ISIS 1105 Diseño y Análisis de Algoritmos
Semestre 2014-10 – Parcial 1
Septiembre 9, 2014
Prof. Rodrigo Cardoso

Nombre: ______________________________
Sección: 1

1 ***[30/100]***

Enriquezca GCL con una instrucción `ForAll`,con sintaxis

`ForAll(z,m,n,k,p)`

donde `z:`**`bool`**, `m,n,k:`**`int`** y `p` es una expresión booleana. El significado pretendido para la nueva instrucción es que se porte de manera equivalente a una instrucción como:

`z,k:= (∀j| m≤j<n: p[k:= j]),n`

1a *[15/30]* Simule `ForAll(z,m,n,k,p)` con un programa `GCL` que no utilice esta instrucción. Suponga que el lenguaje no permite usar expresiones booleanas con cuantificadores, como la mostrada en la asignación.

```
ForAll     ≅     k,z:= m,true;
                 do k≠n →    z,k:= z∧p,k+1   od
```

[15/15]

1b *[15/30]* Enriquezca el Cálculo de Hoare con una regla de inferencia que permita concluir la corrección de afirmaciones como

`{Q}  ForAll(z,m,n,k,p) {R}`

Como en la simulación hay un **do** … **od** involucrado, es necesario considerar un invariante `P` y una cota `t` adecuadas. Parafraseando la regla de inferencia de un **do … od**, la regla de Hoare nueva sería así:

ForAll:		
	(1) Q ⇒ P[k,z:= m,true]	z: **bool**
	(2) P ∧ k=n ⇒ R	m,n,k: **int**
	(3) P ∧ k≠n ⇒ P[z,k:= z∧p,k+1]	
	{Q} ForAll(z,m,n,k,p) {R}	

La terminación está garantizada porque `k` cuenta desde `m` hasta `n` (un intervalo finito que, incluso, puede ser vacío).

[15/15]

Variante:

Regla exigiendo terminación

[12/15]

2 ***[40/100]***

El siguiente algoritmo sirve para decidir si un número `n` es un *palíndrome* (i.e., en notación decimal sin `0`'s a la izquierda, se lee igual de izquierda a derecha que de derecha a izquierda):

[Ctx: n: **nat**

{Pre Q: n = $(d_{S-1}d_{S-2}...d_0)_{10}$}

{Pos R: pal ≡ (∀j| 0≤j<S: $d_j = d_{S-j-1}$)}

⌈ (1) s,a:= 0,1;

{Inv P0: a≤n ∧ (∃s|: a = 10^s)}

{Cota t0: n-10*a]

```
(2)    do n≥10*a → (2.1) a:= 10*a od;

       {Q1:  a≤n<10*a ∧ (∃s|: a = 10^s)}

(3)    m,k,pal:= n,0,true;

       {Inv P: (∃k| 0≤k≤S: pal ≡ (∀j| 0≤j<k: d_j=d_{S-j-1})
                   ∧ m=(d_{S-k-1}d_{S-k-2}...d_k)_10 ) ∧ a≤m<10*a ∧ (∃s|: a = 10^s) }
       {Cota t: m-10}
(4)    do pal ∧ m>9 →
             (4.1) d0:= m mod 10;
             (4.2) du:= m÷a;          // ÷ es division entera

                   {Q2: "d0: dígito 0 de m" ∧ "du: último dígito de m"}

             (4.3) if d0=du → (4.3.a) m:= (m-10*du)÷10; // quita: d0,du
                              (4.3.b) a:= a÷100

                   [] d0≠du → (4.3.c) pal:= false
                   fi
       od
]|
```

2a *[20/40]* Enuncie todas las *obligaciones de prueba*, i.e., las condiciones que deben cumplirse para garantizar la corrección del algoritmo, con las anotaciones dadas. Use las etiquetas para no reescribir el código, pero no deje indicadas instrucciones que no sean secuencias de asignaciones. *No es necesario incluir las condiciones para demostrar la terminación.*

[1] P0 vale antes	:	{Q} (1) {P0}
[2] P0 sirve	:	P0 ∧ n<10*a ⇒ Q1
[3] P0 invariante	:	{P0 ∧ n≥10*a} (2.1) {P0}
[4] P vale antes	:	{Q1} (3) {P}
[5] P sirve	:	P ∧ ¬(pal ∧ m>9) ⇒ R
[6] P invariante	:	{P0 ∧ n≥10*a} (4.1);(4.2) {Q2}
[6.1]	:	Q2 ⇒ (d0=du ∨ do≠du)
[6.2]	:	{Q2 ∧ d0=du}(4.3a);(4.3b) {P}
[6.3]	:	{Q2 ∧ d0≠du}(4.3c) {P}

[20/20]

2b *[20/40]* Encuentre una función f que estime T(n), la complejidad temporal del algoritmo en el peor caso, como θ(f(n)). Considere como operaciones básicas las multiplicaciones, divisiones enteras y módulos.

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

$T(1)$ y $T(3)$ no hacen operaciones aritméticas, i.e.,

$T(1)=0$
$T(3)=0$.

$T(2)$ hace s iteraciones, cada una con 2 multiplicaciones (1 en la guarda, 1 en el cuerpo), desde 1 hasta 10^r, con $10^s \le n < 10^{s+1}$. Es decir, $s \le \log_{10} n < s+1$, o bien:

$T(2) = 2 * \lfloor \log_{10} n \rfloor = \theta(\log n)$.

$T(4)$ es un ciclo que se repite tantas veces como la mitad de los dígitos de n. Es decir, se repite $\theta(\log n)$ veces. En el peor caso, el cuerpo del ciclo hace

(4.1): 1 módulo
(4.2): 1 división
(4.3.a): 1 multiplicación + 1 división
(4.3.ab): 1 división

O sea: 5 operaciones. Por tanto:

$$T(4) = \theta(\log n) * \theta(1) = \theta(\log n).$$

Es decir:

$$T(n) = \theta(\log n) + \theta(\log n) = \theta(\log n)$$

[20/20]

3 ***[30/100]***

El inventario de una tienda obedece a unas reglas de mercado conocidas.

Sean $p(n)$, $c(n)$ y $v(n)$ el inventario, las compras y las ventas en el día n. La regla $r1$ describe el inventario en el día $n+1$ así:

[r1] $p(n+1) = p(n) + c(n) - v(n), \quad n \ge 0$

Las compras y las ventas en $n+1$ se comportan así (α es una constante positiva, $\alpha \ne 1$):

[r2] $c(n+1) = \alpha c(n) + v(n), \quad n \ge 0$
[r3] $v(n+2) = v(n), \quad n \ge 0$

Y se sabe que:

[r4] $p(0) = X$
[r5] $c(0) = Y$
[r6] $v(0) = Z$
[r7] $v(1) = W$

3a ***[10/30]*** Encuentre un expresión cerrada para $v(n)$, $n \ge 0$.

Por $r3$:

$v(n+2) - v(n) = 0, \quad n \ge 0$

$\equiv$

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

$\equiv$

$(E-1)(E+1)v = 0$

Deben existir constantes A, B tales que

$v(n) = A + B*(-1)^n$, $n \geq 0$.

Usando r6 y r7:

$v(0) = Z = A + B$
$v(1) = W = A - B$

i.e., $v(n) = (Z+W)/2 + (-1)^n (Z-W)/2$, $n \geq 0$.

[10/10]

3b *[10/30]* Muestre que $c(n) = O(\alpha^n)$.

Por r2 las compras cumplen que

$c(n+1) - \alpha c(n) = v(n)$, $n \geq 0$

$\equiv$

$(E-\alpha)c = v$

$\equiv$ $\langle (E-1)(E+1)v = 0 \rangle$

$(E-1)(E+1)(E-\alpha)c = 0$

Es decir, (note que $0<\alpha\neq 1$, de modo que no hay repeticiones de raíces) deben existir constantes C,D,F tales que

$c(n) = C + (-1)^n D + F\alpha^n$, $n \geq 0$

Dos casos:

Si $\alpha>1$, se tiene que $\lim_{n\to\infty} \alpha^n/c(n) = F$. En el caso general F no es 0 y $c(n) = O(\alpha^n)$.

[10/10]

Si $\alpha<1$, se tiene que $\lim_{n\to\infty} \alpha^n = 0$ y $c(n) = O(|C+D|)$, o bien, $c(n) = O(1)$.

[Bono +3/3]

3c *[10/30]* Muestre que, en general, $p(n)$ es lineal si $\alpha<1$ y es exponencial si $\alpha>1$.

Por r1:

$p(n+1) = p(n) + c(n) - v(n)$, $n \geq 0$

$\equiv$

$(E-1)p = c - v$

$\equiv$ $\langle (E-1)(E+1)v = 0,\ (E-1)(E+1)(E-\alpha)c = 0 \rangle$

$(E-1)(E-1)(E+1)(E-\alpha)p = 0$

$\equiv$

$(E-1)^2(E+1)(E-\alpha)p = 0$

De nuevo, recuérdese que $0<\alpha\neq 1$, de manera que α no es una raíz repetida. Deben existir constantes F,G,H,J, tales que

$p(n) = F + Gn + (-1)^n H + J\alpha^n$, $n \geq 0$

Entonces:

$p(n) = O(\max(1,n,(-1)^n,\alpha^n)$
$\quad\; = O(\max(n, \alpha^n))$

Es decir:

$p(n) = O(n)$, si $\alpha<1$, porque $\lim_{n\to\infty} \alpha^n/n = 0$

$p(n) = O(\alpha^n)$, si $\alpha>1$, porque $\lim_{n\to\infty} n/\alpha^n = 0$.

[10/10]