Verificação Formal

Teste

20 de Maio de 2022

1. Considere o seguinte algoritmo de exclusão mútua para N processos proposto por Szymański. Cada processo $1 \leq i \leq N$ tem uma variável $flag_i$, inicialmente com o valor 0, em que apenas i escreve, mas que pode ser lida por todos os outros. O algoritmo executado por cada processo i é o seguinte.

```
                 while (true) {
idle ::            ...
                   flag_i ← 1
wait_0 ::          await (∀ 1 ≤ k ≤ N · flag_k ∈ {0,1,2})
                   flag_i ← 3
check ::           if (∃ 1 ≤ k ≤ N · flag_k = 1) {
                     flag_i ← 2
wait_1 ::            await (∃ 1 ≤ k ≤ N · flag_k = 4)
                   }
                   flag_i ← 4
wait_2 ::          await (∀ 1 ≤ k < i · flag_k ∈ {0,1})
critical ::        ...
                   await (∀ i < k ≤ N · flag_k ∈ {0,1,4})
                   flag_i ← 0
                 }
```

   (a) *(6 pontos)* Especifique este algoritmo em TLA+ ou nuXmv (neste caso para $N = 3$). Considere apenas as transições entre as *labels* assinaladas.

   (b) *(1 ponto)* Qual o número mínimo de valores necessários para a variável de controlo (o *program counter*) para especificar este algoritmo? Justifique.

   (c) *(1 ponto)* Como verificar que este algoritmo garante a exclusão mútua?

   (d) *(1 ponto)* Como verificar que nenhum processo espera indefinidamente para entrar na região crítica?

   (e) *(1 ponto)* Como pode obter um exemplo de execução onde todos os N processos entram recorrentemente na região crítica?

   (f) *(1 ponto)* Como verificar que a espera para entrar na região crítica é limitada, mais concretamente, que enquanto há outros processos à esspera um processo não pode entrar duas vezes seguidas na região crítica?

   (g) *(1 ponto)* A propriedade anterior é uma propriedade de *safety* ou *liveness*? Justifique.

2. Considere um canal de comunicação entre dois processos implementado com um array circular de tamanho N, suportando duas operações send(v) e read(v), onde $0 \leq v < V$ é um dos V valores possíveis que podem ser enviados.

   (a) *(4 pontos)* Especifique este canal de comunicação em TLA+ ou nuXmv (neste caso para $N = 3$ e $V = 4$).

   (b) *(1 ponto)* Como verificar que o canal nunca contém valores inválidos?

   (c) *(1 ponto)* Como pode obter um exemplo de execução onde o canal fica totalmente cheio e depois volta a ficar vazio?

   (d) *(1 ponto)* Como verificar que todos os valores lidos foram previamente enviados?

   (e) *(1 ponto)* Como verificar que, se numa execução deixarem de ocorrer envios, então o canal vai necessariamente ficar vazio?

Nota: quando forem necessárias condições de justiça para verificar uma propriedade, coloque as mesmas como pré-condição dessa propriedade.