

Algoritmos e Complexidade

LEI/LCC (2º ano)

1º Ficha Prática

Ano Lectivo de 2009/10

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) $\{J = A\} J := J + 1 \{A = J + 1\}$
- (b) $\{I = J\} I := J + I \{I > J\}$
- (c) $\{J = A + B\} I := B; J := A \{J = 2 * A\}$
- (d) $\{I > J\} J := I + 1; I := J + 1 \{I > J\}$
- (e) $\{I \neq J\} \text{If } I > J \text{ Then } M := I - J \text{ Else } M := J - I \{M > 0\}$
- (f) $\{I = 3 * J\} \text{If } I > J \text{ Then } M := I - J \text{ Else } M := J - I \{M - 2 * J = 0\}$
- (g) $\{X = B\} \text{While } X > A \text{ Do } X := X - 1 \{B = A\}$

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 mínimo 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) $\{P\} S \{\text{true}\}$
- (b) $[P] S [\text{true}]$
- (c) $\{P\} S \{\text{false}\}$
- (d) $[P] S [\text{false}]$
- (e) $\{\text{false}\} S \{Q\}$
- (f) $[\text{false}] S [Q]$
- (g) $\{\text{true}\} S \{Q\}$
- (h) $[\text{true}] S [Q]$

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

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

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

```
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 booleanas, 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 $L1 = false \vee L2 = false$. 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 ($L1 := false$ e $L2 := false$). 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.