



UNIVERSIDADE DO MINHO

DEPARTAMENTO DE INFORMÁTICA

MESTRADO EM ENGENHARIA INFORMÁTICA

MÉTODOS FORMAIS DE PROGRAMAÇÃO

Programação Cíber-Física
Teste Assíncrono

Ariana Lousada (PG47034)

31 de março de 2022

Conteúdo

1	Exercício 1	3
1.1	Parte 1.1	3
1.2	Parte 1.2	3
2	Exercício 2	3
2.1	Parte 2.1	3
2.2	Parte 2.2	3
3	Exercício 3	5
4	Exercício 4	6
4.1	Parte 4.1	6
4.2	Parte 4.2	7

1 Exercício 1

1.1 Parte 1.1

- $c.(a.0 \parallel b.0)$

Esta expressão representa uma comunicação aberta através do canal c , seguida de duas comunicações abertas em paralelo através dos canais a e b . De notar que apenas é recebida informação, nunca enviada.

1.2 Parte 1.2

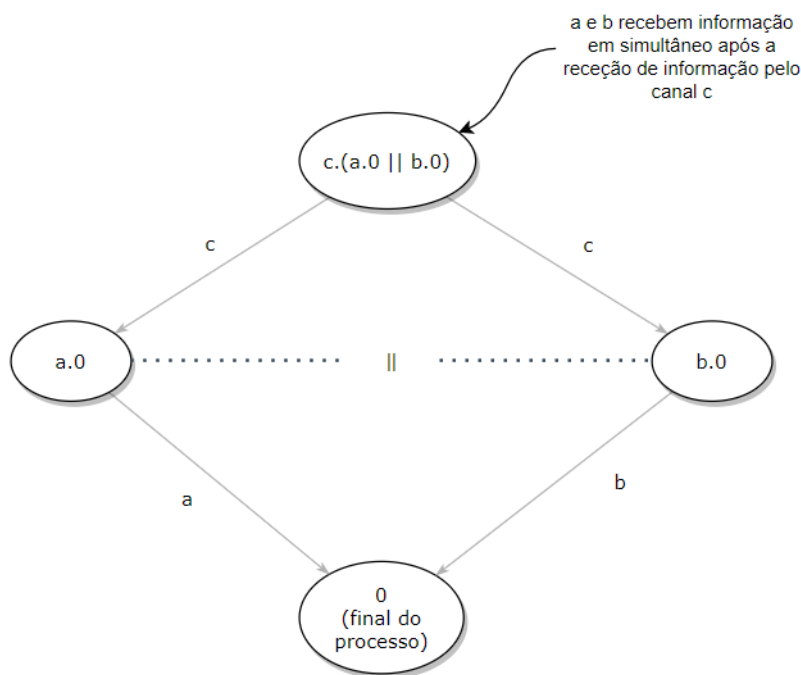


Figura 1

2 Exercício 2

2.1 Parte 2.1

- $recX.(a.X + a.a.X)$

É recebida informação através do canal a ou recebida duas vezes(sequencialmente) também através do canal a . Após uma destas hipóteses o mesmo processo repete-se.

- $recX.a.X$

É recebida informação através do canal a ; após a receção de informação o mesmo processo volta-se a repetir.

2.2 Parte 2.2

De modo a provar a equivalência entre as duas expressões, é necessário desenvolver cada uma individualmente.¹

- $recX.(a.X + a.a.X)$

¹Para este exercício, foram usadas as regras de prova apresentadas nos slides "Transition Systems" da unidade curricular (<https://haslab.github.io/MFP/PCF/2122/transitionSystems.pdf>)

$$\begin{array}{c}
\begin{array}{cc}
(1) & (2) \\
\hline
a. (\text{rec } X. P) & + a.a. (\text{rec } X. P)
\end{array} \\
= \\
\hline
\frac{(a.X + a.a.X) [\text{rec } X. (a.X + a.a.X) / X]}{\text{rec } X. \underbrace{(a.X + a.a.X)}_P}
\end{array}$$

Slides 16,
Regra da recursividade

Figura 2

$$\begin{array}{c}
a. (\text{rec } X. P) \xrightarrow{a} (\text{rec } X. P) \\
\hline
(1) \quad a. (\text{rec } X. P) + a.a. (\text{rec } X. P) \xrightarrow{a} (\text{rec } X. P)
\end{array}$$

Slides 16,
Regra da soma

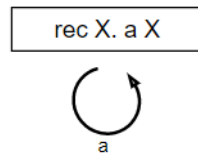


Figura 3

$$\begin{array}{c}
a. (\text{rec } X. P) \xrightarrow{a} (\text{rec } X. P) \\
\hline
a.a. (\text{rec } X. P) \xrightarrow{a} a.(\text{rec } X. P) \\
\hline
(2) \quad a. (\text{rec } X. P) + a.a. (\text{rec } X. P) \xrightarrow{a} (\text{rec } X. P)
\end{array}$$

Slides 16,
Regra da soma

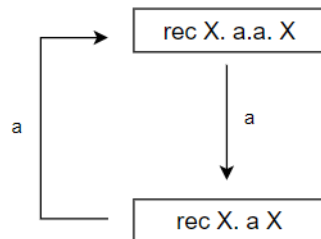


Figura 4

Se analisarmos os processos (1) e (2), podemos concluir que de ambos resulta uma "lista infinita" de a 's.

Portanto, em termos de resultados, na expressão

$$\text{rec } X. (\underbrace{a.X}_1 + \underbrace{a.a.X}_2)$$

executar a opção 1 ou a opção 2 vai dar exatamente o mesmo resultado.

- $\text{rec } X. a.X$

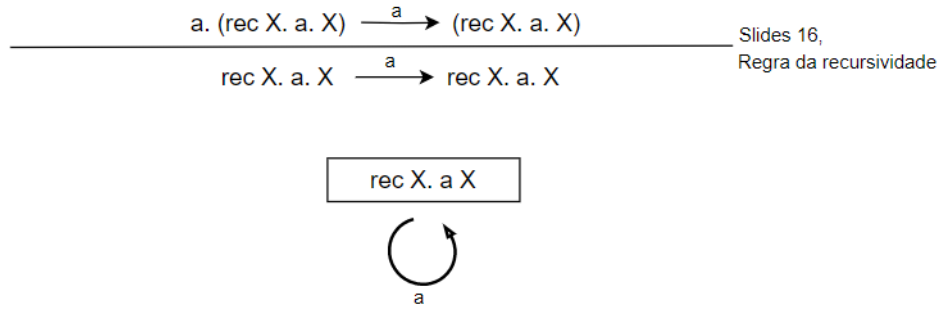


Figura 5

A partir da conclusão exposta na secção anterior e analisando a figura 3 podemos então concluir que

$$\text{rec } X. (a.X + a.a.X) \sim \text{rec } X. a.X \quad (1)$$

3 Exercício 3

Se considerarmos a terceira regra de paralelismo entre processos do slide 16 temos²:

$$\frac{Q \xrightarrow{\alpha} Q' \quad P \xrightarrow{\bar{\alpha}} P'}{Q \parallel P \xrightarrow{\tau} Q' \parallel P'} \quad \text{Slides 16, 3a Regra do Paralelismo}$$

$$\frac{P \xrightarrow{\alpha} P' \quad Q \xrightarrow{\bar{\alpha}} Q'}{P \parallel Q \xrightarrow{\tau} P' \parallel Q'} \quad \text{Slides 16, 3a Regra do Paralelismo}$$

Figura 6

Assim temos:

$$Q \xrightarrow{\alpha} Q'$$

$$Q \xrightarrow{\bar{\alpha}} Q'$$

e

$$P \xrightarrow{\alpha} P'$$

²Para este exercício, foram usadas as regras de prova apresentadas nos slides "Transition Systems" da unidade curricular (<https://haslab.github.io/MFP/PCF/2122/transitionSystems.pdf>)

$$P \xrightarrow{\bar{\alpha}} P'$$

Com isto, podemos chegar à conclusão que, neste caso

$$\alpha \sim \bar{\alpha}$$

Assim podemos inferir:

$$P \xrightarrow{\alpha} P' \sim P \xrightarrow{\bar{\alpha}} P'$$

$$Q \xrightarrow{\alpha} Q' \sim Q \xrightarrow{\bar{\alpha}} Q'$$

Assim,

$$Q \parallel P \xrightarrow{\tau} Q' \parallel P' \sim P \parallel Q \xrightarrow{\tau} P' \parallel Q'$$

Consequentemente,

$$Q \parallel P \sim P \parallel Q$$

4 Exercício 4

4.1 Parte 4.1

De acordo com o enunciado, o processo P corresponde a

$$P = (I \parallel S \parallel P1 \parallel \dots \parallel P4) \setminus \{st1, \dots, st4, end\} \quad (2)$$

Começando por analisar I :

$$I = \overline{st_1} \dots \overline{st_4}.0 \quad (3)$$

Como é possível analisar I tem um papel semelhante a um tipo de "inicializador" do sistema. Começa por mandar um sinal a cada processo, de modo a permitir o começo das tarefas de cada um. Após o envio de um sinal pelos canais respectivos a cada processo, termina.

Analisando S :

$$S = rec\ X. end.end.end.end.\overline{st_1} \dots \overline{st_4}.X \quad (4)$$

Neste sistema, S tem um papel de *scheduler*. É responsável pela manutenção da ordem de execução dos processos. Este espera por quatro sinais de término de tarefa (aguarda que todos os quatro processos terminem a sua execução) para mandar sinais de recomeço para os mesmos. O *scheduler* repete esta receção e envio recursivamente.

Analisando P_i :

$$P_i = rec\ Y_i.st_i.a_i.b_i.\overline{end}.Y_i \quad (1 \leq i \leq 4) \quad (5)$$

Cada processo, recursivamente, recebe o seu sinal para começar a sua tarefa, executa-a e envia um sinal de finalização.

Se considerarmos o sistema P como um sistema simplificado no qual as comunicações são instantâneas, uma vez que os sinais são enviados sequencialmente para o P_1 , P_2 , P_3 e P_4 por parte do I e do S , são recebidos também pela mesma ordem; logo, os processos começam e recomeçam sempre por ordem crescente.

Com isto, pode-se então concluir que P corresponde à descrição exposta no enunciado.

4.2 Parte 4.2

Para retirar o *scheduler* do sistema e garantir que todos processos terminam as suas respectivas tarefas antes do seu recomeço assim como a sua execução ordenada, a resolução deste exercício foi baseada no protocolo *peer-to-peer* aplicado na área de Redes de Computadores.

Nesta definição do processo P , tal como na anterior, os processos começam os seus ciclos através de um inicializador I . Os quatro envios de sinalização final são enviados de modo a possibilitar o começo do processo 1. Este inicializador executa apenas uma vez.

Para a execução dos processos, o primeiro só recomeça quando recebe quatro sinais de término de tarefas, o que garante que este só executa após a finalização das tarefas de todos os processos, inclusive dele próprio.

Cada processo, ao receber o seu sinal de começo, envia um sinal de começo para o processo seguinte e executa a sua tarefa, enviando um sinal de finalização para o primeiro processo. O processo quatro, como não tem qualquer processo a seguir dele mesmo, apenas envia um sinal de término após conclusão da sua tarefa.

$$P = (I \parallel P1 \parallel \dots \parallel P4) \setminus \{st1, \dots, st4, end\} \quad (6)$$

$$I = \overline{end}.\overline{end}.\overline{end}.\overline{end}.\overline{st_1}.0 \quad (7)$$

$$P_1 = rec\ Y_1.end.end.end.end.st_1.\overline{st_2}.a_1.b_1.\overline{end}.Y_1 \quad (8)$$

$$P_i = rec\ Y_i.st_i.\overline{st_{i+1}}.a_i.b_i.\overline{end}.Y_i \quad (2 \leq i \leq 3) \quad (9)$$

$$P_4 = rec\ Y_4.st_4.a_4.b_4.\overline{end}.Y_4 \quad (10)$$