ISIS 1105 Diseño y Análisis de Algoritmos
Semestre 2011-10 – Parcial 1
Septiembre 2, 2011
Prof. Rodrigo Cardoso

1 ***[25/100]***

El patio de la casa de María tiene un camino de baldosas. Las baldosas del camino se numeran 1, 2, …, n, para cierto n>0. Estando en la baldosa k, María puede saltar a la k+1 o a la baldosa k+2. Se quiere estimar de cuántas maneras diferentes puede María recorrer todo el camino.

Ejemplo: Si fueran 4 baldosas, serían 5 maneras (el primer paso es "desde la baldosa 0"): $0\to1\to2\to3\to4$, $0\to1\to2\to4$, $0\to1\to3\to4$, $0\to2\to3\to4$, $0\to2\to4$.

1a [10/25] Sea m(n)el número de recorridos posibles cuando hay n baldosas. Plantee una recurrencia que defina m(n) en términos de valores anteriores de n.

```
m(1) = 1                        // Recorrido : 0→1
m(2) = 2                        // Recorridos: 0→1→2, 0→2
m(n) = m(n-1) + m(n-2), n≥3     // Recorridos: 0→…(n-1)→n, 0→…(n-2)→n
```

Variante 1a1

```
m(0) = 1                        // Recorrido : 0 (vacío)
m(1) = 1                        // Recorridos: 0→1
m(n) = m(n-1) + m(n-2), n≥2     // Recorridos: 0→…(n-1)→n, 0→…(n-2)→n
```

[10/10]

1b [15/25] Pruebe que $m(n) = \theta(f(n))$ para cierta función exponencial f.

La recurrencia se puede expresar en forma estándar:

$m(n+2) - m(n+1) - m(n) = 0, \; n\geq 0$

$\equiv$

$(E^2 - E - 1)m = 0$

$\equiv \quad \langle\, \varphi=(1+\sqrt{5})/2, \; \psi=(1-\sqrt{5})/2 \,\rangle$

$(E - \varphi)(E - \psi)m = 0$

$\Rightarrow$

Existen A,B constantes, tales que:

$m(n) = A\varphi^n + B\psi^n \quad , \; n\geq 0$

Ahora:

$m(1) = 1 = A\varphi + B\psi$

$m(2) = 2 = A\varphi^2 + B\psi^2$

Nótese que $\varphi>1$, $\psi<-1$. Si A=0, debe pasar que $B = 1/\psi = 2/\psi^2$, de modo que también $\psi=2$, lo que es falso. Por tanto:

$m(n) = \theta(A\varphi^n + B\psi^n)$

$\Rightarrow$

$m(n) = \theta(A\varphi^n)$

$\Rightarrow$

$m(n) = \theta(\varphi^n)$

Se puede llegar al mismo resultado usando las definiciones de la Variante 1a1.

Variante 1b1

Establecer A, B con exactitud

$$A = (\psi-2-\varphi)/\varphi$$
$$B = (2-\varphi)/\psi$$

y comprobar que $A\neq 0$, señalando que $\varphi>1$, $\psi<-1$. La contribución del término $B\psi^n$ no es significativa (tiende a 0), de modo que $m(n) = \theta(\varphi^n)$.

[15/15]

2 ***[25/100]***

Considere un programa GCL de la forma

```
⌈ n,b: nat

⌈ {Pre Q: true}

   b:= 0;
   do n≠0 → if par(n)   → n:= n/2
            [] impar(n) → b,n:= b+1,n-1
            fi;
   od

   {Pos R: b = "No. de bits en la representación binaria de n"}
 ⌋
```

Sea $T(n)$ la complejidad temporal del programa, contando como operación básica el cambiar el valor de la variable b.

2a [10/25] Establezca una recurrencia que defina $T(n)$.

```
T(0) = 0
T(n) = T(n/2)      , si n≠0, par(n)
T(n) = 1+T(n-1)    , si n≠0, impar(n)
```

[10/10]

2b [5/25] Muestre que T no es una función creciente.

Basta mostrar dos argumentos, n, n', con n<n', pero $T(n)>T(n')$.

Por ejemplo:

$$T(3) = 1 + T(2) = 1 + T(1) = 2 + T(0) = 2$$
$$T(4) = T(2) = T(1) = 1 + T(0) = 1$$

[5/10]

2c [5/25] Muestre que $T(2^k) = 1$, si $k\geq 0$.

Por inducción:

Predicado de inducción: $P(k) \equiv T(2^k) = 1$, $k\geq 0$

Caso base: $P(0)$

$T(2^0)$

=

$T(1)$
$=$
$1 + T(0)$
$=$
1

Caso inductivo: $P(k+1)$
HI: $P(k),\ k\geq 0$
$T(2^{k+1})$
$=$ $\langle 2^{k+1}$ es par$\rangle$
$T(2^k)$
$=$ $\langle$HI$\rangle$
1

[5/10]

2d [10/25] Muestre que $T(2^k-1) = k$ ¸si $k\geq 0$.

Por inducción:

Predicado de inducción: $Q(k) \equiv T(2^k-1) = k$ ¸ $k\geq 0$

Caso base: $Q(0)$
$T(2^0-1)$
$=$
$T(0)$
$=$
0

Caso inductivo: $Q(k+1)$
HI: $Q(k),\ k\geq 0$
$T(2^{k+1}-1)$
$=$ $\langle 2^{k+1}-1$ es impar$\rangle$
$1 + T(2^{k+1}-2)$
$=$ $\langle 2^{k+1}-2$ es par$\rangle$
$1 + T(2^k-1)$
$=$ $\langle$HI$\rangle$
$1 + k$

[10/10]

2e [Bono: +10] Muestre que $T(n) = O(\log n)$ ¸si $n>0$.

Para $n>0$, sea 2^k la menor potencia de 2 que sea mayor que n. Entonces: $2^{k-1}\leq n<2^k$, con $k\geq 1$. Se probará por inducción que $T(n)\leq k$.

Si esto se muestra, se tendrá que $k-1 \leq \log n < k$, de modo que $T(n) \leq 1 + \log n$. Por tanto, $T(n) = O(\log n)$.

Para la prueba:

Predicado de inducción: $R(n) \equiv T(n)\leq k$, si $2^{k-1}\leq n<2^k$

Caso base: $R(1)$
Nótese que: $2^{1-1}\leq 1<2^1$, i.e., $k=1$.
Ahora:
$T(1) = 1 \leq 1.$

Caso inductivo: R(n+1)
HI: $T(j) \leq k$, si $2^{k-1} \leq j < 2^k$, para $j \leq n$
Sea k tal que $2^{k-1} \leq n+1 < 2^k$

Ahora:

Caso par(n+1):

```
   T(n+1)
=        ⟨Caso: par(n+1) ⟩
   T((n+1)/2)
≤        ⟨HI: 2^(k-2)≤(n+1)/2<2^(k-1)
   k-1
<
   k.
```

Caso impar(n+1):

```
   T(n+1)
=        ⟨Caso: impar(n+1) ⟩
   1+T(n)
=        ⟨ par(n)⟩
   1+T(n/2)
≤        ⟨HI: 2^(k-2)≤n/2<2^(k-1)
   1+k-1
=
   k.
```

[+10]

3 ***[25/100]***

Enriquezca GCL con una instrucción de asignación condicional AC(x,B,e1,e2), con sintaxis

x:= B ? e1 : e2

donde x es una variable, B un predicado y e1,e2 dos expresiones. La semántica operacional de la instrucción se entiende definida así:

Si B es indefinido, e1 es indefinido o e2 es indefinido, abortar. En otro caso:
Si B es verdadero, asignar x con el valor de e1.
Si B es falso, asignar x con el valor de e2.

3a [10/25] Simule AC(x,B,e1,e2) con un programa GCL que no utilice esta instrucción.

```
AC(x,B,e1,e2) ≅   if  B →     x:= e1
                  [] ¬B →     x:= e2
                  fi
```

[10/10]

3b [10/25] Enriquezca el cálculo de Hoare para GCL con una regla de inferencia que permita concluir la corrección de afirmaciones como

{Q} x:= B ? e1 : e2 {R}

De **3a** se observa que la corrección de AC corresponde a la de un condicional. Una regla de inferencia correspondiente sería:

$$\text{R\_AC:} \quad \frac{Q \wedge B \Rightarrow R[x:= e1], \quad Q \wedge \neg B \Rightarrow R[x:= e2]}{\{Q\}\ x:= B\ ?\ e1 : e2 \quad \{R\}}$$

[10/10]

3c [5/25] Use el resultado de **3b** para mostrar que

`{Q : x=3 ∨ x=5}   x:= x>3 ? x-1 : x+1   {R : x=4}`

Será necesario mostrar que se satisfacen las dos hipótesis de R_AC.

Para $Q \wedge B \Rightarrow R[x:= e1]$:

```
   R[x:= e1]
=
   (x = 4)[x:= x-1]
=
   x-1 = 4
=
   x = 5
=
   x=5 ∧ x>3
=
   (x=3 ∨ x=5) ∧ x>3
=
   Q ∧ B
```

Para $Q \wedge \neg B \Rightarrow R[x:= e2]$:

```
   R[x:= e2]
=
   (x = 4)[x:= x+1]
=
   x+1 = 4
=
   x = 3
=
   x=3 ∧ ¬(x>3)
=
   (x=3 ∨ x=5) ∧ ¬(x>3)
=
   Q ∧ ¬B
```

[5/25]

4 ***[25/100]***

En la teoría de la semántica axiomática de programas, considere los siguientes enunciados

P1: $\{Q\}S\{R1\} \vee \{Q\}S\{R2\} \Rightarrow \{Q\}S\{R1 \vee R2\}$

P2: $\{Q\}S\{R1 \vee R2\} \Rightarrow \{Q\}S\{R1\} \vee \{Q\}S\{R2\}$

4a [15/25] Justifique operacionalmente que P1 vale siempre.

Supóngase cierto que $\{Q\}S\{R1\} \vee \{Q\}S\{R2\}$.

Sea x un estado que cumple Q. Por hipótesis, es cierto que una ejecución que parta de x debe terminar en R1 o en R2. Por tanto, debe terminar en $R1 \vee R2$.

[15/15]

4b [10/25] Muestre un contraejemplo que indique que P2 no vale siempre.

Sean

$Q \equiv x=0$

```
R1 ≡ x=0
R2 ≡ x=1

S: if    x=0 → skip
   []    x=0 → x:= 1
   fi
```

Ahora:

```
{Q} S {R1 ∨ R2} es verdadero.
{Q} S {R1} es falso
{Q} S {R2} es falso
```

Todo contraejemplo debe ser no-determinístico.

[10/10]

4b [Bono: +10] Defina que un programa sea *determinístico* si y solo si, para cualquier trío de condiciones Q, R1, R2 se cumple la condición P2. Justifique operacionalmente esta definición.

Cuando S es determinístico, cada ejecución desde un estado dado sigue un camino único.

Supóngase que S cumple P2. Sea x un estado que satisface Q y que hay dos ejecuciones de S, a partir de x, que terminan en estados diferentes y1,y2.

Ahora:

```
   true
⇒     ⟨Hipótesis⟩
   {x}S{y1,y2}
⇒     ⟨P2 vale⟩
   {x}S{y1} ∨ {x}S{y2}
⇒     ⟨y1≠y2 ⇒ ¬{x}S{y1}; y1≠y2 ⇒ ¬{x}S{y2}⟩
   false
```

Entonces, y1=y2 y la ejecución es determinística.

[+10]