Types in Programming Languages (3)

Christian Urban

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

Last Week

- Terms:

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

- Types:

$$
\begin{array}{rcll}
T & ::= & X & \text{type variables} \\
 & \mid & T \to T & \text{function types}
\end{array}
$$

Simple Type-System

- Variables

$$\frac{\mathsf{valid}\ \Gamma \quad (x : T) \in \Gamma}{\Gamma \vdash x : T}$$

- Applications

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

- Lambdas

$$\frac{x : T_1, \Gamma \vdash e : T_2 \quad x \notin \mathsf{dom}\,\Gamma}{\Gamma \vdash \lambda x.e : T_1 \to T_2}$$

- Lets

$$\frac{\Gamma \vdash e_1 : T_1 \quad x : T_1, \Gamma \vdash e_2 : T_2 \quad x \notin \mathsf{dom}\,\Gamma}{\Gamma \vdash \mathsf{let}\, x = e_1\, \mathsf{in}\, e_2 : T_2}$$

A Problem

- $\lambda y.\lambda x.x$:

$$
\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\\ \mathsf{valid}\,\{x\!:\!T_2, y\!:\!T_1\}\end{array} \quad (x\!:\!T_2) \in \{x\!:\!T_2, y\!:\!T_1\}}{\{x\!:\!T_2, y\!:\!T_1\} \vdash x : T_2} \qquad x \notin \mathsf{dom}\,\{y\!:\!T_1\}}{\{y\!:\!T_1\} \vdash \lambda x.x : T_2 \to T_2} \qquad y \notin \mathsf{dom}\,\varnothing}{\varnothing \vdash \lambda y.\lambda x.x : T_1 \to T_2 \to T_2}
$$

A Problem

- $\lambda y.\lambda x.x$:

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots \\ \mathsf{valid}\,\{x\!:\!T_2, y\!:\!T_1\} \\ (x\!:\!T_2) \in \{x\!:\!T_2, y\!:\!T_1\}\end{array}}{\{x\!:\!T_2, y\!:\!T_1\} \vdash x : T_2} \quad x \notin \mathsf{dom}\,\{y\!:\!T_1\}}{\{y\!:\!T_1\} \vdash \lambda x.x : T_2 \to T_2} \quad y \notin \mathsf{dom}\,\varnothing}{\varnothing \vdash \lambda y.\lambda x.x : T_1 \to T_2 \to T_2}$$

- $\lambda x.\lambda x.x$:

$$\dfrac{\dfrac{\{x\!:\!T_2, x\!:\!T_1\} \vdash x : T_2 \quad \color{red}{x \notin \mathsf{dom}\,\{x\!:\!T_1\}}}{\{x\!:\!T_1\} \vdash \lambda x.x : T_2 \to T_2} \quad y \notin \mathsf{dom}\,\varnothing}{\varnothing \vdash \lambda x.\lambda x.x : T_1 \to T_2 \to T_2}$$

Alpha-Equivalence

$$\frac{}{x \approx x} \qquad \frac{e_1 \approx s_1 \quad e_2 \approx s_2}{e_1\, e_2 \approx s_1\, s_2}$$

$$\frac{e \approx s}{\lambda x.e \approx \lambda x.s} \qquad \frac{x \neq y \quad e \approx (x\, y)\bullet s \quad x \notin \mathsf{fv}(s)}{\lambda x.e \approx \lambda y.s}$$

$$\frac{e_1 \approx s_1 \quad e_2 \approx s_2}{\mathsf{let}\, x = e_1\, \mathsf{in}\, e_2 \approx \mathsf{let}\, x = s_1\, \mathsf{in}\, s_2}$$

$$\frac{x \neq y \quad e_1 \approx s_1 \quad e_2 \approx (x\, y)\bullet s_2 \quad x \notin \mathsf{fv}(s_2)}{\mathsf{let}\, x = e_1\, \mathsf{in}\, e_2 \approx \mathsf{let}\, y = s_1\, \mathsf{in}\, s_2}$$

Swapping

- $$\begin{array}{lcll} e & ::= & x & \text{variables} \\ & \mid & e\ e & \text{applications} \\ & \mid & \lambda x.e & \text{lambda-abstractions} \\ & \mid & \mathsf{let}\ x = e\ \mathsf{in}\ e & \text{lets} \end{array}$$
- A swapping operation:

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

$$(x\,y)\bullet(e_1\,e_2) \stackrel{\text{def}}{=} ((x\,y)\bullet e_1)\ ((x\,y)\bullet e_2)$$

$$(x\,y)\bullet(\lambda z.e) \stackrel{\text{def}}{=} \lambda((x\,y)\bullet z).((x\,y)\bullet e)$$

$$(x\,y)\bullet(\mathsf{let}\, z = e_1 \,\mathsf{in}\, e_2) \stackrel{\text{def}}{=} \mathsf{let}\,(x\,y)\bullet z = (x\,y)\bullet e_1 \,\mathsf{in}\,(x\,y)\bullet e_2$$

Swapping

- $e \quad ::= \quad x$ variables
  $\quad\quad\mid\quad e\ e$ applications
  $\quad\quad\mid\quad \lambda x.e$ lambda-abstractions
  $\quad\quad\mid\quad \mathsf{let}\ x = e\ \mathsf{in}\ e$ lets

- A swapp We have $(x\,y)\bullet(x\,y)\bullet e = e$.

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

$$(x\,y)\bullet(e_1\,e_2) \stackrel{\mathsf{def}}{=} ((x\,y)\bullet e_1)\ ((x\,y)\bullet e_2)$$

$$(x\,y)\bullet(\lambda z.e) \stackrel{\mathsf{def}}{=} \lambda((x\,y)\bullet z).((x\,y)\bullet e)$$

$$(x\,y)\bullet(\mathsf{let}\ z = e_1\ \mathsf{in}\ e_2) \stackrel{\mathsf{def}}{=} \mathsf{let}\ (x\,y)\bullet z = (x\,y)\bullet e_1\ \mathsf{in}\ (x\,y)\bullet e_2$$

A Property

- Weakening Lemma

  If $\Gamma_1 \vdash e : T$ and valid Γ_2 and $\Gamma_1 \subseteq \Gamma_2$, then $\Gamma_2 \vdash e : T$.

- By induction over $\Gamma_1 \vdash e : T$.

$$P\,\Gamma_1\,e\,T = \forall \Gamma_2.\ \text{valid}\,\Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash e : T$$

In the literature the proof of this lemma is often labelled as “trivial”, “obvious” etc.

Weakening

$$\frac{x : T_1, \Gamma \vdash M : T_2 \quad x \notin \mathsf{dom}\,\Gamma}{\Gamma \vdash \lambda x.M : T_1 \to T_2}$$

- If $\Gamma_1 \vdash M : T_2$ thn $\forall \Gamma_2.\ \mathsf{valid}\ \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

- We know:
  $\forall \Gamma_2.\ \mathsf{valid}\ \Gamma_2 \ \wedge\ x : T_1, \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$
  $x \notin \mathsf{dom}\ \Gamma_1$

- We have to show:
  $\forall \Gamma_2.\ \mathsf{valid}\ \Gamma_2 \ \wedge\ \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash \lambda x.M : T_1 \to T_2$

Weakening

$$\frac{x:T_1, \Gamma \vdash M:T_2 \quad x \notin \mathsf{dom}\,\Gamma}{\Gamma \vdash \lambda x.M:T_1 \to T_2}$$

- If $\Gamma_1 \vdash M:T_2$ thn $\forall \Gamma_2.\,\mathsf{valid}\,\Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M:T_2$

For all Γ_1, x, M, T_1 and T_2:

- We know:

  $\forall \Gamma_2.\,\mathsf{valid}\,\Gamma_2 \;\wedge\; x:T_1, \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M:T_2$

  $x \notin \mathsf{dom}\,\Gamma_1$

- We have to show:

  $\forall \Gamma_2.\,\mathsf{valid}\,\Gamma_2 \;\wedge\; \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash \lambda x.M:T_1 \to T_2$

Weakening

$$\frac{x : T_1, \Gamma \vdash M : T_2 \quad x \notin \mathrm{dom}\,\Gamma}{\Gamma \vdash \lambda x.M : T_1 \to T_2}$$

- If $\Gamma_1 \vdash M : T_2$ thn $\forall \Gamma_2.\,\mathrm{valid}\,\Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

For all Γ_1, x, M, T_1 and T_2:

- We know:

  $\forall \Gamma_2.\,\mathrm{valid}\,\Gamma_2 \;\wedge\; x : T_1, \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

  $x \notin \mathrm{dom}\,\Gamma_1$

  $\mathrm{valid}\,\Gamma_2 \;\wedge\; \Gamma_1 \subseteq \Gamma_2$

- We have to show:

  $\Gamma_2 \vdash \lambda x.M : T_1 \to T_2$

Weakening

$$\frac{x : T_1, \Gamma \vdash M : T_2 \quad x \notin \mathsf{dom}\, \Gamma}{\Gamma \vdash \lambda x.M : T_1 \to T_2}$$

- If $\Gamma_1 \vdash M : T_2$ thn $\forall \Gamma_2.\, \mathsf{valid}\, \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

For all Γ_1, x, M, T_1 and T_2:

- We know:

  $\forall \Gamma_2.\, \mathsf{valid}\, \Gamma_2 \;\wedge\; x : T_1, \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

  $x \notin \mathsf{dom}\, \Gamma_1$

  $\mathsf{valid}\, \Gamma_2 \;\wedge\; \Gamma_1 \subseteq \Gamma_2$

- We have to show:

  $\Gamma_2 \vdash \lambda x.M : T_1 \to T_2$

Weakening

$$\frac{x : T_1, \Gamma \vdash M : T_2 \quad x \notin \mathsf{dom}\,\Gamma}{\Gamma \vdash \lambda x.M : T_1 \to T_2}$$

- If $\Gamma_1 \vdash M : T_2$ thn $\forall \Gamma_2.\, \mathsf{valid}\,\Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

For all Γ_1, x, M, T_1 and T_2:

instantiate $\Gamma_2 \mapsto x : T_1, \Gamma_2$

- We know:

  $\forall \Gamma_2.\, \mathsf{valid}\,\Gamma_2 \;\wedge\; x : T_1, \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

  $x \notin \mathsf{dom}\,\Gamma_1$

  $\mathsf{valid}\,\Gamma_2 \;\wedge\; \Gamma_1 \subseteq \Gamma_2$

- We have to show:

  $\Gamma_2 \vdash \lambda x.M : T_1 \to T_2$

Weakening

$$\frac{x : T_1, \Gamma \vdash M : T_2 \quad x \notin \mathsf{dom}\,\Gamma}{\Gamma \vdash \lambda x.M : T_1 \to T_2}$$

- If $\Gamma_1 \vdash M : T_2$ thn $\forall \Gamma_2.\ \mathsf{valid}\ \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

For all Γ_1, x, M, T_1 and T_2:

instantiate $\Gamma_2 \mapsto x : T_1, \Gamma_2$

- We know:

$\forall \Gamma_2.\ \mathsf{valid}\ \Gamma_2 \ \wedge\ x : T_1, \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

$x \notin \mathsf{dom}\ \Gamma_1$

$\mathsf{valid}\ \Gamma_2 \ \wedge\ \Gamma_1 \subseteq \Gamma_2 \ \Rightarrow x : T_1, \Gamma_1 \subseteq x : T_1, \Gamma_2$

- We have to show:

$\Gamma_2 \vdash \lambda x.M : T_1 \to T_2$

Weakening

$$\frac{x : T_1, \Gamma \vdash M : T_2 \quad x \notin \mathrm{dom}\,\Gamma}{\Gamma \vdash \lambda x.M : T_1 \to T_2}$$

- If $\Gamma_1 \vdash M : T_2$ thn $\forall \Gamma_2.\ \text{valid}\ \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

For all Γ_1, x, M, T_1 and T_2:

instantiate $\Gamma_2 \mapsto x : T_1, \Gamma_2$

- We know:

  $\forall \Gamma_2.\ \text{valid}\ \Gamma_2 \ \wedge\ x : T_1, \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

  $x \notin \mathrm{dom}\,\Gamma_1$

  $\text{valid}\ \Gamma_2 \ \wedge\ \Gamma_1 \subseteq \Gamma_2 \ \Rightarrow x : T_1, \Gamma_1 \subseteq x : T_1, \Gamma_2$

  valid $(x : T_1, \Gamma_2)$???

- We have to show:

  $\Gamma_2 \vdash \lambda x.M : T_1 \to T_2$

Weakening

$$\frac{x : T_1, \Gamma \vdash M : T_2 \quad x \notin \mathrm{dom}\,\Gamma}{\Gamma \vdash \lambda x.M : T_1 \to T_2}$$

- If $\Gamma_1 \vdash M : T_2$ thn $\forall \Gamma_2.\ \mathrm{valid}\ \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

For all Γ_1, x, M, T_1 and T_2:

instantiate $\Gamma_2 \mapsto x : T_1, \Gamma_2$

- We know:

  $\forall \Gamma_2.\ \mathrm{valid}\ \Gamma_2 \ \wedge\ x : T_1, \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

  $x \notin \mathrm{dom}\ \Gamma_1$

  $\mathrm{valid}\ \Gamma_2 \ \wedge\ \Gamma_1 \subseteq \Gamma_2 \ \Rightarrow x : T_1, \Gamma_1 \subseteq x : T_1, \Gamma_2$

  valid $(x : T_1, \Gamma_2)$???

  $x \notin \mathrm{dom}\ \Gamma_2$???

- We have to show:

  $\Gamma_2 \vdash \lambda x.M : T_1 \to T_2$

A Proof that Works I

- Extend swapping to contexts

$$(x\,y)\bullet\Gamma \stackrel{\mathsf{def}}{=} \{((x\,y)\bullet z : T) \mid (z : T) \in \Gamma\}$$

- We have valid $(x\,y)\bullet\Gamma$ iff valid Γ!!
  - If valid Γ then valid $(x\,y)\bullet\Gamma$, and
  - if valid $(x\,y)\bullet\Gamma$ then valid Γ.

Note this holds for swappings, but not for substitutions!

A Proof that Works II

- We can show for every x and y

$$\Gamma \vdash e : T \Rightarrow (x\,y) \bullet \Gamma \vdash (x\,y) \bullet e : T$$

and

$$(x\,y) \bullet \Gamma \vdash (x\,y) \bullet e : T \Rightarrow \Gamma \vdash e : T$$

The first is by induction on the definition of typing and the second follows from $(x\,y) \bullet (x\,y) \bullet \Gamma = \Gamma$ and $(x\,y) \bullet (x\,y) \bullet e = e$.

A Proof that Works III

- We can also show:

$$P\,\Gamma\,e\,T \Rightarrow P\,(x\,y)\bullet\Gamma\,(x\,y)\bullet e\,T$$

holds for all Γ, e and T.

- We have:

$$\forall\Gamma_2.\ \mathsf{valid}\,\Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash e : T$$

- We have to show:

$$\forall\Gamma_2.\ \mathsf{valid}\,\Gamma_2 \wedge (x\,y)\bullet\Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash (x\,y)\bullet e : T$$

A Proof that Works III

- We can also show:

$$P\,\Gamma\,e\,T \Rightarrow P\,(x\;y)\bullet\Gamma\,(x\;y)\bullet e\,T$$

holds for all Γ, e and T.

- We have:

$$\forall\Gamma_2.\ \mathsf{valid}\,\Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash e : T$$
$$\mathsf{valid}\,\Gamma_2 \qquad (x\;y)\bullet\Gamma_1 \subseteq \Gamma_2$$

- We have to show:

$$\Gamma_2 \vdash (x\;y)\bullet e : T$$

A Proof that Works III

- We can also show:

$$P\,\Gamma\,e\,T \Rightarrow P\,(x\,y)\bullet\Gamma\,(x\,y)\bullet e\,T$$

holds for all Γ, e and T.

- We have:

$$\forall\Gamma_2.\ \mathsf{valid}\,\Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash e : T$$
$$\mathsf{valid}\,\Gamma_2 \qquad (x\,y)\bullet\Gamma_1 \subseteq \Gamma_2$$

- We have to show:

$$(x\,y)\bullet\Gamma_2 \vdash (x\,y)\bullet(x\,y)\bullet e : T$$

A Proof that Works III

- We can also show:

$$P\,\Gamma\,e\,T \Rightarrow P\,(x\,y)\bullet\Gamma\,(x\,y)\bullet e\,T$$

holds for all Γ, e and T.

- We have:

$$\forall\Gamma_2.\mathsf{valid}\,\Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash e : T$$
$$\mathsf{valid}\,\Gamma_2 \qquad (x\,y)\bullet\Gamma_1 \subseteq \Gamma_2$$

- We have to show:

$$(x\,y)\bullet\Gamma_2 \vdash e : T$$

A Proof that Works III

- We can also show:

$$P\,\Gamma\,e\,T \Rightarrow P\,(x\,y)\bullet\Gamma\,(x\,y)\bullet e\,T$$

holds for all Γ, e and T.

- We have:

$$\mathsf{valid}\,(x\,y)\bullet\Gamma_2 \wedge \Gamma_1 \subseteq (x\,y)\bullet\Gamma_2 \Rightarrow (x\,y)\bullet\Gamma_2 \vdash e : T$$

$$\mathsf{valid}\,\Gamma_2 \qquad (x\,y)\bullet\Gamma_1 \subseteq \Gamma_2$$

- We have to show:

$$(x\,y)\bullet\Gamma_2 \vdash e : T$$

A Proof that Works III

- We can also show:

$$P\,\Gamma\,e\,T \Rightarrow P\,(x\,y)\bullet\Gamma\,(x\,y)\bullet e\,T$$

holds for all Γ, e and T.

- We have:

$$\text{valid}\,(x\,y)\bullet\Gamma_2 \wedge \Gamma_1 \subseteq (x\,y)\bullet\Gamma_2 \Rightarrow (x\,y)\bullet\Gamma_2 \vdash e : T$$

$$\text{valid}\,(x\,y)\bullet\Gamma_2 \qquad (x\,y)\bullet\Gamma_1 \subseteq \Gamma_2$$

- We have to show:

$$(x\,y)\bullet\Gamma_2 \vdash e : T$$

A Proof that Works III

- We can also show:

$$P\,\Gamma\,e\,T \Rightarrow P\,(x\;y)\bullet\Gamma\,(x\;y)\bullet e\,T$$

  holds for all Γ, e and T.

- We have:

$$\text{valid}\,(x\;y)\bullet\Gamma_2 \wedge \Gamma_1 \subseteq (x\;y)\bullet\Gamma_2 \Rightarrow (x\;y)\bullet\Gamma_2 \vdash e:T$$
$$\text{valid}\,(x\;y)\bullet\Gamma_2 \qquad (x\;y)\bullet(x\;y)\bullet\Gamma_1 \subseteq (x\;y)\bullet\Gamma_2$$

- We have to show:

$$(x\;y)\bullet\Gamma_2 \vdash e:T$$

A Proof that Works III

- We can also show:

$$P\;\Gamma\;e\;T \Rightarrow P\;(x\;y)\bullet\Gamma\;(x\;y)\bullet e\;T$$

  holds for all Γ, e and T.

- We have:

$$\text{valid}\;(x\;y)\bullet\Gamma_2 \wedge \Gamma_1 \subseteq (x\;y)\bullet\Gamma_2 \Rightarrow (x\;y)\bullet\Gamma_2 \vdash e : T$$
$$\text{valid}\;(x\;y)\bullet\Gamma_2 \qquad \Gamma_1 \subseteq (x\;y)\bullet\Gamma_2$$

- We have to show:

$$(x\;y)\bullet\Gamma_2 \vdash e : T$$

A Proof that Works III

- We can also show:

$$P\,\Gamma\,e\,T \Rightarrow P\,(x\,y)\bullet\Gamma\,(x\,y)\bullet e\,T$$

  holds for all Γ, e and T.

- We have:

$$(x\,y)\bullet\Gamma_2 \vdash e : T$$
$$\text{valid}\,(x\,y)\bullet\Gamma_2 \qquad \Gamma_1 \subseteq (x\,y)\bullet\Gamma_2$$

- We have to show:

$$(x\,y)\bullet\Gamma_2 \vdash e : T$$

Lambda Case Again

$$\frac{x : T_1, \Gamma \vdash M : T_2 \quad x \notin \mathrm{dom}\,\Gamma}{\Gamma \vdash \lambda x.M : T_1 \to T_2}$$

- If $\Gamma_1 \vdash M : T_2$ thn $\forall \Gamma_2.\, \mathrm{valid}\,\Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

For all Γ_1, x, M, T_1 and T_2:

- We know:

  $\forall \Gamma_2.\, \mathrm{valid}\,\Gamma_2 \;\wedge\; x : T_1, \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$
  $x \notin \mathrm{dom}\,\Gamma_1$
  $\mathrm{valid}\,\Gamma_2 \;\wedge\; \Gamma_1 \subseteq \Gamma_2$

- We have to show:

  $\Gamma_2 \vdash \lambda x.M : T_1 \to T_2$

Lambda Case Again

$$\frac{x : T_1, \Gamma \vdash M : T_2 \quad x \notin \mathrm{dom}\,\Gamma}{\Gamma \vdash \lambda x.M : T_1 \to T_2}$$

- If $\Gamma_1 \vdash M : T_2$ thn $\forall \Gamma_2.\, \mathrm{valid}\,\Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

For all Γ_1, x, M, T_1 and T_2:

- We know:

  $\forall \Gamma_2.\, \mathrm{valid}\,\Gamma_2 \;\wedge\; x : T_1, \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

  $x \notin \mathrm{dom}\,\Gamma_1$

  $\mathrm{valid}\,\Gamma_2 \;\wedge\; \Gamma_1 \subseteq \Gamma_2$

  choose a y s. t. $y \neq x$, $y \notin \mathrm{dom}\,\Gamma_1, \mathrm{dom}\,\Gamma_2$, $y \notin \mathrm{fv}(M)$

- We have to show:

  $\Gamma_2 \vdash \lambda x.M : T_1 \to T_2$

Lambda Case Again

$$\frac{x : T_1, \Gamma \vdash M : T_2 \quad x \notin \mathrm{dom}\,\Gamma}{\Gamma \vdash \lambda x.M : T_1 \to T_2}$$

- If $\Gamma_1 \vdash M : T_2$ thn $\forall \Gamma_2.\, \mathrm{valid}\,\Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

For all Γ_1, x, M, T_1 and T_2:

- We know:

  $\forall \Gamma_2.\, \mathrm{valid}\,\Gamma_2 \;\wedge\; x : T_1, \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

  $x \notin \mathrm{dom}\,\Gamma_1$

  $\mathrm{valid}\,\Gamma_2 \;\wedge\; \Gamma_1 \subseteq \Gamma_2$

  choose a y s. t. $y \neq x$, $y \notin \mathrm{dom}\,\Gamma_1, \mathrm{dom}\,\Gamma_2$, $y \notin \mathrm{fv}(M)$

- We have to show:

  $\Gamma_2 \vdash \lambda y.(x\,y) \bullet M : T_1 \to T_2$

Lambda Case Again

$$\frac{x : T_1, \Gamma \vdash M : T_2 \quad x \notin \mathsf{dom}\,\Gamma}{\Gamma \vdash \lambda x.M : T_1 \to T_2}$$

- If $\Gamma_1 \vdash M : T_2$ thn $\forall \Gamma_2.\,\mathsf{valid}\,\Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

For all Γ_1, x, M, T_1 and T_2:

- We know:

  $\forall \Gamma_2.\,\mathsf{valid}\,\Gamma_2 \;\wedge\; x : T_1, \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

  $x \notin \mathsf{dom}\,\Gamma_1$

  $\mathsf{valid}\,\Gamma_2 \;\wedge\; \Gamma_1 \subseteq \Gamma_2$

  choose a y s. t. $y \neq x$, $y \notin \mathsf{dom}\,\Gamma_1, \mathsf{dom}\,\Gamma_2$, $y \notin \mathsf{fv}(M)$

- We have to show:

  $y : T_1, \Gamma_2 \vdash (x\,y) \bullet M : T_2 \qquad y \notin \mathsf{dom}\,\Gamma_2$

Lambda Case Again

$$\frac{x\!:\!T_1, \Gamma \vdash M\!:\!T_2 \quad x \notin \mathrm{dom}\,\Gamma}{\Gamma \vdash \lambda x.M\!:\!T_1 \to T_2}$$

- If $\Gamma_1 \vdash M\!:\!T_2$ thn $\forall \Gamma_2.\ \mathrm{valid}\,\Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M\!:\!T_2$

For all Γ_1, x, M, T_1 and T_2:

- We know:

  $\forall \Gamma_2.\ \mathrm{valid}\,\Gamma_2 \ \wedge\ x\!:\!T_1, \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M\!:\!T_2$

  $x \notin \mathrm{dom}\,\Gamma_1$

  $\mathrm{valid}\,\Gamma_2 \ \wedge\ \Gamma_1 \subseteq \Gamma_2$

  choose a y s. t. $y \neq x$, $y \notin \mathrm{dom}\,\Gamma_1, \mathrm{dom}\,\Gamma_2$, $y \notin \mathrm{fv}(M)$

- We have to show:

  $y\!:\!T_1, \Gamma_2 \vdash (x\ y) \bullet M\!:\!T_2$

Lambda Case Again

$$\frac{x:T_1,\Gamma \vdash M:T_2 \quad x \notin \mathrm{dom}\,\Gamma}{\Gamma \vdash \lambda x.M : T_1 \rightarrow T_2}$$

- If $\Gamma_1 \vdash M:T_2$ thn $\forall \Gamma_2.\ \mathrm{valid}\ \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M:T_2$

For all Γ_1, x, M, T_1 and T_2:

- We know:

  $\forall \Gamma_2.\ \mathrm{valid}\ \Gamma_2 \wedge y:T_1,\Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash (x\ y) \bullet M : T_2$

  $x \notin \mathrm{dom}\,\Gamma_1$

  $\mathrm{valid}\ \Gamma_2 \ \wedge\ \Gamma_1 \subseteq \Gamma_2$

  choose a y s. t. $y \neq x$, $y \notin \mathrm{dom}\,\Gamma_1$, $\mathrm{dom}\,\Gamma_2$, $y \notin \mathrm{fv}(M)$

- We have to show:

  $y:T_1,\Gamma_2 \vdash (x\ y) \bullet M : T_2$

Lambda Case Again

$$\frac{x : T_1, \Gamma \vdash M : T_2 \quad x \notin \mathrm{dom}\,\Gamma}{\Gamma \vdash \lambda x.M : T_1 \to T_2}$$

- If $\Gamma_1 \vdash M : T_2$ thn $\forall \Gamma_2.$ valid $\Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : T_2$

For all Γ_1, x, M, T_1 and T_2:

- We know:

valid $(y : T_1, \Gamma_2) \wedge y : T_1, \Gamma_1 \subseteq y : T_1, \Gamma_2 \Rightarrow y : T_1, \Gamma_2 \vdash (x\,y) \bullet M : T_2$

$x \notin \mathrm{dom}\,\Gamma_1$

valid $\Gamma_2 \quad \wedge \quad \Gamma_1 \subseteq \Gamma_2$

choose a y s. t. $y \neq x$, $y \notin \mathrm{dom}\,\Gamma_1, \mathrm{dom}\,\Gamma_2$, $y \notin \mathrm{fv}(M)$

- We have to show:

$y : T_1, \Gamma_2 \vdash (x\,y) \bullet M : T_2$

Polymorphism

- Given the typing rule

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

- what can we say about the type of

$$g\ (f\ a)\ (f\ b)?$$

Polymorphism

- Given the typing rule

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

- what can we say about the type of

$$g\ (f\ a)\ (f\ b)?$$

Every expression has a unique type. This is an irksome limitation. Solution...

Polymorphism

- Given the typing rule

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

- wh

Parametric Polymorphism - same expression belongs to a family of types (in Java and C#, this concept is called **generics**).

Every expression has a unique type. This is an irksome limitation. Solution. . .

Type-Schemes

- So far we have used types of the form:

$$\begin{array}{rcll} T & ::= & X & \text{type variables} \\ & | & T \to T & \text{function types} \end{array}$$

- We now introduce type-schemes:

$$S \quad ::= \quad \forall A.T$$

Where A ranges over a finite set of type-variables.
When $A = \{X_1, \ldots, X_n\}$ we write $\forall A.T$ as

$$\forall\{X_1, \ldots, X_n\}.T$$

Type-Schemes

- So far we have used types of the form:

$$\begin{array}{rcll} T & ::= & X & \text{type variables} \\ & | & T \to T & \text{function types} \end{array}$$

- We now introduce type-schemes:

$$S \quad ::= \quad \forall A.T$$

Where A ranges over a finite set of type-variables. When $A = \{X_1, \ldots, X_n\}$ we write $\forall A.T$ as

$$\forall\{X_1, \ldots, X_n\}.T$$

$\forall\{\}.T$ is possible; $\forall A.\forall B.T$ is not. Note that type-schemes are not types!

Alpha-Equivalence

- We also only interested in alpha-equivalence classes of type-schemes, that is $[\forall A.T]_\alpha$:

$$
\begin{aligned}
\mathsf{tv}(X) &\stackrel{\mathsf{def}}{=} \{X\} \\
\mathsf{tv}(T_1 \to T_2) &\stackrel{\mathsf{def}}{=} \mathsf{tv}(T_1) \cup \mathsf{tv}(T_2) \\
\mathsf{ftv}(\forall A.T) &\stackrel{\mathsf{def}}{=} \mathsf{tv}(T) - A
\end{aligned}
$$

$$
(X\,Y) \bullet Z \stackrel{\mathsf{def}}{=} \begin{cases} Y & \mathsf{if}\ Z = X \\ X & \mathsf{if}\ Z = Y \\ Z & \mathsf{o'wise} \end{cases}
$$

$$
(X\,Y) \bullet (T_1 \to T_2) \stackrel{\mathsf{def}}{=} (X\,Y) \bullet T_1 \to (X\,Y) \bullet T_2
$$

Alpha Equivalence

$$\frac{}{\forall\{\}.T \approx \forall\{\}.T}$$

$$\frac{\forall\{X_2,\ldots\}.T \approx \forall\{Y_2,\ldots\}.T'}{\forall\{X_1, X_2,\ldots\}.T \approx \forall\{X_1, Y_2,\ldots\}.T'}$$

$$\frac{X_1 \neq Y_1 \quad X_1 \notin \mathsf{tv}(T') \qquad \forall\{X_2,\ldots\}.T \approx \forall\{Y_2,\ldots\}.(X_1\,Y_1)\bullet T'}{\forall\{X_1, X_2,\ldots\}.T \approx \forall\{Y_1, Y_2,\ldots\}.T'}$$

For example:

$$\forall\{X\}.Y \to X \approx \forall\{X'\}.Y \to X'$$

Alpha-equivalence classes of type-schemes:

$$[\forall A.T]_\alpha \stackrel{\text{def}}{=} \{S \mid S \approx \forall A.T\}$$

More General

- Def: We say that a type-scheme $S = \forall\{X_1, \ldots, X_n\}.T'$ generalises a type T, written $S \succ T$ if T can be obtained from T' by substituting for the type-variables X_i types T_i. That is

$$T = T'[X_1 := T_1, \ldots, X_n := T_n]$$

- Examples:

$$\begin{array}{lcl}
\forall\{X\}.(X \to X) & \succ & X \to X \\
\forall\{X\}.(X \to X) & \succ & (T_1 \to T_2) \to (T_1 \to T_2) \\
\forall\{X\}.(Y \to X) & \succ & Y \to T \\
\\
\forall\{X\}.(X \to X) & \not\succ & (X \to X) \to X
\end{array}$$

Typing-Judgement

$$\Gamma \vdash e : T$$

So far we had:

- Γ is a set of (variable,type)-pairs
- e is an expression (alpha-equivalence class)
- T is a type

Typing-Judgement

$$\Gamma \vdash e : T$$

Now we have:

- Γ is a set of (variable,type-scheme)-pairs
- e is an expression (alpha-equivalence class)
- T is a type

ML Type-System

- Variables

$$\frac{\mathsf{valid}\ \Gamma \quad (x : S) \in \Gamma \quad S \succ T}{\Gamma \vdash x : T}$$

- Applications

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

- Lambdas

$$\frac{x : \forall\{\}.T_1, \Gamma \vdash e : T_2 \quad x \not\in \mathsf{dom}\,\Gamma}{\Gamma \vdash \lambda x.e : T_1 \to T_2}$$

- Lets

$$\frac{\Gamma \vdash e_1 : T_1 \quad x : \forall A.T_1, \Gamma \vdash e_2 : T_2 \quad x \not\in \mathsf{dom}\,\Gamma}{\Gamma \vdash \mathsf{let}\, x = e_1\, \mathsf{in}\, e_2 : T_2}$$

ML Type-System

- Variables

$$\frac{\text{valid } \Gamma \quad (x : S) \in \Gamma \quad S \succ T}{\Gamma \vdash x : T}$$

- Applications

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

- Lambdas

$$\frac{x : \forall\{\}.T_1, \Gamma \ldots}{\Gamma \vdash \lambda x.e : T_1 \to T_2}$$

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

- Lets

$$\frac{\Gamma \vdash e_1 : T_1 \quad x : \forall A.T_1, \Gamma \vdash e_2 : T_2 \quad x \notin \text{dom } \Gamma}{\Gamma \vdash \text{let } x = e_1 \text{ in } e_2 : T_2}$$

Example

- $$\Gamma = \left\{ \begin{array}{l} g : \forall\{\}.\boldsymbol{X}_1 \rightarrow \boldsymbol{X}_2 \rightarrow \boldsymbol{X}_3, \\ a : \forall\{\}.\boldsymbol{X}_1, \\ b : \forall\{\}.\boldsymbol{X}_2 \end{array} \right\}$$

- $\Gamma \vdash \mathsf{let}\ f = \lambda x.x\ \mathsf{in}\ g\ (f\ a)\ (f\ b) : \boldsymbol{X}_3$

  f is the identity function, once applied to an argument of type $\boldsymbol{X}_1$ and once to a type $\boldsymbol{X}_2$

Example

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

- $\Gamma \vdash \text{let } f = \lambda x.x \text{ in } g\ (f\ a)\ (f\ b) : X_3$

  f is the identity function, once applied to an argument of type X_1 and once to a type X_2

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

Typing Problem

- Given a Γ and a e, can we find a T such that

$$\Gamma \vdash e : T$$

(if there exists such a T)

- It turns out that: if there exists one such T then in general there might be many. For example

$$\varnothing \vdash \lambda x.x : \ldots$$

- However, if there exists such a type at all, then there will be always a most general type-scheme from which all types can be derived.

Unification

- Give two types, say T_1 and T_2. They contain some type-variables. Can one find a substitution θ, which substitutes types for type-variables, such that T_1 and T_2 become equal. That is

$$\theta(T_1) = \theta(T_2)$$

  where θ is called a unifier.
- For example

$$\begin{array}{ll}
X\,,\,X & \Rightarrow \theta = \varepsilon = [\,] \quad \text{(identity subst.)}\\
X \rightarrow Y\,,\,Z & \Rightarrow \theta = [Z := X \rightarrow Y]\\
X \rightarrow Y\,,\,X \rightarrow Z & \Rightarrow \theta = [Y := Z]\\
X \rightarrow Y\,,\,X \rightarrow Z & \Rightarrow \theta = [X := V, Y := V, Z := V]
\end{array}$$

  $X\,,\,X \rightarrow X$ no

MGU

- If there exists a unifier, then there exists a most general one, MGU.
- A substitution θ_1 is more general than θ_2, $\theta_1 \succ \theta_2$, provided there exists a substitution σ such that

$$\theta_2 = \sigma \circ \theta_1$$

where $\sigma \circ \theta_1 \stackrel{\text{def}}{=} \sigma \cup \sigma(\theta_1)$;

$$\sigma(\theta_1) \stackrel{\text{def}}{=} \text{apply } \sigma \text{ to the codomain of } \theta_1$$

- How to calculate the MGU for T_1 and T_2?

  Start with the unification problem $\{T_1 =^? T_2\}$

MGU

- If there exists a unifier, then there exists a most general one, MGU.

Unification algorithm:

Input: (finite) unification problem

$$\{T_1 =^? U_1, T_2 =^? U_2, \ldots\}$$

Output: either FAIL or substitution θ

If θ then $\theta(T_i) = \theta(U_i)$ and θ is an MGU.

- How to calculate the MGU for T_1 and T_2?

  Start with the unification problem $\{T_1 =^? T_2\}$

Transformations

- TVar-TVar

  $\{X =^? X\} \uplus P \stackrel{\varepsilon}{\Longrightarrow} P$

- Fun-Fun

  $\{T_1 \to T_2 =^? U_1 \to U_2\} \uplus P \stackrel{\varepsilon}{\Longrightarrow} \{T_1 =^? U_1, T_2 =^? U_2\} \cup P$

- TVar-Ty

  $\{X =^? T\} \uplus P \stackrel{[X:=T]}{\Longrightarrow} P[X := T]$

  Ty-TVar

  $\{T =^? X\} \uplus P \stackrel{[X:=T]}{\Longrightarrow} P[X := T]$

  both transformation only if $X \not\in \mathsf{tv}(T)$

- transform until you reach $\varnothing$; if stuck, no unifier

Unifier

- If $\varnothing$ is reached you have a sequence:

$$P_1 \xRightarrow{\theta_1} P_2 \xRightarrow{\theta_2} \ldots \xRightarrow{\theta_n} \varnothing$$

- The (most general) unifier for the problem P_1 is then

$$\theta = \theta_n \circ \ldots \circ \theta_2 \circ \theta_1$$

- Let's do two examples
  - $\{X \to X \to Y =^? Y \to Z \to Y\}$
  - $\{X \to X =^? Y \to Y \to Y\}$

Typing Algorithm

- Input: a context Γ and an expression e
- Output: FAIL or a substitution θ and type T

If a θ and T, then

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

Typing Algorithm

- Input: a context Γ and an expression e
- Output: FAIL or a substitution θ and type T

  If a θ and T, then

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

- Output': FAIL or a substitution θ and a type-scheme $S = \forall A.T$ where

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

we call S a principal type-scheme.

Typing Algorithm

- Input: a context Γ and an expression e
- Output: FAIL or a substitution θ and type T

  If a θ and T then

If e is closed, $\Gamma = \varnothing$ and the typing algorithm returns (θ, S), then S is called the principal type-scheme of e.

- Outp a
  type-scheme $S = \forall A.T$ where

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

we call S a principal type-scheme.

Homework

- Type into your favourite functional language:

$$\begin{array}{l}
\mathsf{let}\ pair = \lambda x.\lambda y.\lambda z.\ z\,x\,y\ \mathsf{in} \\
\quad \mathsf{let}\ x_1 = \lambda y.pair\,y\,y\ \mathsf{in} \\
\quad\quad \mathsf{let}\ x_2 = \lambda y.x_1\,(x_1\,y)\ \mathsf{in} \\
\quad\quad\quad \mathsf{let}\ x_3 = \lambda y.x_2\,(x_2\,y)\ \mathsf{in} \\
\quad\quad\quad\quad \mathsf{let}\ x_4 = \lambda y.x_3\,(x_3\,y)\ \mathsf{in} \\
\quad\quad\quad\quad\quad \mathsf{let}\ x_5 = \lambda y.x_4\,(x_4\,y)\ \mathsf{in} \\
\quad\quad\quad\quad\quad\quad x_5\,(\lambda y.y)
\end{array}$$

and let it check what the type is.

Possible Question

- 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:

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

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

- 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.