

Programação Concorrente - Exercícios 5

Bisimilaridade Fraca (ou Equivalência Observável) e Congruência Observável

1. Sendo

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

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

2. Sendo a bisimilaridade fraca \approx dado por

$$\approx = \bigcup_{R \text{ é bisimulação fraca}} R$$

Mostra que

- (a) A relação \approx é de equivalência.
- (b) A relação \approx é maior bisimulação fraca.
- (c) A bisimilaridade fraca é estritamente mais grosseira que a bisimilaridade forte, i.e. $\sim \subseteq \approx$.

3. Seja

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

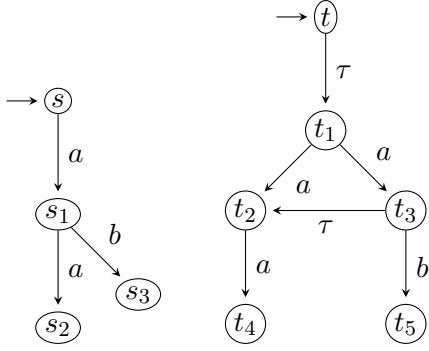
Será que $Spec \approx Unibad$?

4. Mostra que para qualquer $P, Q \in CCS$ e $\alpha \in Act$ as seguintes equivalências verificam-se (As chamamos leis τ de Milner):

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

Nota: constrói bisimulações fracas apropriadas.

5. Considera o seguinte sistema de transições e mostra que $s \not\approx t$ mostrando que um jogo de bisimilaridade fraca onde o atacante tem uma estratégia universal ganhadora (isto é ganha para todas as possíveis escolhas do defensor).



6. Considera o seguinte protocolo $Protocol := acc?.del!.Protocol$ que corresponde a um canal que se envia e recebe mensagens (como um buffer de uma posição). Seja a seguinte implementação

$$\begin{aligned}
 Send &:= acc?.Sending \\
 Sending &:= send!.Wait \\
 Wait &:= ack?.Send + error?.Sending \\
 Rec &:= trans?.Del \\
 Del &:= del!.Ack \\
 Ack &:= ack!.Rec \\
 Med &:= send?.Med1 \\
 Med1 &:= \tau.Err + trans!.Med \\
 Err &:= error!.Med
 \end{aligned}$$

Mostra que $(Send|Med|Rec)\{send, error, trans, ack\} \approx Protocol$. Verifica no Pseuco.Com.

7. Mostra que $a.0 + 0 \not\approx a.0 + \tau.0$ e conclui que \approx não é uma congruência em CCS.
8. Sendo \simeq a congruência observável mostra que
- $\tau.a \not\approx a$
 - $P|\tau.Q \not\approx \tau.(P|Q)$
9. Modelar o algoritmo de Peterson com dois processos para a exclusão mútua em CCS.

Para o processo $P_i, j, i = 1, 2$ e $i \neq j$.

```

while true do
  noncritical actions
   $b_i \leftarrow \text{true};$ 
   $k \leftarrow j;$ 
  while  $b_j \wedge k = j$  do
    skip;
  critical actions
   $b_i \leftarrow \text{false}$ 

```