Cálculo lambda II

Extensiones del cálculo lambda

Paradigmas de Lenguajes de Programación

Departamento de Computación - FCEyN

12 de mayo de 2020

Prefacio

Hasta ahora vimos:

- Introducción al cálculo lambda, a través de la identificación de términos y convenciones. ✓
- Verificación de **juicios de tipado** usando los axiomas y reglas del **sistema de tipos**. ✓
- Identificación de términos en forma normal y valores usando los axiomas y reglas de la **semántica operacional**. ✓
- Extensión del cálculo lambda para los números naturales, manteniendo la propiedad del determinismo. ✓

Hoy el plan es:

- Definir nuevas extensiones al cálculo lambda con sus respectivos términos.
- Extender el conjunto de reglas de tipado verificando así nuevos ejemplos de juicios de tipado.
- Extender el conjunto de reglas de semántica para reducir términos a forma normal.
- Atender a la propiedad del determinismo.

Primera extensión - Tipado

Comencemos entonces definiendo una extensión sencilla:

Extensión de 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.

$$\frac{\Gamma \triangleright M : \sigma \quad \Gamma \triangleright N : \tau}{\Gamma \triangleright < M, N > : \sigma \times \tau} \text{(T-PAR)}$$

$$\frac{\Gamma \triangleright M : \sigma \times \tau}{\Gamma \triangleright \pi_1(M) : \sigma} \text{(T-}\pi_1\text{)} \qquad \frac{\Gamma \triangleright M : \sigma \times \tau}{\Gamma \triangleright \pi_2(M) : \tau} \text{(T-}\pi_2\text{)}$$

Primera extensión - Semántica

Extensión de pares

- $M, N ::= \ldots \mid\ < M, N > \ \mid \pi_1(M) \mid \pi_2(M)$
- $\sigma, \tau ::= \ldots \mid \sigma \times \tau$

☆ Extender el conjunto de valores y determinar las nuevas reglas de semántica.

$$V ::= \ldots \mid\ < V, V >$$

$$\frac{M \ \rightarrow \ M'}{< M, N > \rightarrow < M', N >} \ (\text{E-PAR1})$$

$$\frac{N \ \rightarrow \ N'}{< V, N > \rightarrow < V, N' >} \ (\text{E-PAR2})$$

Primera extensión - Semántica

Extensión de pares

- $M, N ::= \ldots \mid\ < M, N > \ \mid \pi_1(M) \mid \pi_2(M)$
- $\sigma, \tau ::= \ldots \mid \sigma \times \tau$

☆ Extender el conjunto de valores y determinar las nuevas reglas de semántica.

$$\frac{M \ \to \ M'}{\pi_1(M) \ \to \ \pi_1(M')}\ (\text{E-}\pi_1) \qquad \frac{}{\pi_1(< V, W >) \ \to \ V}\ (\text{E-}\pi_1\text{Par})$$

$$\frac{M \ \to \ M'}{\pi_2(M) \ \to \ \pi_2(M')}\ (\text{E-}\pi_2) \qquad \frac{}{\pi_2(< V, W >) \ \to \ W}\ (\text{E-}\pi_2\text{Par})$$

Primera extensión - Nuevas reglas

Extensión de pares

- $M, N ::= \ldots \mid\ < M, N >\ \mid \pi_1(M) \mid \pi_2(M)$
- $\sigma, \tau ::= \ldots \mid \sigma \times \tau$

- ¿Qué problema introduce agregar la siguiente regla?

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

¿Qué ocurre con el siguiente término al agregarla?

$$\pi_1(< (\lambda x : \text{Bool}.x)\ \text{True}, \text{False} >)$$

¿Si reemplazamos otra regla por ésta solucionamos el problema?

Primera extensión - Juicios de tipado

Extensión de **pares**

- $M, N ::= \ldots \mid < M, N > \mid \pi_1(M) \mid \pi_2(M)$
- $\sigma, \tau ::= \ldots \mid \sigma \times \tau$

☆ Verificar si el siguiente juicio de tipado es derivable:

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

$$\dfrac{\dfrac{\dfrac{\dfrac{\checkmark\ x : \mathsf{Nat} \in \{x : \mathsf{Nat}\}}{\{x : \mathsf{Nat}\} \triangleright x : \mathsf{Nat}}\,(\text{T-Var}) \qquad \dfrac{\tau = \mathsf{Bool}}{\{x : \mathsf{Nat}\} \triangleright \mathsf{True} : \tau}\,(\text{T-True})}{\{x : \mathsf{Nat}\} \triangleright < x, \mathsf{True} >: \mathsf{Nat} \times \tau}\,(\text{T-Par})}{\emptyset \triangleright (\lambda x : \mathsf{Nat}. < x, \mathsf{True} >) : \mathsf{Nat} \to \mathsf{Nat} \times \tau}\,(\text{T-Abs}) \qquad \dfrac{\checkmark}{\emptyset \triangleright 0 : \mathsf{Nat}}\,(\text{T-Zero})}{\emptyset \triangleright (\lambda x : \mathsf{Nat}. < x, \mathsf{True} >)\ 0 : \mathsf{Nat} \times \tau}\,(\text{T-App})}{\emptyset \triangleright \pi_1((\lambda x : \mathsf{Nat}. < x, \mathsf{True} >)\ 0) : \mathsf{Nat}}\,(\text{T-}\pi_1)$$

Primera extensión - Reducción

Extensión de **pares**

- $M, N ::= \ldots \mid\ < M, N >\ \mid \pi_1(M) \mid \pi_2(M)$
- $\sigma, \tau ::= \ldots \mid \sigma \times \tau$

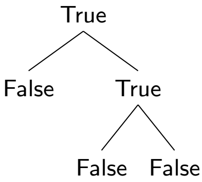
☆ Reducir el siguiente término a un valor:
$\pi_1((\lambda x : \text{Nat}. < x, \text{True} >)\ 0)$ $\pi_1((\lambda x : \text{Nat}. < x, \text{True} >)\ 0) \rightarrow_{E-\pi_1,\beta}$
$\pi_1(< 0, \text{True} >) \rightarrow_{E-\pi_1 Par}$
0

Segunda extensión - Ejemplos

Extensión de **árboles binarios**

- $M, N, O ::= \dots \mid \mathsf{Nil}_\sigma \mid \mathsf{Bin}(M, N, O) \mid \mathsf{raíz}(M) \mid \mathsf{der}(M) \mid \mathsf{izq}(M) \mid \mathsf{esNil}(M)$
- $\sigma ::= \dots \mid \mathsf{AB}_\sigma$

- Definir como ejemplo un árbol de términos de tipo Bool.
  Por ejemplo, para el siguiente árbol:



Tenemos el término:

$\mathsf{Bin}(\mathsf{Bin}(\mathsf{Nil}_{\mathsf{Bool}}, \mathsf{False}, \mathsf{Nil}_{\mathsf{Bool}}), \mathsf{True}, \mathsf{Bin}(\mathsf{Bin}(\mathsf{Nil}_{\mathsf{Bool}}, \mathsf{False}, \mathsf{Nil}_{\mathsf{Bool}}), \mathsf{True}, \mathsf{Bin}(\mathsf{Nil}_{\mathsf{Bool}}, \mathsf{False}, \mathsf{Nil}_{\mathsf{Bool}})))$

Segunda extensión - Tipado

Extensión de **árboles binarios**

- $M, N, O ::= \ldots \mid \mathsf{Nil}_\sigma \mid \mathsf{Bin}(M, N, O) \mid \mathsf{raíz}(M) \mid \mathsf{izq}(M) \mid \mathsf{der}(M) \mid \mathsf{esNil}(M)$
- $\sigma ::= \ldots \mid AB_\sigma$

☆ Definir las reglas de tipado de esta extensión.

$$\frac{}{\Gamma \triangleright \mathsf{Nil}_\sigma : AB_\sigma}\ (\text{T-Nil})$$

$$\frac{\Gamma \triangleright M : AB_\sigma \quad \Gamma \triangleright N : \sigma \quad \Gamma \triangleright O : AB_\sigma}{\Gamma \triangleright Bin(M, N, O) : AB_\sigma}\ (\text{T-Bin})$$

¿Es Nil un término? ¿Por qué hace falta agregarle su tipo como subíndice?

Segunda extensión - Tipado

Extensión de árboles binarios

- $M, N, O ::= \dots \mid \mathsf{Nil}_\sigma \mid \mathsf{Bin}(M, N, O) \mid \mathsf{raíz}(M) \mid \mathsf{izq}(M) \mid \mathsf{der}(M) \mid \mathsf{esNil}(M)$
- $\sigma ::= \dots \mid \mathsf{AB}_\sigma$

☆ Definir las reglas de tipado de esta extensión.

$$\frac{\Gamma \triangleright M : \mathsf{AB}_\sigma}{\Gamma \triangleright \mathsf{raíz}(M) : \sigma} \ (\text{T-}\textsc{raíz})$$

$$\frac{\Gamma \triangleright M : \mathsf{AB}_\sigma}{\Gamma \triangleright \mathsf{izq}(M) : \mathsf{AB}_\sigma} \ (\text{T-}\textsc{izq}) \qquad \frac{\Gamma \triangleright M : \mathsf{AB}_\sigma}{\Gamma \triangleright \mathsf{der}(M) : \mathsf{AB}_\sigma} \ (\text{T-}\textsc{der})$$

$$\frac{\Gamma \triangleright M : \mathsf{AB}_\sigma}{\Gamma \triangleright \mathsf{esNil}(M) : \mathsf{Bool}} \ (\text{T-}\textsc{esNil})$$

Segunda extensión - Semántica

Extensión de árboles binarios

- $M, N, O ::= \dots \mid \mathsf{Nil}_\sigma \mid \mathsf{Bin}(M, N, O) \mid \mathsf{raíz}(M) \mid \mathsf{der}(M) \mid \mathsf{izq}(M) \mid \mathsf{esNil}(M)$
- $\sigma ::= \dots \mid \mathsf{AB}_\sigma$

☆ Definir las reglas de semántica y nuevos valores.

$$V ::= \dots \mid \mathsf{Bin}(V, V, V) \mid \mathsf{Nil}_\sigma$$

$$\frac{M \to M'}{\mathsf{Bin}(M, N, O) \to \mathsf{Bin}(M', N, O)} \text{(E-BIN1)}$$

$$\frac{N \to N'}{\mathsf{Bin}(V, N, O) \to \mathsf{Bin}(V, N', O)} \text{(E-BIN2)}$$

$$\frac{O \to O'}{\mathsf{Bin}(V, W, O) \to \mathsf{Bin}(V, W, O')} \text{(E-BIN3)}$$

Segunda extensión - Semántica

Extensión de 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 las reglas de semántica y nuevos valores.

$$\frac{M \to M'}{\mathsf{raíz}(M) \to \mathsf{raíz}(M')}\ (\text{E-RAÍZ})$$

$$\frac{}{\mathsf{raíz}(Bin(V, W, Y)) \to W}\ (\text{E-RAÍZBIN})$$

¿Podemos reducir
$\mathsf{raíz}(\mathsf{Bin}(\mathsf{Nil}_{\mathsf{Bool}}, \mathsf{True}, (\lambda x : \mathsf{Bool}.\mathsf{Bin}(\mathsf{Nil}_{\mathsf{Bool}}, x, \mathsf{Nil}_{\mathsf{Bool}}))\ \mathsf{False}))$
en un paso a True? ¿Por qué?

Segunda extensión - Semántica

Extensión de 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 las reglas de semántica y nuevos valores.

$$\frac{M \to M'}{\mathsf{izq}(M) \to \mathsf{izq}(M')}\ (\textsc{E-izq}) \qquad \frac{}{\mathsf{izq}(Bin(V, W, Y)) \to V}\ (\textsc{E-izqBin})$$

$$\frac{M \to M'}{\mathsf{der}(M) \to \mathsf{der}(M')}\ (\textsc{E-der}) \qquad \frac{}{\mathsf{der}(Bin(V, W, Y)) \to Y}\ (\textsc{E-derBin})$$

Extensión de 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 las reglas de semántica y nuevos valores.

$$\frac{M \to M'}{\mathsf{esNil}(M) \to \mathsf{esNil}(M')} \ (\text{E-}\textsc{esNil})$$

$$\frac{}{\mathsf{esNil}(\mathsf{Nil}_\sigma) \to \mathsf{True}} \ (\text{E-}\textsc{esNilTrue})$$

$$\frac{}{\mathsf{esNil}(Bin(V, W, Y)) \to \mathsf{False}} \ (\text{E-}\textsc{esNilFalse})$$

¿A qué reduciríamos en un paso $\mathsf{esNil}(\mathsf{Bin}(\mathsf{Nil}_{\mathsf{Nat}}, \mathsf{succ}(\mathsf{pred}(0)), \mathsf{Nil}_{\mathsf{Nat}}))$ si no tuviésemos la regla (E-PredZero)?

Segunda extensión bis - Notación

Tenemos otra manera de representar los proyectores u observadores de esta extensión de una manera más prolija y con menos reglas, aunque tengamos una construcción más sofisticada:

Extensión de árboles binarios bis

- $M, N, O ::= \ldots \mid Nil_\sigma \mid \text{Bin}(M, N, O) \mid \text{case}_{\text{AB}_\sigma}\ M\ of\ Nil \rightsquigarrow N\ ; \text{Bin}(i, r, d) \rightsquigarrow O$

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.
  Subtérminos: $\text{case}_{\text{AB}_\sigma}\ M\ of\ Nil \rightsquigarrow N ; \text{Bin}(i, r, d) \rightsquigarrow O$
  Anotaciones de tipo: $\text{case}_{\text{AB}_\sigma}\ M\ of\ Nil \rightsquigarrow N\ ; \text{Bin}(i, r, d) \rightsquigarrow O$

Segunda extensión bis - Tipado y Semántica

Extensión de árboles binarios bis

- $M, N, O ::= \ldots \mid Nil_\sigma \mid \text{Bin}(M, N, O) \mid \text{case}_{\text{AB}_\sigma}\ M\ of\ Nil \rightsquigarrow N\ ; \text{Bin}(i, r, d) \rightsquigarrow O$

Importante: las minúsculas i, r y d representan los nombres de las variables que pueden aparecer libres en O.

- Agregar las reglas de tipado y semántica necesarias para soportar la nueva extensión.

$$\frac{\Gamma \triangleright M : \text{AB}_\sigma \quad \Gamma \triangleright N : \tau \quad \Gamma \cup \{i : \text{AB}_\sigma, r : \sigma, d : \text{AB}_\sigma\} \triangleright O : \tau}{\Gamma \triangleright \text{case}_{\text{AB}_\sigma}\ M\ of\ Nil \rightsquigarrow N\ ; \text{Bin}(i, r, d) \rightsquigarrow O : \tau}\ (\text{T-CASE})$$

$$\frac{M \to M'}{\text{case}_{\text{AB}_\sigma}\ M\ of\ Nil \rightsquigarrow N\ ; \text{Bin}(i, r, d) \rightsquigarrow O \to \text{case}_{\text{AB}_\sigma}\ M'\ of\ Nil \rightsquigarrow N\ ; \text{Bin}(i, r, d) \rightsquigarrow O}\ (\text{E-CASE})$$

Segunda extensión bis - Tipado y Semántica

Extensión de árboles binarios bis

- $M, N, O ::= \ldots \mid Nil_\sigma \mid \text{Bin}(M, N, O) \mid \text{case}_{\text{AB}_\sigma}\ M\ of\ Nil \rightsquigarrow N\ ; \text{Bin}(i, r, d) \rightsquigarrow O$

Importante: las minúsculas i, r y d representan los <u>nombres</u> de las variables que pueden aparecer libres en O.

- Agregar las reglas de tipado y semántica necesarias para soportar la nueva extensión.

$$\frac{}{\text{case}_{\text{AB}_\sigma}\ \text{Nil}_\sigma\ of\ Nil \rightsquigarrow N\ ; \text{Bin}(i, r, d) \rightsquigarrow O \to N}\ (\text{E-}\textsc{caseNil})$$

$$\frac{}{\text{case}_{\text{AB}_\sigma}\ \text{Bin}(V, W, Y)\ of\ Nil \rightsquigarrow N\ ; \text{Bin}(i, r, d) \rightsquigarrow O \to O\{i \leftarrow V, r \leftarrow W, d \leftarrow Y\}}\ (\text{E-}\textsc{caseBin})$$

Segunda extensión bis - Macros

Extensión de árboles binarios bis

- $M, N, O ::= \ldots \mid \mathit{Nil}_\sigma \mid \mathrm{Bin}(M, N, O) \mid \mathrm{case}_{\mathrm{AB}_\sigma}\ M\ \mathit{of}\ \mathit{Nil} \leadsto N\ ; \mathrm{Bin}(i, r, d) \leadsto O$

Importante: las minúsculas i, r y d representan los nombres de las variables que pueden aparecer libres en O.

- Redefinir los términos $\mathrm{raíz}(M) \mid \mathrm{der}(M) \mid \mathrm{izq}(M) \mid \mathrm{esNil}(M)$ como macros usando la expresión de case.

$$\mathrm{raíz}(M) \stackrel{def}{=} \mathrm{case}_{\mathrm{AB}_\sigma}\ M\ \mathit{of}\ \mathit{Nil} \leadsto \mathit{fix}\ \mathit{id}_\sigma\ ; \mathrm{Bin}(i, r, d) \leadsto r$$
$$\mathrm{esNil}(M) \stackrel{def}{=} \mathrm{case}_{\mathrm{AB}_\sigma}\ M\ \mathit{of}\ \mathit{Nil} \leadsto \mathrm{True}\ ; \mathrm{Bin}(i, r, d) \leadsto \mathrm{False}$$

Segunda extensión bis - Map

Extensión de árboles binarios bis

- $M, N, O ::= \ldots \mid Nil_\sigma \mid \mathrm{Bin}(M, N, O) \mid \mathrm{case}_{\mathrm{AB}_\sigma}\ M\ of\ Nil \rightsquigarrow N\ ; \mathrm{Bin}(i, r, d) \rightsquigarrow O$

Importante: las minúsculas i, r y d representan los <u>nombres</u> de las variables que pueden aparecer libres en O.

- 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.

$$M, N ::= \ldots \mid map(M, N) \qquad \frac{\Gamma \triangleright M : \sigma \to \tau \quad \Gamma \triangleright N : \mathrm{AB}_\sigma}{\Gamma \triangleright map(M, N) : \mathrm{AB}_\tau}\ (\textsc{T-Map})$$

$$\frac{M \to M'}{map(M, N) \to map(M', N)}\ (\textsc{E-Map1}) \qquad \frac{N \to N'}{map(V, N) \to map(V, N')}\ (\textsc{E-Map2})$$

Segunda extensión bis - Map

Extensión de árboles binarios bis

- $M, N, O ::= \ldots \mid Nil_\sigma \mid \text{Bin}(M, N, O) \mid \text{case}_{\text{AB}_\sigma}\ M\ of\ Nil \rightsquigarrow N\ ; \text{Bin}(i, r, d) \rightsquigarrow O$

Importante: las minúsculas i, r y d representan los <u>nombres</u> de las variables que pueden aparecer libres en O.

- ☆ 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.

$$\frac{\emptyset \triangleright V : \sigma \to \tau}{map(V, W) \to \text{case}_{\text{AB}_\sigma}\ W\ of\ Nil \rightsquigarrow \text{Nil}_\tau\ ; \text{Bin}(i, r, d) \rightsquigarrow \text{Bin}(map(V, i), V\ r, map(V, d))}\ (\text{E-MapVar})$$

One more for the record

Recordemos la extensión de registros que vieron en la teórica:

Extensión de registros

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

Tipado:

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

Semántica:

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

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

One more for the record

Recordemos la extensión de registros que vieron en la teórica:

Extensión de registros

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

- ☆ 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.

$$M, N ::= \ldots \mid union(M, N)$$

$$\frac{\Gamma \triangleright M : \{l_1\colon \sigma_1, \ldots, l_n\colon \sigma_n\} \quad \Gamma \triangleright N : \{t_1\colon \tau_1, \ldots, t_m\colon \tau_m\} \quad (\forall_{1 \le i \le n})\ (\forall_{1 \le j \le m})\ l_i \neq t_j}{\Gamma \triangleright union(M, N) : \{l_1 : \sigma_1, \ldots, l_n : \sigma_n, t_1 : \tau_1, \ldots, t_m : \tau_m\}} \text{(T-UNION)}$$

One more for the record

Recordemos la extensión de registros que vieron en la teórica:

Extensión de registros

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

☆ 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.

$$\frac{M \to M'}{union(M, N) \to union(M', N)} \text{(E-UNION1)} \qquad \frac{N \to N'}{union(V, N) \to union(V, N')} \text{(E-UNION2)}$$

$$\frac{\dfrac{}{union(\{l_1 = M_1, \ \ldots, \ l_n = M_n\}, \{t_1 = N_1, \ \ldots, \ t_m = N_m\}) \to \ldots} \to}{\cdots \to \{l_1 = M_1, \ \ldots, \ l_n = M_n, \ t_{n+1} = N_{n+1}, \ \ldots, \ t_{n+m} = N_{n+m}\}} \text{(E-UNIONREG)}$$

One more for the record

Extensión de registros

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

- ☆ Usando la extensión anterior, definir otra que permita obtener el "reverso" de un registro, es decir, uno con las mismas etiquetas y valores pero en orden inverso.

$$M ::= \ldots \mid \mathit{reverso}(M)$$

$$\frac{\Gamma \triangleright M : \{l_1 \colon \sigma_1, \; \ldots, \; l_n \colon \sigma_n\}}{\Gamma \triangleright \mathit{reverso}(M) : \{l_n : \sigma_n, \; \ldots, \; l_1 : \sigma_1\}} \; (\text{T-}\textsc{Reverso})$$

$$\frac{M \to M'}{\mathit{reverso}(M) \to \mathit{reverso}(M')} \; (\text{E-}\textsc{Reverso})$$

One more for the record

Extensión de registros

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

☆ Usando la extensión anterior, definir otra que permita obtener el "reverso" de un registro, es decir, uno con las mismas etiquetas y valores pero en orden inverso.

$$\frac{}{\mathit{reverso}(\{\}) \to \{\}}\ (\text{E-ReversoCero})$$

$$\frac{n > 0}{\mathit{reverso}(\{l_1 = M_1,\ \ldots,\ l_n = M_n\}) \to \mathit{union}(\{l_n = M_n\}, \mathit{reverso}(\{l_1 = M_1,\ \ldots,\ l_{n-1} = M_{n-1}\}))}\ (\text{E-ReversoMulti})$$

¿Alguna duda sobre la clase?