① a) Derivar con recursividad

$$f.xs = \langle \forall as, e, bs : xs = as ++ [e] ++ bs : \langle \forall i : 0 \leq i < \#bs : divide.e.(bs!i) \rangle \rangle$$

defino func. auxiliar $t.e.bs = \langle \forall i : 0 \leq i < \#bs : divide.e.(bs!i) \rangle$ ⊛

c. base: $t.e.[\,] = \langle \forall i : False : divide.e.(bs!i) \rangle$ $\equiv$ {elem neutro}

$= True$

p. induc. asumo ⊛, $t.e.(b \triangleright bs) = \langle \forall i : 0 \leq i \leq \#bs : divide.e.((b \triangleright bs)!i) \rangle$

$\equiv$ {logica / aritmetica}

$= \langle \forall i : 0 = i \vee 0 < i \leq \#bs : divide.e.(b \triangleright bs!i) \rangle$

$\equiv$ {Partición de rango, rango unitario}

$= divide.e.b \wedge \langle \forall i : 0 \leq i < \#bs : divide.e.bs!i \rangle$

$\equiv$ {HI}

$= divide.e.b \wedge t.e.bs$

Ahora, volviendo a f.xs, ajustaré la especificación

$\langle \forall as, e, bs : xs = as ++ (e \triangleright bs) : t.e.bs \rangle$

$\equiv$ {aritm.}

$\langle \forall as, e, bs : xs = e \triangleright bs \vee as \neq [\,] : t.e.bs \rangle$

Ahora calculo, asumiendo que $f.(xs) = \langle \forall as, e, bs : xs = e \triangleright bs \vee as \neq [\,] : t.e.bs \rangle$

caso base: $f\,[\,] = \langle \forall as, e, bs : [\,] = e \triangleright bs \vee as \neq [\,] : t.e.bs \rangle$

$\equiv$ {Rango unitario}

True

caso inductivo $f\,(x \triangleright xs) = \langle \forall as, e, bs : (x \triangleright xs) = (e \triangleright bs) \vee as \neq [\,]\ t.e.bs$

Pedro Salas piñero

44273547

② Const N: Int;
Var A: array [0, M) of Int;
r: Nat;

$\{M \geq 0\}$

S

$\{r = \langle N i : 0 \leq i \leq M : \langle \Sigma j : 0 \leq j < i : A.j \rangle = \langle \Pi j : 0 \leq j < i : A.j \rangle \}$

a) calcular para A = [3, 1, 2, 0, 6].

$(i=0) \to 3 = 3 \to +1$

$(i=1) \to 3+1 = 3 \cdot 1 \to +0$

$(i=2) \to 3+1+2 = 3 \cdot 2$

$6 = 6$

$\equiv$ True $\to +1$

$(i=3) \to 3+1+2+0 = 6 \cdot 0$

$\equiv$ False $\to +0$

$(i=4) \to 6+6 = 0 \cdot 6$

$\equiv$ False $\to +0$

para A := [3, 1, 2, 0, 6], el estado final r vale $(r \mapsto 2)$

b) Este programa cuenta cuantas veces sucede que la sumatoria y la productoria de los primeros n elementos de un array tiene el mismo valor

c) Derivar (usando un ciclo).

voy a desarrollar una idea de cómo se debería ver el programa (intuición)

```
{P}
i, r, s, p := 0, 0, 0, 0
do  i ≤ M →
    s, p := s + A.i , p * A.i ;
    i := i+1 ;
    if s = p →
        r := r + 1
    fi ;
od
{Q}
```

como ahora tengo una idea de lo que necesito, prosigo con la derivación

Necesito un invariante $I: (r = \langle N i : 0 \leq i \leq l : (\text{termino igual a } Q) \rangle$ $\wedge\ 0 \leq l \leq M$
y propongo

y propongo una Guarda $B = l \leq M$

pedro Salas Piñero
44273597

luego, $\{M \geq 0\}$

S_0

$\{I: r = \langle N i : 0 \leq i \leq l : \langle \Sigma j : 0 \leq j < i : A.j \rangle = \langle \Pi j : 0 \leq j < i \rangle \rangle \wedge 0 \leq l \leq M\}$

do $B \rightarrow$

$\{I \wedge B\}$

S_1

$\{I\}$

od

$\{I \wedge \neg B\}$

antes, debo probar que $I \wedge \neg B \Rightarrow Q$

$$r = \langle N i : 0 \leq i \leq l : \langle \Sigma j : 0 \leq j \leq i : A.j \rangle = \langle \Pi j : 0 \leq j < i \rangle \rangle \wedge 0 \leq l \wedge \underbrace{l \leq M \wedge l > M}_{False} \Rightarrow Q$$

$\equiv$ {Absorción}

False $\Rightarrow Q$

$\equiv$ {Logica}

True

Este invariante no me funcionó

Pedro Salas Piñero

44273547

③ a) Dado arreglo A de $N \geq 0$ enteros, decidir si algún elemento es igual al triple de la suma de todos los otros elem del arreglo

Const N: Int; A: array [0, N) of Int

Var b: Bool

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

S

$\{Q: b = \langle \exists i : 0 \leq i \leq N : \langle \sum j : 0 \leq j \leq N \wedge i \neq j : A_j \rangle * 3 = A.i \rangle\}$

b) Dado un arreglo A de $N > 0$ enteros, decidir si el producto de los elem en posiciones pares es impar

Const N: Nat ; A: array [0, N) of Int

Var b: Bool

$\{P: N > 0\}$

S

$\{Q: b = (\langle \prod p : 0 \leq p \leq N \wedge p \% 2 = 0 : A_p \rangle \% 2 \neq 0)\}$

nota: si el tipo es Nat, por lo general no incluye 0, pero nunca está de más la Precondición

nota 2: no recuerdo el formalismo, pero uso la forma en c para módulo

Pedro Salas Piñero

44273547

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

Pedro Salas Piñero
DNI: 44273547