Algoritmos e Complexidade
LEI (2º ano)

1º Ficha Prática

Ano Lectivo de 2011/12

O objectivo desta ficha é treinar o aluno na especificação formal da funcionalidade de programas através de contratos expressos como triplos de Hoare.

Especificação Formal de Programas

1. Discuta a validade dos seguintes triplos de Hoare:
   (a) $\{\texttt{J} = \texttt{A}\}\ \texttt{J} := \texttt{J} + 1\ \{\texttt{A} = \texttt{J} + 1\}$
   (b) $\{\texttt{I} = \texttt{J}\}\ \texttt{I} := \texttt{J} + \texttt{I}\ \{\texttt{I} > \texttt{J}\}$
   (c) $\{\texttt{J} = \texttt{A} + \texttt{B}\}\ \texttt{I} := \texttt{B};\ \texttt{J} := \texttt{A}\ \{\texttt{J} = 2 * \texttt{A}\}$
   (d) $\{\texttt{I} > \texttt{J}\}\ \texttt{J} := \texttt{I} + 1;\ \texttt{I} := \texttt{J} + 1\ \{\texttt{I} > \texttt{J}\}$
   (e) $\{\texttt{I} != \texttt{J}\}$ IF I > J THEN M := I − J ELSE M := J − I $\{\texttt{M} > 0\}$
   (f) $\{\texttt{I} = 3 * \texttt{J}\}$ IF I > J THEN M := I − J ELSE M := J − I $\{\texttt{M} - 2 * \texttt{J} = 0\}$
   (g) $\{\texttt{X} = \texttt{B}\}$ WHILE X > A DO X := X − 1 $\{\texttt{B} = \texttt{A}\}$
2. Escreva especificações (contratos para triplos de Hoare) para os seguintes programas:
   (a) Um programa que coloca na variável `Z` a soma dos valores das variáveis `X` e `Y`.
   (b) Um programa que coloca na variável `Z` o máximo divisor comum das variáveis `X` e `Y`.
   (c) Um programa que coloca na variável `Z` o minimo múltiplo comum das variáveis `X` e `Y`.
   (d) Um programa que recebe dois arrays `A` e `B` como parâmetros, e verifica que eles têm um elemento em comum.
   (e) Um programa que recebe dois arrays `A` e `B` como parâmetros, e retorna o prefixo mais longo que os dois têm em comum.
   (f) Um programa que ordena um array `A`.
3. Sejam P, Q dois predicados arbitrários, e seja S um programa arbitrário. O que significam as seguintes especificações (tendo em conta que a notação [ ] representa correcção total):
   (a) $\{P\}\ S\ \{\mathsf{true}\}$
   (b) $[P]\ S\ [\mathsf{true}]$
   (c) $\{P\}\ S\ \{\mathsf{false}\}$
   (d) $[P]\ S\ [\mathsf{false}]$
   (e) $\{\texttt{false}\}\ S\ \{Q\}$

(f) [false] S $[Q]$

(g) {true} S $\{Q\}$

(h) [true] S $[Q]$

4. Dê, para cada programa, uma pré-condição que seja suficiente para garantir a pós-condição. Sugestão: tente encontrar a pré-condição mais fraca que permita atingir esse objectivo.

   (a) $\{?\}$ $\mathtt{X} := \mathtt{X} + 1;\ \mathtt{S} := \mathtt{S} + \mathtt{X}\ \{\mathtt{S} > 0\}$

   (b) $\{?\}$ $\mathtt{Yold} := \mathtt{Y};$ IF $\mathtt{I} = 0$ THEN $\mathtt{Y} := 0$ ELSE $\mathtt{Y} := \mathtt{Y}/\mathtt{X}\ \{\mathtt{X} < \mathtt{Yold}\}$

5. Considere o seguinte programa, para o qual vamos utilizar a pós-condição $\mathtt{R} < \mathtt{Y} \wedge \mathtt{X} = \mathtt{R} + (\mathtt{Y} * \mathtt{Q})$.

```
R:=X;
Q:=0;
WHILE Y <= R DO
   BEGIN R:=R-Y; Q:=Q+1; END
```

   (a) Escreva uma pré-condição para uma especificação de correcção parcial para este programa.

   (b) Escreva uma pré-condição para uma especificação de correcção total para este programa.

6. Considere o seguinte programa, em que L1 e L2 são variáveis boleanas, e n1 e n2 são variáveis inteiras.

```
L1 := false; L2 := false;
WHILE (N1 + N2 > 0) DO
BEGIN
   IF (L2 = false) THEN
   BEGIN
       IF (N1 > 0)
       THEN BEGIN L1 := true; N1 := N1 - 1; END
       ELSE L1 := false;
   END
   IF (L1 = false) THEN
   BEGIN
        IF (n2 > 0)
        THEN BEGIN L2 := true; N2 := N2 - 1; END
        ELSE L2 := false;
   END
   L2 := ! L2;
   L1 := ! L1;
END
```

   (a) Encontre uma pré-condição para este programa que permita garantir a pós-condição $\mathtt{L1} = \mathtt{false} \vee \mathtt{L2} = \mathtt{false}$. Sugestão: tente encontrar a pré-condição mais fraca que permite satisfazer esse objectivo.

   (b) É possível que o ciclo não termine? Se sim, em que condições?

   (c) Suponha que este programa é codificado, e compilado, com um erro que elimina as primeiras duas atribuições ($\mathtt{L1} := \mathtt{false}$ e $\mathtt{L2} := \mathtt{false}$). Será possível, apenas através da realização de testes ao código compilado (executando-o com valores iniciais diferentes para as quatro variáveis e observando o estado final) detectar este erro de implementação? Justifique a sua resposta.