Ejercicios Teoría de la Programación

Mario Calvarro Marines

Índice general

El lenguaje while

Ejercicio 1

Enunciado

Siendo la sintaxis para n

$$n \ ::= \ \mathit{0} \ | \ \mathit{1} \ | \ \mathit{0} \ n \ | \ \mathit{1} \ n$$

¿Se puede definir $\mathcal{N}$ correctamente?

Solución:

La semántica $\mathcal{N}$ para esta sintaxis se podría obtener trivialmente utilizando la misma que para el caso en que los constructores compuestos tuviesen el orden inverso. De esta forma, lo único que cambiaría sería que "la lectura"del número se haría de izquierda a derecha. Si intuitivamente pensamos en n como un número binario, diríamos que los bits menos significativos son los de la izquierda en contra de lo usual (que son los de la derecha).

Sin embargo, si queremos mantener el convenio de lectura de estos numerales, debemos hacer uso de una función auxiliar. Esta se deberá definir composicionalmente y simplemente es la longitud:

$$\begin{gathered}\mathrm{long} : \mathrm{Num} \to \mathbb{Z} \\ \mathrm{long}[[0]] = \mathrm{long}[[1]] = 1 \\ \mathrm{long}[[0\ n]] = \mathrm{long}[[1\ n]] = 1 + \mathrm{long}[[n]]\end{gathered}$$

En definitiva, la función semántica será:

$$\begin{gathered}\mathcal{N} : \mathrm{Num} \to \mathbb{Z} \\ \mathcal{N}[[0]] = 0 \\ \mathcal{N}[[0]] = 1 \\ \mathcal{N}[[0\ n]] = \mathcal{N}[[n]] \\ \mathcal{N}[[1\ n]] = 2^{\mathrm{long}[[n]]} \cdot \mathcal{N}[[n]]\end{gathered}$$

Ejercicio 2

Enunciado

Explicitar la equivalencia sintáctica entre las dos formas vistas de definir sintácticamente los numerales (en un orden y el otro) mediante dos biyecciones, $M : \mathrm{Num}' \to \mathrm{Num}$ y $M' : \mathrm{Num} \to \mathrm{Num}'$, tales que $M' \circ M = \mathrm{Id}_{\mathrm{Num}'}$ y $M \circ M' = \mathrm{Id}_{\mathrm{Num}}$.

Solución:

En primer lugar, de manera natural podemos definir las funciones M y M' tal que:

$$\begin{aligned} M : \text{Num}' &\rightarrow \text{Num} \\ M(0') &= 0 \\ M(1') &= 1 \\ M(0'\ n') &= M(n')\ \ 0 \\ M(1'\ n') &= M(n')\ \ 1 \end{aligned} \qquad \begin{aligned} M' : \text{Num} &\rightarrow \text{Num}' \\ M'(0) &= 0' \\ M'(1) &= 1' \\ M'(n\ 0) &= 0'\ M'(n) \\ M'(n\ 1) &= 1'\ M'(n) \end{aligned}$$

Veamos ahora por inducción que $M \circ M' = \text{Id}_{\text{Num}}$:

1. Casos base:
   - $n = 0$: $M'(0) = 0'$ y $M(0') = 0 \Rightarrow M \circ M'(0) = 0$.
   - $n = 1$: $M'(1) = 1'$ y $M(1') = 1 \Rightarrow M \circ M'(1) = 1$.

   Por tanto, se cumple para los casos base.
2. Caso inductivo: Supongamos por hipótesis de inducción que esta composición es la identidad al aplicarse sobre los constituyentes de los constructores compuestos de Num, que denominaremos n. Es decir, $M \circ M'(n) = n$. Veamos ahora que ocurre ahora en los operadores:
   - $n := n\ 0$: Como $M'(n\ 0) = 0'\ \ M'(n)$, aplicando ahora M tenemos que $M(0'\ \ M'(n)) = M(M'(n))\ \ 0 = n\ 0$ (esta última igualdad por hipótesis de inducción).
   - $n := n\ 1$: Similar al anterior punto.

Para ver la composición inversa podemos utilizar el mismo razonamiento y con esto tenemos el resultado que buscábamos.

Ejercicio 3

Enunciado

Definir el conjunto de variables libres para una expresión booleana y demostrar un resultado similar al Lema 1.

Solución:

Utilizaremos una definición inductiva sobre las expresiones:

$$\begin{aligned} \text{FV}(\mathsf{true}) = \text{FV}(\mathsf{false}) &= \emptyset \\ \text{FV}(a_1 = a_2) &= \text{FV}(a_1) \cup \text{FV}(a_2) \\ \text{FV}(a_1 \leq a_2) &= \text{FV}(a_1) \cup \text{FV}(a_2) \\ \text{FV}(\neg b) &= \text{FV}(b) \\ \text{FV}(b_1 \wedge b_2) &= \text{FV}(b_1) \cup \text{FV}(b_2) \end{aligned}$$

El resultado similar al Lema 1 será entonces:

Lema

Sean $s, s' \in \text{State}$ *tales que* $\forall x \in \text{FV}(b).s\ x = s'\ x$, *entonces* $\mathcal{B}[[b]]\,s = \mathcal{B}[[b]]s'$.

Demostración:

Realizaremos una demostración por inducción estructural.

- Casos base: ($b :=$ true ó false)

  En este caso, $\mathcal{B}[[b]]\text{s} = \text{tt}$ ó ff, respectivamente, $\forall s \in$ State.

- Casos inductivos: Supongamos, por hipótesis de inducción, que se cumple el lema para los componentes de los constructores compuestos (que serán a_1, a_2, b, b_1, b_2), entonces:

  1. $\boxed{b := a_1 = a_2}$. Sabemos que en este caso,

  $$\mathcal{B}[[b]]s = \begin{cases} \text{tt}, & \text{si } \mathcal{A}[[a_1]]s = \mathcal{A}[[a_2]]s \\ \text{ff}, & \text{c.c.} \end{cases}$$

  Por tanto, como $\text{FV}(b) = \text{FV}(a_1) \cup \text{FV}(a_2)$ y por la hipótesis del lema, tenemos que a_1 y a_2 cumplen la hipótesis del lema 1 y, es decir, $\mathcal{A}[[a_i]]s = \mathcal{A}[[a_i]]s',\ \forall i \in \{1, 2\}$. Con esto último, podemos simplemente sustituir en la definición de la semántica s por s' y vemos que el resultado es directo.

  2. $\boxed{b := a_1 \leq a_2}$. Razonamiento similar al anterior apartado.

  3. $\boxed{b := \neg b_1}$. Recordando la definición de la semántica de este operador tenemos que:

  $$\mathcal{B}[[\neg b_1]]s = \begin{cases} \text{tt}, & \text{si } \mathcal{B}[[b_1]]s = \text{ff} \\ \text{ff}, & \text{si } \mathcal{B}[[b_1]]s = \text{tt} \end{cases}$$

  Pero como podemos aplicar la hipótesis de inducción sobre b_1, tenemos que $\mathcal{B}[[b_1]]s = \mathcal{B}[[b_1]]s'$ y, por tanto, podemos simplemente sustituir en la definición de la semántica s por s' y tendremos el resultado buscado.

  4. $\boxed{b := b_1 \wedge b_2}$. La definición de la semántica para esta expresión, recordemos, es:

  $$\mathcal{B}[[b_1 \wedge b_2]]s = \begin{cases} \text{tt}, & \text{si } \mathcal{B}[[b_1]]s = \text{tt y } \mathcal{B}[[b_2]]s = \text{tt} \\ \text{ff}, & \text{c.c} \end{cases}$$

  De nuevo, simplemente podemos utilizar la hipótesis de inducción sobre b_1 y b_2 y, sustituyendo s por s', tendremos el resultado que buscamos.