ISIS 211 Diseño de algoritmos
Semestre 2003-2 * Parcial No. 1
Septiembre 25, 2003
Prof. Rodrigo Cardoso

1 **[30/100]**

Suponga que GCL se enriquece con una nueva instrucción

`S1 [] S2`

cuya semántica operacional sea "ejecutar no determinísticamente `S1` o `S2`"

1a ***[15/30]*** Defina una regla de inferencia que aumente el Cálculo de Hoare para poder probar la corrección de esta nueva instrucción con respecto a una especificación.

1b ***[15/30]*** Use el cálculo extendido según **1a** para mostrar que, si puede demostrar que

`{Q} S1 [] (S2 [] S3) {R}`

también debe poderse demostrar que

`{Q} (S1 [] S2) [] S3 {R}`

---

2 **[30/100]**

2a ***[25/30]*** Escriba código GCL que cumpla la siguiente especificación (si lo requiere, use como convención: $0^0 = 1$):

```
[ Ctx C:  n:nat ∧ x:int
  {Pre Q: T}

  ...

  {Inv P : 0≤k≤n ∧ z(x-1) = x^k-1}
  {Cota t: ...}

  do ... od

  {R: z(x-1) = x^n-1}
]
```

Compruebe que su solución mantiene el invariante `P`.

AYUDAS:

- Para averiguar cómo variar `z`, sumar y restar `x` en algún momento, y factorizar `x-1`.
- $1 + x + x^2 + ... + x^i = ?$

2b ***[5/30]*** Cuente asignaciones como operaciones básicas y estime el orden de complejidad de su solución (como $\theta(...)$).

---

3 `[40/100]`

Sea `f[0..n-1]` un arreglo de numeros enteros diferentes, ordenado en forma ascendente. Sea `x` un número entero. Considere el problema de determinar si existen dos números diferentes, dentro del arreglo `f`, cuya suma sea `x`.

Se quiere desarrollar un algoritmo de acuerdo con el siguiente esquema:

```
[Ctx: f[0..n-1] of int  ∧ x:int

 ...

 {Inv P: 0≤i≤icent≤n ∧ 0≤j<n
         ∧ (∀u,v| 0≤u≤i ∧ 0≤j≤v<n ∧ (u,v)≠(i,j): f[u]+f[v]≠x)
         ∧ (icent=n cor f[i]+f[j]=x)}
 {cota t: ... }

 do ... od

 {R1: 0≤i=icent≤n ∧ 0≤j<n ∧
     (∀u,v| 0≤u≤i ∧ 0≤j≤v<n ∧ (u,v)≠(i,j): f[u]+f[v]≠x)
    ∧ (icent=n cor f[i]+f[j]=x)}

 ...

 {Pos R: resp ≡ (∃u,v| 0≤u<n ∧ 0≤v<n : f[u]+f[v] = x) }

]
```

3a *`[5/40]`* Explique qué relación hay entre el invariante y la poscondición (v.gr., "es el resultado de eliminar una conjunción ...").

3b *`[5/40]`* Defina una función cota correspondiente a su estrategia de solución.

3c *`[25/40]`* Escriba una solución correspondiente. Si es necesario, agregue aserciones e instrucciones al final del **`do ... od`**.

3d *`[5/30]`* Cuente asignaciones como operaciones básicas y estime el orden de complejidad de su solución (como `θ(...)`).

____

ISIS 211 Diseño de algoritmos
Semestre 2003-2 * Parcial No. 1
Septiembre 25, 2003
Prof. Rodrigo Cardoso

1 [30/100]

Suponga que GCL se enriquece con una nueva instrucción

`S1 [] S2`

cuya semántica operacional sea "ejecutar no determinísticamente `S1` o `S2`"

1a [15/30] Defina una regla de inferencia que aumente el Cálculo de Hoare para poder probar la corrección de esta nueva instrucción con respecto a una especificación.

Se puede simular la nueva instrucción con código GCL así:

```
S1 [] S2 ≅     if   true → S1
               []   true → S2
               fi
```

lo que sugiere una regla de inferencia derivada de la de los condicionales:

$$\frac{\{Q\}\ S1\ \{R\},\ \{Q\}\ S2\ \{R\}}{\{Q\}\ S1\ []\ S2\ \{R\}}$$

1b [15/30] Use el cálculo extendido según **1a** para mostrar que, si puede demostrar que

`{Q} S1 [] (S2 [] S3) {R}`

también debe poderse demostrar que

`{Q} (S1 [] S2) [] S3 {R}`

Demostrar `{Q} S1 [] (S2 [] S3) {R}`
requiere mostrar `{Q} S1 {R}, {Q} S2 [] S3 {R}`
requiere mostrar `{Q} S1 {R}, {Q} S2 {R}, {Q} S3 {R}`

[8/15]

Comentario: Si se razona operacionalmente: 5/15

La necesidad de estas hipótesis viene de que la regla nueva es la única de la que se dispone para mostrar la corrección de la nueva instrucción.

Ahora:

```
Hip: {Q} S1 {R}, {Q} S2 {R}, {Q} S3 {R}
     true
⇒        ⟨Hip⟩
     {Q} S1 {R} ∧ {Q} S2 {R}
⇒        ⟨Regla []⟩
     {Q} (S1 [] S2) {R}
⇒        ⟨Hip⟩
     {Q} (S1 [] S2) {R}  ∧  {Q} S3 {R}
⇒        ⟨Regla []⟩
     {Q} (S1 [] S2) [] S3 {R}.
```

[7/15]

---

2 **[30/100]**

2a **[25/30]** Escriba código GCL que cumpla la siguiente especificación (si lo requiere, use como convención: $0^0 = 1$):

```
[ Ctx C:  n:nat ∧ x:int
  {Pre Q: T}

  ...

  {Inv P : 0≤k≤n ∧ z(x-1) = x^k-1}
  {Cota t: ...}

  do ... od

  {R: z(x-1) = x^n-1}
]
```

Compruebe que su solución mantiene el invariante P.

Ayudas:

- Para averiguar cómo variar z, sumar y restar x en algún momento, y factorizar x-1.
- $1 + x + x^2 + ... + x^i = ?$

```
[ Ctx C:  n:nat ∧ x:int
  {Pre Q: T}

  z,k:= 0,0;                                         [ 5/25]

  {Inv P : 0≤k≤n ∧ z(x-1) = x^k-1}
  {Cota t: n-k}                                      [ 3/25]

  do   k≠n                                           [ 2/25]
       →        z,k:= A,k+1
  od

       {R: z(x-1) = x^n-1}
]
```

Comentario: Nótese que aquí se usa la convención: $0^0=1$.

Comentario: Nótese que esto no se califica todavía, mientras no se sepa cuánto es A.

***Variante 1**: (usando la primera ayuda)*

Para determinar A, debe valer que

{P ∧ k≠n} z,k:= A,k+1 {P}

es decir:

$0\le k\le n \land z(x-1) = x^k-1 \land k\ne n \Rightarrow 0\le k+1\le n \land A(x-1) = x^{k+1}-1$ **[5/25]**

Comentario: Para saber el valor de A, se busca una expresión que haga que el invariante se satisfaga.

Ahora:

Hip: $0\le k\le n,\ z(x-1) = x^k-1,\ k\ne n$

$A(x-1) = x^{k+1}-1$

=

$A(x-1) = x^{k+1} - x + x - 1$

=

$A(x-1) = x(x^k - 1) + (x - 1)$

Comentario: Aquí se usa el truco de sumar y restar x. En cualquier variante de demostración parece requerirse un truco similar.

$=\qquad \langle \text{Hip: } z(x-1) = x^k-1\rangle$

$\quad A(x-1) = xz(x-1) + (x-1)$

$\Leftarrow$

$\quad A = xz+1$

[10/25]

El código que resulta es:

```
z,k:= 0,0;
do k≠n  →   z,k:= xz+1,k+1  od
```

Variante 2 *(usando la segunda ayuda):*

Se puede haber reconocido que z está calculando una suma geométrica, si se recuerda que, para $x\neq 1$:

$$(+i|\ 0\leq i<n\ :\ x^i) = \frac{x^n-1}{x-1}$$

[5/25]

Por tanto, la poscondición se puede refrasear en (obsérvese que la ecuación vale, incluso, si $x=1$):

$$z=(+i|\ 0\leq i<n\ :\ x^i) \wedge z(x-1) = x^n-1$$

Por su parte, el invariante se puede expresar como

$$P:\ 0\leq k\leq n \wedge z=(+i|\ 0\leq i<k\ :\ x^i) \wedge z(x-1) = x^k-1$$

Ahora, el mantenimiento del invariante sugiere efectuar $z:= z+x^k$.

[3/25]

Esto redunda en la solución:

```
z,k:= 0,0;
do k≠n  →   z,k:= z+x^k,k+1  od
```

Para mostrar que el invariante se mantiene, debe valer que

$$\{P \wedge k\neq n\}\ z,k:= z+x^k,k+1\ \{P\}$$

es decir:

$$0\leq k\leq n \wedge z=(+i|\ 0\leq i<k\ :\ x^i) \wedge z(x-1) = x^k-1 \wedge k\neq n$$
$$\Rightarrow\quad 0\leq k+1\leq n \wedge z+x^k = (+i|\ 0\leq i<k+1\ :\ x^i) \wedge (z+x^k)(x-1) = x^{k+1}-1$$

Ahora:

Hip: $0\leq k\leq n,\ z(x-1) = x^k-1,\ z=(+i|\ 0\leq i<k\ :\ x^i),\ k\neq n$

$\quad \text{true}$

$\Rightarrow\qquad \langle \text{Hip: } 0\leq k\leq n,\ k\neq n\rangle$

$\quad 0\leq k+1\leq n$

$\Rightarrow\qquad \langle \text{Hip: } z = (+i|\ 0\leq i<k\ :\ x^i)\rangle$

$\quad 0\leq k+1\leq n \wedge z+x^k = (+i|\ 0\leq i<k+1\ :\ x^i)$

$\Rightarrow\qquad \langle \text{Hip: } z(x-1) = x^k-1\rangle$

$\quad 0\leq k+1\leq n \wedge z+x^k = (+i|\ 0\leq i<k+1\ :\ x^i) \wedge (z+x^k)(x-1) = x^{k+1}-1$

[7/25]

La solución se puede expresar así, solo que parece exigir elevar a la potencia k. Se podría fortalecer el invariante, pero es más sencillo observar que, si el invariante vale, entonces

$$z(x-1) = x^k-1 \equiv z+x^k = zx+1$$

[+5/25]

lo que simplifica la solución encontrada y, de paso, muestra que equivale a la primera solución.

2b **[5/30]** Cuente asignaciones como operaciones básicas y estime el orden de complejidad de su solución (como $\theta(...)$).

La cota natural es `t = n-k`. Empieza en `n` y, en cada iteración, rebaja en `1`. La complejidad es $\theta(n)$.

[5/30]

3 **[40/100]**

Sea `f[0..n-1]` un arreglo de numeros enteros diferentes, ordenado en forma ascendente. Sea `x` un número entero. Considere el problema de determinar si existen dos números diferentes, dentro del arreglo `f`, cuya suma sea `x`.

Se quiere desarrollar un algoritmo de acuerdo con el siguiente esquema:

```
[Ctx: f[0..n-1] of int  ∧ x:int

 ...

 {Inv P: 0≤i≤icent≤n ∧ 0≤j<n
          ∧ (∀u,v| 0≤u≤i ∧ 0≤j≤v<n  ∧ (u,v)≠(i,j): f[u]+f[v]≠x)
         ∧ (icent=n cor f[i]+f[j]=x)}
 {cota t: ... }

 do ... od

 {R1: 0≤i=icent≤n ∧ 0≤j<n
         ∧ (∀u,v| 0≤u≤i ∧ 0≤j≤v<n  ∧ (u,v)≠(i,j): f[u]+f[v]≠x)
    ∧ (icent=n cor f[i]+f[j]=x)}

 ...

 {Pos R: resp ≡ (∃u,v| 0≤u<n ∧ 0≤v<n : f[u]+f[v] = x) }

]
```

[+5/40]

Comentario: Bomo de +5/40 por el error en la anotación (Inv y Pos con errores).

3a **[5/40]** Explique qué relación hay entre el invariante y R1 (v.gr., "es el resultado de eliminar una conjunción ...").

El invariante es el resultado de aumentar el rango de variables de la poscondición. Tanto `i` como `icent` tienen un rango de variación más grande en el invariante.

3b **[5/40]** Defina una función cota correspondiente a su estrategia de solución.

```
cota t: if f[i]+f[j]≠x then (n-i)(j+1) else 0 fi
```

[5/40]

Comentario: Si contesta (n-i)(j+1): 4/40.

Corresponde al área de los índices no explorados en la búsqueda de la condición.

3c **[25/40]** Escriba una solución correspondiente. Si es necesario, agregue aserciones e instrucciones al final del **do ... od**.

```
[Ctx: f[0..n-1] of int  ∧ x:int

 i,j,icent:= 0,n-1,n;                                                 [ 3/25]

 {Inv P: 0≤i≤icent≤n ∧ 0≤j<n
          ∧ (∀u,v| 0≤u≤i ∧ 0≤j≤v<n  ∧ (u,v)≠(i,j): f[u]+f[v]≠x)
         ∧ (icent=n cor f[i]+f[j]=x)}
 {cota t: (n-i)(j+1)}

 do i≠icent                                                           [ 2/25]

       →   if   f[i]+f[j] = x    →    icent:= i                       [ 5/25]

           []   f[i]+f[j] < x    →    if  j=0 → i:= n                 [ 5/25]
                                      []  j≠0 → j:= j-1
                                      fi

           []   f[i]+f[j] > x    →    i:= i+1                         [ 5/25]
           fi
 od

 {R1: 0≤i=icent≤n ∧ 0≤j<n
          ∧ (∀u,v| 0≤u≤i ∧ 0≤j≤v<n  ∧ (u,v)≠(i,j): f[u]+f[v]≠x)
         ∧ (icent=n cor f[i]+f[j]=x)}

 resp:= (i≠n ∧ i≠j)                                                   [ 5/25]

 {Pos R: resp ≡ (∃u,v| 0≤u<n ∧ 0≤v<n : f[u]+f[v] = x) }

]
```

Comentario: Se podría usar 2 centinelas. Si se hiciera así, se simplificaría a "`j:= j-1`" el tratamiento del caso "`f[i]+f[j] < x`", porque `jcent`, el centinela de `j`, se inicializaría en `-1`. Pero este no es el caso, ya que el invariante sólo habla de un centinela.

3d **[5/40]** Estime la complejidad temporal de la solución (cuente asignaciones, $\theta(...)$).

En cada iteración se reduce una fila o una columna. Hay `2n` de ellas.

En total: $T(n) = \theta(n)$.

[5/40]