Cálculo lambda II

Extensiones del cálculo lambda

Paradigmas de Lenguajes de Programación

8 de febrero de 2018

Primera extensión

Extensión con **pares**

- $M, N ::= \ldots \mid < M, N > \mid \pi_1(M) \mid \pi_2(M)$
- $\sigma, \tau ::= \ldots \mid \sigma \times \tau$
- Enunciar las nuevas reglas de tipado.
- Extender el conjunto de valores y determinar las nuevas reglas de semántica.

Primera extensión

Extensión con **pares**

- $M, N ::= \ldots \mid < M, N > \mid \pi_1(M) \mid \pi_2(M)$
- $\sigma, \tau ::= \ldots \mid \sigma \times \tau$
- Enunciar las nuevas reglas de tipado.
- Extender el conjunto de valores y determinar las nuevas reglas de semántica.
- ¿Qué problema introduce agregar la siguiente regla? ¿Y reemplazar otras por esta?
$$\pi_1(< M, N >) \rightsquigarrow M$$

Primera extensión

Extensión con **pares**

- $M, N ::= \ldots \mid < M, N > \mid \pi_1(M) \mid \pi_2(M)$
- $\sigma, \tau ::= \ldots \mid \sigma \times \tau$
- Enunciar las nuevas reglas de tipado.
- Extender el conjunto de valores y determinar las nuevas reglas de semántica.
- ¿Qué problema introduce agregar la siguiente regla? ¿Y reemplazar otras por esta?

$$\pi_1(< M, N >) \rightsquigarrow M$$

Verificar el siguiente juicio de tipado

- $\emptyset \rhd \pi_1((\lambda x : Nat. < x, \mathsf{True} >)\ 0) : Nat$

Primera extensión

Extensión con **pares**

- $M, N ::= \ldots \mid < M, N > \mid \pi_1(M) \mid \pi_2(M)$
- $\sigma, \tau ::= \ldots \mid \sigma \times \tau$
- Enunciar las nuevas reglas de tipado.
- Extender el conjunto de valores y determinar las nuevas reglas de semántica.
- ¿Qué problema introduce agregar la siguiente regla? ¿Y reemplazar otras por esta?

$$\pi_1(< M, N >) \rightsquigarrow M$$

Verificar el siguiente juicio de tipado

- $\emptyset \triangleright \pi_1((\lambda x : Nat. < x, \mathsf{True} >)\ 0) : Nat$

Reducir el término a un valor

- $\pi_1((\lambda x : Nat. < x, \mathsf{True} >)\ 0)$

Segunda extensión

Extensión con **árboles binarios**

- $M, N, O ::= \ldots \mid \mathsf{Nil}_\sigma \mid \mathsf{Bin}(M, N, O) \mid \mathsf{raíz}(M) \mid \mathsf{der}(M) \mid \mathsf{izq}(M) \mid \mathsf{esNil}(M)$
- $\sigma ::= \ldots \mid \mathsf{AB}_\sigma$
- Definir como ejemplo un árbol de funciones de Bool $\to$ Bool.
- Reglas de tipado.
- Reglas de semántica y nuevos valores.

Otra forma de proyectar/observar

Otra forma de representar proyectores u observadores más prolija y que requiere menos reglas (aunque una construcción más sofisticada).

Árboles bis

- $M, N, O ::= \ldots \mid Nil_\sigma \mid \mathrm{Bin}(M, N, O) \mid \mathrm{case}_{AB_\sigma}\ M\ of\ Nil \rightsquigarrow N\ ; \mathrm{Bin}(i, r, d) \rightsquigarrow O$
  Importante: las minúsculas i, r y d representan los <u>nombres</u> de las variables que pueden aparecer libres en O).
- Marcar en la expresión del case: subtérminos y <u>anotaciones</u> de tipos.
- Modificar las reglas de tipado para soportar la nueva extensión.
- Agregar las reglas de semántica necesarias.

Otra forma de proyectar/observar

Otra forma de representar proyectores u observadores más prolija y que requiere menos reglas (aunque una construcción más sofisticada).

Árboles bis

- $M, N, O ::= \ldots \mid Nil_\sigma \mid \mathrm{Bin}(M, N, O) \mid \mathrm{case}_{AB_\sigma}\ M\ of\ Nil \rightsquigarrow N\ ; \mathrm{Bin}(i, r, d) \rightsquigarrow O$ **Importante:** las minúsculas i, r y d representan los nombres de las variables que pueden aparecer libres en O).
- Marcar en la expresión del case: subtérminos y anotaciones de tipos.
- Modificar las reglas de tipado para soportar la nueva extensión.
- Agregar las reglas de semántica necesarias.

Agregado

A partir de la extensión anterior, definir una nueva extensión que incorpore expresiones de la forma map(M, N), donde N es un árbol y M una función que se aplicará a cada uno de los elementos de N.

Algo para registrar

Recordemos la extensión con registros vista en la teórica

$$\begin{array}{rcl}\sigma & ::= & \ldots \mid \{l_1 : \sigma_1, \ldots, l_n : \sigma_n\} \\ M & ::= & \ldots \mid \{l_1 = M_1, \ldots, l_n = M_n\} \mid M.l\end{array}$$

Tipado:

$$\frac{\Gamma \vdash M_1 : \sigma_1 \quad \ldots \quad \Gamma \vdash M_n : \sigma_n}{\Gamma \vdash \{l_1 = M_1, \ldots, l_n = M_n\} : \{l_1 : \sigma_1, \ldots, l_n : \sigma_n\}} \text{(T-RCD)}$$

Semántica:

$$\frac{M_i \to M_i'}{\{l_1 = V_1, \ldots, l_{i-1} = v_{i-1}, l_i = M_i, l_{i+1} = M_{i+1}, \ldots\} \to \{l_1 = V_1, \ldots, l_{i-1} = v_{i-1}, l_i = M_i', l_{i+1} = M_{i+1}, \ldots\}} \text{(E-RCD)}$$

$$\frac{M \to M'}{M.l \to M'.l} \text{(E-PROJ)} \qquad \frac{1 \leq i \leq n}{\{l_1 = V_1, \ldots, l_n = V_n\}.l_i \to V_i} \text{(E-RCD-PROJ)}$$

Algo para registrar

Recordemos la extensión con registros vista en la teórica

$$\begin{array}{rcl} \sigma & ::= & \ldots \mid \{l_1 : \sigma_1, \ldots, l_n : \sigma_n\} \\ M & ::= & \ldots \mid \{l_1 = M_1, \ldots, l_n = M_n\} \mid M.l \end{array}$$

Tipado:

$$\frac{\Gamma \vdash M_1 : \sigma_1 \quad \ldots \quad \Gamma \vdash M_n : \sigma_n}{\Gamma \vdash \{l_1 = M_1, \ldots, l_n = M_n\} : \{l_1 : \sigma_1, \ldots, l_n : \sigma_n\}} \text{(T-RCD)}$$

Semántica:

$$\frac{M_i \to M_i'}{\{l_1 = V_1, \ldots, l_{i-1} = v_{i-1}, l_i = M_i, l_{i+1} = M_{i+1}, \ldots\} \to \{l_1 = V_1, \ldots, l_{i-1} = v_{i-1}, l_i = M_i', l_{i+1} = M_{i+1}, \ldots\}} \text{(E-RCD)}$$

$$\frac{M \to M'}{M.l \to M'.l} \text{(E-PROJ)} \qquad \frac{1 \leq i \leq n}{\{l_1 = V_1, \ldots, l_n = V_n\}.l_i \to V_i} \text{(E-RCD-PROJ)}$$

Definir una extensión que permita "unir" un registro $\{x_1 = M_1, \ldots, x_m = M_m\}$ con otro $\{y_1 = N_1, \ldots, y_n = N_n\}$, de manera tal que el registro resultante contenga todas las etiquetas de ambos, con los mismos valores y en el mismo orden.
Restricción: los registros a unir **no deben** tener etiquetas en común.

¿? ¿? ¿? ¿? ¿? ¿? ¿? ¿? ¿? ¿? ¿? ¿? ¿?