Licenciatura em Engenharia Informática e de Computadores - LEIC
Licenciatura em Engenharia de Redes de Comunicação e Informação - LERCI

Exercícios Teoria de Computação
Cálculo de Hoare

Secção de Lógica e Computação
Departamento de Matemática
Instituto Superior Técnico
2006/2007

1 Correcção total e parcial

1. Considere o programa P correspondente ao corpo da seguinte função:

   Function[{n},Module[{i, r}
   $i = 0$;
   $r = 0$;
   While[$i \neq n$,
   $r = r + 2i + 1$;
   $i = i + 1$];
   r]]

   Usando o Cálculo de Hoare:

   (a) Demonstre que a inicialização garante $r == i^2$, isto é, que a asserção

   $$\{\text{True}\}\ i = 0; r = 0\ \{r == i^2\}$$

   é teorema.

   (b) Demonstre que $r == i^2$ é condição invariante do ciclo, isto é, que a asserção

   $$\{r == i^2 \wedge i \neq n\}\ r = r + 2i + 1; i = i + 1\ \{r == i^2\}$$

   é teorema.

   (c) Mostre que a asserção

   $$\{\text{True}\}\ P\ \{r == n^2\}$$

   é teorema.

   (d) Indique uma expressão variante τ tal que ambas as seguintes asserções sejam verdadeiras (não necessita de as demonstrar)

   - {IntegerQ[n] $\wedge$ $(n \geq 0)$}
     $i = 0; r = 0$
     {IntegerQ[n] $\wedge$ IntegerQ[i] $\wedge$ $(\tau \geq 0)$}
   - {IntegerQ[n] $\wedge$ IntegerQ[i] $\wedge$ $(\tau \geq 0) \wedge (i \neq n) \wedge (\tau == N)$}
     $r = r + 2i + 1; i = i + 1$
     {IntegerQ[n] $\wedge$ IntegerQ[i] $\wedge$ $(\tau \geq 0) \wedge (\tau < N)$}

   (e) Prove que a asserção

   $$\Omega[\text{IntegerQ}[n] \wedge (n \geq 0), P]$$

   é teorema.

2. Considere o programa correspondente ao corpo da seguinte função:

   Function[{w, x},Module[{i, n, m}
   i=1;
   n=0;
   m=0;
   While[$i <$ Length[w] $+ 1$,
   If[$w[[i]] > x, n = n + 1, m = m + 1$];

$i = i + 1]$;
{n,m}]]

Usando o Cálculo de Hoare:

(a) Demonstre que

$$\{n + m == k\}\text{If}[w[[i]] > x, n = n + 1, m = m + 1]\{n + m == k + 1\}$$

(b) Indique uma condição invariante I tal que ambas as seguintes asserções sejam verdadeiras (não necessita de as demonstrar)

- $(I \wedge \neg G) \Rightarrow (n == \#\{j : (w[[j]] > x) \wedge (1 \leq j \leq \text{Length}[w])\})$
- $\{\text{True}\}\ i = 1; n = 0; m = 0\ \{I\}$
- $\{I \wedge G\}\ B\ \{I\}$

onde G é a guarda do ciclo e B é o corpo do ciclo. Demonstre que

- $\{\text{True}\}\ i = 1; n = 0; m = 0\ \{I\}$
- $\{I \wedge G\}\ B\ \{I\}$

(c) Demonstre que

$$\Omega[\text{ListQ}[w], i = 1; n = 0; m = 0]$$

(d) Indique a expressão variante τ do ciclo

(e) Demonstre que

$\Omega[\text{ListQ}[w] \wedge \text{IntegerQ}[i] \wedge (\tau \geq 0),$
$\quad \text{While}[i < \text{Length}[w] + 1, \text{If}[w[[i]] > x, n = n + 1, m = m + 1]; i = i + 1]]$

3. Recorrendo ao Cálculo de Hoare verifique que o programa correspondente ao corpo da função indicada está parcialmente correcto face à condição inicial C_i e à condição final C_f descritas:

(a) Function[$\{n\}$,Module[$\{i, r\}$
$i = 1$;
$r = 0$;
While[$i \leq n$,
$r = r + i$;
$i = i + 1]$;
$r]]$

$C_i = \text{True}$

$C_f = (r == \sum_{k=0}^{n} k)$

(b) Function[$\{w\}$,Module[$\{i, r\}$
$i = 1$;
$r = 0$;
While[$i \neq \text{Length}[w] + 1$,

$\text{If}[w[[i]] > 0, r = r + 1,\text{Null}];$
$i = i + 1];$
$r]]$

$C_i = \text{True}$

$C_f = (r == \#\{j : (w[[j]] > 0) \wedge (1 \leq j \leq \text{Length}[w])\})$

(c) $\text{Function}[\{n\},\text{Module}[\{i, c\}$
$i = 2;$
$c = 1;$
$\text{While}[c \neq n,$
$i = i + 1;$
$\text{If}[\text{PrimeQ}[i], c = c + 1,\text{Null}]];$
$i]]$

$C_i = \text{True}$

$C_f = (i == \text{Prime}[n])$

(d) $\text{Function}[\{w_1, w_2\},\text{Module}[\{i, r\}$
$i = 1;$
$r = \text{False};$
$\text{While}[i \leq \text{Length}[w_1],$
$r = r \vee (w_1[[i]] \neq w_2[[i]]);$
$i = i + 1];$
$r]]$

$C_i = (\text{Length}[w_1] == \text{Length}[w_2])$

$C_f = (r == \bigvee_{k=1}^{\text{Length}[w_1]} w_1[[k]] \neq w_2[[k]])$

(e) $\text{Function}[\{x, y\},\text{Module}[\{n, m, r\}$
$n = x;$
$m = y;$
$r = 1;$
$\text{While}[m \neq 0,$
$\text{If}[\text{OddQ}[m], r = r \times n; m = m - 1, n = n^2; m = \frac{m}{2}]];$
$r]]$

$C_i = \text{True}$

$C_f = (r == x^y)$

4. Recorrendo ao Cálculo de Hoare verifique que o programa correspondente ao corpo da função indicada está totalmente correcto face à condição inicial C_i e à condição final C_f descritas:

   (a) Function[{n},Module[{i, r}
   $i = 1$;
   $r = 0$;
   While[$i \leq n$,
   $r = r + i$;
   $i = i + 1$];
   r]]

   $C_i = \text{IntegerQ}[n]$

   $C_f = (r == \sum_{k=0}^{n} k)$

   (b) Function[{w},Module[{i, r}
   $i = 1$;
   $r = 0$;
   While[$i \neq \text{Length}[w] + 1$,
   If[$w[[i]] > 0, r = r + 1$,Null];
   $i = i + 1$];
   r]]

   $C_i = \text{ListQ}[w]$

   $C_f = (r == \#\{j : (w[[j]] > 0) \wedge (1 \leq j \leq \text{Length}[w])\})$

   (c) Function[{n},Module[{i, c}
   $i = 2$;
   $c = 1$;
   While[$c \neq n$,
   $i = i + 1$;
   If[PrimeQ[i], $c = c + 1$,Null]];
   i]]

   $C_i = \text{IntegerQ}[n] \wedge (n > 0)$

   $C_f = (i == \text{Prime}[n])$

   (d) Function[{w_1, w_2},Module[{i, r}
   $i = 1$;

r =False;
While[$i \leq$Length[w_1],
  $r = r \vee (w_1[[i]] \neq w_2[[i]])$;
  $i = i + 1$];
r]]

$C_i = \text{ListQ}[w_1] \wedge \text{ListQ}[w_2] \wedge (\text{Length}[w_1] == \text{Length}[w_2])$

$C_f = (r == \bigvee_{k=1}^{\text{Length}[w_1]} w_1[[k]] \neq w_2[[k]])$

(e) Function[$\{x, y\}$,Module[$\{n, m, r\}$
  $n = x$;
  $m = y$;
  $r = 1$;
  While[$m \neq 0$,
    If[OddQ[m], $r = r \times n; m = m - 1, n = n^2; m = \frac{m}{2}$]];
  r]]

$C_i = \text{IntegerQ}[x] \wedge \text{IntegerQ}[y] \wedge (y \geq 0)$

$C_f = (r == x^y)$

2 Regras de convergência e de correcção parcial

1. Apresente as regras de convergência e de correcção parcial para os comandos seguintes:

   (a) While[$G_1, P_1, \ldots, G_n, P_n, P_{n+1}$] com a seguinte interpretação: em cada passo do ciclo executa não deterministicamente um dos programas P_i em que a expressão G_i é verdadeira. Caso nenhuma das expressões $G_1, \ldots, G_n$ seja verdadeira o ciclo termina executando P_{n+1}.

   (b) For[$P_{\text{start}}, G, P_{\text{inc}}, P_{\text{corpo}}$] com a seguinte interpretação: executa inicialmente o programa P_{start} e posteriormente enquanto a condição G for verdadeira executa $P_{\text{corpo}}; P_{inc}$.

   (c) Cases[$G_1, P_1, \ldots, G_n, P_n$] com a seguinte interpretação: executa não-deterministicamente um dos programas P_i tal que a expressão G_i é verdadeira. Caso nenhuma das expressões $G_1, \ldots, G_n$ seja verdadeira a execução do Cases termina sem que nenhum dos programas $P_1, \ldots, P_n$ seja executado.

   (d) ConditionalExec[$G_1, P_1, \ldots, G_n, P_n$] com a seguinte interpretação: executa sequencialmente os comandos P_i para os quais a guarda G_i seja verdadeira até se avaliar uma guarda que seja falsa. Caso a expressão G_1 seja falsa a execução do ConditionalExec termina sem que nenhum dos programas $P_1, \ldots, P_n$ seja executado.

(e) Choice$[P_1, P_2]$ com a seguinte interpretação: executa não-deterministicamente o programa P_1 ou o programa P_2.