Inferencia de Tipos

segunda parte

PLP

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

20 de septiembre de 2018

Recordemos algunas reglas de tipado

$$\frac{x : \sigma \in \Gamma}{\Gamma \triangleright x : \sigma}\,(\textrm{T-}\textsc{Var}) \qquad \frac{\Gamma \cup \{x : \sigma\} \triangleright M : \tau}{\Gamma \triangleright \lambda x : \sigma.M : \sigma \to \tau}\,(\textrm{T-}\textsc{Abs})$$

$$\frac{\Gamma \triangleright M : \sigma \to \tau \quad \Gamma \triangleright N : \sigma}{\Gamma \triangleright M\,N : \tau}\,(\textrm{T-}\textsc{App})$$

Tipado vs. Inferencia
Variables

Tipado vs. Inferencia
Variables

$$\frac{x : \sigma \in \Gamma}{\Gamma \rhd x : \sigma}\,(\textsc{T-Var})$$

Tipado vs. Inferencia
Variables

$$\frac{x : \sigma \in \Gamma}{\Gamma \triangleright x : \sigma} \, (\textsc{T-Var})$$

$$\mathbb{W}(x) \quad \stackrel{\text{def}}{=}$$

$$\frac{x : \sigma \in \Gamma}{\Gamma \rhd x : \sigma} \;(\textsc{T-Var})$$

$$\mathbb{W}(x) \quad \stackrel{\text{def}}{=} \quad \{x : t\} \rhd x : t,$$

$$\frac{x : \sigma \in \Gamma}{\Gamma \rhd x : \sigma}\,(\textsc{T-Var})$$

$$\mathbb{W}(x) \quad \stackrel{\text{def}}{=} \quad \{x : t\} \rhd x : t, \quad t \text{ variable fresca}$$

Tipado vs. Inferencia
Abstracciones

$$\frac{\Gamma \cup \{x : \sigma\} \rhd M : \tau}{\Gamma \rhd \lambda x : \sigma.M : \sigma \to \tau} \text{(T-ABS)}$$

- Sea $\mathbb{W}(U)=$

$$\frac{\Gamma \cup \{x : \sigma\} \rhd M : \tau}{\Gamma \rhd \lambda x : \sigma . M : \sigma \to \tau} \text{ (T-ABS)}$$

- Sea $\mathbb{W}(U)$=$\Gamma \rhd M : \rho$
- Si el contexto tiene información de tipos para x (i.e. $x : \alpha \in \Gamma$ para algún α), entonces

$$\frac{\Gamma \cup \{x : \sigma\} \rhd M : \tau}{\Gamma \rhd \lambda x : \sigma . M : \sigma \to \tau} \text{(T-Abs)}$$

- Sea $\mathbb{W}(U) = \Gamma \rhd M : \rho$
- Si el contexto tiene información de tipos para x (i.e. $x : \alpha \in \Gamma$ para algún α), entonces

$$\mathbb{W}(\lambda x . U) \stackrel{\text{def}}{=} \Gamma \setminus \{x : \alpha\} \rhd \lambda x : \alpha . M : \alpha \to \rho$$

- Si el contexto no tiene información de tipos para x

$$\frac{\Gamma \cup \{x : \sigma\} \rhd M : \tau}{\Gamma \rhd \lambda x : \sigma . M : \sigma \to \tau} \text{ (T-ABS)}$$

- Sea $\mathbb{W}(U) = \Gamma \rhd M : \rho$
- Si el contexto tiene información de tipos para x (i.e. $x : \alpha \in \Gamma$ para algún α), entonces

$$\mathbb{W}(\lambda x . U) \stackrel{\text{def}}{=} \Gamma \setminus \{x : \alpha\} \rhd \lambda x : \alpha . M : \alpha \to \rho$$

- Si el contexto no tiene información de tipos para x (i.e. $x \notin \mathrm{Dom}(\Gamma)$) elegimos una variable fresca t y entonces

$$\mathbb{W}(\lambda x . U) \stackrel{\text{def}}{=} \Gamma \rhd \lambda x : t . M : t \to \rho$$

Otra forma de escribirlo: Sea $\mathbb{W}(U)=$

Otra forma de escribirlo: Sea $\mathbb{W}(U)$=$\Gamma \triangleright M : \rho$

$$\frac{\Gamma \cup \{x : \sigma\} \triangleright M : \tau}{\Gamma \triangleright \lambda x : \sigma . M : \sigma \to \tau} \text{(T-A}\textsc{bs}\text{)}$$

$$\beta = \begin{cases} \alpha \text{ si } x : \alpha \in \Gamma \\ \text{variable fresca en otro caso.} \end{cases}$$

$\Gamma' = \Gamma \ominus \{x\}$

$$\mathbb{W}(\lambda x.U) \stackrel{\text{def}}{=} \Gamma' \triangleright \lambda x : \beta . M : \beta \to \rho$$

Tipado vs. Inferencia
Aplicación

$$\frac{\Gamma \rhd M : \sigma \to \tau \quad \Gamma \rhd N : \sigma}{\Gamma \rhd M\, N : \tau}\,(\text{T-App})$$

- Sea

$$\frac{\Gamma \triangleright M : \sigma \to \tau \quad \Gamma \triangleright N : \sigma}{\Gamma \triangleright M\, N : \tau}\,(\textsc{T-App})$$

- Sea
    - $\mathbb{W}(U)=$

$$\frac{\Gamma \rhd M : \sigma \to \tau \quad \Gamma \rhd N : \sigma}{\Gamma \rhd M\,N : \tau}\,(\textsc{T-App})$$

- Sea
  - $\mathbb{W}(U) = \Gamma_1 \rhd M : \tau$
  - $\mathbb{W}(V) =$

$$\frac{\Gamma \rhd M : \sigma \to \tau \quad \Gamma \rhd N : \sigma}{\Gamma \rhd M\, N : \tau} \text{(T-App)}$$

- Sea
  - $\mathbb{W}(U) = \Gamma_1 \rhd M : \tau$
  - $\mathbb{W}(V) = \Gamma_2 \rhd N : \rho$
- Sea

$$\frac{\Gamma \rhd M : \sigma \to \tau \quad \Gamma \rhd N : \sigma}{\Gamma \rhd M\, N : \tau} \text{(T-APP)}$$

- Sea
  - $\mathbb{W}(U) = \Gamma_1 \rhd M : \tau$
  - $\mathbb{W}(V) = \Gamma_2 \rhd N : \rho$
- Sea

$$S = MGU$$

$$\frac{\Gamma \rhd M : \sigma \to \tau \quad \Gamma \rhd N : \sigma}{\Gamma \rhd M\, N : \tau} \text{(T-App)}$$

- Sea
  - $\mathbb{W}(U) = \Gamma_1 \rhd M : \tau$
  - $\mathbb{W}(V) = \Gamma_2 \rhd N : \rho$
- Sea

$$S = MGU\{\sigma_1 \doteq \sigma_2 \mid x : \sigma_1 \in \Gamma_1 \wedge x : \sigma_2 \in \Gamma_2\}$$
$$\cup$$
$$\{\tau \doteq \rho \to t\} \text{ con } t \text{ una variable fresca}$$

Tipado vs. Inferencia
Aplicación

$$\frac{\Gamma \triangleright M : \sigma \to \tau \quad \Gamma \triangleright N : \sigma}{\Gamma \triangleright M\,N : \tau} \text{(T-App)}$$

- Sea
  - $\mathbb{W}(U) = \Gamma_1 \triangleright M : \tau$
  - $\mathbb{W}(V) = \Gamma_2 \triangleright N : \rho$
- Sea

$$S = MGU\{\sigma_1 \doteq \sigma_2 \mid x : \sigma_1 \in \Gamma_1 \wedge x : \sigma_2 \in \Gamma_2\}$$
$$\cup$$
$$\{\tau \doteq \rho \to t\} \text{ con } t \text{ una variable fresca}$$

- Entonces

$$\mathbb{W}(U\,V) \stackrel{\text{def}}{=}$$

$$\frac{\Gamma \rhd M : \sigma \to \tau \quad \Gamma \rhd N : \sigma}{\Gamma \rhd M\, N : \tau} \text{(T-App)}$$

- Sea
  - $\mathbb{W}(U) = \Gamma_1 \rhd M : \tau$
  - $\mathbb{W}(V) = \Gamma_2 \rhd N : \rho$
- Sea

$$S = MGU\{\sigma_1 \doteq \sigma_2 \mid x : \sigma_1 \in \Gamma_1 \wedge x : \sigma_2 \in \Gamma_2\}$$
$$\cup$$
$$\{\tau \doteq \rho \to t\} \text{ con } t \text{ una variable fresca}$$

- Entonces

$$\mathbb{W}(U\, V) \stackrel{\text{def}}{=} S\Gamma_1 \cup S\Gamma_2 \rhd S(M\, N) : St$$

Extensiones al algoritmo

En general

- Agregar casos nuevos al algoritmo.
- Menos frecuentemente, modificar casos existentes.

Extensiones al algoritmo

En general

- Agregar casos nuevos al algoritmo.
- Menos frecuentemente, modificar casos existentes.

Para incorporar nuevos términos

- Nuevas reglas de tipado $\Rightarrow$ nuevos casos del algoritmo $\mathbb{W}$.
- Anotar las expresiones con sus tipos.

$$M ::= \ldots | M \overset{x\,:\,\sigma}{\texttt{WTF}} \;(N\;\; O)$$

$$\frac{\Gamma,\; x : \sigma \triangleright M \;:\tau \qquad \Gamma \triangleright N \;:\sigma \qquad \Gamma \triangleright O \;:\sigma}{\Gamma \triangleright M \overset{x\,:\,\sigma}{\texttt{WTF}} \;(N\;\; O) \;:\tau}$$

Extender el algoritmo

$\mathbb{W}(M \;\texttt{WTF}\; (N\;\; O)) \stackrel{\text{def}}{=} ?$

$\sigma ::= \ldots \mid [\sigma]$
$M, N, O ::= \ldots \mid [\,]_\sigma \mid M :: N \mid \mathit{Case}\ M\ \mathit{of}\ [\,] \leadsto N\ ; h :: t \leadsto O$

$$\frac{}{\Gamma \triangleright [\,]_\sigma \ : [\sigma]} \qquad \frac{\Gamma \triangleright M : \sigma \qquad \Gamma \triangleright N : [\sigma]}{\Gamma \triangleright M :: N : [\sigma]}$$

$$\frac{\begin{array}{c}\Gamma \triangleright M : [\sigma] \qquad \Gamma \triangleright N : \tau \\ \Gamma \cup \{h : \sigma, t : [\sigma]\} \triangleright O : \tau\end{array}}{\Gamma \triangleright \mathit{Case}\ M\ \mathit{of}\ [\,] \leadsto N\ ; h :: t \leadsto O : \tau}$$

$\sigma ::= \ldots \mid [\sigma]$
$M, N, O ::= \ldots \mid [\,]_\sigma \mid M :: N \mid \mathit{Case}\ M\ \mathit{of}\ [\,] \leadsto N\ ; h :: t \leadsto O$

$$\frac{}{\Gamma \triangleright [\,]_\sigma\ : [\sigma]} \qquad \frac{\Gamma \triangleright M : \sigma \qquad \Gamma \triangleright N : [\sigma]}{\Gamma \triangleright M :: N : [\sigma]}$$

$$\frac{\begin{array}{c}\Gamma \triangleright M : [\sigma] \qquad \Gamma \triangleright N : \tau \\ \Gamma \cup \{h : \sigma, t : [\sigma]\} \triangleright O : \tau\end{array}}{\Gamma \triangleright \mathit{Case}\ M\ \mathit{of}\ [\,] \leadsto N\ ; h :: t \leadsto O : \tau}$$

Extender el algoritmo

$\mathbb{W}([\,]) \stackrel{\text{def}}{=} ?$
$\mathbb{W}(U_1 :: U_2) \stackrel{\text{def}}{=} ?$
$\mathbb{W}(\mathit{Case}\ U_1\ \mathit{of}\ [\,] \leadsto U_2\ ; h :: t \leadsto U_3) \stackrel{\text{def}}{=} ?$

Switch

Extender el algoritmo de inferencia $\mathbb{W}$ para que soporte el tipado del *switch* de números naturales, similar al de C o C++. La extensión de la sintaxis es la siguiente:

$M = \ldots \mid \texttt{switch}\ M\ \{\texttt{case}\ \underline{n_1} :\ M_1\ \ \ldots\ \texttt{case}\ \underline{n_k} :\ M_k\ \texttt{default} :\ M_{k+1}\}$

donde cada $\underline{n_i}$ es un numeral (un *valor* de tipo Nat, como 0, succ(0), succ(succ(0)), etc.). Esto forma parte de la sintaxis y no hace falta verificarlo en el algoritmo.

Switch

Extender el algoritmo de inferencia $\mathbb{W}$ para que soporte el tipado del *switch* de números naturales, similar al de C o C++. La extensión de la sintaxis es la siguiente:

$M = \ldots \mid \texttt{switch}\ M\ \{\texttt{case}\ \underline{n_1} :\ M_1\ \ \ldots\ \texttt{case}\ \underline{n_k} :\ M_k\ \texttt{default} :\ M_{k+1}\}$

donde cada $\underline{n_i}$ es un numeral (un *valor* de tipo Nat, como 0, succ(0), succ(succ(0)), etc.). Esto forma parte de la sintaxis y no hace falta verificarlo en el algoritmo.

La regla de tipado es la siguiente:

$$\frac{\Gamma \triangleright M : Nat \quad \forall i, j (1 \leq i, j \leq k \wedge i \neq j \Rightarrow n_i \neq n_j) \qquad \Gamma \triangleright N_1 : \sigma \quad \ldots \quad \Gamma \triangleright N_k : \sigma \quad \Gamma \triangleright N : \sigma}{\Gamma \triangleright \texttt{switch}\ M\ \{\texttt{case}\ \underline{n_1} :\ N_1\ \ldots\ \texttt{case}\ \underline{n_k} :\ N_k\ \texttt{default} : N\} : \sigma}$$

En este ejercicio modificaremos el algoritmo de inferencia para incorporar la posibilidad de utilizar letrec en nuestro cálculo.

$M ::= \ldots |$ letrec $f = M$ in N

Permite por ejemplo representar el factorial de 10 de la siguiente manera:

letrec $f = (\lambda x : \mathsf{Nat}\,.\,\mathsf{if}\ \mathsf{isZero}(x)\ \mathsf{then}\ \underline{1}\ \mathsf{else}\ x \times f\ (\mathsf{Pred}(x)))$ in $f\ \underline{10}$

Para ello se agrega la siguiente regla de tipado:

$$\frac{\Gamma \cup \{f : \pi \to \tau\} \rhd M : \pi \to \tau \qquad \Gamma \cup \{f : \pi \to \tau\} \rhd N : \sigma}{\Gamma \rhd \mathsf{letrec}\ f = M\ \mathsf{in}\ N : \sigma}$$

Extendemos el algoritmo

$$\frac{\Gamma \cup \{f : \pi \to \tau\} \rhd M : \pi \to \tau \qquad \Gamma \cup \{f : \pi \to \tau\} \rhd N : \sigma}{\Gamma \rhd \mathsf{letrec}\ f = M\ \mathsf{in}\ N : \sigma}$$

$\mathbb{W}(\mathsf{letrec}\ f = U_1\ \mathsf{in}\ U_2) \stackrel{\text{def}}{=}$?

donde

Extendemos el algoritmo

$$\frac{\Gamma \cup \{f : \pi \to \tau\} \rhd M : \pi \to \tau \qquad \Gamma \cup \{f : \pi \to \tau\} \rhd N : \sigma}{\Gamma \rhd \mathsf{letrec}\ f = M\ \mathsf{in}\ N : \sigma}$$

$\mathbb{W}(\mathsf{letrec}\ f = U_1\ \mathsf{in}\ U_2) \stackrel{\text{def}}{=} ?$
donde

- $\mathbb{W}(U_1) = \Gamma_1 \rhd M : \gamma$
- $\mathbb{W}(U_2) = \Gamma_2 \rhd N : \sigma$

Extendemos el algoritmo

$$\frac{\Gamma \cup \{f : \pi \to \tau\} \triangleright M : \pi \to \tau \qquad \Gamma \cup \{f : \pi \to \tau\} \triangleright N : \sigma}{\Gamma \triangleright \mathsf{letrec}\ f = M \ \mathsf{in}\ N : \sigma}$$

$\mathbb{W}(\mathsf{letrec}\ f = U_1\ \mathsf{in}\ U_2) \stackrel{\mathrm{def}}{=} ?$
donde

- $\mathbb{W}(U_1) = \Gamma_1 \triangleright M : \gamma$
- $\mathbb{W}(U_2) = \Gamma_2 \triangleright N : \sigma$
- $\rho = \begin{cases} \alpha_1 \text{ si } f : \alpha_1 \in \Gamma_1 \\ \text{variable fresca en otro caso.} \end{cases}$
- $\delta = \begin{cases} \alpha_2 \text{ si } f : \alpha_2 \in \Gamma_2 \\ \text{variable fresca en otro caso.} \end{cases}$

Extendemos el algoritmo

$$\frac{\Gamma \cup \{f : \pi \to \tau\} \rhd M : \pi \to \tau \qquad \Gamma \cup \{f : \pi \to \tau\} \rhd N : \sigma}{\Gamma \rhd \mathsf{letrec}\ f = M \ \mathsf{in}\ N : \sigma}$$

$\mathbb{W}(\mathsf{letrec}\ f = U_1\ \mathsf{in}\ U_2) \stackrel{\mathrm{def}}{=} ?$
donde

- $\mathbb{W}(U_1) = \Gamma_1 \rhd M : \gamma$
- $\mathbb{W}(U_2) = \Gamma_2 \rhd N : \sigma$
- $\rho = \begin{cases} \alpha_1 \text{ si } f : \alpha_1 \in \Gamma_1 \\ \text{variable fresca en otro caso.} \end{cases}$
- $\delta = \begin{cases} \alpha_2 \text{ si } f : \alpha_2 \in \Gamma_2 \\ \text{variable fresca en otro caso.} \end{cases}$
- $\Gamma_1' = \Gamma_1 \ominus \{f\}$ y $\Gamma_2' = \Gamma_2 \ominus \{f\}$

Extendemos el algoritmo

$$\frac{\Gamma \cup \{f : \pi \to \tau\} \rhd M : \pi \to \tau \qquad \Gamma \cup \{f : \pi \to \tau\} \rhd N : \sigma}{\Gamma \rhd \mathsf{letrec}\ f = M \ \mathsf{in}\ N : \sigma}$$

$\mathbb{W}(\mathsf{letrec}\ f = U_1 \ \mathsf{in}\ U_2) \stackrel{\mathrm{def}}{=} S\,\Gamma'_1 \cup S\,\Gamma'_2 \rhd S\,(\mathsf{letrec}\ f = M \ \mathsf{in}\ N) : S\,\sigma$
donde

- $\mathbb{W}(U_1) = \Gamma_1 \rhd M : \gamma$
- $\mathbb{W}(U_2) = \Gamma_2 \rhd N : \sigma$
- $\rho = \begin{cases} \alpha_1 \ \text{si}\ f : \alpha_1 \in \Gamma_1 \\ \text{variable fresca en otro caso.} \end{cases}$
- $\delta = \begin{cases} \alpha_2 \ \text{si}\ f : \alpha_2 \in \Gamma_2 \\ \text{variable fresca en otro caso.} \end{cases}$
- $\Gamma'_1 = \Gamma_1 \ominus \{f\}$ y $\Gamma'_2 = \Gamma_2 \ominus \{f\}$
- $\begin{array}{rcl} S & = & \mathtt{mgu}\ \{\rho \doteq \delta, \gamma \doteq t_1 \to t_2, \gamma \doteq \rho\} \\ & \cup & \{\sigma_1 \doteq \sigma_2 \mid x : \sigma_1 \in \Gamma'_1, x : \sigma_2 \in \Gamma'_2\} \\ & & \quad t_1 \ \text{y}\ t_2 \ \text{variables frescas} \end{array}$

Otra forma de escribirlo

$$\frac{\Gamma \cup \{f : \pi \to \tau\} \rhd M : \pi \to \tau \qquad \Gamma \cup \{f : \pi \to \tau\} \rhd N : \sigma}{\Gamma \rhd \mathsf{letrec}\ f = M\ \mathsf{in}\ N : \sigma}$$

$\mathbb{W}(\mathsf{letrec}\ f = U_1\ \mathsf{in}\ U_2) \stackrel{\text{def}}{=} S\,\Gamma_1' \cup S\,\Gamma_2' \rhd S\,(\mathsf{letrec}\ f = M\ \mathsf{in}\ N) : S\,\sigma$
donde

- $\mathbb{W}(U_1) = \Gamma_1 \rhd M : \gamma$
- $\mathbb{W}(U_2) = \Gamma_2 \rhd N : \sigma$
- $\rho = \begin{cases} \alpha_1 \text{ si } f : \alpha_1 \in \Gamma_1 \\ \alpha_2 \text{ si } f \notin \mathtt{dom}(\Gamma_1) \text{ y } f : \alpha_2 \in \Gamma_2 \\ \text{variable fresca en otro caso.} \end{cases}$
- $\Gamma_1' = \Gamma_1 \ominus \{f\}$ y $\Gamma_2' = \Gamma_2 \ominus \{f\}$
- $\begin{array}{rcl} S & = & \mathtt{mgu}\ \{\gamma \doteq t_1 \to t_2, \gamma \doteq \rho\} \\ & \cup & \{\sigma_1 \doteq \sigma_2 \mid x : \sigma_1 \in \Gamma_1, x : \sigma_2 \in \Gamma_2\} \end{array}$

  t_1 y t_2 variables frescas

Moraleja

Algunas conclusiones

- Los llamados recursivos devuelven un contexto, un término anotado y un tipo. No podemos asumir nada sobre ellos.
- Cuando la regla tiene tipos iguales: unificar.
- Si hay contextos repetidos en las premisas, unificarlos.
- Cuando la regla liga variables:
  - Obtener su tipo del Γ obtenido recursivamente.
  - Si no figuran: variable fresca.
  - Sacarlas del Γ del resultado (y del que se vaya a unificar).
- Decorar los términos según corresponda.
- Si la regla tiene restricciones adicionales, se incorporan como posibles casos de falla.