

Nome: Frederico Emanuel Almeida Lopes

1. Considera a especificação de um buffer com capacidade 1.

$$B := put?.get?.B$$

Para  $n \geq 1$  podemos iterativamente definir um buffer de capacidade  $n$ , onde  $B_i^n$  indica um buffer de capacidade  $n$  com  $0 \leq i \leq n$  elementos.

$$\begin{aligned} B_0^n &:= put?.B_1^n \\ B_i^n &:= put?.B_{i+1}^n + get?.B_{i-1}^n, \quad 0 < i < n \\ B_n^n &:= get?.B_{n-1}^n \end{aligned}$$

- (a) Verifica que  $B \sim_{iso} B_0^1$  (desenha os seus diagramas).
- (b) Verifica que  $B_0^2 \sim B_0^1 | B_0^1$  (desenha os seus diagramas).
- (c) Mostra que para  $n \geq 1$ ,  $B_0^n \sim \underbrace{B_0^1 | B_0^1 | \dots | B_0^1}_n$ .

Nota: mostrar que a seguinte relação é uma bisimulação

$$\mathcal{R} = \{(B_i^n, B_{i_1}^1 | B_{i_2}^1 | \dots | B_{i_n}^1) \mid i_j \in \{0, 1\} \wedge \sum_{j=1}^n i_j = i\}$$

2. Considera o algoritmo de Hyman para a exclusão mútua. As variáveis  $b_i$  são booleanas e  $k$  é inteira. Para o processo  $P_i$ ,  $j, i = 1, 2$  e  $i \neq j$ .

```

while true do
  noncritical actions
  b_i ← true;
  while k ≠ j do
    while b_j do
      skip;
    k ← i;
  critical actions;
  b_i ← false;

```

- (a) Implementa o algoritmo directamente em CCS explicando sucintamente os processos usados e qual o processo que corresponde ao algoritmo (que deverá ter o nome Hyman). Assinala a zona crítica com as ações  $enter_i$  e  $exits_i$  para  $i = 1, 2$ .
- (b) Fornece um ficheiro correspondente para o pseuco.com e testa para verificar (traços aleatórios) que realmente a exclusão mútua ocorre (ou não). Qual o problema caso não?
- (c) Supõe que se modela a entrada e a saída da zona crítica por

$$MutexSpec := enter1.exit1.MutexSpec + enter2.exit2.MutexSpec$$

Será verdade que  $Hyman \approx MutexSpec$ ?

- (d) Implementa o algoritmo na linguagem Pseuco e testa também a exclusão mútua usando traços aleatórios.
3. Considera o problema do Jantar dos Filósofos, como descrito em "The Little Book of Semaphores", Cap. 4.4.
    - (a) Estuda e tenta resolver os puzzles propostos nesse capítulo. Caso queiras podes apresentar as tuas soluções se diferentes das indicadas
    - (b) Implementa a(s) soluções #1 e # 2 indicadas usando a linguagem Pseuco.
    - (c) Justifica a verificação das propriedades pretendidas usando a ferramenta.