(14) Potencias

1(a)

$hocho.xs = \langle \forall as, bs : xs = as ++ bs : \langle \sum i : 0 \leq i < \#bs : (bs!i)^2 \rangle \leq 88 \rangle$

Bueno, primero que todo el :

$(bs!i)^2$ no tiene sentido en nuestro Formalismo:

Modularizamos: $exp :: Int \to Int$

$exp2 . x = x . x$

La modularización vale ya que $x^2 = x . x$

$hocho.xs = \langle \forall as, bs : xs = as ++ bs : \langle \sum i ; 0 \leq i < \#bs : exp2.(bs!i) \rangle \leq 88 \rangle$

Caso Base $xs = []$

$\langle \forall as, bs : [] = as ++ bs : \langle \sum i : \ldots \rangle \leq 88 \rangle$

Como xs es vacío, por propiedad de listas

$bs = [] \wedge as = []$

Hacemo elim de variables y nos queda

$\langle \forall : [] = [] ++ [] : \langle \sum i : 0 \leq i < \#[] : exp2.([]!i) \rangle \leq 88 \rangle$

Def de $\#[]$ y rango vacío

$\langle \forall : true : 0 \leq 88 \rangle$

Termino constante

True

Caso inductivo (HI) $hocho.xs : \langle \forall as, bs : xs = as \mathbin{++} bs : \langle \Sigma i : 0 \le i < \#bs : exp2.(bs!i) \rangle \le 88 \rangle$

$hocho.(x:xs) = \langle \forall as, bs : x:xs = as \mathbin{++} bs : \langle \Sigma i : 0 \le i < \#bs : exp2.(bs!i) \rangle \le 88 \rangle$

☆ • P. Listas : $as = [\,] \vee as \ne [\,]$. distribuimos el $x:xs = as \mathbin{++} bs$ sobre este "$\vee$" y partimos el rango (Rango disjunto)

$\langle \forall as, bs : \underline{as = [\,]} \wedge x:xs = as \mathbin{++} bs : \langle \Sigma i : 0 \le i < \#bs : exp2.(bs!i) \rangle \le 88 \rangle$
$\wedge \langle \forall as, bs : as \ne [\,] \wedge x:xs = as \mathbin{++} bs : \langle \Sigma i : 0 \le i < \#bs : exp2.(bs!i) \rangle \le 88 \rangle$

□ • elim variable $as = [\,]$ y como $as \ne [\,]$ $as = a:as$

$\langle \forall bs : x:xs = bs : \langle \Sigma i : 0 \le i \ldots \rangle \le 88 \rangle \wedge$
$\langle \forall a, as, bs : x:xs = a:as \mathbin{++} bs : \langle \Sigma i \ldots \rangle \le 88 \rangle$

• P. Listas $x:xs = a:(as \mathbin{++} bs) \to x = a \wedge xs = as \mathbin{++} bs$
• Elim variable $x = a$
$\langle \forall bs : x:xs = bs : \langle \Sigma i \ldots \rangle \le 88 \rangle \wedge$
$\underline{\langle \forall as, bs : xs = as \mathbin{++} bs : \langle \Sigma i : 0 \le i < \#bs : exp2(bs!i) \rangle \le 88 \rangle}$

• HI
$\underline{\langle \forall bs : x:xs = bs : \langle \Sigma i : 0 \le i < \#bs : exp2.(bs!i) \rangle \le 88 \rangle} \wedge$
$hocho.xs$

• Usamos ☆ pero sobre bs y luego □ sobre bs tmb
• (def $\#[\,]$)
$\underline{\langle \forall bs : x:xs = [\,] : \langle \Sigma i : 0 \le i < 0 : exp2([\,]!i) \rangle \le 88 \rangle} \wedge$
$\langle \forall bs, b : x:xs = b:bs : \langle \Sigma i : 0 \le i < \#b:bs : exp(b:bs!i) \rangle \le 88 \rangle \wedge$
$hocho.xs$

Rango vacío $\rightarrow$ $\{True \wedge a \equiv a\}$
$\hookrightarrow$ si xs tiene un elemento x No puede ser vacío

$\langle \forall b, bs : \underline{x:xs = b:bs} : \langle \sum i : 0 \leq i < \# b:bs : exp2(b:bs!i) \rangle \leq 88 \rangle$

$\wedge$ hocho.xs

R. Unitario

$\langle \sum i : 0 \leq i < \underline{\# x:xs} : exp2(x:xs!i) \rangle \leq 88 \wedge hocho.xs$

(Def #)

$\langle \sum i : 0 \leq i < 1 + \# xs : exp2(x:xs!i) \rangle \leq 88 \wedge hocho.xs$

$\langle \sum i : 0 = i \vee 1 \leq i < 1 + \# xs : exp2(x:xs!i) \rangle \leq 88 \wedge hocho.xs$

P Rango y R unitario

$exp2.x + \langle \sum i : \underline{1 \leq i < 1 + \# xs} : exp2(x:xs!i) \rangle \leq 88 \wedge hocho.xs$

C. Variable $\rightarrow$ resto 1 en el rango (a cada termino o/1 rango)

$exp2.x + \langle \sum j : 0 \leq j < \# xs : exp2(x:xs!j+1) \rangle \leq 88 \wedge hocho.xs$

def $(x:xs!j+1) = xs!j$

$exp2.x + \langle \sum j : 0 \leq j < \# xs : exp2(xs!j) \rangle \leq 88 \wedge hocho.xs$

Debemos modularizar el cuantificador.

$sumExp2.xs = \langle \sum i : 0 \leq i < \#xs : exp2(xs!i) \rangle$

Caso base $sumExp2\,[\,]$

$sumExp2.[\,] = \langle \sum i : \underline{0 \leq i < 0} : exp2(xs!i) \rangle$

{R. vacío}

$sumExp2[\,] = 0$

Caso Inductivo

(HI) $: \langle \sum i : 0 \leq i < \#ys : exp2(ys!i) \rangle$

$sumExp2.(x:xs) = \langle \sum i : 0 \leq i < \#x:xs : exp2(x:xs!i) \rangle$

{Mismos pasos que recién}

$exp2.x + \underbrace{\langle \sum j : 0 \leq j < \#xs : exp(xs!j) \rangle}_{HI}$

$\boxed{exp2.x + sumExp2.xs}$

Reemplazemos entonces en nuestra definición

$\underline{exp2.x + \langle \sum j : 0 \leq j < \#xs : exp2(xs!j) \rangle} \leq 88 \wedge hocho.xs$

$sumExp2\,(x:xs) \leq 88 \wedge hocho.xs$

$hocho\,[\,] = True$

$hocho\,(x:xs) = sumExp2(x:xs) \leq 88 \wedge hocho.xs$

16 $f \cdot [-1, 8, -2, 5]$

$[-1, 8, -2, 5] = [\,] \mathbin{++} [-1, 8, -2, 5] \rightarrow 65 + 29 \leq 88 \rightarrow$ False

$= [-1] \mathbin{++} [8, -2, 5] \rightarrow 64 + 29 \leq 88 \rightarrow$ False

$= [-1, 8] \mathbin{++} [-2, 5] \rightarrow 29 \leq 88 \rightarrow$ True

$= [1, 8, -2] \mathbin{++} [5] \rightarrow 25 \leq 88 \rightarrow$ True

$= [-1, 8, -2, 5] \mathbin{++} [\,] \rightarrow 0 \leq 88 \rightarrow$ True

Resultado = False

7 c

$$hocho[-1,8,-2,5] = Exp2(-1) + SumExp2([8,-2,5]) \geq 88 \wedge hocho[8,-2,5]$$

Acá No deríamos de resolver las modularizaciones hasta que lleguemos al final de la recursión. Pero a fines de lectura del examen lo voy a ir simplificando

$$\equiv 1 + 93 \leq 88 \wedge Exp(8) + SumExp2([-2,5]) \leq 88 \wedge hocho([-2,5])$$

$$\equiv 94 \leq 88 \wedge 64 + 29 \leq 88 \wedge exp2(-2) + sumExp2([5])^{\leq 88} \wedge hocho([5])$$

$$= 94 \leq 88 \wedge 93 \leq 88 \wedge 29 \leq 88 \wedge exp2(5) + sumExp2[\,] \wedge hocho[\,]$$

$$= 94 \leq 88 \wedge 91 \leq 88 \wedge 29 \leq 88 \wedge 25 + 0 \leq 88 \wedge True$$

$$= False \wedge False \wedge True \wedge True \wedge True$$

$$= False$$

(22)

$A = [3, -1, -1, -1]$

Como A tiene 4 elementos usamos $N = 4$

$4 \geq 0 \rightarrow$ 4 cumple la Precondición

$\langle \exists i : 0 \leq i < 4 : \langle \Sigma j : 0 \leq j < i : A_j \rangle = -1 \rangle$

Caso $i = 0 \rightarrow$ Rango del existencial = True. → elim variables

$\langle \Sigma j : \underline{0 \leq j < 0} : A_j \rangle = -1$

{R. vacio}

$0 = -1$

False

Caso $i = 1 \rightarrow$ mismo que antes

$\langle \Sigma j : 0 \leq j < 1 : A_j \rangle = -1$

j puede ser 0 solamente

$\rightarrow A.0 = -1$

$3 = -1$

False

Caso $i = 2 \rightarrow$ mismo que en $i = 0$

$\langle \Sigma j : 0 \leq j < 2 : A_j \rangle = -1$

j puede ser 0 ó 1

$\rightarrow A.0 + A.1 = -1$

$\rightarrow 3 - 1 = -1$

$2 = -1$

False

Caso $i = 3 \rightarrow$ lo mismo que $i = 0$

$\langle \Sigma j : 0 \leq j < 3 : A_j \rangle = -1$

j puede ser 0, 1, 2

$A.0 + A.1 + A.2 = -1$

$3 - 1 - 1 = -1$

$1 = -1$

False

Caso $i = 4 \rightarrow$ mismo que $i = 0$

$\langle \Sigma j : 0 \leq j < 4 : A_j \rangle = -1$

$A.0 + A.1 + A.2 + A.3 = -1$

$3 - 1 - 1 - 1$

$0 = -1$

False

Respuesta = False $\vee$ False $\vee$ False $\vee$ False $\vee$ False = False

6 Este programa calcula si existe la suma de los elementos anteriores a cierto "i" (el cuales indice de A), la cual de como resultado -1

2c

Const N : Int ;

Var A : Array $[0, N)$ of Int ;

r : Bool ;

$P: \{N \geq 0\}$

S

$R: \{r = \langle \exists i : 0 \leq i \leq N : \langle \Sigma j : 0 \leq j < i : A.j \rangle = -1 \rangle\}$.

La estructura de nuestro pograma va a ser

Const N : Int ;

Var A : Array $[0, N)$ of Int;

r : Bool ;

$\{N \geq 0\}$

S_1

do $B \rightarrow$

S_2

od

$\{r : \langle \exists i : 0 \leq i \leq N : \langle \Sigma j : 0 \leq j < i : A.j \rangle = -1 \rangle\}$

Reemplazo cont N por var n

Tomamos elementos de una disyunción y deducimos guarda e Inv

Añado var n : Int

$\{N \geq 0\}$

S_1

do $B \rightarrow$

S_2

od

$\{r = \langle \exists i : 0 \leq i \leq n : \langle \Sigma j : 0 \leq j < i : A.j \rangle = -1 \rangle \wedge n = N\}$

Tomo $Inv : \langle \exists i : 0 \leq i \leq n : \langle \Sigma j : 0 \leq j < i : A.j \rangle = -1 \rangle$

$B : n \neq N$

$\{0 \leq N\}$

S_1

$\{Inv\}$

do $n \neq N$

$r, u, n := u = -1 \vee r \ , A_n + u \ , n+1;$

od

$\{r : \langle \exists i : 0 \leq i < n : \langle \Sigma \ j : 0 \leq j < i : A_j \rangle = -1 \rangle \wedge n = N\}$

Ahora, para encontrar S_1 buscamos forzar un R. vacío / unitario en la inv

$\{ Inv : u : \langle \Sigma j : 0 \leq j < n : A_j \rangle \wedge r = \langle \exists i : 0 \leq i < n : \langle \Sigma j : 0 \leq j < i : A_j \rangle = -1 \rangle$

Si $n = 0$ tenemos

$u = 0$ {rango vacío}

$r = 0 = -1 = False$ {rango unitario, rango vacío en el Termino, lógica}

Resultado

Const $N : Int;$

Var $A : Array [0, N)$ of Int; $r : Bool$; $u : Int$; $n : Int$;

$\{N \geq 0\}$

$n, u, r := 0, 0, False$

do $n \neq N \rightarrow$

$r, u, n := (u = -1 \vee r), (A_n + u), (n+1);$

od

$\{r = \langle \exists i : 0 \leq i < n : \langle \Sigma j : 0 \leq j < i : A_j \rangle = -1 \rangle \wedge n = N\}$

(3a)

Const N : Int ;

Var r : Bool ; A : Array $[0, N)$ of Int;

$\{P: 0 \leq N\}$

S

$\{R: r = \langle \forall i : 0 \leq i < N \wedge (A.i \geq 0) ; \langle \exists j : 0 \leq j \leq A_i : A_i = 2^j \rangle \rangle \}$

(3b) Const N : Int ;

Var r : Bool ; A : Array $[0, N)$ of Int ;

$\{P: 0 < N\}$

S

$\{r : \langle \exists i : 0 < i < N : \langle \Sigma j : i \leq j < N ; A_j \rangle > \langle Max\, v : 0 < v < N$
$: A_v \rangle \rangle \}$