El lenguaje WHILE. Ejercicios

Ejercicio 1.4

Enunciado

Siendo la sintaxis para n

```
n ::= 0 | 1 | 0 n | 1 n
```

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

Resolución

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 a diferencia 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:

$$\begin{gathered}\text{long} : \text{Num} \to \mathbb{Z} \\ \text{long}\,(0) = \text{long}\,(1) = 1 \\ \text{long}\,(0\ n) = \text{long}\,(1\ n) = 1 + \text{long}\,(n)\end{gathered}$$

En definitiva, la función semántica será:

$$\begin{gathered}\mathcal{N} : \text{Num} \to \mathbb{Z} \\ \mathcal{N}[\![0]\!] = 0 \\ \mathcal{N}[\![0]\!] = 1 \\ \mathcal{N}[\![0\ n]\!] = \mathcal{N}[\![n]\!] \\ \mathcal{N}[\![1\ n]\!] = 2^{\text{long}(n)} + \mathcal{N}[\![n]\!]\end{gathered}$$

Ejercicio 1.10

Enunciado

Probar que la definición dada para la función semántica de expresiones booleanas es total.

$$\mathcal{B} : \mathbf{Bexp} \to (\mathbf{State} \to \mathbf{T})$$

Resolución

Para que esta función semántica sea total, simplemente necesitamos que para toda expresión booleana, b, y todo estado posible, s, existe un solo $t \in \mathbf{T}$ tal que $\mathcal{B}[\![b]\!]s = t$. Lo probaremos por inducción estructural.

- Casos base:
  - Si $b \equiv \text{true}$, entonces $\forall s \in \mathbf{State}$ tenemos que $\mathcal{B}[\![b]\!]s = \text{tt}$.
  - Si $b \equiv \text{false}$, entonces $\forall s \in \mathbf{State}$ tenemos que $\mathcal{B}[\![b]\!]s = \text{ff}$.
  - Si $b \equiv a_1 = a_2$, entonces tenemos solo dos casos posibles:
    - Si $\mathcal{A}[\![a_1]\!]s = \mathcal{A}[\![a_2]\!]s \Rightarrow \mathcal{B}[\![b]\!]s = \text{tt}$.
    - Si $\mathcal{A}[\![a_1]\!]s \neq \mathcal{A}[\![a_2]\!]s \Rightarrow \mathcal{B}[\![b]\!]s = \text{ff}$.
  - Si $b \equiv a_1 \leq a_2$, entonces tenemos solo dos casos posibles:
    - Si $\mathcal{A}[\![a_1]\!]s \leq \mathcal{A}[\![a_2]\!]s \Rightarrow \mathcal{B}[\![b]\!]s = \text{tt}$.
    - Si $\mathcal{A}[\![a_1]\!]s > \mathcal{A}[\![a_2]\!]s \Rightarrow \mathcal{B}[\![b]\!]s = \text{ff}$.

  Con lo que los casos base se cumplen.

- Casos inductivos:
  - Si $b \equiv \neg b_0$ y suponemos que se cumple que $\mathcal{B}[\![b_0]\!]s,\ \forall s \in \mathbf{State}$ tiene un solo valor, entonces solo tenemos dos posibilidades:
    - Si $\mathcal{B}[\![b_0]\!]s = \text{tt} \Rightarrow \mathcal{B}[\![b]\!]s = \text{ff}$.
    - Si $\mathcal{B}[\![b_0]\!]s = \text{ff} \Rightarrow \mathcal{B}[\![b]\!]s = \text{tt}$.
  - Si $b \equiv b_0 \wedge b_1$ y suponemos que se cumple que $\mathcal{B}[\![b_i]\!]s,\ \forall s \in \mathbf{State}, i \in \{0, 1\}$ tiene un solo valor, entonces solo tenemos dos posibilidades:
    - Si $\mathcal{B}[\![b_0]\!]s = \text{tt y } \mathcal{B}[\![b_1]\!]s = \text{tt} \Rightarrow \mathcal{B}[\![b]\!]s = \text{tt}$.
    - Si $\mathcal{B}[\![b_0]\!]s = \text{ff ó } \mathcal{B}[\![b_1]\!]s = \text{ff} \Rightarrow \mathcal{B}[\![b]\!]s = \text{ff}$.

  Con lo que también se cumple para los casos recursivos (ya que los operadores son funciones completas).

Ejercicio 1.11

Enunciado

Dar una función semántica $\mathcal{B}$ para la siguiente categoría sintáctica $\mathbf{Bexp}'$:

```
b ::= true | false | a1 = a2 | a1 != a2 | a1 <= a2 | a1 >= a2
    | a1 < a2 | a1 > a2 | ¬b | b1 and b2 | b1 or b2
    | b1 implies b2 | b1 if and only if b2
```

Demostrar que para todo $b' \in \mathbf{Bexp}'$, existe un $b \in \mathbf{Bexp}$ tal que son equivalentes.

Resolución

En primer lugar, extendemos de manera natural y composicionalmente la función semántica de $\mathbf{Bexp}$ para que cubra todo $\mathbf{Bexp}'$. Los términos comunes no los repetiremos:

$$\mathcal{B} : \mathbf{Bexp}' \to (\mathbf{State} \to \mathbf{T})$$

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

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

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

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

$$\mathcal{B}[\![b_1 \vee b_2]\!]s = \begin{cases} \mathbf{tt}, & \text{si } \mathcal{A}[\![b_1]\!]s = \mathbf{tt} \text{ ó } \mathcal{A}[\![b_2]\!]s = \mathbf{tt} \\ \mathbf{ff}, & \text{c.c.} \end{cases}$$

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

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

Para demostrar ahora la equivalencia tenemos que buscar expresiones de $\mathbf{Bexp}$

equivalentes a las de $\mathbf{Bexp}'$ por inducción estructural. Para las que están en común será directo. Empezamos por los casos base:

- Si $b' \equiv a_1 \neq a_2$ tomamos $b \equiv \neg\left(a_1 = a_2\right)$.
- Si $b' \equiv a_1 \geq a_2$ tomamos $b \equiv \neg\left[\left(a_1 \leq a_2\right) \wedge \neg\left(a_1 = a_2\right)\right]$.
- Si $b' \equiv a_1 < a_2$ tomamos $b \equiv \left(a_1 \leq a_2\right) \wedge \neg\left(a_1 = a_2\right)$.
- Si $b' \equiv a_1 > a_2$ tomamos $b \equiv \neg\left(a_1 \leq a_2\right)$.

Veamos ahora los casos inductivos. Asumimos que se cumple para los constituyentes b_1 y b_2:

- Si $b' \equiv b_1 \vee b_2$ tomamos $b \equiv \neg\left(\neg b_1 \wedge \neg b_2\right)$ (De Morgan).
- Si $b' \equiv b_1 \Rightarrow b_2$ tomamos $b \equiv \neg\left(b_1 \wedge \neg b_2\right)$.
- Si $b' \equiv b_1 \Leftrightarrow b_2$ tomamos $b \equiv \neg\left(b_1 \wedge \neg b_2\right) \wedge \neg\left(\neg b_1 \wedge b_2\right)$.

Para probarlo de forma completamente rigurosa sería necesario calcular de forma exacta la función semántica de estos términos y ver que son iguales en ambos casos, pero en este caso se puede apreciar simplemente utilizando las reglas lógicas de sobra conocidas o realizando tablas de verdad.

Ejercicio 1.13

Enunciado

Sean dos estados s y s' tales que $\forall x \in \mathbf{FV}\left(b\right)$ se cumple que $s\ x = s'\ x$. Demostrar que $\mathcal{B}[\![b]\!]s = \mathcal{B}[\![b]\!]s'$.

Resolución

Lo realizaremos por inducción estructural. Veamos primero los casos base:

- Si $b \equiv \mathrm{true}$ ó false, entonces se cumple trivialmente.
- Si $b \equiv a_1 = a_2$ ó $a_1 \leq a_2$, entonces, si escribimos $op \in \{=, \leq\}$, tenemos que:

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

  pero como $\mathcal{A}[\![a]\!]s = \mathcal{A}[\![a]\!]s'$ (por el lema 1.12),

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

y, por tanto, se cumple el resultado.

Veamos ahora los casos composicionales. Supongamos que se cumple para los constituyentes inmediatos b_1 y b_2:

- Si $b \equiv \neg b_1$. Como $\mathrm{FV}\,(b) = \mathrm{FV}\,(\neg b)$ podemos aplicar la hipótesis de inducción y $\mathcal{B}[\![b_1]\!]s = \mathcal{B}[\![b_1]\!]s'$. Ahora, con esto, tenemos que:

$$\mathcal{B}[\![b]\!]s = \begin{cases} \mathbf{tt}, & \text{si } \mathcal{B}[\![b_1]\!]s = \mathbf{tt} \\ \mathbf{ff}, & \text{c.c} \end{cases} = \begin{cases} \mathbf{tt}, & \text{si } \mathcal{B}[\![b_1]\!]s' = \mathbf{tt} \\ \mathbf{ff}, & \text{c.c} \end{cases} = \mathcal{B}[\![b]\!]s'$$

  con lo que probamos el resultado.

- Si $b \equiv b_1 \wedge b_2$. Como $\mathrm{FV}\,(b) = \mathrm{FV}\,(b_1) \cup \mathrm{FV}\,(b_2)$ tenemos que $\mathrm{FV}\,(b_i) \subset \mathrm{FV}$ por lo que podemos aplicar la hipótesis de inducción, es decir, $\mathcal{B}[\![b_i]\!]s = \mathcal{B}[\![b_i]\!]s',\ i \in \{1, 2\}$. Con esto,

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

  con lo que probamos el resultado.

Ejercicio 1.14

Enunciado (Teorema fundamental de sustituciones)

Demostrar que, para todo estado s, se cumple:

$$\mathcal{A}[\![a\,[y \mapsto a_0]]\!]s = \mathcal{A}[\![a]\!]\left(s\,[y \mapsto \mathcal{A}[\![a_0]\!]s]\right).$$

Resolución

Lo demostraremos por inducción estructural sobre a. Casos base:

- $a \equiv n \Rightarrow [\![n\,[y \mapsto a_0]]\!]s = n,\ \forall s$. Por otro lado, $[\![n]\!]\underbrace{(s\,[y \mapsto [\![a_0]\!]s])}_{s'} = n$, por lo que tenemos el resultado.

- $a \equiv x \in \mathrm{Var} \Rightarrow [\![x\,[y \mapsto a_0]]\!]s = \begin{cases} [\![a_0]\!]s, & \text{si } x = y \\ s\ x, & \text{si } x \neq y \end{cases}$. Por otro lado, $[\![x]\!]\underbrace{(s\,[y \mapsto [\![a_0]\!]s])}_{s'}$ donde:

$$s'\ x = \begin{cases} s\ x, & \text{si } x \neq y \\ [\![a_0]\!]s, & \text{si } x = y \end{cases}.$$

  Por lo que tenemos el resultado.

Veamos que se da ahora para los casos composicionales. Supongamos que se da para los constituyentes inmediatos a_1 y a_2 y que op se refiere a $+, -$ ó $*$.

- $a \equiv a_1\ op\ a_2 \Rightarrow [\![(a_1\ op\ a_2)\,[y \mapsto a_0]]\!]s =$ $[\![a_1\,[y \mapsto a_0]]\!]s\ op\ [\![a_2\,[y \mapsto a_0]]\!]s$. Por otro lado,

$$[\![a_1\ op\ a_2]\!]\underbrace{(s\,[y \mapsto [\![a_0]\!]s])}_{s'} = [\![a_1]\!]s'\ op\ [\![a_2]\!]s'.$$

  Pero por hipótesis de inducción tenemos que $[\![a_i\,[y \mapsto a_0]]\!]s = [\![a_i]\!]s',\ i \in \{1, 2\}$, por lo que el resultado buscado se cumple.