Tecnología de la Programación
Ingeniería Técnica en Informática de Sistemas. Curso 2007/2008

EJERCICIOS PROPUESTOS

Tema 4: Uso de aserciones en la documentación de programas.

Tema 5: El transformador de predicados *wp*.

Tema 6: El lenguaje GCL.

1. Demostrar $\{Q\}S\{P\}\{R\}$ equivale a probar:
   a) $S \Rightarrow wp(P, R)$ y $R \Rightarrow P$
   b) $P \Rightarrow wp(S, Q)$
   c) $Q \Rightarrow wp(S, P)$ y $P \Rightarrow R$
   d) $P \Rightarrow R$
2. Sea S un programa determinista. Entonces $wp(S, \neg\alpha \vee \beta)$ equivale a probar:
   a) $wp(S, \neg\alpha) \vee wp(S, \beta)$
   b) $wp(S, \neg\alpha \Rightarrow \beta)$
   c) $wp(S, \alpha \Rightarrow \beta)$
   d) Ninguna de las anteriores
3. La expresión $\{T\}S\{T\}$:
   a) es falsa si S puede no terminar
   b) es siempre cierta, termine o no S
   c) es equivalente a $wp(S, T)$
   d) Ninguna de las anteriores
4. Si $\{Q\}S\{R\}$ es cierto, entonces:
   a) Si $Q \Rightarrow A$, entonces $\{A\}S\{R\}$
   b) Si $A \Rightarrow Q$, entonces $\{A\}S\{R\}$
   c) $Q\{S\}R$
   d) $\{Q\}S\{A \vee R\}$
5. Si tenemos que $\{Q\}S\{i > 0\}$, entonces:
   a) $\{Q\}S\{i \geq 0\}$
   b) $\{Q \wedge i = 0\}S\{i > 0\}$
   c) $\{Q \vee A\}S\{i > 0\}$
   d) $wp(S, i > 0) \Rightarrow wp(S, i \neq 0)$
6. Si $\{Q\}S_0\{P\}$ do $B_1 \rightarrow S_1\{P\}$ od $\{R\}$ es cierto entonces:
   a) $\{P \wedge B_1\}S_1\{P\}$
   b) $\{P \wedge B_1\}S_1\{R\}$
   c) Si $\{Q\}S_0\{\neg B_1\}$ entonces $\{Q\}S_0\{R\}$

d) Si $S_1 = skip$ entonces el programa no termina.

7. Calcular $wp(\text{"}a[i], i := a[0], i + 1\text{"}, (\forall j : 0 \leq j < i - 1 : a[j] \leq a[j + 1]))$.

8. Calcular $wp(\text{"}a[i], a[j] := 1, 2\text{"}, a[i] = a[j])$.

9. Calcular $wp(\text{"}b[i], b[j] := b[j], b[i]; b[j], b[3] := b[3], b[j]\text{"}, b[i] = b[3])$.

10. Dado $b[0 : 5] : integer$, demostrar formalmente que: $\{b[0] = 0\}b[0], b[1] := b[1], b[0]\{b[0] = b[b[1]]\}$. Nota: Calcular antes el wp correspondiente por completo.

11. Calcular $wp(\text{"}i := j - k; k := i * j/(i - j)\text{"}, j \leq k \wedge -i \leq k \wedge i > 0)$.

12. Calcula y simplifica $wp(\text{"}b[j], b[i] := i, b[j]\text{"}, b[b[j]] = b[j])$.

13. Se ha comprobado que en algunas ocasiones, en el estado resultante de ejecutar la instrucción $S : b[b[i]] := b[j]$, la postcondición $R : b[b[i]] = b[j]$ no se satisface. Calcula la precondición más débil $wp(S, R)$. Una vez calculada, propón un contraejemplo, esto es, un estado inicial en el que ejecutar S nos lleva a que R es falso al terminar.

14. El programa $\text{if } b[i] = 0 \rightarrow i := 1 \mid b[i] \neq 0 \rightarrow i := 1 \text{ fi}$:

    a) Equivale a ejecutar $i := 1$ sin más.

    b) Es determinista.

    c) A veces es no determinista.

    d) Es equivalente a $\text{if } b[i] = 0 \vee b[i] \neq 0 \rightarrow i := 1 \text{ fi}$.

15. El programa $\text{do } i \neq n \rightarrow \text{ if } b[i] \geq 0 \rightarrow x := 1 \mid b[i] < 0 \rightarrow x := 2 \text{ fi od}$ siempre equivale a:

    a) $\text{do } i \neq n \wedge b[i] \geq 0 \rightarrow x := 1 \mid i \neq n \wedge b[i] < 0 \rightarrow x := 2 \text{ od}$

    b) $\text{do } i \neq n \text{ cand } b[i] \geq 0 \rightarrow x := 1 \mid i \neq n \text{ cand } b[i] < 0 \rightarrow x := 2 \text{ od}$

    c) $\text{do } b[i] \geq 0 \text{ cand } i \neq n \rightarrow x := 1 \mid b[i] < 0 \text{ cand } i \neq n \rightarrow x := 2 \text{ od}$

    d) $\text{do } i \neq n \rightarrow \text{ if } b[i] < 0 \rightarrow x := 2 \mid b[i] \geq 0 \rightarrow x := 1 \text{ fi od}$.

16. El programa $\text{if } x * z > y \rightarrow x := 1 \mid x * z \leq y \rightarrow x := 1 \text{ fi}$:

    a) Es equivalente a $x := 1$.

    b) Es no determinista.

    c) Es equivalente a $\text{if } x > y/z \rightarrow x := 1 \mid x \leq y/z \rightarrow x := 1 \text{ fi}$

    d) Es equivalente a $\text{if } z = 0 \rightarrow \text{ abort fi}$.

17. El programa $\text{do } i \neq n \rightarrow \text{ if } i \geq n \rightarrow i := i + 1 \mid i \leq n \rightarrow i := i + 1 \text{ fi od}$:

    a) Puede no terminar.

    b) Es no determinista.

    c) Es equivalente a $\text{do } i \neq n \rightarrow i := i + 1 \text{ od}$

    d) Es equivalente a $\text{do } i \geq n \rightarrow i := i + 1 \mid i \leq n \rightarrow i := i + 1 \text{ od}$

18. El programa $i, j := 0, 0; \text{ do } i \neq 8 \rightarrow i := i + 1 \mid j \neq 11 \rightarrow j := j + 1 \text{ od}$:

    a) es equivalente a $i, j := 8, 11$.

    b) es equivalente a $i, j := 8, 8$.

    c) es equivalente a $i, j := 0, 0; \text{ do } i \neq 8 \vee j \neq 11 \rightarrow i, j := i + 1, j + 1 \text{ od}$

    d) tiene una ejecución no determinista.

19. El programa $\text{do } i \neq n \text{ cand } b[i] = 0 \rightarrow i := i + 1 \text{ od}$ equivale a:

    a) $\text{do } i \neq n \rightarrow \text{ if } b[i] = 0 \rightarrow i := i + 1 \text{ fi od}$

    b) $\text{do } i \neq n \rightarrow \text{ if } b[i] = 0 \rightarrow i := i + 1 \mid b[i] \neq 0 \rightarrow skip \text{ fi od}$

c) if $i \neq n \rightarrow$ do $b[i] = 0 \rightarrow i := i + 1$ od fi

d) Ninguna de las anteriores.

20. El programa if $B_1 \rightarrow S; S_1 \mid B_2 \rightarrow S; S_2$ fi es equivalente a:

a) S; if $B_1 \rightarrow S_1 \mid B_2 \rightarrow S_2$ fi

b) if $B_1 \vee B_2 \rightarrow S$; if $B_1 \rightarrow S_1 \mid B_2 \rightarrow S_2$ fi fi

c) if $B_1 \wedge \neg B_2 \rightarrow S; S_1 \mid B_2 \wedge \neg B_1 \rightarrow S; S_2 \mid \neg B_1 \wedge \neg B_2 \rightarrow$ abort fi

d) Ninguno de las anteriores.

21. El programa do $i \geq i + 1 \rightarrow i := i + 1 \mid i \leq i + 1 \rightarrow i := i + 1$ od :

a) Puede abortar.

b) Es no determinista.

c) No termina.

d) Equivale a $i := i + 1$.

22. Demostrar formalmente el siguiente algoritmo:

$\{b \geq 0\}$
$x, y, z := a, b, 0;$
do $y > 0 \wedge par(y) \rightarrow y, x := y/2, x + x$
$\parallel impar(y) \rightarrow y, z := y - 1, z + x$
od
$\{R : z = a * b\}$

23. Demostrar formalmente que el siguiente algoritmo calcula la suma en s de todos los elementos de $b[1 : 10]$.

$\{T\}$
$i, s := 10, 0$
$\{inv$ P $: 0 \leq i \leq 10 \wedge s = (\sum k : i + 1 \leq k \leq 10 : b[k])\}$
$\{cota$ t $: i\}$
do $i \neq 0 \rightarrow i, s := i - 1, s + b[i]$
od
$\{R : s = (\sum k : 1 \leq k \leq 10 : b[k])\}$

24. Demostrar formalmente que el siguiente algoritmo busca la posición i de x en el array $b[0 : n-1]$ si $x \in b[0 : n - 1]$ y pone i a n si x no está en dicho array.

$\{n \geq 0\}$
$i := 0$
$\{invP : 0 \leq i \leq n \wedge x \notin b[0 : i - 1]\}$
$\{cotat : n - i\}$
do $i < n$ cand $x \neq b[i] \rightarrow i := i + 1$
od $\{R : (0 \leq i < n \wedge x = b[i]) \vee (i = n \wedge x \notin b[0 : n - 1]\}$

25. Demostrar formalmente que el siguiente algoritmo pone en i la potencia más alta de 2 que es como mucho n.

$\{0 < n\}$
$i := 1$
$\{inv$ P: $0 < i \leq n \wedge (\exists p : i = 2^p)\}$
$\{cota$ t: $n - i\}$
do $2 * i \leq n \rightarrow i := 2 * i$
od
$\{R : 0 < i \leq n < 2 * i \wedge (\exists p : i = 2^p)\}$

26. Demostrar formalmente que el siguiente algoritmo obtiene el término n de la serie de Fibonacci f_n para $n > 0$. La serie se define como $f_0 = 0$, $f_1 = 1$ y $f_n = f_{n-1} + f_{n-2}$ para $n > 1$.

$\{n > 0\}$
$i, a, b := 1, 1, 0$
$\{inv$ P: $1 \leq i \leq n \wedge a = f_i \wedge b = f_{i-1}\}$
$\{cota$ t: $n - i\}$
do $i < n \rightarrow i, a, b := i + 1, a + b, a$
od
$\{R : a = f_n\}$

27. Demostrar formalmente que el siguiente algoritmo calcula el cociente q y el resto r de dividir x entre y.

$\{x \geq 0 \wedge y > 0\}$
$q, r := 0, x$
$\{inv$ P: $r \geq 0 \wedge y > 0 \wedge q * y + r = x\}$
$\{cota$ t: $r\}$
do $r \geq y \rightarrow r, q := r - y, q + 1$
od
$\{R : 0 \leq r < y \wedge q * y + r = x\}$

28. Demostrar formalmente que el siguiente algoritmo busca un entero k tal que $b[k]$ es el valor máximo de $b[0 : n - 1]$. Si el valor máximo aparece más de una vez, el programa es no determinista.

$\{n > 0\}$
$i, k := 1, 0$
$\{inv$ P: $0 < i \leq n \wedge b[k] \geq b[0 : i - 1]\}$
$\{cota$ t: $n - i\}$
do $i < n \rightarrow$ if $b[i] \leq b[k] \rightarrow skip$
$\|\ b[i] \geq b[k] \rightarrow k := i$
fi
$i := i + 1$
od $\{R : b[k] \geq b[0 : n - 1]\}$