Ficha de Exercícios 5: Processos e Concorrência

Luís Soares Barbosa

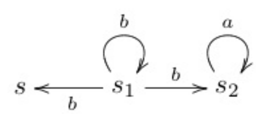
Exercício 1

Mostre que

$$\mathsf{true} \stackrel{\text{abv}}{=} \nu X\,.\,X \quad \text{e} \quad \mathsf{false} \stackrel{\text{abv}}{=} \mu X\,.\,X$$

Exercício 2

Considere o seguinte sistema de transição de estados Determine que conjuntos $S \subseteq \{s, s_1, s_2\}$ são solução das seguintes

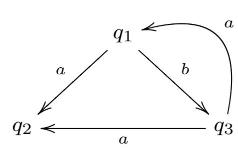


equações em $\mathcal{P}\mathbb{P}$:

$$\begin{aligned} \|X\| &= \langle a \rangle\, \mathsf{true} \vee [b]\, X \\ \|X\| &= \langle a \rangle\, \mathsf{true} \vee ([b]\, X \wedge \langle b \rangle\, \mathsf{true}) \end{aligned}$$

Exercício 3

Calcule $\|[b]\,\mathsf{false} \wedge [a]\,X\|(\{q2\})$ relativamente ao seguinte sistema de transição:



Exercício 4

Mostre que a função $\|.\|$ é monótona. Discuta o efeito de introduzir negação na lógica sobre a monotonia de $\|.\|$ (com a óbvia extensão a formulas $\neg\phi$).

Exercício 5

Um sistema de segurança residencial é suposto fazer soar um alarme (acção modelada por alm) logo que detecta a presença de um intruso (situação modelada por int).

1. Será que a fórmula $[int]\,(\langle alm\rangle\,\mathsf{true}\;\wedge\;[-alm]\,\mathsf{false})$ representa adequadamente essa propriedade comportamental?
2. Caso pense que não, represente-a correctamente.

Exercício 6

Formule em μ-*calculus* a propriedade seguinte sobre o comportamento de uma máquina de venda automática de bebidas: *O depósito de uma ou duas fichas conduz à aquisição de um café ou um chá*.

Exercício 7

Uma propriedade importante em sistemas que controlam linhas de montagem indutriais é a garantia de que

ϕ = *sempre que uma situação de erro grave ocorre, o sistema pára*.

Note, porém, que, regra geral, não pára instantaneamente: por exemplo, pode ser necessário que, antes de parar, o sistema desligue certos circuitos, active indicadores luminosos num painel, etc.

1. Supondo que a acção $erro$ modela a ocorrência de um erro grave, codifique a propriedade ϕ em μ-*calculus*.
2. Recorde a classificação das propriedades modais e temporais. Em que classe incluiría ϕ? Justifique.

Exercício 8

Suponha que num processo que especifica o comportamento de uma máquina de azar a acção $ganha(x)$ modela o facto do jogador ganhar uma quantia de x moedas. Alguém sugeriu que o processo deveria satisfazer uma das seguintes propriedades:

$$\begin{aligned}\phi_1 &= \nu X\,.\,(\mu Y\,.\,(\langle ganha(1000)\rangle\,\mathsf{true} \vee \langle -\rangle\,Y) \wedge [-]\,X)\\ \phi_2 &= \nu X\,.\,(\mu Y\,.\,\langle -\rangle\,Y) \vee \langle ganha(1000)\rangle\,X\end{aligned}$$

Alguém, porém, argumentou que ϕ_1 e ϕ_2 eram equivalentes.

1. Explique o significado destas propriedades e discuta se serão ou não equivalentes.
2. Recorde a classificação das propriedades modais e temporais. Em que classe incluiría ϕ_1? E ϕ_2? Justifique.

Exercício 9

Quando um número limitado de recursos é partilhado por um número elevado de processos existe a possibilidade de alguns desses processos clientes terem de aguardar permissão para utilizar um recurso até que este seja libertado pelo processo que o está correntemente a usar. A alocação de recursos aos clientes é um problema típico, por exemplo, na concepção de sistemas operativos, onde é conhecido como o problema de *scheduling*.

Para que a alocação seja feita com sucesso é necessário saber quais os processos que aguardam permissão de acesso. Assim, o acesso não pode ser modelado por uma única acção atómica, mas antes por um par de acções: $request$ (para requerer o acesso) e $permission$ (para indicar que este foi concedido). Quando o recurso é libertado o processo realiza a acção $release$ para sinalizar que terminou a tarefa e que o recurso está de novo disponível.

Um algoritmo de alocação muito popular é similar ao usado em certas lojas em que cada utilizador tira um *ticket* com um número, emitido, por ordem estritamente crescente, por um dispositivo próprio. Os recursos partilhados são, neste exemplo, os funcionários da loja e, usualmente, existe um número fixo de funcionários que podem atender clientes simultaneamente. Sempre que um deles está livre chama o cliente com o número mais baixo e que ainda aguarda ser servido.

1. Suponha que se pretende modelar uma situação em que existem R recursos (similares) disponíveis, o que significa que R processos podem ser "atendidos" ao mesmo tempo. Complete a especificação do processo:

$$\begin{aligned} Sch &\triangleq Sch(0,0,0) \\ Sch(p,t,r) &\triangleq \text{if} \quad \ldots \quad \text{then } \overline{permission}_t \;. \quad \ldots \\ &\quad\;\; \text{else } \overline{request(p)} \;. \quad \ldots \;+\; release(x) \;. \quad \ldots \end{aligned}$$

   que realiza a alocação de processos a recursos de acordo com o protocolo acima descrito. O processo tem 3 parâmetros:

   - p é o número a atribuir ao próximo processo que solicite acesso (coincidente com o número de processos que já realizaram $request$);
   - t é o número do próximo processo a ser atendido (coincidente com o número de processos que já realizaram $permission$);
   - r é o número de processos que já foram servidos e libertaram o recurso utilizado realizando $release$.

2. Os 3 contadores — p, t e r — podem, eventualmente, acabar por atingir o mesmo valor $n > 0$. Diga quando é que tal situação pode ocorrer e modifique a definição de Sch de modo a que, nessas circunstâncias, os contadores sejam re-inicializados com o valor 0.

3. Um dos objectivos deste protocolo é garantir que nunca existe, simultaneamente, um processo à espera de permissão de acesso e um recurso livre. Especifique essa propriedade em μ-*calculus*.

4. Outra propriedade que é desejável garantir é a ausência de *esperas infinitas*, *i.e.*, todo o processo cliente que pede permissão acaba por conseguir aceder ao recurso. Especifique essa propriedade em μ-*calculus* e diga como poderia demonstrar que Sch a verifica.