Typage monomorphe à la Curry

Typage monomorphe

Expressions à la Curry

$$
\begin{array}{rll}
M ::= & x & | \\
 & cte & | \\
 & \langle M, M \rangle & | \\
 & M\ M & | \\
 & \lambda x.M & | \\
 & \textbf{let } x = M \textbf{ in } M &
\end{array}
$$

Quelques exemples

$$\textbf{let } x : \mathit{int} = 3 \textbf{ in } x + 1$$

s'écrit maintenant

$$\textbf{let } x = 3 \textbf{ in } x + 1$$

$$\textbf{let } x : \mathit{int} = 4 \textbf{ in } (\textbf{let } y : \mathit{int} = x + 1 \textbf{ in } x * y))$$

s'écrit maintenant

$$\textbf{let } x = 4 \textbf{ in } (\textbf{let } y = x + 1 \textbf{ in } x * y))$$

Règles du typage monomorphe à la Curry

$$\Gamma \vdash x : \Gamma(x) \qquad \Gamma \vdash cte : TC(cte)$$

$$\frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash M\ N : B} \qquad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B}$$

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash \langle M, N \rangle : A \times B} \qquad \frac{\Gamma \vdash M : A \quad \Gamma, x : A \vdash N : B}{\Gamma \vdash \textbf{let}\ x = M\ \textbf{in}\ N : B}$$

Propriétés ?

L'**unicité** n'est plus vraie :

$$\vdash \lambda x.x : int \to int \qquad \vdash \lambda x.x : bool \to bool$$

Mais les deux fonctions sont une instance d'une même fonction identité qui se comporte de la même façon :

Polymorphisme

Vers un algorithme de typage

- Les substitutions
- Les unificateurs d'un système d'équations
- Algorithme d'unification d'un système d'équations
- La construction d'un système d'équations à partir d'un terme
- Algorithme de typage d'un terme à l'aide de l'unification

Types à la Curry : difficultés de l'algorithme de typage

$$\mathit{Type}(\Gamma, \lambda x.M) = A \to B \qquad \text{s'il existe } A \text{ t.q. } \mathit{Type}((\Gamma, x : A), M) = B$$

$$\mathit{Type}(\Gamma, \textbf{let } x = M \textbf{ in } N) = B \qquad \text{s'il existe } A \text{ t.q. } \mathit{Type}(\Gamma, M) = A \text{ et } \mathit{Type}((\Gamma, x : A), N) = B$$

Vers un algorithme de typage

- Soit M un terme à typer. Pour chaque variable x de M on introduit une variable de type α_x et pour chaque sous-expression N de M on introduit une variable de type α_N.
- On introduit un tableau de schémas de types pour les constantes, appelé STC. Ainsi par exemple $STC(fst) = \alpha \times \beta \to \alpha$.

On associe à M un système d'équations $SE(M)$ comme suit :

M	$SE(M)$
x	$\{\alpha_M \doteq \alpha_x\}$
cte	$\{\alpha_M \doteq STC(cte)\}$
$\langle N, L\rangle$	$\{\alpha_M \doteq \alpha_N \times \alpha_L\} \cup SE(N) \cup SE(L)$
$N\ L$	$\{\alpha_N \doteq \alpha_L \to \alpha_M\} \cup SE(N) \cup SE(L)$
$\lambda x.N$	$\{\alpha_M \doteq \alpha_x \to \alpha_N\} \cup SE(N)$
let $x = N$ **in** L	$\{\alpha_M \doteq \alpha_L; \alpha_x \doteq \alpha_N\} \cup SE(N) \cup SE(L)$

Correction et complétude de l'algorithme de typage

Théorème : (Correction) Si σ est une solution de $SE(M)$, alors $\Delta \vdash M : \tau(\alpha_M)$, où $\Delta = \{x : \tau(\alpha_x) \mid x \in FV(M)\}$ et τ est une instance de σ.

Théorème : (Complétude) S'il existent Δ et A t.q. $\Delta \vdash M : A$, alors $SE(M)$ admet une solution.

Principalité de l'algorithme de typage

Théorème : (Principalité) Si M est typable, i.e., $\Delta \vdash M : A$, alors A est une instance du type principal $\sigma(\alpha_M)$, où σ est une solution principale du système $SE(M)$.