Por la presente declaro que la resolucion de este examen es obra de mi exclusiva autoría y respetando las pautas y criterios fijados en los enunciados. Asimismo, declaro conocer el regimen de infraccion de los estudiantes cuyo texto ordenado se encuentra en el apéndice de la Res. Rec 1554/2018

Molina
Franco

Molina Franco
44192153

AYED 1

9/2/21

1) $f.xs.ys = \langle \text{Max } as, bs, cs : xs = as \mathbin{+\!\!+} bs \wedge ys = as \mathbin{+\!\!+} cs : \#as \rangle$

a) f calcula el largo maximo de un posible prefijo que pertenegca tanto a xs como a ys

b). $f.[1,2,3,4,1,2].[1,2,4] =$

$= \langle \text{Max } as, bs, cs : [1,2,3,4,1,2] = as \mathbin{+\!\!+} bs \wedge [1,2,4] = as \mathbin{+\!\!+} cs : \#as \rangle$

para esto calcularemos las diferentes formas de separar xs y ys tal que entren en el rango

- $[1,2,3,4,1,2] = [\,] \mathbin{+\!\!+} [1,2,3,4,1,2] \wedge [1,2,4] = [\,] \mathbin{+\!\!+} [1,2,4]$
- $[1,2,3,4,1,2] = [1] \mathbin{+\!\!+} [2,3,4,1,2] \wedge [1,2,4] = [1] \mathbin{+\!\!+} [2,4]$
- $[1,2,3,4,1,2] = [1,2] \mathbin{+\!\!+} [3,4,1,2] \wedge [1,2,4] = [1,2] \mathbin{+\!\!+} [4]$

ahora calculamos el tamaño de los posibles prefijos

$\#[\,] = 0$ $\#[1] = 1$ $\#[1,2] = 2$

y determinamos que el maximo es 2

$\boxed{f.[1,2,3,4,1,2].[1,2,4] = 2}$

c) Para derivar una funcion recursiva primero trabajare con el caso base (CB) de dicha funcion, y luego tomando la especificacion dada como hipotesis inductiva (HI), trabajare en su caso caso inductivo (CI)

CB: $f.[\,].ys$

$\{esp\}$

$= \langle \text{Max } as, bs, cs : [\,] = as \mathbin{++} bs \wedge ys = as \mathbin{++} cs : \# as \rangle$

• {logica de la concat.}

$= \langle \text{Max } as, bs, cs : as = [\,] \wedge bs = [\,] \wedge y = cs : \# as \rangle$

• {rango unitario}

$= \#[\,]$

• {caso base de #}

$= 0$

HI: $f.xs.ys = \langle \text{Max } as, bs, cs : xs = as \mathbin{++} bs \wedge ys = as \mathbin{++} cs : \# as \rangle$

CI: $f.(x:xs).ys$

• {esp}

$= \langle \text{Max } as, bs, cs : (x:xs) = as \mathbin{++} bs \wedge ys = as \mathbin{++} cs : \# as \rangle$

• {tercer excluido}

$= \langle \text{Max } as, bs, cs : (x:xs) = as \mathbin{++} bs \wedge ys = as \mathbin{++} cs \wedge (as = [\,] \vee as \neq [\,]) : \# as \rangle$

• {distributividad, particion de rango, para $as = [\,] \rightarrow$ eliminacion de variable
para $as \neq [\,] \rightarrow$ cambio de variable $as \rightarrow a:as$}

$= \langle \text{Max } bs, cs : (x:xs) = bs \wedge ys = cs : \#[\,] \rangle \max$

$\langle \text{Max } a, as, bs, cs : (x:xs) = (a:as) \mathbin{++} bs \wedge ys = (a:as) \mathbin{++} cs : \#(a:as) \rangle$

• {rango unitario, termino constante, logica}

$= 0 \max \langle \text{Max } a, as, bs, cs : (x = a) \wedge xs = as \mathbin{++} bs \wedge ys = [a] \mathbin{++} as \mathbin{++} cs : \# [a] \mathbin{++} as \rangle$

• {eliminacion de variable}

$= 0 \max \langle \text{Max } as, bs, cs : xs = as \mathbin{++} bs \wedge ys = [x] \mathbin{++} as \mathbin{++} cs : \# [x] \mathbin{++} as \rangle$

• Trabajaremos por casos

Molina Franco
44192153

3

- { trabajo por casos }
- $ys = []$

$= 0 \max \langle \text{Max } as, bs, cs : xs = as ++ bs \wedge [] = [x] ++ as ++ cs : \#[x] ++ as \rangle$

- { logica del concat }

$= 0 \max \langle \text{Max } as, bs, cs : xs = as ++ bs \wedge [x] = [] = as = cs : \#[] \rangle$

- { termino constante }

$= 0 \max \#[]$

- { caso base # }

$= 0 \max 0$

- { algebra }

$= 0$

- $ys \neq []$
- { cambio de variable $ys \Rightarrow (y:ys)$ }

$= 0 \max \langle \text{Max } as, bs, cs : xs = as ++ bs \wedge (y:ys) = [x] ++ as ++ bs : \#[x] ++ as \rangle$

- { logica }

$= 0 \max \langle \text{Max } as, bs, cs : xs = as ++ bs \wedge ys = as ++ bs \wedge y = x : \#[x] ++ as \rangle$

- { trabajo de nuevo por casos }
- $y = x$
- { $y = x \Rightarrow$ True , neutro de $\wedge$, logica de # }

$= 0 \max \langle \text{Max } as, bs, cs : xs = as ++ bs \wedge ys = as ++ bs : \#as + 1 \rangle$

- { distributividad y HI }

$= 0 \max (f.xs.ys + 1)$

Ultimo caso

- $y \neq x$

$= 0 \max \langle \text{Max } as, bs, cs : xs = as{+}{+}bs \wedge ys = as{+}{+}cs \wedge False : \#[x]{+}{+}as \rangle$

$\{$absorvente del $\wedge\}$

$= 0 \max \langle \text{Max } as, bs, cs : False : \#[x]{+}{+}as \rangle$

$\{$rango vacio$\}$

$= 0 \max -\infty$

$\{$algebra$\}$

$= 0$

Resultado :

- $f.[].ys = 0$
- $f.(x:xs).[] = 0$
- $f.(x:xs).(y:ys) = \begin{cases} (x = y) \to 0 \max (f.xs.ys + 1) \\ (x \neq y) \to 0 \end{cases}$

notar que $f.(x:xs).[]$ se puede anotar $f.xs.[]$

y que en el caso inductivo, $(0 \max f.xs.ys + 1)$ siempre es max la derecha

Resultado mejor:

- $f.[].ys = 0$
- $f.xs.[] = 0$
- $f.(x:xs).(y:ys) = \begin{cases} (x = y) \to f.xs.ys + 1 \\ (x \neq y) \to 0 \end{cases}$

Molina Franco
44192153

2)

$h.xs = \langle \text{Max } as, bs, cs : xs = as +\!\!+ bs +\!\!+ cs \wedge \langle \forall i : 0 \leq i < \#bs : par.(bs!i) \rangle : \#bs \rangle$

ó (sin par.())

$h.xs = \langle \text{Max } as, bs, cs : xs = as +\!\!+ bs +\!\!+ cs \wedge \langle \forall i : 0 \leq i < \#bs : \ldots$

$\ldots \quad \langle \exists j : 0 \neq j : bs!i = 2j \rangle \rangle : \#bs \rangle$

teniendo en cuenta que 0 no es par

3) a) $A = [3, -1, 1, -1] \rightarrow$ array $(0, 4]$

$n = \langle \text{Max } i : 0 \leq i < 4 \wedge \langle \Sigma j : 0 \leq j < i : Aj \rangle < A.i : A.i \rangle$

enumero los distintos i

- $(\langle \Sigma j : 0 \leq j < 0 : Aj \rangle < A.0) = 0 < 3 = True$
- $(\langle \Sigma j : 0 \leq j < 1 : Aj \rangle < A.1) = 3 < (-1) = False$
- $(\langle \Sigma j : 0 \leq j < 2 : Aj \rangle < A.2) = 2 < 1 = False$
- $(\langle \Sigma j : 0 \leq j < 3 : Aj \rangle < A.3) = 3 < -1 = False$

ahora calculo el termino de los rangos posibles

- $A.0 = 3$

al ser el unico resultado, es el max.

$\boxed{\text{El resultado es } 3}$

b) el programa calcula el maximo numero del array que cumple con tener un valor mayor a la suma de todos los anteriores.

c) para empezar notamos la necesidad de un ciclo para recorrer el array. y una inisialización planteo lo que sabemos y las espesificaciones necesarias para trabajar

Const M: Int, A: array $[0, M)$ of Int
Var n: Int
$\{M \geq 0\}$
S_0;
$\{I\}$
do $(b) \rightarrow$
$\{I \wedge b \wedge t = T\}$
S_1;
$\{I \wedge t < T\}$
od
$\{n = \langle \text{Max } i : 0 \leq i < M \wedge \langle \Sigma j : 0 \leq j < i : A.j \rangle < A.i : A.i \rangle\}$

ahora necesitamos calcular el invariante, para esto utilizaré, cambio de constante por variable.

$\{I : n = \langle \text{Max } i : 0 \leq i < m \wedge \langle \Sigma j : 0 \leq j < i : A.j \rangle < A.i : A.i \rangle \wedge 0 \leq m \leq M\}$

con esto sabemos que m se inicializará en su minimo e irá creciendo hasta ser igual a M.

con esto podemos calcular nuestra guarda y cota tal que:

$b = (m \neq M)$, $t = (M - m)$

con todo esto podemos empezar a derivar la asignación dentro del ciclo. con la siguiente terna

$\{n = \langle \text{Max } i : 0 \leq i < m \wedge \langle \Sigma j : 0 \leq j < i : A.j \rangle < A.i : A.i \rangle \wedge 0 \leq m \leq M \wedge m \neq M \wedge (M - m) = T\}$
$n, m := E, m+1$
$\{I \wedge (M - m) < T\}$

tomo la primera espesificación como hipotesis

y trabajo en la WP de la asignación

$WP. \{I \wedge (M - m) < T\} (n, m := E, m+1)$

Molina Franco
44192153

• {asigno}

$E = \langle Max\ i : 0 \leq i < m+1 \wedge \langle \Sigma j : 0 \leq j < i : A.j \rangle < A.i : A.i \rangle \wedge 0 \leq m+1 \leq M \wedge M-(m+1) < T$

• {por HI $0 \leq m \leq M \wedge m \neq M \rightarrow 0 \leq m < M$ por lo que $0 \leq m+1 \leq M$ = True}

• {por HI $T = (M-m)$, remplazo}

$E = \langle Max\ i : 0 \leq i \leq m \wedge \langle \Sigma j : 0 \leq j < i : A.j \rangle < A.i : A.i \rangle \wedge True \wedge (M-m-1 < M-m)$

• {algebra y neutro del $\wedge$}

• {logica $i = m \vee 0 \leq i < m$, particion de rango}

$E = \langle Max\ i : 0 \leq i < m \wedge \langle \Sigma j : 0 \leq j < i : A.j \rangle < A.i : A.i \rangle \max \langle Max\ i : i = m \wedge \langle \Sigma j : 0 \leq j < i : A.j \rangle < A.i : A.i \rangle$

• {por HI de def. de n}

• {rango unitario con condicion}

$E = n \max (\langle \Sigma j : 0 \leq j < m : A.j \rangle < A.m \Rightarrow A.m) [\,] (\langle \Sigma j : 0 \leq j < m : A.j \rangle \geq A.m \Rightarrow -\infty)$

para continuar necesito fortalecer el invariante

$\{ I' = I \wedge k = \langle \Sigma j : 0 \leq j < m : A.j \rangle \}$

planteamos de nuevo la terna a laburar

$\{ I' \wedge m \neq M \wedge (M-m) = T \}$
$n, m, k = E, m+1, F$
$\{ I' \wedge (M-m) < T \}$

tomo la primera de hipotesis y laburaré con la wp

$wp \{ I' \wedge (M-m) < T \} (n, m, k = E, m+1, F)$

• {asigno}

$(E = \langle Max\ i : 0 \leq i < m+1 \wedge \langle \Sigma j : 0 \leq j < i : A.j \rangle < A.i : A.i \rangle \wedge \underline{0 \leq m+1 \leq M \wedge M-(m+1) < T} \wedge F = \langle \Sigma j : 0 \leq j < m+1 : A.j \rangle)$

toda la **parte ya laburada** la adelanto a donde nos quedamos, y para mayor comodidad, trabajo E y F por separado.

Molina Franco
44192153

Continuo...

$E = n \max\ (\langle \Sigma j : 0 \leq j < m : A_j \rangle < A.m \Rightarrow A_m)\ [\,]\ (\langle \Sigma j : 0 \leq j < m : A_j \rangle \geq A.m \Rightarrow -\infty)$

• {por HI nueva, def de k}

$E = n \max\ (k < A_m \Rightarrow A.m)\ [\,]\ (k \geq A.m \Rightarrow -\infty)$

• {distributividad, y logica del max}

$E = (k < A.m \Rightarrow n \max A.m)\ [\,]\ (k \geq A_m \Rightarrow n)$

{resultado de E}

ahora trabajo F

$F = \langle \Sigma_j : 0 \leq j < m+1 : A.j \rangle$

• {logica $j = m \vee 0 \leq j < m$, partición de rango}

$F = \langle \Sigma_j : 0 \leq j < m : A.j \rangle + \langle \Sigma_j : j = m : A.j \rangle$

• {HI por def de k y rango unitario}

$F = k + A.m$

{resultado de F}

con estos resultados podemos plantear la asignacion dentro del ciclo.

teniendo en cuenta que necesitamos un condicional gracias a el resultado de E, aparte ya comprobamos la guarda y cota.

```
do (m ≠ M) →
    {I' ∧ b ∧ (M-m) = T}
    if (k < Am) →
        n,m,k = (n max A.m), m+1, (k+A.m)
    elif (k ≥ Am) →
        n,m,k = n, m+1, (k+A.m)
    fi
    {I' ∧ (M-m) < T}
od
```

Nota que las guardas del condicional son completamente opuestas lo que nos asegura una de las dos opciones.

Teniendo el cuerpo del ciclo lo único que nos queda por hacer es derivar la inicialización con la sig. terna, (m se init. en su mínimo)

$\{M \geq 0\}$
$n, m, k = R, \emptyset, S$
$\{I'\}$

$\emptyset \rightarrow$ cero

tomamos la primera condición como HI y laburamos en la wp de la asignación.

$WP.\ \{I'\}\ (n, m, k = R, \emptyset, S)$

- {asigno}

$R = \langle Max\ i : 0 \leq i < 0 \wedge \langle \Sigma j : 0 \leq j < i : A.j \rangle < A.i : A.i \rangle \wedge 0 \leq 0 \leq M \wedge$
$\wedge\ S = \langle \Sigma j : 0 \leq j < 0 : A.j \rangle$

- { logica $0 \leq i < 0$ y $0 \leq j < 0$ = False, absorvente de $\wedge$ y rango vacio}

$R = -\infty \wedge 0 \leq M \wedge S = 0$

- { por HI $0 \leq M$ y ya tenemos los resultados de $R_y S$}

Resultado

Const $M : Int,\ A : array\ [0, M)\ of\ Int$
Var $n : Int,\ k : Int,\ m : Int$

$\{M \geq 0\}$

$n, m, k = -\infty, 0, 0$

$\{I' : n = \langle Max\ i : 0 \leq i < m \wedge \langle \Sigma_j : 0 \leq j < i : A.j \rangle < A.i : A.i \rangle \wedge 0 \leq m \leq M \wedge k = \langle \Sigma_j : 0 \leq j < m : A.j \rangle\}$

```
do (m ≠ M) →
    if (k < A.m) →
        n, m, k = (n max A.m), m+1, (k + A.m)
    elif (k ≥ A.m) →
        n, m, k = n, m+1, (k + A.m)
    fi
od
```

$\{n = \langle Max\ i : 0 \leq i < M \wedge \langle \Sigma_j : 0 \leq j < i : A.j \rangle < A.i : A.i \rangle$

Molina Franco 44192153

4)

Const $M : Int$, $A : array\ [0, M]\ of\ Int$, $k : Int$;

Var $r : Bool$

$\{P : M \geq 2\}$

S_1;

$\{Q : r = \langle \exists i, j : 0 \leq i < j < M \wedge j = i+1 : \langle \exists x, y : k \cdot x = A.i \wedge k \cdot y = A.j : True \rangle \rangle \}$