Cyber-Physical Programming
TPC-1

Melânia Pereira
pg47520@alunos.uminho.pt

30 de março de 2022

Exercise 1

Part 1.1.

O processo $c.(a.0 \parallel b.0)$ comunica com o exterior através de 3 canais: a, b e c. Recebe uma comunicação no canal c e, depois disso, pode receber comunicação no canal a, e depois termina a sua execução, ou no canal b, terminando também de seguida a sua execução.

Part 1.2.

$$\frac{}{c.(a.0 \parallel b.0)}\;^{(pre)} \xrightarrow{c} \frac{\dfrac{}{a.0 \xrightarrow{a} 0}\;^{(\text{pre})}}{(a.0 \parallel b.0) \xrightarrow{c} 0}\;^{(comp.par.)}$$

$$\frac{}{c.(a.0 \parallel b.0)}\;^{(pre)} \xrightarrow{c} \frac{\dfrac{}{b.0 \xrightarrow{b} 0}\;^{(pre)}}{(a.0 \parallel b.0) \xrightarrow{b} 0}\;^{(comp.par.)}$$

Exercise 2

Part 2.1.

O processo `rec` $X.(a.X + a.a.X)$ é um processo recursivo, que tem um canal de comunicação - a - e que se pode desenvolver por dois diferentes caminhos: pela esquerda da soma, onde recebe uma comunicação por a e volta ao estado inicial; ou pela direita da soma, onde recebe uma comunicação por a, seguida de outra comunicação por a e depois retorna ao estado inicial, onde pode novamente se desenvolver por uma das opções descritas.

Já o processo `rec` $X.a.X$ é também um processo recursivo com apenas um canal de comunicação - a - mas que se desenvolve apenas por um caminho que é a receção de uma comunicação por a e volta ao estado inicial, onde fica pronta a, novamente, receber uma comunicação por a.

Part 2.2.

Para provar a equivalência entre estes dois processos, começo por encontrar o sistema de transições de cada um seguindo as regras da semântica da linguagem CCS. Em cada um dos sistemas será possível distinguir o conjunto das suas transições.

Comecemos pelo processo `rec` **X.a.X**:

$$\cfrac{\cfrac{\cfrac{}{a.(\mathtt{rec}X.a.X) \xrightarrow{a} \mathtt{rec}X.a.X}\;^{(\text{pre})}}{(a.X)[\mathtt{rec}X.a.X/X] \xrightarrow{a} \mathtt{rec}X.a.X}\;^{(\text{substituição})}}{\mathtt{rec}X.a.X \xrightarrow{a} \mathtt{rec}X.a.X}\;^{(\text{rec})}$$

Conclui-se que existe uma transição possível, a partir do estado $\texttt{rec}X.a.X$, por a, para ele próprio.

Agora, para o processo **rec X.(a.X + a.a.X)**, por uma questão de simplificação, seja $P = \texttt{rec}X.(a.X + a.a.X)$. Assim, temos:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{a.P \xrightarrow{a} P}\,{\scriptstyle (pre)}}{a.P + a.a.P \xrightarrow{a} P}\,{\scriptstyle (soma)}}{(a.X + a.a.X)[P/X] \xrightarrow{a} P}\,{\scriptstyle (substituição)}}{P \xrightarrow{a} P}\,{\scriptstyle (rec)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{a.a.P \xrightarrow{a} a.P}\,{\scriptstyle (pre)}}{a.P + a.a.P \xrightarrow{a} a.P}\,{\scriptstyle (soma)}}{(a.X + a.a.X)[P/X] \xrightarrow{a} a.P}\,{\scriptstyle (substituição)}}{P \xrightarrow{a} P}\,{\scriptstyle (rec)} \qquad \dfrac{}{a.P \xrightarrow{a} P}\,{\scriptstyle (pre)}$$

Com a aplicação destas regras, conseguimos encontrar os seguintes sistemas de transições para cada processo:

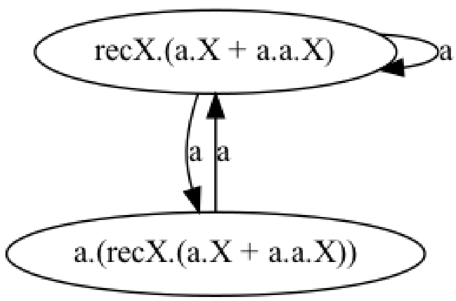

Figura 1: Sistema de transições do processo $\texttt{rec}X.(a.X + a.a.X)$

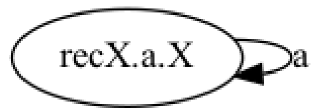

Figura 2: Sistema de transições do processo $\texttt{rec}X.a.X$

Sabendo o sistema de transições dos dois processos, pode-se começar a prova de que são equivalentes.

Para provar que dois sistemas são equivalentes, é necessário provar que a observação do estado inicial de um dos sistemas é igual à observação do estado inicial do outro e que o estado seguinte num sistema é equivalente ao estado seguinte no outro.

Isto traduz-se num "jogo"que se demonstra a seguir.

- Começa por se provar a equivalência dos estados iniciais: $\texttt{rec}X.(a.X + a.a.X) \sim \texttt{rec}X.a.X$
  1. $\texttt{rec}X.(a.X + a.a.X) \xrightarrow{s} a.(\texttt{rec}X.(a.X + a.a.X))$
     $\Longrightarrow$ $\texttt{rec}X.a.X \xrightarrow{a} \texttt{rec}X.a.X$
     $\Longrightarrow$ $a.(\texttt{rec}X.(a.X + a.a.X)) \sim \texttt{rec}X.a.X \rightarrow$ necessário provar
  2. $\texttt{rec}X.(a.X + a.a.X) \xrightarrow{s} \texttt{rec}X.(a.X + a.a.X)$
     $\Longrightarrow$ $\texttt{rec}X.a.X \xrightarrow{a} \texttt{rec}X.a.X$
     $\Longrightarrow$ $\texttt{rec}X.(a.X + a.a.X) \sim \texttt{rec}X.a.X \rightarrow$ voltamos à equivalência inicial, aquela que queremos provar
- Agora, queremos provar $a.(\texttt{rec}X.(a.X + a.a.X)) \sim \texttt{rec}X.a.X$
  1. $a.(\texttt{rec}X.(a.X + a.a.X)) \xrightarrow{s} \texttt{rec}X.(a.X + a.a.X)$
     $\Longrightarrow$ $\texttt{rec}X.a.X \xrightarrow{a} \texttt{rec}X.a.X$
     $\Longrightarrow$ $\texttt{rec}X.(a.X + a.a.X) \sim \texttt{rec}X.a.X \rightarrow$ voltamos à equivalência inicial, aquela que queremos provar; por isso, o jogo termina aqui.

Foi provada a equivalência entre todos os estados dos sistemas de transição, logo, conclui-se que os dois sistemas são equivalentes.

Exercise 4

Part 4.1.

O processo P corresponde à execução em paralelo de um conjunto de outros processos - I, S, P_1, P_2, P_3, P_4 - com um conjunto de canais privados - st_1, st_2, st_3, st_4, end. De acordo com a descrição dada, estes processos devem ser, de facto, executados em paralelo.

O processo I inicia a execução enviando sinais de *start* nos canais st_i, por ordem, enquanto isso, os processos P_i recebem esse sinal e iniciam também eles a sua execução, realizando as suas tarefas e enviando sinais nos canais a_i e b_i ($1 <= i <= 4$), quando terminam, enviam sinais de término no canal end. Estes sinais são então recebidos pelo processo S que funciona como um *scheduler*, esperando por 4 sinais no canal end e, só depois de receber os 4, envia novamente sinais de *start* (da mesma forma que o processo de inicialização I); com isto, acontece o que é descrito no enunciado, em que o processo P_1 só reinicia quando todos os outros processos tiverem terminado a sua execução.

Todos estes processos são recursivos (à exceção do processo de inicialização I, pois a reinicialização será responsabilidade do processo S), ou seja, realizam as suas tarefas de forma repetitiva, tal como descrito.

Assim, é seguro dizer que o processo P corresponde, de facto, à descrição do enunciado.

Part 4.2.

Para que os processos se possam sincronizar entre si, precisam de comunicar uns com os outros, então, sendo P_1 o primeiro processo a começar, este deve comunicar com o processo seguinte (P_2) para que este saiba que pode também iniciar a sua execução. E o mesmo para os restantes processos. A ideia é que cada processo deve comunicar ao seguinte que este pode iniciar a execução.

Assim, P_1 inicia e deve, em paralelo com a execução das suas tarefas, enviar uma mensagem pelo canal st_2 para o processo P_2, este irá iniciar e fazer o mesmo para o processo P_3. Assim, todos os processos iniciam por ordem.

O processo P_1, sendo o primeiro a (re)iniciar, deve saber quando todos os outros terminaram as suas tarefas, por isso, terá que receber essa informação dos restantes 3 processos pelo canal end, que a enviarão quando terminarem a sua execução. Quando receber as 3 comunicações, P_1 saberá que pode reiniciar a sua execução.

Desta forma, todos os processos iniciam por ordem e P_1 só reinicia quando todos tiverem terminado.

Então, P terá a nova forma $P = (P_1 \parallel P_2 \parallel P_3 \parallel P_4)\backslash\{st_2, st_3, st_4, end\}$ onde:

$$\begin{aligned}
P_1 &= \texttt{rec}\ Y_1.(\overline{st_2} \parallel a_1.b_1.end.end.end.Y_1) \\
P_2 &= \texttt{rec}\ Y_2.(st_2.(\overline{st_3} \parallel a_2.b_2.\overline{end}.Y_2)) \\
P_3 &= \texttt{rec}\ Y_3.(st_3.(\overline{st_4} \parallel a_3.b_3.\overline{end}.Y_3)) \\
P_4 &= \texttt{rec}\ Y_4.st_4.a_4.b_4.\overline{end}.Y_4
\end{aligned}$$