E.T.S.I. Informática. Primer Parcial, 19 Octubre 2006
Lenguajes de Programación (tercer curso)

Apellidos: SOLUCIONES

Nombre: ______________________

1	2	3	4	5	6	Total
2.0	2.0	1.5	1.5	1.5	1.5	10.0

□ si
□ no
} deseo que se publique mi calificación

---

1 Prueba la siguiente identidad en todo espacio de estados:

$$[(A \Rightarrow M \wedge N) \equiv (A \Rightarrow M) \wedge (A \Rightarrow N)]$$

SOL Tenemos, *ptle*

$A \Rightarrow M \wedge N$

$\equiv\because$ oro, def. negación

$\neg A \vee (M \wedge N)$

$\equiv\because$ distributiva

$(\neg A \vee M) \wedge (\neg A \vee N)$

$\equiv\because$ oro, def. $\neg$ dos veces

$(A \Rightarrow M) \wedge (A \Rightarrow N)$

---

2 Siendo S un programa sano, prueba e interpreta la siguiente propiedad para tripletes de Dijkstra:

$$\{P\}S\{Falso\} \qquad \Rightarrow \qquad [P \equiv Falso]$$

La interpretación es: Si algún estado ι satisface el predicado P, entonces toda ejecución de S desde ι termina en un estado satisfaciendo $Falso$, lo que es imposible.

Una demostración sería:

$\{P\}S\{Falso\}$

$\equiv\because$ definición de triplete de Dijkstra

$[P \Rightarrow S.Falso]$

$\equiv\because$ Al ser S sano satisface la Ley del milagro excluido: $[S.Falso \equiv Falso]$, y por sustitutividad

$[P \Rightarrow Falso]$

$\equiv\because$ regla de oro

$[P \wedge Falso \equiv Falso]$

$\equiv\because$ CP

$[P \equiv Falso]$

Obsérvese que todos los pasos son equivalencias, y lo demostrado es la equivalencia: $\{P\}S\{Falso\} \equiv [P \equiv Falso]$

Sea Jim un programa con el siguiente transformador:

$$[Jim.Z \doteq y > 6 \wedge x := y.Z \vee y \leq 6 \wedge Z]$$

3 Prueba que Jim termina siempre. Hay que demostrar: $[Jim.Cierto \equiv Cierto]$

En efecto: *ptle*,

$Jim.Cierto$

$\equiv\because$ def. de Jim

$y > 6 \wedge x := y.Cierto \vee y \leq 6 \wedge Cierto$

$\equiv\because$ sustitución, CP

$y > 6 \wedge Cierto \vee y \leq 6$

$\equiv\because$ CP

$Cierto$

Queremos **confirmar** que Jim tiene el mismo comportamiento que la sentencia: "Si $y > 6$ realizar la acción $x := y$, pero en otro caso no hacer nada". Para ello prueba las dos afirmaciones descritas en $\boxed{\mathbf{4}}$ y $\boxed{\mathbf{5}}$.

$\boxed{\mathbf{4}}$ Para todo estado inicial satisfaciendo $y \leq 6$, Jim termina sin alterar ninguna variable.
En efecto; hay que demostrar, en términos de transformadores, que para cada variable $t \not\equiv x$,

$$[y \leq 6 \wedge (x,t) = (a,b) \qquad \Rightarrow \qquad Jim.(x,t) = (a,b)]$$

Una prueba sería: ptle

$y > 6 \wedge x := y.Z \vee y \leq 6 \wedge Z$

$Jim.(x,t) = (a,b)$

$\equiv\because$ def. de Jim

$y > 6 \wedge x := y.(x,t) = (a,b) \vee y \leq 6 \wedge (x,t) = (a,b)$

$\equiv\because$ sustitución

$y > 6 \wedge (y,t) = (a,b) \vee y \leq 6 \wedge (x,t) = (a,b)$

$\Leftarrow\because [A \vee B \Leftarrow B]$

$y \leq 6 \wedge (x,t) = (a,b)$

$\boxed{\mathbf{5}}$ Para todo estado inicial satisfaciendo $y > 6$, Jim termina alterando solamente la variable x, asignándole el contenido que tenía la variable y al principio del programa.
En efecto; hay que demostrar, en términos de transformadores, que para cada variable $t \not\equiv x$,

$$[y > 6 \wedge (x,t,y) = (a,b,c) \qquad \Rightarrow \qquad Jim.(x,t,y) = (c,b,c)]$$

Una prueba sería: ptle

$Jim.(x,t,y) = (c,b,c)$

$\equiv\because$ def. de Jim

$y > 6 \wedge x := y.(x,t,y) = (c,b,c) \vee y \leq 6 \wedge (x,t,y) = (c,b,c)$

$\equiv\because$ sustitución

$y > 6 \wedge (y,t,y) = (c,b,c) \vee y \leq 6 \wedge (x,t,y) = (c,b,c)$

$\Leftarrow\because [A \vee B \Leftarrow A]$

$y > 6 \wedge (y,t,y) = (c,b,c)$

$\Leftarrow\because$obsérvese que $(y,t,y) = (c,b,c) \equiv (t,y) = (b,c) \qquad \Rightarrow \qquad (a,t,y) = (a,b,c)$

$y > 6 \wedge (x,t,y) = (a,b,c)$

$\boxed{\mathbf{6}}$ Sea T un mecanismo determinista en sentido operacional ("para cada estado inicial ι, si alguna ejecución de T termina, todas lo hacen en el mismo estado final"). Justifica entonces la siguiente afirmación: El predicado $\neg T.Cierto$ representa al conjunto de estados iniciales para los que T no termina.

$\boxed{SOL}$ Hay que demostrar:

1. Si ι satisface el predicado $\neg T.Cierto$, entonces ninguna ejecución desde el estado ι puede terminar.

   En efecto: Si ι verifica el predicado $\neg T.Cierto$, entonces ι no satisface $T.Cierto$, luego, no se garantiza la terminación desde ι; es decir, existe una ejecución desde ι que no termina, y por ser determinista, ninguna ejecución desde el estado ι puede terminar.

2. Si ninguna ejecución desde el estado ι puede terminar, entonces ι satisface $\neg T.Cierto$.

   En efecto: Por reducción al absurdo: Si ι no satisface $\neg T.Cierto$, por "tercio excluido", ι satisface $T.Cierto$, de donde toda ejecución desde ι termina.