Aula 5: Case Studies

Interaction & Concurrency Course Unit:: Reactive Systems Module

April 21, 2023

Case 1

Consider a description in CCS of a simple protocol, transferring data one at a time from one port to another, as follows:

$$Protocol \stackrel{\text{def}}{=} (Sender \mid Receiver)\backslash\{a, b\}$$

$$Sender \stackrel{\text{def}}{=} in.\overline{a}.b.Sender$$

$$Receiver \stackrel{\text{def}}{=} a.out.\overline{b}.Receiver$$

where $\overline{a}$ represents transmission of a message from the sender to the receiver, and $\overline{b}$ is an acknowledgement from the receiver to the sender, signalling that the latest message has been passed.
Prove that *Protocol* is observationally equivalent to the following specification:

$$PSpec \stackrel{\text{def}}{=} in.out.PSpec$$

Case 2

Consider the following specification of a control system for a crossing between a road and a railway. Events *car* and *train* modelled, respectively, a car or a train approaching the cross. Actions *up* e *dw* stand for the opening and closing of the protection bar to prevent cars to cross. Similarly, *green* and *red* model the semaphore for trains. Finally, events *ccross* and *tcross* come from sensors which register the actual cross of a car or a train, respectively.

$$Road \stackrel{\text{def}}{=} car.up.\overline{ccross}.\overline{dw}.Road$$

$$Rail \stackrel{\text{def}}{=} train.green.\overline{tcross}.\overline{red}.Rail$$

$$Signal \stackrel{\text{def}}{=} \overline{green}.red.Signal + \overline{up}.dw.Signal$$

$$C \stackrel{\text{def}}{=} (Road \mid Rail \mid Signal)\backslash\{green, red, up, dw\}$$

- Explain the behaviour of this process.
- Present the transition graph corresponding to process C.

Case 3

Consider the following solution to a simple mutual exclusion problem, expressed in CCS:

$$Mutex \stackrel{\text{def}}{=} (Proc \mid Sem \mid Proc)\backslash\{p, v\}$$

$$Proc \stackrel{\text{def}}{=} \overline{p}.enter.exit.\overline{v}.Proc$$

$$Sem \stackrel{\text{def}}{=} p.v.Sem$$

- Define a transition graph of the system.
- Present a possible specification, observationally equivalent, of Mutex.
- Prove that equivalence.

Case 4

Define a variant of $Mutex$, where one of the processes is faulty and may deadlock after exiting the critical region (the deadlock behaviour modelled as Nil). Is the faulty system observationally equivalent to $Mutex$? Has it the same specification?

Defina uma variante de Mutex, onde um dos processos está com defeito e pode travar após sair da região crítica (o comportamento de impasse modelado como Nil). O sistema defeituoso é observacionalmente equivalente ao Mutex? Tem a mesma especificação?

CASE 1:

$\text{Protocol} \stackrel{def}{=} (\text{Sender} \mid \text{Receiver}) \setminus \{a, b\}$

$\text{Sender} \stackrel{def}{=} in \cdot \bar{a} \cdot b \cdot \text{Sender}$

$\text{Receiver} \stackrel{def}{=} a \quad out \cdot \bar{b} \cdot \text{Receiver}$

→ Processos em paralelo (comunicação entre si)

① $(\text{Sender} \mid \text{Receiver}) \setminus \{a, b\}$

$\downarrow in$

② $(\bar{a} \cdot b \cdot \text{Sender} \mid \text{Receiver}) \setminus \{a, b\}$

$\downarrow \tau$

Nunca poderia por uma trans. por a ou ā pois estes não são visíveis do ext.

③ $(b \cdot \text{Sender} \mid out \cdot \bar{b} \cdot \text{Receiver}) \setminus \{a, b\}$

$\downarrow out$

④ $(b \cdot \text{Sender} \mid \bar{b} \cdot \text{Receiver}) \setminus \{a, b\}$

$④ \xrightarrow{\tau} ①$

$\text{Pspec} = in \cdot out \cdot \text{Pspec}$

⑤ Pspec

$\downarrow in$

⑥ $out \cdot \text{Pspec}$

$⑥ \xrightarrow{out} ⑤$

Uma relação de Bissimilação não tem que ser simétrica

(①, ⑤)

$① \xrightarrow{in} ②$ $\qquad ⑤ \xRightarrow{in} ⑥$ $\qquad (②, ⑥)$

(②, ⑥)

$② \xrightarrow{\tau} ③$ $\qquad ⑥ \xRightarrow{\tau} ⑥$ $\qquad (③, ⑥)$

(③, ⑥)

$③ \xrightarrow{out} ④$ $\qquad ⑥ \xRightarrow{out} ⑤$ $\qquad (④, ⑤)$

(④, ⑤)

④ $\xrightarrow{J}$ ①　　　⑤ $\xrightarrow{J}$ ⑤　　　(①, ⑤)

CASE 2:

$C \xrightarrow{tram} (Road \mid green \cdot \overline{tcross} \cdot \overline{red} \cdot Rail \mid signal) \backslash 2 \ldots 4$

$\xrightarrow{J\ (green,\ \overline{green})} (Road \mid \overline{tcross} \cdot \overline{red} \cdot Rail \mid red \cdot signal)$

$\xrightarrow{\overline{tcross}} (Road \mid \overline{red} \cdot Rail \mid red \cdot signal)$

$\xrightarrow{J\ (red,\ \overline{red})} C$

$C \xrightarrow{car} (up \cdot \overline{cross} \cdot \overline{dw} \cdot Road \mid Rail \mid signal) \backslash 2 \ldots 4$

$\xrightarrow{J\ (up \mid \overline{up})} (\overline{cross} \cdot \overline{dw} \cdot Road \mid Rail \mid dw \cdot signal) \backslash 2 \ldots 4$

$\xrightarrow{\overline{cross}} (\overline{dw} \cdot Road \mid Rail \mid dw \cdot signal) \backslash 2 \ldots 4$

$\xrightarrow{J\ (dw \mid \overline{dw})} C$

Case 3

$$\mathrm{Mutex} = (\overset{P}{\mathrm{PROC}} \mid \overset{S}{\mathrm{sem}} \mid \overset{P}{\mathrm{PROC}}) \setminus \{p, v\}$$

$$\mathrm{PROC} = \bar{p} \cdot \mathrm{enter} \cdot \mathrm{exit} \cdot \bar{v} \cdot \mathrm{PROC}$$

$$\mathrm{sem} = p \cdot v \cdot \mathrm{sem}$$

$$(P \mid S \mid P) \setminus \{p, v\}$$

$$(P \mid S \mid P) \setminus \{p, v\} \xrightarrow{\tau} (\mathrm{enter} \cdot \mathrm{exit} \cdot \bar{v} \cdot \mathrm{PROC}) \mid v \cdot \mathrm{sem} \mid P$$

$$(P \mid S \mid P) \setminus \{p, v\} \xrightarrow{\tau} (P \mid v \cdot \mathrm{sem} \mid \mathrm{enter} \cdot \mathrm{exit} \cdot \mathrm{PROC})$$

$$(\mathrm{enter} \cdot \mathrm{exit} \cdot \bar{v} \cdot \mathrm{PROC}) \mid v \cdot \mathrm{sem} \mid P \xrightarrow{\mathrm{enter}} \mathrm{exit} \cdot \bar{v} \cdot \mathrm{PROC} \mid v \cdot \mathrm{spec} \mid P$$

$$\mathrm{exit} \cdot \bar{v} \cdot \mathrm{PROC} \mid v \cdot \mathrm{spec} \mid P \xrightarrow{\mathrm{exit}} \bar{v} \cdot \mathrm{PROC} \mid v \cdot \mathrm{spec} \mid P$$

$$\bar{v} \cdot \mathrm{PROC} \mid v \cdot \mathrm{spec} \mid P \xrightarrow{\tau} (P \mid S \mid P) \setminus \{p, v\}$$

case 4 imp.

FMutex = (Proc | sem | FProc) \ {...}

FProc = $\bar{p}$ · enter · exit · ($\bar{v}$ · Nil + $\bar{v}$ · FProc)

falha ou funciona

(Proc | sem | FProc) \ {...}

$\downarrow \bar{v}$

Proc | v · sem | enter · exit · ($\bar{v}$ · Nil + $\bar{v}$ · FProc)

$\downarrow$ enter

...

$\downarrow$ exit

(Proc | v · sem | $\bar{v}$ Nil + $\bar{v}$ · FProc)

$\swarrow \bar{v}$ $\searrow \bar{v}$

(Proc | sem | Nil) (Proc | sem | FProc)

$\downarrow$?

exit · ($\bar{v}$ · Nil + $\bar{v}$ · FProc)

$\downarrow$ exit

(Proc | v · sem | $\bar{v}$ · Nil + $\bar{v}$ · FProc)

$\downarrow \bar{v}$

Proc | v · sem | FProc

Temos que ter no caso que falha:

$\bar{v}$ · Nil → senão não conseguimos transitar para Nil.

Transições visíveis:

SpecMutex = enter . exit . SpecMutex

Temos então,

Mutex ≈ SpecMutex

FMutex ≈ SpecMutex

Mutex ≠ FMutex