

Resolução e-fólio A

Esta resolução é apenas uma possível resposta às questões colocadas, não sendo de forma nenhuma a única possibilidade.

1. A abstração em Sistemas Computacionais consiste em apresentar um modelo de um sistema (ex. computador) que, escondendo os detalhes da sua implementação ou construção, oferece ao programador ou engenheiro uma redução de complexidade na forma de tratar com aquele sistema. Esse modelo pode ser simplesmente uma redução a caraterísticas essenciais para determinado fim, ou uma perspetiva diferente sobre o sistema em questão, podendo mesmo introduzir conceitos novos. Frequentemente, uma abstração abrange sistemas diferentes. Como exemplos mais flagrantes, as linguagens de programação de alto nível são abstrações sobre o funcionamento do computador subjacente, e constituem mecanismos para construir programas que correm sobre vários tipos de hardware, sem que o programador tenha de conhecer em detalhe ou distinguir as máquinas em que o programa corre. Notavelmente, linguagens como o C++ introduzem os conceitos de "objecto" e "classe" que não existem ao nível do código máquina do processador.

2.

a)

Tendo em conta que o tempo de execução é dado pela expressão:

$$T = I * CPI / f$$

o tempo para cada tipo de instrução é, tendo em conta a respetiva percentagem, é:

$$T_a = 0,6 * 4 \times 10^9 * 1 / 3,33 \times 10^9 = 2,4s$$

$$T_l = 0,3 * 4 \times 10^9 * 9 / 3,33 \times 10^9 = 10,8s$$

$$T_b = 0,1 * 4 \times 10^9 * 3 / 3,33 \times 10^9 = 1,2s$$

No total, o tempo de execução é de **4,32s**

b)

Considera-se a seguinte expressão, sendo T_r o tempo com a redução pretendida, T o tempo de execução num único processador, p o número de processadores, e o o overhead:

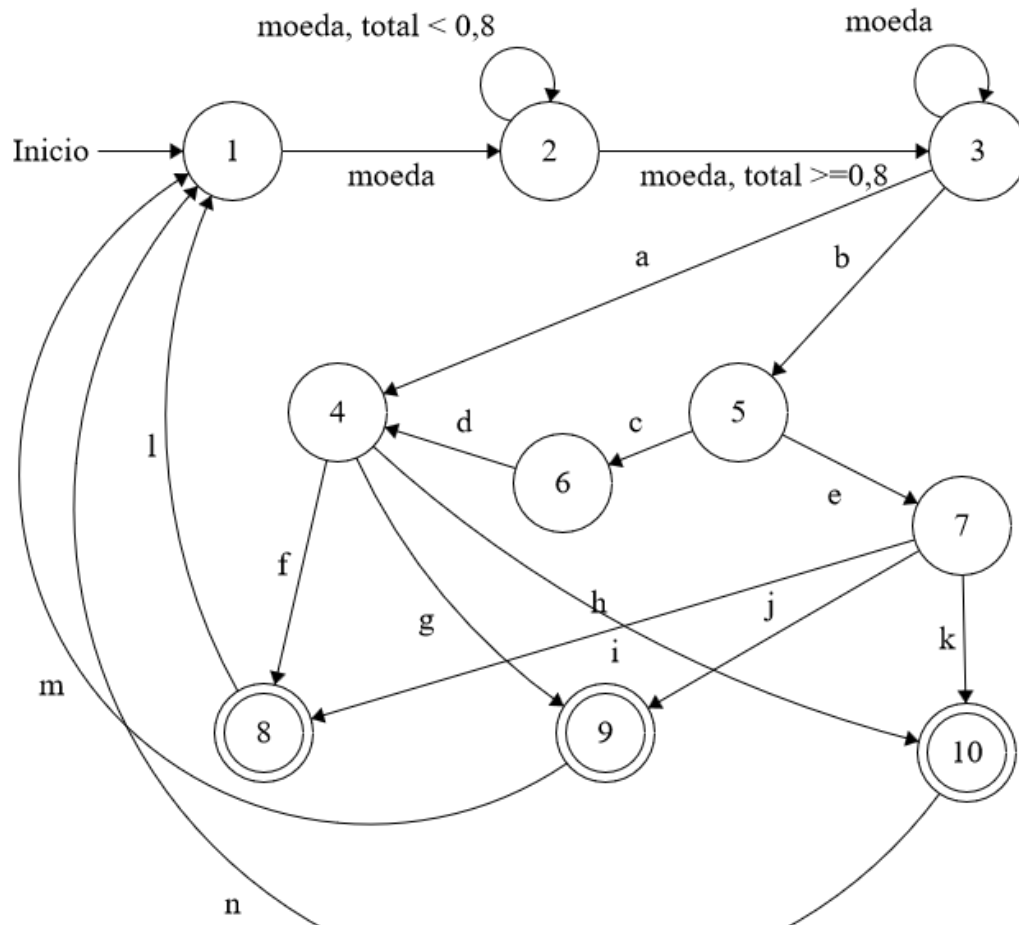
$$T_r = T/p + o$$

Sendo $T_r = T - 0,85 * T = 0,15 * T$, temos:

$$o = 0,15 * T - T/p = (0,15 - 1/p) * T$$

$$o = (0,15 - 1/16) * 4,32 = \mathbf{0,378 s}$$

3. Uma máquina de estados possível para representar o parquímetro é a seguinte (por razões de espaço, os estados e transições são identificados na legenda subsequente):

**Legenda:**

- 1 - aguarda (*idle*)
- 2 - moedas inseridas ($< 0,80\text{€}$)
- 3 - moedas inseridas ($\geq 0,80\text{€}$)
- 4 - pronto a emitir (quantia exata)
- 5 - se quantia $< 2\text{€}$, pergunta troco ou tempo
- 6 - pronto a devolver troco
- 7 - pronto a emitir (calcula tempo excedente)
- 8 - ticket azul
- 9 - ticket laranja
- 10 - ticket rosa

- a - emitir, com quantia exata
- b - emitir, com quantia não exata
- c - troco ou quantia $> 2\text{€}$
- d - devolve troco
- e - tempo
- f - quantia = $0,80\text{€}$
- g - quantia = $1,5\text{€}$
- h - quantia = 2€
- i - $0,8\text{€} < \text{quantia} < 1,5\text{€}$
- j - $1,5\text{€} < \text{quantia} < 2\text{€}$
- k - quantia $> 2\text{€}$
- l, m, n - imprimir ticket

4. As máquinas de estados podem classificar-se em deterministas e não-deterministas. Numa máquina determinista, o par estado-entrada (ou estado-condição de transição) determina inequivocamente o estado seguinte, enquanto que, numa máquina não-determinista, o mesmo par estado-entrada pode levar a mais do que um estado (ou mesmo a nenhum). Por outras palavras, numa máquina determinista, a transição é definida por uma função do par estado-entrada, e numa não-determinista, a transição é uma relação mais geral. Note-se, no entanto, que os dois tipos de máquinas são equivalentes, na medida em que uma máquina não-determinista que admite determinadas sequências de entrada é convertível numa máquina determinista que admite exatamente as mesmas sequências.