Exercícios resolvidos da Ficha 4

Exercício 3

Relembre a questão da Ficha 1 em que se estendia a linguagem **While** com ciclos *for*:

$$\mathbf{Stm} \ni C ::= \ \ldots \ \mid \ \texttt{for}\ (C_1; b\,; C_3)\ \texttt{do}\ C_2$$

com a seguinte semântica informal

> *O comando C_1 é executado; em seguida, a expressão booleana b é testada, e caso seja verdadeira é executada uma iteração de C_2, seguida de C_3, seguida de novo teste de b; enquanto b for verdadeiro são executadas iterações de C_2 seguido de C_3; se b for falso termina a execução.*

1. Escreva uma regra da lógica de Hoare para os ciclos $\texttt{for}\ (C_1; b\,; C_3)$, sem recorrer a outras formas de ciclo.
2. Prove a correcção da regra que propôs. *Sugestão:* tenha em conta a equivalência semântica entre

$$\texttt{for}\ (C_1; b\,; C_3)\ \texttt{do}\ C_2 \qquad \text{e} \qquad C_1;\ \texttt{while}\ b\ \texttt{do}\ \{C_2\ ;\ C_3\}$$

Resolução

1. Uma regra possível é a seguinte

$$\frac{\{\phi\}\, C_1\, \{\theta\} \qquad \{\theta \wedge b\}\, C_2; C_3\, \{\theta\}}{\{\phi\}\, \texttt{for}\ (C_1; b\,; C_3)\ \texttt{do}\ C_2\, \{\theta \wedge \neg b\}}$$

2. A regra acima é correcta se a validade das premissas da regra implicar a validade da sua conclusão. Vamos então assumir que

$$\models \{\phi\}\, C_1\, \{\theta\} \tag{1}$$

$$\models \{\theta \wedge b\}\, C_2; C_3\, \{\theta\} \tag{2}$$

tendo por objectivo provar que $\models \{\phi\}\, \texttt{for}\ (C_1; b\,; C_3)\ \texttt{do}\ C_2\, \{\theta \wedge \neg b\}$.

A equivalência semântica, que vimos na Ficha 1, garante que para todo o $s, s' \in \mathbf{State}$,

$$\langle \texttt{for}\ (C_1; b\,; C_3)\ \texttt{do}\ C_2, s\rangle \to s' \quad \text{sse} \quad \langle C_1;\ \texttt{while}\ b\ \texttt{do}\ \{C_2\ ;\ C_3\}, s\rangle \to s' \tag{3}$$

Por outro lado, podemos construir a seguinte árvore de derivação

$$\frac{\{\phi\}\, C_1\, \{\theta\} \qquad \dfrac{\{\theta \wedge b\}\, C_2; C_3\, \{\theta\}}{\{\theta\}\, \texttt{while}\ b\ \texttt{do}\ (C_2; C_3)\, \{\theta \wedge \neg b\}}}{\{\phi\}\, C_1;\ \texttt{while}\ b\ \texttt{do}\ (C_2; C_3)\, \{\theta \wedge \neg b\}}$$

Ou seja, podemos derivar a seguinte regra

$$\frac{\{\phi\}\, C_1\, \{\theta\} \qquad \{\theta \wedge b\}\, C_2; C_3\, \{\theta\}}{\{\phi\}\, C_1;\ \texttt{while}\ b\ \texttt{do}\ (C_2; C_3)\, \{\theta \wedge \neg b\}}$$

que é necessáriamente correcta, porque o sistema de inferência da lógica de Hoare é correcto. Logo, com base em (1) e (2), podemos concluir que

$$\models \{\phi\}\, C_1;\ \texttt{while}\ b\ \texttt{do}\ (C_2; C_3)\, \{\theta \wedge \neg b\}$$

e daqui por (3) que

$$\models \{\phi\}\, \texttt{for}\ (C_1; b\,; C_3)\ \texttt{do}\ C_2\, \{\theta \wedge \neg b\}.$$

Exercício 5

Considere o seguinte programa que calcula o quadrado de um número natural.

```
r := 0;
i := 0;
a := 1;
while i < x do {
  i := i + 1;
  r := r + a;
  a := a + 2
}
```

Escreva a especificação que descreve de forma adequada o que este programa faz, e encontre um invariante do ciclo que lhe permita provar a correcção do programa face à especificação. Apresente a árvore de prova.

Resolução

A especificação deste programa deve indicar que o valor final da variável r é igual ao quadrado do valor inícial da variável x. Para nos referirmos ao valor inicial da variável x temos que usar uma variável auxiliar (uma variável que não ocorre no programa). Vamos chamar-lhe x_0. Dado que a indicação que temos é que *o programa calcula o quadrado de um número natural*, teremos a seguinte especificação:

Pré-condição: $x \geq 0 \wedge x = x_0$
Pós-condição: $r = x_0^2$

O programa faz o cálculo de x^2 pela soma dos x primeiros números ímpares. Analisando o código, podemos ver que no início de cada iteração i do cíclo: $r = i^2$, $a = 2i + 1$ (ou seja, a contém o próximo número ímpar), e r guarda o valor acumulado da soma dos i primeiros ímpares.

Vejamos uma simulação da execução do programa para $x = 3$, registando o valor das variáveis, `i`, `r` e `a`, à entrada de cada iteração do ciclo while:

i	r	a
0	0	1
1	1	3
2	4	5
3	9	7

Vamos ainda precisar colocar no invariante condições que nos garantam que o valor de x não é alterado e que nos permitam concluir que à saida do ciclo o valor de i é igual a x. Assim um invariante, θ, que parece anotar corretamente o programa face à especificação apresentada é o seguinte:

$$\theta \quad \equiv \quad x = x_0 \ \wedge \ r = i^2 \ \wedge \ a = 2i + 1 \ \wedge \ i \leq x$$

Note que o invariante tem de ter as seguintes características:

1. ser suficientemente forte para garantir a pós-condição à saída do ciclo;
2. ser válido à entrada do ciclo;
3. ser preservado pelo ciclo.

Apresentamos a seguir um esquema da árvore de prova para o triplo $\{\phi\}\, A;$ `while` b `do` $C\,\{\psi\}$, com:

$$\begin{array}{lll} \phi \ \equiv \ x \geq 0 \wedge x = x_0 & A \ \equiv \ r := 0;\ i := 0;\ a := 1 & b \ \equiv \ i < x \\ \psi \ \equiv \ r = x_0^2 & C \ \equiv \ i := i + 1;\ r := r + a;\ a := a + 2 & \end{array}$$

$$
\dfrac{
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\cdots}{}\,[\text{ass}]}{\vdots}\,[\text{cons}](2)}{\vdots}\;\ldots\ [\text{ass}]\ \ldots\ [\text{comp}]\ \ldots}{\{\phi\}\,A\,\{\theta\}}
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\cdots}{}\,[\text{ass}]}{\vdots}\,[\text{cons}](3)}{\vdots}\;\ldots\ [\text{ass}]\ \ldots\ [\text{comp}]\ \ldots}{\{\theta \wedge b\}\,C\,\{\theta\}}}{\{\theta\}\,A;\ \mathtt{while}\ b\ \mathtt{do}\ C\,\{\theta \wedge \neg b\}}\,[\text{while}]
}{
\dfrac{\{\phi\}\,A;\ \mathtt{while}\ b\ \mathtt{do}\ C\,\{\theta \wedge \neg b\}}{\{\phi\}\,A;\ \mathtt{while}\ b\ \mathtt{do}\ C\,\{\psi\}}\,[\text{cons}](1)
}\,[\text{comp}]
$$

em que:

(1) $\theta \wedge \neg b \,\rightarrow\, \psi$

(2) $\phi \,\rightarrow\, \theta[1/a][0/i][0/r]$

(3) $\theta \wedge b \,\rightarrow\, \theta[(a+2)/a][(r+a)/r][(i+1)/i]$

Note como a regra da consequência é aqui aplicada nas extremidades da árvore:

- Antes de atingir a conclusão (a raiz da árvore), no sentido de enfraquecer a pós-condição, o que exige que a condição (1) seja válida.
- Nas folhas da árvore, no sentido de fortalecer pré-condições relativamente à pré-condição mais fraca que se obtém por propagação para trás de uma pós-condição face a uma sequência de atribuições. Neste caso, isso exige que as condições (2) e (3) sejam válidas.

Note ainda como as condições laterais que resultam da aplicação da regra de consequência correspondem às propriedades do invariante anteriomente enunciadas: (1) utilidade, (2) inicialização e (3) preservação. Como estas condições são válidas, temos uma árvore de prova.