Types in Programming Languages (5)

Christian Urban

http://www4.in.tum.de/lehre/vorlesungen/types/WS0607/

Algorithm W

The original algorithm of Damas and Milner:

- **Variables:**

$$W(\Gamma, x) = (\varepsilon, T[X_1 := Y_1, \ldots, X_n := Y_n])$$

where $(x : \forall\{X_1, \ldots, X_n\}.T) \in \Gamma$ and the Y_i are distinct and fresh (w.r.t. Γ and T).

- **Lambdas:** calculate first

$$W((x : \forall\{\}.Y, \Gamma), e) = (\theta, T_1)$$

where Y is fresh (w.r.t. Γ). Then return

$$W(\Gamma, \lambda x.e) = (\theta, \theta(Y) \rightarrow T_1)$$

Algorithm W

- **Applications:** Calculate first

$$W(\Gamma, e_1) = (\theta_1, T_1)$$

then

$$W(\theta_1(\Gamma), e_2) = (\theta_2, T_2)$$

and then

$$\text{unify}\ \{\theta_2(T_1) \stackrel{?}{=} T_2 \to Y\} = \theta_3$$

where Y is fresh (w.r.t. Γ). Finally return

$$W(\Gamma, e_1\ e_2) = (\theta_3 \circ \theta_2 \circ \theta_1, \theta_3(Y))$$

Algorithm W

- **Lets:** Calculate first

$$W(\Gamma, e_1) = (\theta_1, T_1)$$

then

$$A = \mathsf{tv}(T_1) - \mathsf{ftv}(\theta_1(\Gamma))$$

and then

$$W((x : \forall A.T_1, \theta_1(\Gamma)), e_2) = (\theta_2, T_2)$$

Finally return

$$W(\Gamma, \mathsf{let}\, x = e_1 \,\mathsf{in}\, e_2) = (\theta_2 \circ \theta_1, T_2)$$

Example

- $\Gamma \vdash \mathsf{let}\ f = \lambda x.x\ \mathsf{in}\ g\ (f\ a) :\quad ?$

  We expect $X_2 \to X_3$.

- $\Gamma = \left\{ \begin{array}{l} g : \forall\{\}.X_1 \to X_2 \to X_3, \\ a : \forall\{\}.X_1, \end{array} \right\}$

$$W(\Gamma; \mathsf{let}\ f = \lambda x.x\ \mathsf{in}\ g\ (f\ a)) =$$

$W(\Gamma; \text{let } f = \lambda x.x \text{ in } g\ (f\ a)) =$

$\boldsymbol{W(\Gamma; \lambda x.x) =}$

$W(\Gamma; \mathsf{let}\ f = \lambda x.x\ \mathsf{in}\ g\ (f\ a)) =$

$W(\Gamma; \lambda x.x) =$

$\boldsymbol{W((x : \forall\{\}.Y), \Gamma; x) =}$

$W(\Gamma; \text{let } f = \lambda x.x \text{ in } g\ (f\ a)) =$

$W(\Gamma; \lambda x.x) =$

$\boldsymbol{W((x : \forall\{\}.Y), \Gamma; x) = \ (\varepsilon, Y)}$

$W(\Gamma; \mathsf{let}\ f = \lambda x.x\ \mathsf{in}\ g\ (f\ a)) =$

$\boldsymbol{W(\Gamma; \lambda x.x) = \ (\varepsilon, \varepsilon(Y) \rightarrow Y)}$

$W((x : \forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$

$W(\Gamma; \mathsf{let}\ f = \lambda x.x\ \mathsf{in}\ g\ (f\ a)) =$

$\boldsymbol{W(\Gamma; \lambda x.x) = \ (\varepsilon, Y \to Y)}$

$W((x : \forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$

$W(\Gamma; \mathsf{let}\ f = \lambda x.x\ \mathsf{in}\ g\ (f\ a)) =$

$W(\Gamma; \lambda x.x) = (\varepsilon, Y \to Y)$

$W((x{:}\forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$

$A = \mathsf{tv}(Y \to Y) - \mathsf{ftv}(\Gamma)$

$W(\Gamma; \mathsf{let}\ f = \lambda x.x\ \mathsf{in}\ g\ (f\ a)) =$

$W(\Gamma; \lambda x.x) = (\varepsilon, Y \to Y)$

$W((x : \forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$

$\boldsymbol{A = \{Y\} - \{X_1, X_2, X_3\}}$

$W(\Gamma; \mathsf{let}\ f = \lambda x.x\ \mathsf{in}\ g\ (f\ a)) =$

$W(\Gamma; \lambda x.x) = (\varepsilon, Y \to Y)$

$W((x : \forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$

$\boldsymbol{A = \{Y\}}$

$$W(\Gamma; \mathsf{let}\ f = \lambda x.x\ \mathsf{in}\ g\ (f\ a)) =$$

$$W(\Gamma; \lambda x.x) = (\varepsilon, Y \to Y)$$

$$W((x : \forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$$

$$A = \{Y\}$$

$$\boldsymbol{W(f : \forall\{Y\}.Y \to Y, \varepsilon(\Gamma); g\ (f\ a)) =}$$

$$W(\Gamma; \text{let } f = \lambda x.x \text{ in } g\ (f\ a)) =$$

$$W(\Gamma; \lambda x.x) = (\varepsilon, Y \to Y)$$

$$W((x : \forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$$

$$A = \{Y\}$$

$$\boldsymbol{W(f : \forall\{Y\}.Y \to Y, \Gamma; g\ (f\ a)) =}$$

$$W(\Gamma; \mathsf{let}\ f = \lambda x.x\ \mathsf{in}\ g\ (f\ a)) =$$

$$W(\Gamma; \lambda x.x) = (\varepsilon, Y \to Y)$$

$$W((x : \forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$$

$$A = \{Y\}$$

$$W(f : \forall\{Y\}.Y \to Y, \Gamma; g\ (f\ a)) =$$

$$\boldsymbol{W(f : \forall\{Y\}.Y \to Y, \Gamma; g) =}$$

$$W(\Gamma; \mathsf{let}\ f = \lambda x.x\ \mathsf{in}\ g\ (f\ a)) =$$

$$W(\Gamma; \lambda x.x) = (\varepsilon, Y \to Y)$$

$$W((x : \forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$$

$$A = \{Y\}$$

$$W(f : \forall\{Y\}.Y \to Y, \Gamma; g\ (f\ a)) =$$

$$\boldsymbol{W(f : \forall\{Y\}.Y \to Y, \Gamma; g) = (\varepsilon, X_1 \to X_2 \to X_3)}$$

$$W(\Gamma; \mathsf{let}\ f = \lambda x.x\ \mathsf{in}\ g\ (f\ a)) =$$

$$W(\Gamma; \lambda x.x) = (\varepsilon, Y \to Y)$$

$$W((x{:}\forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$$

$$A = \{Y\}$$

$$W(f{:}\forall\{Y\}.Y \to Y, \Gamma; g\ (f\ a)) =$$

$$W(f{:}\forall\{Y\}.Y \to Y, \Gamma; g) = (\varepsilon, X_1 \to X_2 \to X_3)$$

$$\boldsymbol{W(f{:}\forall\{Y\}.Y \to Y, \Gamma; f a) =}$$

$W(\Gamma; \text{let } f = \lambda x.x \text{ in } g\ (f\ a)) =$

$W(\Gamma; \lambda x.x) = (\varepsilon, Y \to Y)$

$W((x : \forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$

$A = \{Y\}$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; g\ (f\ a)) =$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; g) = (\varepsilon, X_1 \to X_2 \to X_3)$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; f a) =$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; f) =$

$W(\Gamma; \mathsf{let}\ f = \lambda x.x\ \mathsf{in}\ g\ (f\ a)) =$

$W(\Gamma; \lambda x.x) = (\varepsilon, Y \to Y)$

$W((x{:}\forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$

$A = \{Y\}$

$W(f{:}\forall\{Y\}.Y \to Y, \Gamma; g\ (f\ a)) =$

$W(f{:}\forall\{Y\}.Y \to Y, \Gamma; g) = (\varepsilon, X_1 \to X_2 \to X_3)$

$W(f{:}\forall\{Y\}.Y \to Y, \Gamma; f a) =$

$\boldsymbol{W(f{:}\forall\{Y\}.Y \to Y, \Gamma; f) = (\varepsilon, Y' \to Y')}$

$$W(\Gamma; \mathsf{let}\ f = \lambda x.x\ \mathsf{in}\ g\ (f\ a)) =$$

$$W(\Gamma; \lambda x.x) = (\varepsilon, Y \to Y)$$

$$W((x{:}\forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$$

$$A = \{Y\}$$

$$W(f{:}\forall\{Y\}.Y \to Y, \Gamma; g\ (f\ a)) =$$

$$W(f{:}\forall\{Y\}.Y \to Y, \Gamma; g) = (\varepsilon, X_1 \to X_2 \to X_3)$$

$$W(f{:}\forall\{Y\}.Y \to Y, \Gamma; f a) =$$

$$W(f{:}\forall\{Y\}.Y \to Y, \Gamma; f) = (\varepsilon, Y' \to Y')$$

$$\boldsymbol{W(f{:}\forall\{Y\}.Y \to Y, \Gamma; a) =}$$

$W(\Gamma; \text{let } f = \lambda x.x \text{ in } g\ (f\ a)) =$

$W(\Gamma; \lambda x.x) = (\varepsilon, Y \rightarrow Y)$

$W((x : \forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$

$A = \{Y\}$

$W(f : \forall\{Y\}.Y \rightarrow Y, \Gamma; g\ (f\ a)) =$

$W(f : \forall\{Y\}.Y \rightarrow Y, \Gamma; g) = (\varepsilon, X_1 \rightarrow X_2 \rightarrow X_3)$

$W(f : \forall\{Y\}.Y \rightarrow Y, \Gamma; f a) =$

$W(f : \forall\{Y\}.Y \rightarrow Y, \Gamma; f) = (\varepsilon, Y' \rightarrow Y')$

$\boldsymbol{W(f : \forall\{Y\}.Y \rightarrow Y, \Gamma; a) = (\varepsilon, X_1)}$

$W(\Gamma; \mathsf{let}\ f = \lambda x.x\ \mathsf{in}\ g\ (f\ a)) =$

$W(\Gamma; \lambda x.x) = (\varepsilon, Y \to Y)$

$W((x : \forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$

$A = \{Y\}$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; g\ (f\ a)) =$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; g) = (\varepsilon, X_1 \to X_2 \to X_3)$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; f a) =$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; f) = (\varepsilon, Y' \to Y')$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; a) = (\varepsilon, X_1)$

$\mathsf{unify}(\varepsilon(Y' \to Y') \stackrel{?}{=} X_1 \to Y'') =$

$W(\mathit{\Gamma}; \text{let } f = \lambda x.x \text{ in } g\ (f\ a)) =$

$W(\mathit{\Gamma}; \lambda x.x) = (\varepsilon, Y \to Y)$

$W((x : \forall\{\}.Y), \mathit{\Gamma}; x) = (\varepsilon, Y)$

$A = \{Y\}$

$W(f : \forall\{Y\}.Y \to Y, \mathit{\Gamma}; g\ (f\ a)) =$

$W(f : \forall\{Y\}.Y \to Y, \mathit{\Gamma}; g) = (\varepsilon, X_1 \to X_2 \to X_3)$

$W(f : \forall\{Y\}.Y \to Y, \mathit{\Gamma}; f a) =$

$W(f : \forall\{Y\}.Y \to Y, \mathit{\Gamma}; f) = (\varepsilon, Y' \to Y')$

$W(f : \forall\{Y\}.Y \to Y, \mathit{\Gamma}; a) = (\varepsilon, X_1)$

$\text{unify}(Y' \to Y' \stackrel{?}{=} X_1 \to Y'') =$

$W(\Gamma; \mathsf{let}\ f = \lambda x.x\ \mathsf{in}\ g\ (f\ a)) =$

$W(\Gamma; \lambda x.x) = (\varepsilon, Y \to Y)$

$W((x{:}\forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$

$A = \{Y\}$

$W(f{:}\forall\{Y\}.Y \to Y, \Gamma; g\ (f\ a)) =$

$W(f{:}\forall\{Y\}.Y \to Y, \Gamma; g) = (\varepsilon, X_1 \to X_2 \to X_3)$

$W(f{:}\forall\{Y\}.Y \to Y, \Gamma; f a) =$

$W(f{:}\forall\{Y\}.Y \to Y, \Gamma; f) = (\varepsilon, Y' \to Y')$

$W(f{:}\forall\{Y\}.Y \to Y, \Gamma; a) = (\varepsilon, X_1)$

$\mathsf{unify}(Y' \to Y' =^{?} X_1 \to Y'') = [Y' := X_1, Y'' := X_1]$

$W(\Gamma; \text{let } f = \lambda x.x \text{ in } g\ (f\ a)) =$

$W(\Gamma; \lambda x.x) = (\varepsilon, Y \to Y)$

$W((x : \forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$

$A = \{Y\}$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; g\ (f\ a)) =$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; g) = (\varepsilon, X_1 \to X_2 \to X_3)$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; f a) =$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; f) = (\varepsilon, Y' \to Y')$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; a) = (\varepsilon, X_1)$

$\text{unify}(Y' \to Y' \stackrel{?}{=} X_1 \to Y'') = [Y' := X_1, Y'' := X_1]$

$[Y' := X_1, Y'' := X_1] \circ \varepsilon \circ \varepsilon = [Y' := X_1, Y'' := X_1]$

$Y''[Y' := X_1, Y'' := X_1] = X_1$

$W(\Gamma; \text{let } f = \lambda x.x \text{ in } g\ (f\ a)) =$

$W(\Gamma; \lambda x.x) = (\varepsilon, Y \to Y)$

$W((x : \forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$

$A = \{Y\}$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; g\ (f\ a)) =$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; g) = (\varepsilon, X_1 \to X_2 \to X_3)$

$\boldsymbol{W(f : \forall\{Y\}.Y \to Y, \Gamma; f a) = ([Y' := X_1, Y'' := X_1], X_1)}$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; f) = (\varepsilon, Y' \to Y')$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; a) = (\varepsilon, X_1)$

$\text{unify}(Y' \to Y' \stackrel{?}{=} X_1 \to Y'') = [Y' := X_1, Y'' := X_1]$

$[Y' := X_1, Y'' := X_1] \circ \varepsilon \circ \varepsilon = [Y' := X_1, Y'' := X_1]$

$Y''[Y' := X_1, Y'' := X_1] = X_1$

$W(\Gamma; \mathsf{let}\ f = \lambda x.x\ \mathsf{in}\ g\ (f\ a)) =$

$W(\Gamma; \lambda x.x) = (\varepsilon, Y \to Y)$

$W((x : \forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$

$A = \{Y\}$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; g\ (f\ a)) =$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; g) = (\varepsilon, X_1 \to X_2 \to X_3)$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; f a) = ([Y' := X_1, Y'' := X_1], X_1)$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; f) = (\varepsilon, Y' \to Y')$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; a) = (\varepsilon, X_1)$

$\mathsf{unify}(Y' \to Y' \stackrel{?}{=} X_1 \to Y'') = [Y' := X_1, Y'' := X_1]$

$[Y' := X_1, Y'' := X_1] \circ \varepsilon \circ \varepsilon = [Y' := X_1, Y'' := X_1]$

$Y''[Y' := X_1, Y'' := X_1] = X_1$

$\boldsymbol{X_1 \to X_2 \to X_3[Y' := X_1, Y'' := X_1] = X_1 \to X_2 \to X_3}$

$W(\Gamma; \mathsf{let}\ f = \lambda x.x\ \mathsf{in}\ g\ (f\ a)) =$

$W(\Gamma; \lambda x.x) = (\varepsilon, Y \rightarrow Y)$

$W((x : \forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$

$A = \{Y\}$

$W(f : \forall\{Y\}.Y \rightarrow Y, \Gamma; g\ (f\ a)) =$

$W(f : \forall\{Y\}.Y \rightarrow Y, \Gamma; g) = (\varepsilon, X_1 \rightarrow X_2 \rightarrow X_3)$

$W(f : \forall\{Y\}.Y \rightarrow Y, \Gamma; f a) = ([Y' := X_1, Y'' := X_1], X_1)$

$W(f : \forall\{Y\}.Y \rightarrow Y, \Gamma; f) = (\varepsilon, Y' \rightarrow Y')$

$W(f : \forall\{Y\}.Y \rightarrow Y, \Gamma; a) = (\varepsilon, X_1)$

$\mathsf{unify}(Y' \rightarrow Y' =^? X_1 \rightarrow Y'') = [Y' := X_1, Y'' := X_1]$

$[Y' := X_1, Y'' := X_1] \circ \varepsilon \circ \varepsilon = [Y' := X_1, Y'' := X_1]$

$Y''[Y' := X_1, Y'' := X_1] = X_1$

$X_1 \rightarrow X_2 \rightarrow X_3[Y' := X_1, Y'' := X_1] = X_1 \rightarrow X_2 \rightarrow X_3$

$\mathsf{unify}(X_1 \rightarrow X_2 \rightarrow X_3 =^? X_1 \rightarrow Y''') = [Y''' := X_2 \rightarrow X_3]$

$W(\Gamma; \mathsf{let}\ f = \lambda x.x\ \mathsf{in}\ g\ (f\ a)) =$

$W(\Gamma; \lambda x.x) = (\varepsilon, Y \to Y)$

$W((x : \forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$

$A = \{Y\}$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; g\ (f\ a)) =$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; g) = (\varepsilon, X_1 \to X_2 \to X_3)$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; f a) = ([Y' := X_1, Y'' := X_1], X_1)$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; f) = (\varepsilon, Y' \to Y')$

$W(f : \forall\{Y\}.Y \to Y, \Gamma; a) = (\varepsilon, X_1)$

$\mathsf{unify}(Y' \to Y' =^? X_1 \to Y'') = [Y' := X_1, Y'' := X_1]$

$[Y' := X_1, Y'' := X_1] \circ \varepsilon \circ \varepsilon = [Y' := X_1, Y'' := X_1]$

$Y''[Y' := X_1, Y'' := X_1] = X_1$

$X_1 \to X_2 \to X_3[Y' := X_1, Y'' := X_1] = X_1 \to X_2 \to X_3$

$\mathsf{unify}(X_1 \to X_2 \to X_3 =^? X_1 \to Y''') = [Y''' := X_2 \to X_3]$

$$[Y''' := X_2 \to X_3] \circ [Y' := X_1, Y'' := X_1] \circ \varepsilon = [Y' := X_1, Y'' := X_1, Y''' := X_2 \to X_3] \mapsto \theta$$

$$W(\Gamma; \text{let } f = \lambda x.x \text{ in } g\ (f\ a)) =$$

$$W(\Gamma; \lambda x.x) = (\varepsilon, Y \to Y)$$

$$W((x : \forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$$

$$A = \{Y\}$$

$$\boldsymbol{W(f : \forall\{Y\}.Y \to Y, \Gamma; g\ (f\ a)) = \ (\theta, X_2 \to X_3)}$$

$$W(f : \forall\{Y\}.Y \to Y, \Gamma; g) = (\varepsilon, X_1 \to X_2 \to X_3)$$

$$W(f : \forall\{Y\}.Y \to Y, \Gamma; f a) = ([Y' := X_1, Y'' := X_1], X_1)$$

$$W(f : \forall\{Y\}.Y \to Y, \Gamma; f) = (\varepsilon, Y' \to Y')$$

$$W(f : \forall\{Y\}.Y \to Y, \Gamma; a) = (\varepsilon, X_1)$$

$$\text{unify}(Y' \to Y' =^? X_1 \to Y'') = [Y' := X_1, Y'' := X_1]$$

$$[Y' := X_1, Y'' := X_1] \circ \varepsilon \circ \varepsilon = [Y' := X_1, Y'' := X_1]$$

$$Y''[Y' := X_1, Y'' := X_1] = X_1$$

$$X_1 \to X_2 \to X_3[Y' := X_1, Y'' := X_1] = X_1 \to X_2 \to X_3$$

$$\text{unify}(X_1 \to X_2 \to X_3 =^? X_1 \to Y''') = [Y''' := X_2 \to X_3]$$

$$[Y''' := X_2 \to X_3] \circ [Y' := X_1, Y'' := X_1] \circ \varepsilon =$$
$$[Y' := X_1, Y'' := X_1, Y''' := X_2 \to X_3] \mapsto \theta$$

$$W(\Gamma; \text{let } f = \lambda x.x \text{ in } g\ (f\ a)) = (\theta, X_2 \to X_3)$$

$$W(\Gamma; \lambda x.x) = (\varepsilon, Y \to Y)$$

$$W((x : \forall\{\}.Y), \Gamma; x) = (\varepsilon, Y)$$

$$A = \{Y\}$$

$$W(f : \forall\{Y\}.Y \to Y, \Gamma; g\ (f\ a)) = (\theta, X_2 \to X_3)$$

$$W(f : \forall\{Y\}.Y \to Y, \Gamma; g) = (\varepsilon, X_1 \to X_2 \to X_3)$$

$$W(f : \forall\{Y\}.Y \to Y, \Gamma; f a) = ([Y' := X_1, Y'' := X_1], X_1)$$

$$W(f : \forall\{Y\}.Y \to Y, \Gamma; f) = (\varepsilon, Y' \to Y')$$

$$W(f : \forall\{Y\}.Y \to Y, \Gamma; a) = (\varepsilon, X_1)$$

$$\text{unify}(Y' \to Y' =^? X_1 \to Y'') = [Y' := X_1, Y'' := X_1]$$

$$[Y' := X_1, Y'' := X_1] \circ \varepsilon \circ \varepsilon = [Y' := X_1, Y'' := X_1]$$

$$Y''[Y' := X_1, Y'' := X_1] = X_1$$

$$X_1 \to X_2 \to X_3[Y' := X_1, Y'' := X_1] = X_1 \to X_2 \to X_3$$

$$\text{unify}(X_1 \to X_2 \to X_3 =^? X_1 \to Y''') = [Y''' := X_2 \to X_3]$$

$$[Y''' := X_2 \to X_3] \circ [Y' := X_1, Y'' := X_1] \circ \varepsilon =$$
$$[Y' := X_1, Y'' := X_1, Y''' := X_2 \to X_3] \mapsto \theta$$

Modified Algorithm

- **Input:** a (valid) context Γ, an expression e, a type-constraint ρ (this is a type)
- **Output:** FAIL or a substitution θ
- **Start of the Algorithm:**

$$M(\Gamma, e, X)$$

where X is a type-variable; the result will be the substitution θ applied to X.

Algorithm M

- **Variables:**

  $$M(\Gamma, x, \rho) = \mathsf{unify}\{\rho \stackrel{?}{=} T[X_1 := Y_1, .., X_n := Y_n]\}$$

  where $(x : \forall\{X_1, \ldots, X_n\}.T) \in \Gamma$ and the Y_i are distinct and fresh (w.r.t. Γ and T).

Algorithm M

- **Lambdas:** unify first

$$\mathsf{unify}\{\rho =^? Y_1 \to Y_2\} = \theta_1$$

where Y_1 and Y_2 are fresh (w.r.t. Γ,ρ).

Then calculate

$$M(x : \theta_1(Y_1), \Gamma\ , e, \theta_1(Y_2)) = \theta_2$$

Finally return

$$M(\Gamma, \lambda x.e, \rho) = \theta_2 \circ \theta_1$$

Algorithm M

- **Applications:** Calculate first

$$M(\Gamma, e_1, Y \to \rho) = \theta_1$$

where Y is fresh (w.r.t. Γ,ρ).

Then

$$M(\theta_1(\Gamma), e_2, \theta_1(Y)) = \theta_2$$

Finally return

$$M(\Gamma, e_1\ e_2, \rho) = \theta_2 \circ \theta_1$$

Algorithm M

- **Lets:** Calculate first

$$M(\Gamma, e_1, Y) = \theta_1$$

where Y is fresh (w.r.t. Γ,ρ). Then

$$A = \mathsf{tv}(\theta_1(Y)) - \mathsf{ftv}(\theta_1(\Gamma))$$

and then

$$M((x : \forall A.\theta_1(Y), \theta_1(\Gamma)), e_2, \theta_1(\rho)) = \theta_2$$

Finally return

$$M(\Gamma, \mathsf{let}\, x = e_1 \,\mathsf{in}\, e_2, \rho) = \theta_2 \circ \theta_1$$

Soundness and Completeness

We can show soundness and completeness relative to $\boldsymbol{W}$:

- $\boldsymbol{M(\Gamma, e, \rho) = \theta \Rightarrow W(\Gamma, e) = (\theta, \theta(\rho))}$

- $\boldsymbol{W(\Gamma, e) = (\theta, T) \Rightarrow M(\Gamma, e, Y) = \theta}$

  will not work!

Soundness and Completeness

We can show soundness and completeness relative to W:

- $M(\Gamma, e, \rho) = \theta \Rightarrow W(\Gamma, e) = (\theta, \theta(\rho))$

- $W(\Gamma, e) = (\theta, T) \Rightarrow$
  $\forall \rho.\ \mathsf{unify}\{\rho =^? T\} \Rightarrow M(\Gamma, e, \rho) = \theta$

Soundness and Completeness

We can show soundness and completeness relative to $\boldsymbol{W}$:

- $M(\Gamma, e, \rho) = \theta \Rightarrow W(\Gamma, e) = (\theta, \theta(\rho))$

- $W(\Gamma, e) = (\theta, T) \Rightarrow$
  $$\forall \rho.\, \mathsf{unify}\{\rho \stackrel{?}{=} T\} \Rightarrow M(\Gamma, e, \rho) = \theta$$

Such proofs are important! (cf. "Nearly all Binary Searches and Mergesorts are Broken" by Joshua Bloch)

Constraints

- Problematic when calculating the type is the rule for applications:

$$\frac{\Gamma \vdash e_1 : T_1 \to T_2 \quad \Gamma \vdash e_2 : T_1}{\Gamma \vdash e_1\ e_2 : T_2}$$

- $A(\Gamma, e_1\ e_2) = ?$
- $A(\Gamma, e_1) = S_1$
- $A(\Gamma, e_2) = S_2$
- we have to check that S_1 equals $S_2 \to S_{12}$ and return S_{12} (Why not recording this as a constraint?)

Constraint System

- **Variables:**

$$C(\Gamma, x) = (T[X_1 := Y_1, \ldots, X_n := Y_n], \emptyset)$$

where $(x : \forall\{X_1, \ldots, X_n\}.T) \in \Gamma$ and the Y_i are distinct and fresh (w.r.t. Γ and T).

- **Lambdas:** calculate first

$$C((x : \forall\{\}.Y, \Gamma), e) = (T_1, C)$$

where Y is fresh (w.r.t. Γ). Then return

$$C(\Gamma, \lambda x.e) = (Y \to T_1, C)$$

Constraint System

- **Applications:** Calculate first

$$C(\Gamma, e_1) = (T_1, C_1)$$

then

$$C(\Gamma, e_2) = (T_2, C_2)$$

Finally return

$$C(\Gamma, e_1\ e_2) = (Y, C_1 \cup C_2 \cup \{T_1 \stackrel{?}{=} T_2 \rightarrow Y\})$$

where Y is fresh (w.r.t. Γ, C_1, C_2).

Constraint System

- **Lets:** Calculate first

$$C(\Gamma, e_1) = (T_1, C_1)$$

then $\mathsf{unify}(C_1) = \theta$

$$A = \mathsf{tv}(\theta(T_1)) - \mathsf{ftv}(\theta(\Gamma))$$

and then

$$C((x : \forall A.\theta(T_1), \theta(\Gamma)), e_2) = (T_2, C_2)$$

Finally return

$$C(\Gamma, \mathsf{let}\, x = e_1 \,\mathsf{in}\, e_2) = (T_2, C_1 \cup C_2)$$

Constraint System

- Output: We obtain

$$C(\Gamma, e) = (T, C)$$

and then need to unify$(C) = \theta$.

Finally return $\theta(T)$.

It holds

$$\theta(\Gamma) \vdash e : \theta(T)$$

Runtime Again

- Evaluation rules for our expressions

$$\frac{}{\lambda x.x \Downarrow \lambda x.x}$$

$$\frac{e_1 \Downarrow \lambda x.e' \quad e_2 \Downarrow v' \quad e'[x := v'] \Downarrow v}{e_1\, e_2 \Downarrow v}$$

$$\frac{e_1 \Downarrow v' \quad e_2[x := v'] \Downarrow v}{\mathsf{let}\ x = e_1\ \mathsf{in}\ e_2 \Downarrow v}$$

Failures & Exceptions

- Sometimes Functions need to indicate that they fail and have to handle failure.

$$
\begin{array}{rcll}
e & ::= & \ldots & \\
 & | & \mathsf{error} & \text{error value} \\
 & | & \mathsf{try}\ e_1\ \mathsf{with}\ e_2 & \text{error handling}
\end{array}
$$

$$
\frac{\mathsf{valid}\ \Gamma}{\Gamma \vdash \mathsf{error} : T}
\qquad
\frac{\Gamma \vdash e_1 : T \quad \Gamma \vdash e_2 : T}{\Gamma \vdash \mathsf{try}\ e_1\ \mathsf{with}\ e_2 : T}
$$

Failures & Exceptions

- Evaluation rules:

$$\frac{e_1 \Downarrow \mathsf{error}}{e_1\ e_2 \Downarrow \mathsf{error}} \qquad \frac{e_2 \Downarrow \mathsf{error}}{e_1\ e_2 \Downarrow \mathsf{error}}$$

$$\frac{e_1 \Downarrow v}{\mathsf{try}\ e_1\ \mathsf{with}\ e_2 \Downarrow v}$$

$$\frac{e_1 \Downarrow \mathsf{error} \qquad e_2 \Downarrow v}{\mathsf{try}\ e_1\ \mathsf{with}\ e_2 \Downarrow v}$$

Failures & Exceptions

- Evaluation rules:

$$\frac{e_1 \Downarrow \mathsf{error}}{e_1\ e_2 \Downarrow \mathsf{error}} \qquad \frac{e_2 \Downarrow \mathsf{error}}{e_1\ e_2 \Downarrow \mathsf{error}}$$

$$\frac{e_1 \Downarrow v}{\mathsf{try}\ e_1\ \mathsf{with}\ e_2 \Downarrow v}$$

$$\frac{e_1 \Downarrow \mathsf{error} \qquad e_2 \Downarrow v}{\mathsf{try}\ e_1\ \mathsf{with}\ e_2 \Downarrow v}$$

- error causes a problem with type-checking, since terms might have incomparable principal type-schemes. Solution: subtyping (next week).

Possible Question

- What does it mean to say that an ML type scheme S generalises a type T? Writing $S \succ T$ for this relation and defning

$$S_1 = \forall\{X,Y\}.X \to Y \qquad S_2 = \forall\{X\}.X \to Y$$
$$T_1 = (X \to Y) \to X \qquad T_2 = (Y \to X) \to Y$$

say whether or not $S_i \succ T_j$ holds for each of the four possibilities.

Inductions Again

- Given the language:

$$
\begin{array}{rcll}
e & ::= & x & \text{variables} \\
& | & e\ e & \text{applications} \\
& | & \lambda x.e & \text{lambda-abstractions} \\
& | & \mathsf{let}\ x = e\ \mathsf{in}\ e & \text{lets}
\end{array}
$$

- and the swapping operation:

$$
(x\,y)\bullet z \stackrel{\mathsf{def}}{=} \begin{cases} y & \text{if } z = x \\ x & \text{if } z = y \\ z & \text{o'wise} \end{cases}
$$

$$
\begin{array}{rcl}
(x\,y)\bullet(e_1\,e_2) & \stackrel{\mathsf{def}}{=} & ((x\,y)\bullet e_1)\ ((x\,y)\bullet e_2) \\
(x\,y)\bullet(\lambda z.e) & \stackrel{\mathsf{def}}{=} & \lambda((x\,y)\bullet z).((x\,y)\bullet e) \\
(x\,y)\bullet(\mathsf{let}\ z = e_1\ \mathsf{in}\ e_2) & \stackrel{\mathsf{def}}{=} & \mathsf{let}\ (x\,y)\bullet z = (x\,y)\bullet e_1\ \mathsf{in}\ (x\,y)\bullet e_2
\end{array}
$$

- show by structural induction that $(x\,y)\bullet(x\,y)\bullet e = e$.

More Next Week

- Slides at the end of

  http://www4.in.tum.de/lehre/vorlesungen/types/WS0607/

  There is also an appraisal form where you can complain anonymously.

- You can say whether the lecture was too easy, too quiet, too hard to follow, too chaotic and so on. You can also comment on things I should repeat.