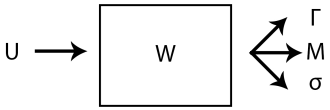
Inferencia de Tipos

PLP

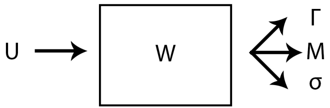
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

14 de Febrero de 2018

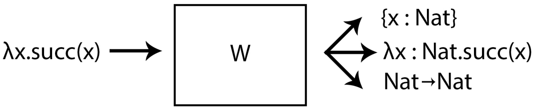
Algoritmo W



Algoritmo W



Por ejemplo:



Algoritmo W

Queremos $\mathbb{W}(\cdot)$ que dado un término U sin anotaciones verifica:

Corrección $\mathbb{W}(U) = \Gamma \rhd M : \sigma$ implica

- $\textsc{Erase}(M) = U$ y
- $\Gamma \rhd M : \sigma$ es derivable

Completitud Si $\Gamma \rhd M : \sigma$ es derivable y $\textsc{Erase}(M) = U$, entonces

- $\mathbb{W}(U)$ tiene éxito y
- $\mathbb{W}(U)$ computa un tipo principal ($\Gamma \rhd M : \sigma$ es instancia del mismo)

Casos base

Casos base:

$$
\begin{aligned}
\mathbb{W}(\mathit{0}) &\stackrel{\text{def}}{=} \emptyset \triangleright \mathit{0} : \mathsf{Nat} \\
\mathbb{W}(\mathsf{true}) &\stackrel{\text{def}}{=} \emptyset \triangleright \mathsf{true} : \mathsf{Bool} \\
\mathbb{W}(\mathsf{false}) &\stackrel{\text{def}}{=} \emptyset \triangleright \mathsf{false} : \mathsf{Bool} \\
\mathbb{W}(x) &\stackrel{\text{def}}{=} \{x : s\} \triangleright x : s, \quad s \text{ variable fresca}
\end{aligned}
$$

Casos naturales

Casos Nat:

- Sea $\mathbb{W}(U) = \Gamma \triangleright M : \tau$
- Sea $S = MGU\{\tau \doteq \mathsf{Nat}\}$
- Entonces

$$\mathbb{W}(succ(U)) \stackrel{\text{def}}{=} S\Gamma \triangleright S\, succ(M) : \mathsf{Nat}$$
$$\mathbb{W}(pred(U)) \stackrel{\text{def}}{=} S\Gamma \triangleright S\, succ(M) : \mathsf{Nat}$$
$$\mathbb{W}(isZero(U)) \stackrel{\text{def}}{=} S\Gamma \triangleright S\, succ(M) : \mathsf{Bool}$$

Caso Abs

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

Sea $\mathbb{W}(U) = \Gamma \rhd M : \rho$

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

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

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

Caso App

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

- 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

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

Aplicando el algoritmo W

Ejercicio 1

Utilizar el algoritmo $\mathbb{W}$ para las siguientes expresiones:

a. $\lambda f\,.\,\lambda x\,.\,f(f\ x)$

b. $x(\lambda x\,.\,x)$

Algoritmo de Martelli-Montanari

1. **Descomposiciń**
   $\{\sigma_1 \to \sigma_2 \doteq \tau_1 \to \tau_2\} \cup G \mapsto \{\sigma_1 \doteq \tau_1, \sigma_2 \doteq \tau_2\} \cup G$
   $\{\mathsf{Nat} \doteq \mathsf{Nat}\} \cup G \mapsto G$
   $\{\mathsf{Bool} \doteq \mathsf{Bool}\} \cup G \mapsto G$
2. **Eliminación de par trivial**
   $\{s \doteq s\} \cup G \mapsto G$
3. **Swap**: si σ no es una variable
   $\{\sigma \doteq s\} \cup G \mapsto \{s \doteq \sigma\} \cup G$
4. **Eliminación de variable**: si $s \notin FV(\sigma)$
   $\{s \doteq \sigma\} \cup G \mapsto_{\sigma/s} G[\sigma/s]$
5. **Falla**
   $\{\sigma \doteq \tau\} \cup G \mapsto \texttt{falla}$, con $(\sigma, \tau) \in T \cup T^{-1}$ y
   $T = \{(\mathsf{Bool}, \mathsf{Nat}), (\mathsf{Nat}, \sigma_1 \to \sigma_2), (\mathsf{Bool}, \sigma_1 \to \sigma_1)\}$
6. **Occur check**: si $s \neq \sigma$ y $s \in FV(\sigma)$
   $\{s \doteq \sigma\} \cup G \mapsto \texttt{falla}$

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.

Extensión del lenguaje

Abstracciones sobre pares

$$M ::= \ldots \,|\lambda\langle x, y\rangle : \langle\sigma \times \tau\rangle.M$$

$$\frac{\Gamma, x : \sigma, y : \tau \rhd M : \rho}{\Gamma \rhd \lambda\langle x, y\rangle : \langle\sigma \times \tau\rangle.M : \langle\sigma \times \tau\rangle \to \rho}$$

Extensión del lenguaje
Abstracciones sobre pares

$$M ::= \ldots \mid \lambda\langle x, y\rangle : \langle \sigma \times \tau\rangle . M$$

$$M' ::= \ldots \mid \lambda\langle x, y\rangle . M'$$

$$\frac{\Gamma, x : \sigma, y : \tau \triangleright M : \rho}{\Gamma \triangleright \lambda\langle x, y\rangle : \langle \sigma \times \tau\rangle . M : \langle \sigma \times \tau\rangle \to \rho}$$

Extensión del lenguaje

Abstracciones sobre pares

$$M ::= \dots | \lambda \langle x, y \rangle : \langle \sigma \times \tau \rangle . M$$

$$M' ::= \dots | \lambda \langle x, y \rangle . M'$$

$$\frac{\Gamma, x : \sigma, y : \tau \triangleright M : \rho}{\Gamma \triangleright \lambda \langle x, y \rangle : \langle \sigma \times \tau \rangle . M : \langle \sigma \times \tau \rangle \to \rho}$$

Ejercicio 2

Extender el algoritmo:

$\mathbb{W}(\lambda \langle x, y \rangle . U) \stackrel{\text{def}}{=}$?

Extensiones del lenguaje
Listas

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

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

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

Extensiones del lenguaje
Listas

$\sigma ::= \ldots \mid [\sigma]$
$M, N, O ::= \ldots \mid [\,]_\sigma \mid M :: N \mid Case\ M\ of\ [\,] \rightsquigarrow N \,;\, h :: t \rightsquigarrow O$

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

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

Ejercicio 3

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

Otra extensión

Switch de naturales

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

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

Ejercicio 4

Extender el algoritmo:

$\mathbb{W}(\texttt{switch}\ U_0\ \{\texttt{case}\ \underline{n_1} :\ U_1\ \ \ldots\ \texttt{case}\ \underline{n_k} :\ U_k\ \texttt{default} :\ U_{k+1}\}) \stackrel{\text{def}}{=}$?

Otra extensión del lenguaje

Letrec

$$M ::= \ldots \mid \text{letrec } f = M \text{ in } N$$

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

Ejercicio 5

Extender el algoritmo:

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

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 o tipos con una forma específica: 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.

PLP ▷ fin clase: consultas

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