ISIS 211 Diseño de algoritmos
Semestre 2003-1 * Parcial No. 1
Febrero 27, 2003
Prof. Rodrigo Cardoso

1 **[30/100]**

Sean $B_1,\ldots, B_n$ predicados, $S_0, S_1 \ldots, S_{n-1}, S_n$ instrucciones, $n>0$.
Enriquezca GCL con una instrucción ITE_n, de la forma

```
ITEn :   if   Bn   then Sn
         elsf Bn-1 then Sn-1
         elsf ...
         elsf B1   then S1
                   else S0
         fi
```

Con la semántica operacional correspondiente a un condicional secuencial, i.e.:
Si B_n vale: se ejecuta S_n y se termina la ejecución.
Si B_n no vale, pero B_{n-1} vale: se ejecuta S_{n-1} y se termina la ejecución.
...
Si B_n , $B_{n-1},\ldots, B_2$ no valen, pero si B_1 vale: se ejecuta S_1 y se termina la ejecución.
En otro caso: se ejecuta S_0 y se termina la ejecución.

1a **[5/30]** Simule ITE_1 con un programa GCL.

1b **[10/30]** Simule ITE_{k+1}, $1<k\leq n$, con un programa GCL (eventualmente, usando una simulación de ITE_k).

1c **[20/30]** Enuncie reglas de inferencia que enriquezcan el Cálculo de Hoare de forma que sea posible concluir la corrección de afirmaciones como

$$\{Q\}\ ITE_n\ \{R\}$$

2 **[30/100]**

2a **[24/30]** Escriba código GCL que cumpla la siguiente especificación

```
[ Ctx C:  b:[0..n-1] of nat ∧ x:nat ∧ b = B
  ...
  {Inv P : (∀i| 0≤i<p : b[i]<x) ∧ (∀i| q≤i<n : b[i]≥x) ∧ 0≤p≤q<n
           ∧ perm(b,B)}
  {Cota t: q-p}

  do ... od

  {R: (∀i| 0≤i<p : b[i]<x) ∧ (∀i| p≤i<n : b[i]≥x) ∧ perm(b,B)}
]
```

2b **[6/30]** Cuente asignaciones como operaciones básicas y estime el orden de complejidad de su solución (como $\theta(\ldots)$).

3 **[50/100]**

Sean `n` un entero positivo y `f[0..n-1]` un arreglo de números naturales no nulos. Para `0≤ i,j≤n`, defínase

```
suma(f;i,j) := (+k: i≤k<j : f[k]).
```

Considere el problema de contar el número de parejas

```
⟨p,q⟩ ∈ 0..n x 0..n,
```

tales que:

```
suma(f;0,p) = suma(f;q,n).
```

Para resolverlo se desea escribir un programa GCL que satisfaga:

```
[Ctx C: n: nat+ ∧ f: array[0..n-1] of nat+
 {Pre Q: T}
 ...
 {Inv P :    0≤p≤n ∧ 0≤q≤n ∧
         r + C(p,q) = C(n,0) ∧ sp = suma(f;0,p) ∧ sq = suma(f;q,n) }
 {Cota t: ...}
 do ...
 od
 {Pos R: r = C(n,0) } }
]
```

donde:

```
C(p,q) = #{⟨j,k⟩:0..p x q..n | suma(f;0,j)=suma(f;k,n)}
```

Note que:

```
C(0,q) = C(p,n) = 1.
```

3a Explique qué relación hay entre el invariante y la poscondición (v.gr., “es el resultado de eliminar una conjunción ...”).

3b Defina una función cota correspondiente a su estrategia de solución.

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

3d Estime la complejidad temporal de la solución (cuente asignaciones, $\theta(...)$).

ISIS 211 Diseño de algoritmos
Semestre 2003-1 * Parcial No. 1
Febrero 27, 2003
Prof. Rodrigo Cardoso

1 [30/30]

Sean $B_1,\ldots, B_n$ predicados, $S_1,\ldots, S_n, S_{n+1}$ instrucciones, $n>0$.
Enriquezca GCL con una instrucción ITE_n, de la forma

```
ITEn :  if   Bn    then Sn
        elsf Bn-1  then Sn-1
        elsf ...
        elsf B1    then S1
                   else S0
        fi
```

Con la semántica operacional correspondiente a un condicional secuencial, i.e.:
Si B_n vale: se ejecuta S_n y se termina la ejecución.
Si B_n no vale, pero B_{n-1} vale: se ejecuta S_{n-1} y se termina la ejecución.
...
Si B_n , $B_{n-1},\ldots, B_2$ no valen, pero si B_1 vale: se ejecuta S_1 y se termina la ejecución.
En otro caso: se ejecuta S_0 y se termina la ejecución.

1a Simule ITE_1 con un programa GCL.
[5/30]

$ITE_1 \cong$ **if** $B_1 \to S_1$ [] $\neg B_1 \to S_0$ **fi**

1b Simule ITE_{k+1}, $1<k\leq n$, con un programa GCL (eventualmente, usando una simulación de ITE_k).
[10/30]

$ITE_k \cong$ **if** $B_k \to S_k$ [] $\neg B_k \to ITE_{k-1}$ **fi**

1c Enuncie reglas de inferencia que enriquezcan el Cálculo de Hoare de forma que sea posible concluir la corrección de afirmaciones como

$$\{Q\}\ ITE_n\ \{R\}$$

[15/30]

[5/30]

Para $k=1$:

$$\frac{\{Q \land B_1\}\ S_1\ \{R\},\ \{Q \land \neg B_1\}\ S_0\ \{R\}}{\{Q\}\ ITE_1\ \{R\}}$$

[10/30]

Para $1<k\leq n$:

$$\frac{\{Q \land B_k\}\ S_k\ \{R\},\ \{Q \land \neg B_k\}\ ITE_{k-1}\ \{R\}}{\{Q\}\ ITE_k\ \{R\}}$$

2 [30/30]

2a Escriba código GCL que cumpla la siguiente especificación
[24/30]

[Ctx C: $b:[0..n-1]$ **of nat** $\land$ x:**nat** $\land$ $b=B$

[6/30]

```
    p,q:= 0,n;

  {Inv P : (∀i| 0≤i<p: b[i]<x) ∧ (∀i| q≤i<n : b[i]≥x) ∧ 0≤p≤q≤n ∧ perm(b,B)}
    {Cota t: q-p}
                                                                    [6/30]
    do p≠q  →
                                                                   [12/30]
                if   b[p]<x    →    i,p:= i+1,p+1
                []   b[p]≥x    →    b[p],b[q-1],q:= b[q-1],b[p],q-1
                fi
    od

    {R: (∀i| 0≤i<p : b[i]<x) ∧ (∀i| p≤i<n : b[i]≥x) ∧ perm(b,B)}
  ]
```

Variante:

```
    p,q:= 0,n;

    {Inv P : (∀i| 0≤i<p: b[i]<x) ∧ (∀i| q≤i<n: b[i]≥x) ∧ 0≤p≤q≤n ∧ perm(b,B)}
    {Cota t: q-p}

    do p≠q  →   if   b[p]<x    →    i,p:= i+1,p+1
                []   b[p]≥x    →    q:= q-1;
                                    b[p],b[q]:= b[q],b[p]
                fi
    od
```

etc. (intercambiar con temporal, ...)

2b Cuente asignaciones como operaciones básicas y estime el orden de complejidad de su solución (como `θ(...)`).

[6/30]

[6/30]

La complejidad es `θ(n)`. Razón: La cota empieza en `n`, rebaja en `1` en cada iteración y realiza una asignación en cada iteración.

3 **[50/100]**

Sean `n` un entero positivo y `f[0..n-1]` un arreglo de números naturales no nulos. Para `0≤i,j≤n`, defínase

`suma(f;i,j) := (+k: i≤k<j : f[k]).`

Considere el problema de contar el número de parejas

`⟨p,q⟩ ∈ 0..n x 0..n,`

tales que:

`suma(f;0,p) = suma(f;q,n).`

Para resolverlo se desea escribir un programa GCL que satisfaga:

```
[Ctx C: n: nat⁺ ∧ f: array[0..n-1] of nat⁺
```

```
 {Pre Q: T}

 ...

 {Inv P : 0≤p≤n ∧ 0≤q≤n ∧
          r + C(p,q) = C(n,0) ∧ sp = suma(f;0,p) ∧ sq = suma(f;q,n) }
 {Cota t: ...}

 do  ...

 od

 {Pos R: r = C(n,0) } }
]
```

donde:

```
     C(p,q) = #{⟨j,k⟩:0..p x q..n | suma(f;0,j)=suma(f;k,n)}
```

Note que:

```
     C(0,q) = C(p,n) = 1.
```

3a Explique qué relación hay entre el invariante y la poscondición (v.gr., "es el resultado de eliminar una conjunción ...").
[6/50]

[6/50]

El invariante es un predicado que enuncia un *balance de información*

```
          0≤p≤n ∧ 0≤q≤n ∧   r + C(p,q) = C(0,n)
```

fortalecido con el mantenimiento de las variables `sp` y `sq`:

```
                    sp = suma(f;0,p) ∧ sq = suma(f;q,n)
```

3b Defina una función cota correspondiente a su estrategia de solución.
[7/50]

[7/50]

```
Cota t: n-p+q
```

3c Escriba una solución correspondiente. Si es necesario, agregue aserciones e instrucciones al final del **do ... od**.
[30/50]

```
[Ctx C: n: nat⁺ ∧ f: array[0..n-1] of nat⁺
 {Pre Q: T}
```

[5/50]

```
 p,q,r:= 0,n,0;
 sp,sq:= 0,0;

 {Inv P:  0≤p≤n ∧ 0≤q≤n ∧
          r + C(p,q) = C(n,0) ∧ sp = suma(f;0,p) ∧ sq = suma(f;q,n) }
 {Cota t: n-p+q}
```

[10/50]

```
 do  p≠n ∧ q≠0 →
```

[15/50]

```
     if   sp<sq    →    p,sp:= p+1,sp+f[p]
     []   sp=sq    →    r,p,q,sp:= r+1,p+1,sp+f[p]
     []   sq<sp    →    q,sq:= q-1,sq+f[q-1]
```

```
        []   sq=sp     →    r,q,sq:= r+1,q-1,sp+f[q-1]
        fi

    od;
    {R1: P ∧ (p=n ∨ q=0)}
    {R2: r + 1 = C(n,0)}
                                                             [+5/50]
     r:= r+1
    {R: r = C(n,0)}
  ]
```

Variante 1

En el cuerpo del ciclo, en condición de igualdad `sp=sq` se pueden adelantar las dos acciones que en la solución mostrada se indica. Esto por el hecho de que los valores de `f` son estrictamente positivos.

Variante 2

La guarda puede obligar a terminar recorriendo los dos arreglos, i.e., `p≠n ∨ q≠0`.

3d Estime la complejidad temporal de la solución (cuente asignaciones, `θ(...)`).
[7/50]

[7/50]

`θ(n).` La cota vale `n` al principio y rebaja en `1` en cada iteración.