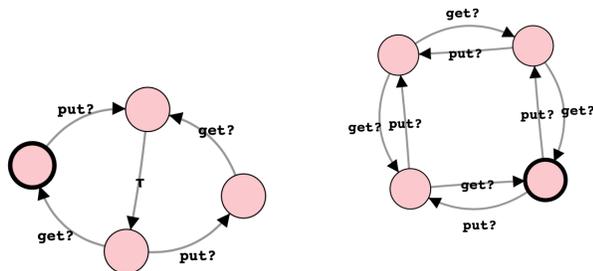


$$BufferL := put?.pass!.BufferL$$

$$BufferR := pass?.get?.BufferR$$

Será que

$(BufferL|BufferR) \setminus \{pass!, pass?\} \sim (Buffer|Buffer)?$



Bissimulação Fraca

Ignora transições por τ . Seja $TS = (S, \longrightarrow, s_0)$. Uma relação $R \in S \times S$ é uma *bissimulação fraca* 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 $(s', t') \in R$.

Onde

Transições fracas

- $s \xRightarrow{\tau} s'$ sse $\exists n \geq 0 : s \xrightarrow{\tau^n} s'$
- $s \xRightarrow{\alpha} s'$ sse $\exists s'', s''' \in S : s \xrightarrow{\tau} s'' \xrightarrow{\alpha} s''' \xRightarrow{\tau} s'$

Nota que

- $\longrightarrow \subseteq \xRightarrow{\quad}$
- na definição de b.f. podemos usar sempre $\xRightarrow{\quad}$.
- $s \xRightarrow{\tau} s$ para todo o s

Bissimilaridade fraca

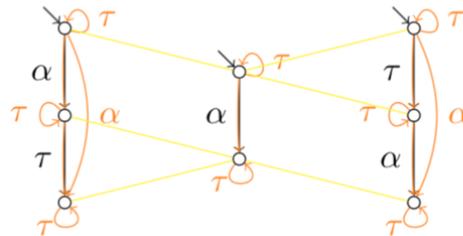
Dois estados s e t são *fracamente bissimilares*, e escreve-se $s \approx t$, se existe uma bissimulação fraca R que contém (s, t) , i.e., $(s, t) \in R$.

Exemplo

Mostrar que

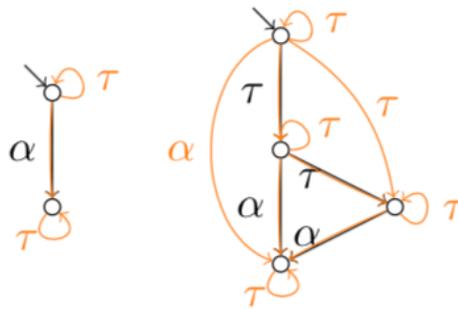
- $a.\tau.0 \approx a.0 \approx \tau.a.0$
- $\tau.(a.0 + \tau.a.0) \approx a.0$

$$a.\tau.0 \approx a.0 \approx \tau.a.0$$



A laranja a relação \Rightarrow e a amarelo a bissimulação R .

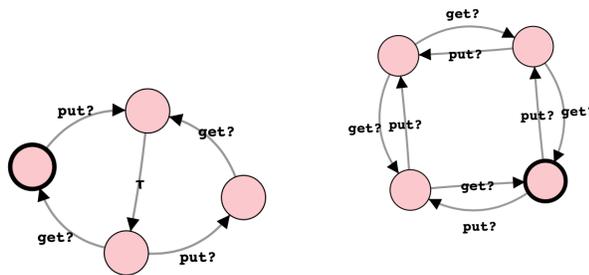
$$\tau.(a.0 + \tau.a.0) \approx a.0$$



Exemplo dos Buffers

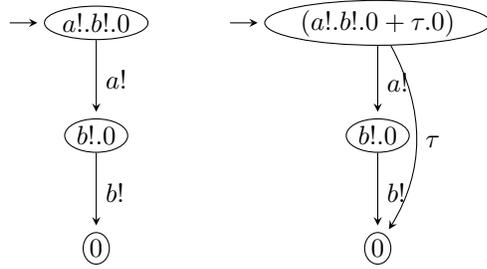
$$\begin{aligned} Buffer &:= put?.get?.Buffer \\ BufferL &:= put?.pass!.BufferL \\ BufferR &:= pass?.get?.BufferR \end{aligned}$$

$$(BufferL|BufferR) \setminus \{pass!, pass?\} \approx (Buffer|Buffer)$$



Processos não fracamente bissimilares

$$(a!.b!.0) \not\approx (a!.b!.0 + \tau.0)$$



- $A: a!.(b!.0 + \tau.0) \xrightarrow{\tau} 0$
- $D: a!.b!.0 \xrightarrow{\tau} a!.b!.0$
- $A: a!.b!.0 \xrightarrow{a!} b!.0$
- $D: \text{n\~o pode jogar com } a! \text{ porque } 0 \text{ n\~o tem transi\~c\~o fraca com } a!.$

Propriedades de \approx

Temos que

$$\approx = \bigcup_{R \text{ \u00e9 bissimula\~c\~o fraca}} R$$

Teorema 9.1. *A rela\~c\~o \approx \u00e9 de equival\u00eancia.*

Notar que se R \u00e9 uma bissimula\~c\~o fraca R^{-1} tamb\u00e9m \u00e9.

Teorema 9.2. *A rela\~c\~o \approx \u00e9 maior bissimula\~c\~o fraca .*

Mostrar que \approx \u00e9 uma bissimula\~c\~o fraca.

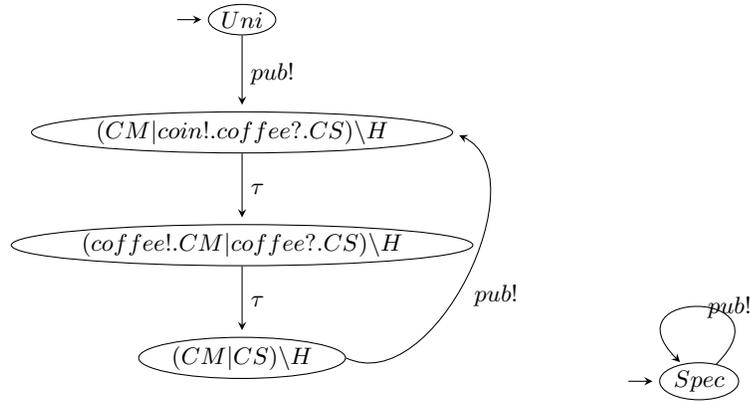
Teorema 9.3. *A bissimilaridade fraca \u00e9 estritamente mais grosseira que a bissimilaridade forte, i.e $\sim \subsetneq \approx$.*

Cada bissimula\~c\~o forte \u00e9 uma bissimula\~c\~o fraca. E vimos exemplos de processos que n\~o s\~o bissimilares fortes mas que s\~o bissimilares fracos. A bissimilaridade fraca tamb\u00e9m \u00e9 chamada de *equival\u00eancia observ\u00e1vel*.

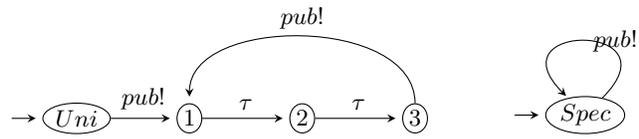
Ser\u00e1 que $Spec \approx Uni$?

Seja $Spec := pub!.Spec$ e

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



$Spec \approx Uni$



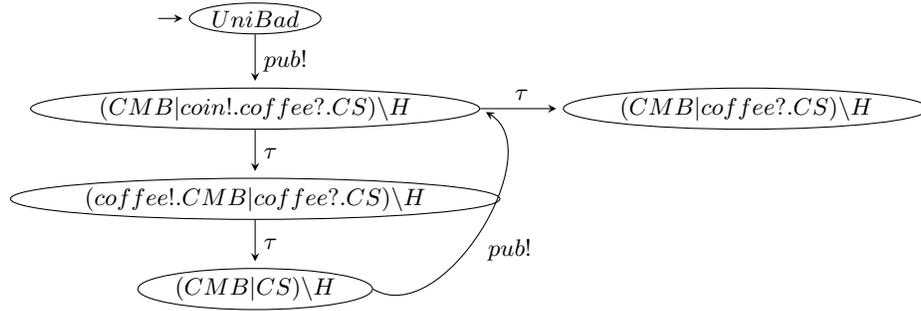
Jogo onde o defesa ganha. Configuração inicial $(Uni, Spec)$

- A: $Spec \xrightarrow{pub!} Spec$
- D: $Uni \xRightarrow{pub!} 3 \quad (3, Spec)$
- A: $3 \xrightarrow{pub!} 1$
- D: $Spec \xRightarrow{pub!} Spec \quad (1, Spec)$
- A: $1 \xrightarrow{\tau} 2$
- D: $Spec \xRightarrow{pub!} Spec \quad (2, Spec)$
- A: $2 \xrightarrow{\tau} 3$
- D: $Spec \xRightarrow{pub!} Spec \quad (3, Spec)$ (ciclo detectado)

Será que $Uni \approx Unibad$?

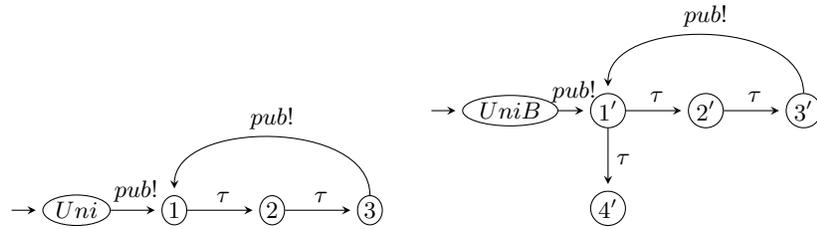
Seja

$$\begin{aligned}
CM &:= \text{coin?}.\text{coffee!}.CM \\
CMB &:= \text{coin?}.\text{coffee!}.CMB + \text{coin?}.CMB \\
CS &:= \text{pub!}.\text{coin!}.\text{coffee?}.CS \\
Uni &:= (CM|CS)\setminus\{\text{coin}, \text{coffee}\} \\
UniBad &:= (CMB|CS)\setminus\{\text{coin}, \text{coffee}\}
\end{aligned}$$



Neste caso $Uni \not\approx UniBad$, o que é bom porque $UniBad$ entra em deadlock e Uni não.

Jogo de bissimilaridade fraca $(Uni, UniBad)$



O atacante ganha. Configuração inicial $(Uni, UniBad)$

- $A: UniBad \xrightarrow{pub!} 1'$
- $D: Uni \xrightarrow{pub!} 3 \quad (3, 1')$
- $A: 1' \xrightarrow{\tau} 4'$
- $D: 3 \xrightarrow{\tau} 3 \quad (3, 4')$
- $A: 3 \xrightarrow{pub!} 1$
- $D: \text{não pode jogar e perde}$

A Bissimilaridade Fraca não é uma Congruência

Para todo $P, Q \in CCS$ se $P \approx Q$ e $\alpha \in Act$, $R \in CCS$ e $H \subseteq Com$,

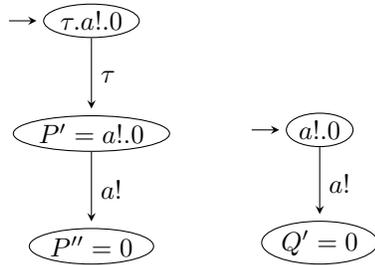
$$\begin{aligned} \alpha.P &\approx \alpha.Q \\ P|R &\approx Q|R \\ R|P &\approx R|Q \\ P \setminus H &\approx Q \setminus H \end{aligned}$$

mas não implica que

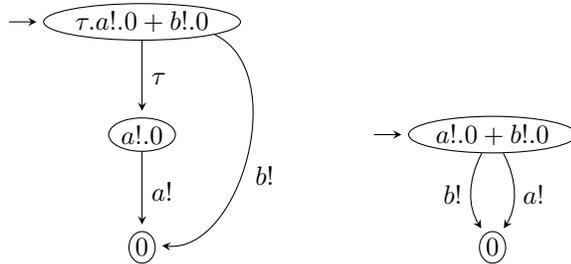
$$\begin{aligned} P + R &\approx Q + R \\ R + P &\approx R + Q \end{aligned}$$

Contraexemplo: $P = \tau.a!.0$, $Q = a!.0$ e $R = b!.0$.

- Mostrar que $P \approx Q$ considerando a relação $\mathcal{R} = \{(P, Q), (P', Q), (P'', Q')\}$ onde $P' = a!.0$ e $P'' = Q' = 0$ (mas em sistemas diferentes).



- Mostrar que $P + R \not\approx Q + R$.



- A: $\tau.a!.0 + b!.0 \xrightarrow{\tau} a!.0$
- D: $a!.0 + b!.0 \xrightarrow{\tau} a!.0 + b!.0$
- A: $a!.0 + b!.0 \xrightarrow{b!} 0$
- D: não pode jogar porque não há nenhuma transição fraca de $a!.0$ por $b!$

O problema está nos processos não guardados!

Impondo a congruência do +

Congruência para a Escolha

Dois processos $P, Q \in CCS$ são *congruentes para a escolha*, e escreve-se $P \approx^+ Q$ sse para todos os $R \in CCS$

$$P + R \approx Q + R$$

Pode-se provar que

- $\approx^+ \subseteq \approx$ (tomando $R = 0$)
- \approx^+ é uma congruência e é a mais grosseira contida em \approx

Embora seja esta a noção de igualdade de processos que se pretende, não é fácil de usar por causa de *para todos os $R \in CCS$* . Vamos ver uma definição que é equivalente mas mais perto das outras definições de bissimulação.

Congruência Observável

Congruência Observável

Dois processos $P, Q \in CCS$ são congruentes para a observação, e escreve-se $P \simeq Q$ sse para todos $\alpha \in Act$ verifica-se que:

- se $P \xrightarrow{\alpha} P'$ então existe Q' tal que $Q \xrightarrow{\tau} \xrightarrow{\alpha} \xrightarrow{\tau} Q'$ e $P' \approx Q'$.
- se $Q \xrightarrow{\alpha} Q'$ então existe P' tal que $P \xrightarrow{\tau} \xrightarrow{\alpha} \xrightarrow{\tau} P'$ e $P' \approx Q'$.

A diferença entre \approx e \simeq está que os passos internos iniciais não podem ser simulados por não haver movimento (lacete de τ).

Teorema 9.4. *Para todos os $P, Q \in CCS$ $P \simeq Q$ sse $P \approx^+ Q$.*

Exemplos

- $\tau.a.0 \not\approx a.0$
- $P|\tau.Q \not\approx \tau.(P|Q)$

Igualdade de comportamento

- Dois processos $P, Q \in CCS$ tem o mesmo comportamento sempre que $P \simeq Q$.

- É chamada a *igualdade de comportamento*
- as propriedades pretendidas:
 - relação de equivalência
 - relação de congruência
 - ter o mesmo conjunto de traços *fracos*
 - seja sensível a deadlocks
- $\sim \subseteq \simeq \subseteq \subseteq \approx$
- em vez de \simeq usa-se $=$.

Regras algébricas de =

Para além das satisfeitas por \sim temos

$$\begin{aligned}\alpha.\tau.P &= \alpha.P \\ P + \tau.P &= \tau.P \\ \alpha.(P + \tau.Q) &= \alpha.(P + \tau.Q) + \alpha.Q\end{aligned}$$

Nota: as regras algébricas permitem provar a igualdade de processos pela aplicação das regras (em especial em CCS_0) ou considerá-las axiomas num sistema de dedução.

Representantes Minimais

- Para cada LTS podemos considerar o *representante minimal* o menor LTS cujo comportamento é igual ao LTS dado.
- O representante minimal é único a menos de isomorfismo se for estados-finito.
- É obtido pelo quociente induzido pela relação de equivalência (\simeq).
- cada classe de equivalência é um estado

Como obter o representante minimal

Dado um processo P do CCS estados-finito:

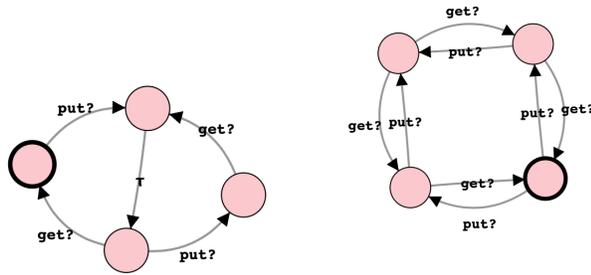
- Construir o LTS atíngivel de P .
- Calcular a relação \approx
- Considerar cada classe de equivalência um estado: $S = \{[s]_{\simeq} \mid s \in Reach(P)\}$

- Adicionar transições entre duas classes sempre que uma transição exista entre os elementos da respectiva classe.
- Podemos apagar as transições que seriam adicionadas por \Rightarrow e não estavam em \longrightarrow .
- Adicionar um lacete com τ se P tem um a τ -transição para a sua classe $[P]_{\approx}$.

Exemplo2 – Buffer

$$\begin{aligned} Buffer & := put?.get?.Buffer \\ BufferL & := put?.pass!.BufferL \\ BufferR & := pass?.get?.BufferR \end{aligned}$$

Será que $(BufferL|BufferR) \setminus \{pass!, pass?\} \sim (Buffer|Buffer)$?



Sim e o LTS mínimo equivalente é

