Karlsruher Institut für Technologie

Einführung in Typinferenz

Simon Bischof, Jakob von Raumer

IPD Snelting

$$\textit{Abs}\;\dfrac{\textit{Abs}\;\dfrac{\textit{App}\;\dfrac{\textit{Var}\;\dfrac{(\mathtt{f}:\alpha_2,\mathtt{x}:\alpha_4)\,(\mathtt{f})=\alpha_6}{\mathtt{f}:\alpha_2,\mathtt{x}:\alpha_4\vdash\mathtt{f}:\alpha_6}\quad\textit{Var}\;\dfrac{(\mathtt{f}:\alpha_2,\mathtt{x}:\alpha_4)\,(\mathtt{x})=\alpha_7}{\mathtt{f}:\alpha_2,\mathtt{x}:\alpha_4\vdash\mathtt{x}:\alpha_7}}{\mathtt{f}:\alpha_2,\mathtt{x}:\alpha_4\vdash\mathtt{f}\;\mathtt{x}:\alpha_5}}{\mathtt{f}:\alpha_2\vdash\lambda\mathtt{x}.\;\mathtt{f}\;\mathtt{x}:\alpha_3}}{\vdash\lambda\mathtt{f}.\;\lambda\mathtt{x}.\;\mathtt{f}\;\mathtt{x}:\alpha_1}$$

$$C=\{\alpha_1=\alpha_2\to\alpha_3,\alpha_3=\alpha_4\to\alpha_5,\alpha_6=\alpha_7\to\alpha_5,\alpha_2=\alpha_6,\alpha_4=\alpha_7\}$$

$$\sigma_C=[\alpha_1\Rightarrow(\alpha_7\to\alpha_5)\to\alpha_7\to\alpha_5,\alpha_2\Rightarrow\alpha_7\to\alpha_5,\alpha_3\Rightarrow\alpha_7\to\alpha_5,\alpha_4\Rightarrow\alpha_7,\alpha_6\Rightarrow\alpha_7\to\alpha_5]$$

$$\sigma_C\,(\alpha_1)=(\alpha_7\to\alpha_5)\to\alpha_7\to\alpha_5$$

Der λ-Kalkül

- Extrem simpel: Es gibt nur Funktionsdefinition und -Anwendung
- Fundament funktionaler Programmiersprachen
- $\lambda \mathtt{x}.\ t$ entspricht Funktion mit Parameter $\mathtt{x}$ und Ergebnis t
- Funktionsanwendung durch Hintereinanderschreiben (z.B. $\mathtt{f}\ \mathtt{x}$)

Beispiele:
$\lambda \mathtt{x}.\ \mathtt{x}$: Identitätsfunktion
$\lambda \mathtt{x}.\ \lambda \mathtt{y}.\ \mathtt{x}$: Erzeugt konstante Funktion
$\lambda \mathtt{f}.\ \lambda \mathtt{g}.\ \lambda \mathtt{x}.\ \mathtt{f}\ (\mathtt{g}\ \mathtt{x})$: Funktionsverkettung

Der λ-Kalkül (formal)

var := [a-zA-Z][a-zA-Z0-9]*
Term := *App* | *Abs* | **(** *Term* **)** | **var**
App := *Term Term*
Abs := $\underline{\lambda}$ **var** **.** *Term*

Klammerung ist wichtig!
Funktions-Applikation implizit **links**geklammert, Abstraktion “greedy”

Beispiele:
$\lambda\mathrm{x.\ x\ x\ x} = \lambda\mathrm{x.\ (x\ x)\ x}$
$\neq \lambda\mathrm{x.\ x\ (x\ x)}$
$\lambda\mathrm{x.\ x\ (}\lambda\mathrm{y.\ y)} = \lambda\mathrm{x.\ (x\ (}\lambda\mathrm{y.\ y))}$
$\neq (\lambda\mathrm{x.\ x})\ (\lambda\mathrm{y.\ y})$

Betareduktion (vereinfacht)

- "Ausführung" von Lambda-Termen
- Redex: Terme der Form $(\lambda \mathtt{x}.\ t_1)\ t_2$
- Betareduktion: Ersetze alle freien Vorkommen von $\mathtt{x}$ in t_1 durch t_2
- Beispiel: $(\lambda \mathtt{x}.\ \lambda \mathtt{y}.\ \mathtt{x}\ (\mathtt{y}\ \mathtt{x}))\ (\lambda \mathtt{z}.\ \mathtt{z}\ \mathtt{z}) \Rightarrow \lambda \mathtt{y}.\ (\lambda \mathtt{z}.\ \mathtt{z}\ \mathtt{z})\ (\mathtt{y}\ (\lambda \mathtt{z}.\ \mathtt{z}\ \mathtt{z}))$

Weitere Details und wie man im Lambda-Kalkül programmiert:
`https://pp.ipd.kit.edu/lehre/WS201718/pse_lambda-ide/lambda-vorlesung.pdf`

Lambda-IDE:
`https://pp.ipd.kit.edu/lehre/misc/lambda-ide/Wavelength.html`

Übungsblätter

Übungsblätter zum λ-Kalkül, Typinferenz etc. unter
`https://pp.ipd.kit.edu/lehre/WS201920/paradigmen/uebung`

Relevante Blätter:

- Blatt 5, Blatt 6 für λ-Kalkül
- Blatt 9 für Typinferenz

Typen

var := $\alpha \mid \beta \mid \ldots \mid \alpha_1 \mid \alpha_2 \mid \ldots$
basetype := **int** | **bool** | …
Fun := *Fun* $\rightarrow$ *Fun* | **var** | **basetype**

Klammerung ist wichtig! Funktionstypen implizit **rechts**geklammert

Beispiele:

`int`

α

$\alpha \rightarrow \beta$

$\alpha_1 \rightarrow \texttt{int} \rightarrow \alpha_2 = \alpha_1 \rightarrow (\texttt{int} \rightarrow \alpha_2)$

$\neq (\alpha_1 \rightarrow \texttt{int}) \rightarrow \alpha_2$

Klammerung λ-Terme vs. Typen

Linksassoziativität von Funktionsanwendung in λ-Termen und Rechtsassoziativität in Funktionstypen gehören zusammen:

Äquivalenz (a) "Funktion mit *n* Parametern"
$\leftrightarrow$
(b) "Funktion mit 1 Parameter, die Funktion mit $n - 1$ Parametern zurückgibt"

Beispiel $\texttt{const} = \lambda \texttt{x}.\ \lambda \texttt{y}.\ \texttt{x}$:

Interpretation	Fkt.-Definition	Fkt.-Anwendung	Fkt.-Typ
(a)	$\lambda \texttt{x}.\ \lambda \texttt{y}.\ \texttt{x}$	$\texttt{const a b}$	$\alpha \to \beta \to \alpha$
(b)	$\lambda \texttt{x}.\ (\lambda \texttt{y}.\ \texttt{x})$	$(\texttt{const a})\ \texttt{b}$	$\alpha \to (\beta \to \alpha)$

Typisierung

Typumgebung = Map von Variablen auf Typen
z.B. $\Gamma = \mathtt{x} : \alpha_1, \mathtt{y} : \textit{bool}, \mathtt{f} : \alpha_2 \to \alpha_3$

Typisierung: $\Gamma \vdash t : \tau$
“Unter Typumgebung (Typannahmen) Γ hat t den Typ τ

Beispiele:
$\mathtt{x} : \alpha \vdash \mathtt{x} : \alpha$
$\mathtt{f} : \mathtt{bool} \to \mathtt{bool} \to \mathtt{int} \vdash \lambda\mathtt{x}.\ \mathtt{f}\ \mathtt{x}\ \mathtt{x} : \mathtt{bool} \to \mathtt{int}$
$\vdash \lambda\mathtt{x}.\ \mathtt{x} : \alpha \to \alpha$ (leere Typumgebung $\Gamma = \emptyset$)

Typregeln

Darstellung von Typregeln:

$$\frac{\texttt{Voraussetzungen}}{\texttt{Folgerung}}$$

Const-Regel:

$$\textsc{Const}\ \frac{\texttt{c ist eine Konstante}}{\Gamma \vdash \mathtt{c} : \tau_{\mathtt{c}}}$$

Var-Regel:

$$\textsc{Var}\ \frac{\Gamma(\mathtt{x}) = \tau}{\Gamma \vdash \mathtt{x} : \tau}$$

App-Regel:

$$\textsc{App}\ \frac{\Gamma \vdash t_1 : \tau_1 \to \tau_2 \qquad \Gamma \vdash t_2 : \tau_1}{\Gamma \vdash t_1\ t_2 : \tau_2}$$

Abs-Regel:

$$\textsc{Abs}\ \frac{\Gamma, \mathtt{x} : \tau_1 \vdash t : \tau_2}{\Gamma \vdash \lambda \mathtt{x}.\ t : \tau_1 \to \tau_2}$$

Typinferenz

- Start mit Gesamtterm als finale Folgerung mit Typvariable als Typ
- Rückwärtsanwendung der Typregeln (es passt immer genau eine Regel!)
- Einführung neuer Typvariablen bei Abs- und App-Regel
- Aufsammeln von Constraints (Gleichungen) für Typvariablen

Beispiel: Typherleitungsbaum

$\vdash \lambda \texttt{x}.\ \lambda \texttt{y}.\ \texttt{y}\ \texttt{x} : \alpha_1$

Contraints:
$C = \{\}$

Beispiel: Typherleitungsbaum

$$\text{Abs}\ \frac{\mathtt{x} : \alpha_2 \vdash \lambda \mathtt{y}.\ \mathtt{y}\ \mathtt{x} : \alpha_3}{\vdash \lambda \mathtt{x}.\ \lambda \mathtt{y}.\ \mathtt{y}\ \mathtt{x} : \alpha_1}$$

Contraints:

$C = \{\alpha_1 = \alpha_2 \to \alpha_3\}$

Beispiel: Typherleitungsbaum

$$\textsc{Abs}\ \dfrac{\textsc{Abs}\ \dfrac{\mathtt{x} : \alpha_2, \mathtt{y} : \alpha_4 \vdash \mathtt{y}\ \mathtt{x} : \alpha_5}{\mathtt{x} : \alpha_2 \vdash \lambda \mathtt{y}.\ \mathtt{y}\ \mathtt{x} : \alpha_3}}{\vdash \lambda \mathtt{x}.\ \lambda \mathtt{y}.\ \mathtt{y}\ \mathtt{x} : \alpha_1}$$

Contraints:

$C = \{\alpha_1 = \alpha_2 \to \alpha_3, \alpha_3 = \alpha_4 \to \alpha_5\}$

Beispiel: Typherleitungsbaum

$$\cfrac{\cfrac{\cfrac{\mathtt{x} : \alpha_2, \mathtt{y} : \alpha_4 \vdash \mathtt{y} : \alpha_6 \qquad \mathtt{x} : \alpha_2, \mathtt{y} : \alpha_4 \vdash \mathtt{x} : \alpha_7}{\mathtt{x} : \alpha_2, \mathtt{y} : \alpha_4 \vdash \mathtt{y}\ \mathtt{x} : \alpha_5}\ \textsc{App}}{\mathtt{x} : \alpha_2 \vdash \lambda \mathtt{y}.\ \mathtt{y}\ \mathtt{x} : \alpha_3}\ \textsc{Abs}}{\vdash \lambda \mathtt{x}.\ \lambda \mathtt{y}.\ \mathtt{y}\ \mathtt{x} : \alpha_1}\ \textsc{Abs}$$

Contraints:

$C = \{\alpha_1 = \alpha_2 \to \alpha_3, \alpha_3 = \alpha_4 \to \alpha_5, \alpha_6 = \alpha_7 \to \alpha_5\}$

Beispiel: Typherleitungsbaum

$$
\textsc{Abs}\ \dfrac{\textsc{Abs}\ \dfrac{\textsc{App}\ \dfrac{\textsc{Var}\ \dfrac{(\mathtt{x} : \alpha_2, \mathtt{y} : \alpha_4)(\mathtt{y}) = \alpha_4}{\mathtt{x} : \alpha_2, \mathtt{y} : \alpha_4 \vdash \mathtt{y} : \alpha_6} \qquad \mathtt{x} : \alpha_2, \mathtt{y} : \alpha_4 \vdash \mathtt{x} : \alpha_7}{\mathtt{x} : \alpha_2, \mathtt{y} : \alpha_4 \vdash \mathtt{y}\ \mathtt{x} : \alpha_5}}{\mathtt{x} : \alpha_2 \vdash \lambda \mathtt{y}.\ \mathtt{y}\ \mathtt{x} : \alpha_3}}{\vdash \lambda \mathtt{x}.\ \lambda \mathtt{y}.\ \mathtt{y}\ \mathtt{x} : \alpha_1}
$$

Contraints:

$$C = \{\alpha_1 = \alpha_2 \to \alpha_3, \alpha_3 = \alpha_4 \to \alpha_5, \alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4\}$$

Beispiel: Typherleitungsbaum

$$
\dfrac{\dfrac{\dfrac{(\mathtt{x} : \alpha_2, \mathtt{y} : \alpha_4)(\mathtt{y}) = \alpha_4}{\mathtt{x} : \alpha_2, \mathtt{y} : \alpha_4 \vdash \mathtt{y} : \alpha_6}\textsc{Var} \qquad \dfrac{(\mathtt{x} : \alpha_2, \mathtt{y} : \alpha_4)(\mathtt{x}) = \alpha_2}{\mathtt{x} : \alpha_2, \mathtt{y} : \alpha_4 \vdash \mathtt{x} : \alpha_7}\textsc{Var}}{\mathtt{x} : \alpha_2, \mathtt{y} : \alpha_4 \vdash \mathtt{y}\ \mathtt{x} : \alpha_5}\textsc{App}}{\dfrac{\mathtt{x} : \alpha_2 \vdash \lambda \mathtt{y}.\ \mathtt{y}\ \mathtt{x} : \alpha_3}{\vdash \lambda \mathtt{x}.\ \lambda \mathtt{y}.\ \mathtt{y}\ \mathtt{x} : \alpha_1}\textsc{Abs}}\textsc{Abs}
$$

Contraints:

$C = \{\alpha_1 = \alpha_2 \to \alpha_3, \alpha_3 = \alpha_4 \to \alpha_5, \alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\}$

Substitution

- Ersetzen von freien Variablen durch Terme
- Muss nicht für alle Variablen eine Ersetzung definieren

Substitution

- Ersetzen von freien Variablen durch Terme
- Muss nicht für alle Variablen eine Ersetzung definieren
- $\sigma = [\alpha \Rightarrow \texttt{int}]$:
  $\sigma(\alpha \to \beta) = \texttt{int} \to \beta$

Substitution

- Ersetzen von freien Variablen durch Terme
- Muss nicht für alle Variablen eine Ersetzung definieren
- $\sigma = [\alpha \Rightarrow \texttt{int}]$:
  $\sigma(\alpha \to \beta) = \texttt{int} \to \beta$
- $\sigma = [\alpha \Rightarrow \texttt{int}, \beta \Rightarrow \gamma \to \texttt{bool}]$:
  $\sigma(\alpha \to \beta) = \texttt{int} \to \gamma \to \texttt{bool}$
  $\sigma(\beta \to \alpha) = (\gamma \to \texttt{bool}) \to \texttt{int}$

Substitution

- Ersetzen von freien Variablen durch Terme
- Muss nicht für alle Variablen eine Ersetzung definieren
- $\sigma = [\alpha \Rightarrow \texttt{int}]$:
  $\sigma(\alpha \to \beta) = \texttt{int} \to \beta$
- $\sigma = [\alpha \Rightarrow \texttt{int}, \beta \Rightarrow \gamma \to \texttt{bool}]$:
  $\sigma(\alpha \to \beta) = \texttt{int} \to \gamma \to \texttt{bool}$
  $\sigma(\beta \to \alpha) = (\gamma \to \texttt{bool}) \to \texttt{int}$
- $\sigma = [\alpha \Rightarrow \texttt{int}, \beta \Rightarrow (\delta \to \delta) \to \texttt{bool}]$:
  $\sigma((\alpha \to \gamma) \to \beta) = (\texttt{int} \to \gamma) \to (\delta \to \delta) \to \texttt{bool}$

Unifikation

- Unifikator: Substitution, die alle Gleichungen löst
- allgemeinster Unifikator: Unifikator, der ”möglichst wenig festlegt“
- $\{\alpha = \beta, \gamma = \texttt{int}\}$
  allgem. Unifikatoren z.B. $\sigma = [\alpha \Rightarrow \beta, \gamma \Rightarrow \texttt{int}]$, $\sigma = [\beta \Rightarrow \alpha, \gamma \Rightarrow \texttt{int}]$,
  dagegen $\sigma = [\alpha \Rightarrow \texttt{bool}, \beta \Rightarrow \texttt{bool}, \gamma \Rightarrow \texttt{int}]$ Unifikator, aber nicht allgemeinster Unifikator

Unifikation

- Unifikator: Substitution, die alle Gleichungen löst
- allgemeinster Unifikator: Unifikator, der ”möglichst wenig festlegt“
- $\{\alpha = \beta, \gamma = \texttt{int}\}$
  allgem. Unifikatoren z.B. $\sigma = [\alpha \Rightarrow \beta, \gamma \Rightarrow \texttt{int}]$, $\sigma = [\beta \Rightarrow \alpha, \gamma \Rightarrow \texttt{int}]$,
  dagegen $\sigma = [\alpha \Rightarrow \texttt{bool}, \beta \Rightarrow \texttt{bool}, \gamma \Rightarrow \texttt{int}]$ Unifikator, aber nicht allgemeinster Unifikator
- $\{\alpha = \texttt{int} \to \beta, \alpha \to \beta = (\texttt{int} \to \gamma) \to \texttt{bool}\}$
  Unifikator $\sigma = [\alpha \Rightarrow \texttt{int} \to \texttt{bool}, \beta \Rightarrow \texttt{bool}, \gamma \Rightarrow \texttt{bool}]$

Unifikation

- Unifikator: Substitution, die alle Gleichungen löst
- allgemeinster Unifikator: Unifikator, der "möglichst wenig festlegt"
- $\{\alpha = \beta, \gamma = \texttt{int}\}$
  allgem. Unifikatoren z.B. $\sigma = [\alpha \Rightarrow \beta, \gamma \Rightarrow \texttt{int}]$, $\sigma = [\beta \Rightarrow \alpha, \gamma \Rightarrow \texttt{int}]$,
  dagegen $\sigma = [\alpha \Rightarrow \texttt{bool}, \beta \Rightarrow \texttt{bool}, \gamma \Rightarrow \texttt{int}]$ Unifikator, aber nicht allgemeinster Unifikator
- $\{\alpha = \texttt{int} \to \beta, \alpha \to \beta = (\texttt{int} \to \gamma) \to \texttt{bool}\}$
  Unifikator $\sigma = [\alpha \Rightarrow \texttt{int} \to \texttt{bool}, \beta \Rightarrow \texttt{bool}, \gamma \Rightarrow \texttt{bool}]$
- $\{\alpha = \beta \to \gamma, \delta \to \alpha = \delta \to \texttt{int}\}$
  Nicht unifizierbar! Widerspruch durch Gleichung $\beta \to \gamma = \texttt{int}$!

Unifikation

- Unifikator: Substitution, die alle Gleichungen löst
- allgemeinster Unifikator: Unifikator, der ”möglichst wenig festlegt“
- $\{\alpha = \beta, \gamma = \texttt{int}\}$
  allgem. Unifikatoren z.B. $\sigma = [\alpha \Rightarrow \beta, \gamma \Rightarrow \texttt{int}]$, $\sigma = [\beta \Rightarrow \alpha, \gamma \Rightarrow \texttt{int}]$,
  dagegen $\sigma = [\alpha \Rightarrow \texttt{bool}, \beta \Rightarrow \texttt{bool}, \gamma \Rightarrow \texttt{int}]$ Unifikator, aber nicht allgemeinster Unifikator
- $\{\alpha = \texttt{int} \to \beta, \alpha \to \beta = (\texttt{int} \to \gamma) \to \texttt{bool}\}$
  Unifikator $\sigma = [\alpha \Rightarrow \texttt{int} \to \texttt{bool}, \beta \Rightarrow \texttt{bool}, \gamma \Rightarrow \texttt{bool}]$
- $\{\alpha = \beta \to \gamma, \delta \to \alpha = \delta \to \texttt{int}\}$
  Nicht unifizierbar! Widerspruch durch Gleichung $\beta \to \gamma = \texttt{int}$!
- $\{\alpha = \beta \to \alpha\}$
  Nicht unifizierbar! Würde unendlichen Typen benötigen
  $(\alpha = \beta \to \beta \to \beta \to \ldots)$!

Unifikationsalgorithmus

$$
\begin{array}{l}
\texttt{unify}(C) \;= \\
\quad \textbf{if}\; C == \emptyset \;\textbf{then}\; [] \\
\quad \textbf{else}\; \texttt{let}\; \{\tau_1 = \tau_2\} \cup C' = C \;\texttt{in} \\
\qquad \textbf{if}\; \tau_1 == \tau_2 \;\textbf{then}\; \texttt{unify}(C') \\
\qquad \textbf{else if}\; \tau_1 == \alpha \;\&\&\; \alpha \notin FV(\tau_2) \;\textbf{then}\; \texttt{unify}([\alpha \Rightarrow \tau_2]C') \circ [\alpha \Rightarrow \tau_2] \\
\qquad \textbf{else if}\; \tau_2 == \alpha \;\&\&\; \alpha \notin FV(\tau_1) \;\textbf{then}\; \texttt{unify}([\alpha \Rightarrow \tau_1]C') \circ [\alpha \Rightarrow \tau_1] \\
\qquad \textbf{else if}\; \tau_1 == (\tau_1' \to \tau_1'') \;\&\&\; \tau_2 == (\tau_2' \to \tau_2'') \\
\qquad\qquad \textbf{then}\; \texttt{unify}(C' \cup \{\tau_1' = \tau_2', \tau_1'' = \tau_2''\}) \\
\qquad \textbf{else}\; \textbf{fail}
\end{array}
$$

$FV(\tau)$: freie Variablen aus τ

$\alpha \notin FV(\tau_i)$ (“occurs check”) verhindert unendliche Typen

Beispiel: Unifikation

Karlsruher Institut für Technologie

$$\mathit{unify}(\{\alpha_1 = \alpha_2 \to \alpha_3, \alpha_3 = \alpha_4 \to \alpha_5, \alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\})$$

Beispiel: Unifikation

Karlsruher Institut für Technologie

$$unify(\{\alpha_1 = \alpha_2 \to \alpha_3, \alpha_3 = \alpha_4 \to \alpha_5, \alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\})$$
$$= unify(\{\alpha_3 = \alpha_4 \to \alpha_5, \alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\})$$
$$\circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]$$

Beispiel: Unifikation

$$unify(\{\alpha_1 = \alpha_2 \to \alpha_3, \alpha_3 = \alpha_4 \to \alpha_5, \alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\})$$
$$= unify(\{\alpha_3 = \alpha_4 \to \alpha_5, \alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\})$$
$$\circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]$$
$$= unify(\{\alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\})$$
$$\circ [\alpha_3 \Rightarrow \alpha_4 \to \alpha_5] \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]$$

Beispiel: Unifikation

$$
\begin{aligned}
&\mathit{unify}(\{\alpha_1 = \alpha_2 \to \alpha_3, \alpha_3 = \alpha_4 \to \alpha_5, \alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\}) \\
&= \mathit{unify}(\{\alpha_3 = \alpha_4 \to \alpha_5, \alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\}) \\
&\quad \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3] \\
&= \mathit{unify}(\{\alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\}) \\
&\quad \circ [\alpha_3 \Rightarrow \alpha_4 \to \alpha_5] \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3] \\
&= \mathit{unify}(\{\alpha_7 \to \alpha_5 = \alpha_4, \alpha_7 = \alpha_2\}) \\
&\quad \circ [\alpha_6 \Rightarrow \alpha_7 \to \alpha_5] \circ [\alpha_3 \Rightarrow \alpha_4 \to \alpha_5] \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]
\end{aligned}
$$

Beispiel: Unifikation

$$\mathit{unify}(\{\alpha_1 = \alpha_2 \to \alpha_3, \alpha_3 = \alpha_4 \to \alpha_5, \alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\})$$
$$= \mathit{unify}(\{\alpha_3 = \alpha_4 \to \alpha_5, \alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\})$$
$$\circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]$$
$$= \mathit{unify}(\{\alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\})$$
$$\circ [\alpha_3 \Rightarrow \alpha_4 \to \alpha_5] \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]$$
$$= \mathit{unify}(\{\alpha_7 \to \alpha_5 = \alpha_4, \alpha_7 = \alpha_2\})$$
$$\circ [\alpha_6 \Rightarrow \alpha_7 \to \alpha_5] \circ [\alpha_3 \Rightarrow \alpha_4 \to \alpha_5] \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]$$
$$= \mathit{unify}(\{\alpha_7 = \alpha_2\})$$
$$\circ [\alpha_4 \Rightarrow \alpha_7 \to \alpha_5] \circ [\alpha_6 \Rightarrow \alpha_7 \to \alpha_5] \circ [\alpha_3 \Rightarrow \alpha_4 \to \alpha_5] \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]$$

Beispiel: Unifikation

$$
\begin{aligned}
&\mathit{unify}(\{\alpha_1 = \alpha_2 \to \alpha_3, \alpha_3 = \alpha_4 \to \alpha_5, \alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\})\\
&= \mathit{unify}(\{\alpha_3 = \alpha_4 \to \alpha_5, \alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\})\\
&\quad \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]\\
&= \mathit{unify}(\{\alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\})\\
&\quad \circ [\alpha_3 \Rightarrow \alpha_4 \to \alpha_5] \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]\\
&= \mathit{unify}(\{\alpha_7 \to \alpha_5 = \alpha_4, \alpha_7 = \alpha_2\})\\
&\quad \circ [\alpha_6 \Rightarrow \alpha_7 \to \alpha_5] \circ [\alpha_3 \Rightarrow \alpha_4 \to \alpha_5] \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]\\
&= \mathit{unify}(\{\alpha_7 = \alpha_2\})\\
&\quad \circ [\alpha_4 \Rightarrow \alpha_7 \to \alpha_5] \circ [\alpha_6 \Rightarrow \alpha_7 \to \alpha_5] \circ [\alpha_3 \Rightarrow \alpha_4 \to \alpha_5] \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]\\
&= \mathit{unify}(\emptyset) \circ [\alpha_7 \Rightarrow \alpha_2] \circ [\alpha_4 \Rightarrow \alpha_7 \to \alpha_5] \circ [\alpha_6 \Rightarrow \alpha_7 \to \alpha_5]\\
&\quad \circ [\alpha_3 \Rightarrow \alpha_4 \to \alpha_5] \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]
\end{aligned}
$$

Beispiel: Unifikation

$$
\begin{aligned}
&\mathit{unify}(\{\alpha_1 = \alpha_2 \to \alpha_3, \alpha_3 = \alpha_4 \to \alpha_5, \alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\})\\
&= \mathit{unify}(\{\alpha_3 = \alpha_4 \to \alpha_5, \alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\})\\
&\quad \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]\\
&= \mathit{unify}(\{\alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\})\\
&\quad \circ [\alpha_3 \Rightarrow \alpha_4 \to \alpha_5] \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]\\
&= \mathit{unify}(\{\alpha_7 \to \alpha_5 = \alpha_4, \alpha_7 = \alpha_2\})\\
&\quad \circ [\alpha_6 \Rightarrow \alpha_7 \to \alpha_5] \circ [\alpha_3 \Rightarrow \alpha_4 \to \alpha_5] \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]\\
&= \mathit{unify}(\{\alpha_7 = \alpha_2\})\\
&\quad \circ [\alpha_4 \Rightarrow \alpha_7 \to \alpha_5] \circ [\alpha_6 \Rightarrow \alpha_7 \to \alpha_5] \circ [\alpha_3 \Rightarrow \alpha_4 \to \alpha_5] \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]\\
&= \mathit{unify}(\emptyset) \circ [\alpha_7 \Rightarrow \alpha_2] \circ [\alpha_4 \Rightarrow \alpha_7 \to \alpha_5] \circ [\alpha_6 \Rightarrow \alpha_7 \to \alpha_5]\\
&\quad \circ [\alpha_3 \Rightarrow \alpha_4 \to \alpha_5] \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]\\
&= [\alpha_7 \Rightarrow \alpha_2] \circ [\alpha_4 \Rightarrow \alpha_7 \to \alpha_5] \circ [\alpha_6 \Rightarrow \alpha_7 \to \alpha_5]\\
&\quad \circ [\alpha_3 \Rightarrow \alpha_4 \to \alpha_5] \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]
\end{aligned}
$$

Beispiel: Unifikation

$$
\begin{aligned}
&\mathit{unify}(\{\alpha_1 = \alpha_2 \to \alpha_3, \alpha_3 = \alpha_4 \to \alpha_5, \alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\})\\
&= \mathit{unify}(\{\alpha_3 = \alpha_4 \to \alpha_5, \alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\})\\
&\quad \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]\\
&= \mathit{unify}(\{\alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\})\\
&\quad \circ [\alpha_3 \Rightarrow \alpha_4 \to \alpha_5] \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]\\
&= \mathit{unify}(\{\alpha_7 \to \alpha_5 = \alpha_4, \alpha_7 = \alpha_2\})\\
&\quad \circ [\alpha_6 \Rightarrow \alpha_7 \to \alpha_5] \circ [\alpha_3 \Rightarrow \alpha_4 \to \alpha_5] \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]\\
&= \mathit{unify}(\{\alpha_7 = \alpha_2\})\\
&\quad \circ [\alpha_4 \Rightarrow \alpha_7 \to \alpha_5] \circ [\alpha_6 \Rightarrow \alpha_7 \to \alpha_5] \circ [\alpha_3 \Rightarrow \alpha_4 \to \alpha_5] \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]\\
&= \mathit{unify}(\emptyset) \circ [\alpha_7 \Rightarrow \alpha_2] \circ [\alpha_4 \Rightarrow \alpha_7 \to \alpha_5] \circ [\alpha_6 \Rightarrow \alpha_7 \to \alpha_5]\\
&\quad \circ [\alpha_3 \Rightarrow \alpha_4 \to \alpha_5] \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]\\
&= [\alpha_7 \Rightarrow \alpha_2] \circ [\alpha_4 \Rightarrow \alpha_7 \to \alpha_5] \circ [\alpha_6 \Rightarrow \alpha_7 \to \alpha_5]\\
&\quad \circ [\alpha_3 \Rightarrow \alpha_4 \to \alpha_5] \circ [\alpha_1 \Rightarrow \alpha_2 \to \alpha_3]\\
&= [\alpha_1 \Rightarrow \alpha_2 \to (\alpha_2 \to \alpha_5) \to \alpha_5, \alpha_3 \Rightarrow (\alpha_2 \to \alpha_5) \to \alpha_5,\\
&\quad\ \alpha_4 \Rightarrow \alpha_2 \to \alpha_5, \alpha_6 \Rightarrow \alpha_2 \to \alpha_5, \alpha_7 \Rightarrow \alpha_2]
\end{aligned}
$$

Beispiel: Ergebnis

Typherleitungsbaum:

$$\textsc{Abs}\ \dfrac{\textsc{Abs}\ \dfrac{\vdots}{\mathtt{x} : \alpha_2 \vdash \lambda \mathtt{y}.\ \mathtt{y}\ \mathtt{x} : \alpha_3}}{\vdash \lambda \mathtt{x}.\ \lambda \mathtt{y}.\ \mathtt{y}\ \mathtt{x} : \alpha_1}$$

Contraints:

$$C = \{\alpha_1 = \alpha_2 \to \alpha_3, \alpha_3 = \alpha_4 \to \alpha_5, \alpha_6 = \alpha_7 \to \alpha_5, \alpha_6 = \alpha_4, \alpha_7 = \alpha_2\}$$

Lösung:

$$\sigma = [\alpha_1 \Rightarrow \alpha_2 \to (\alpha_5 \to \alpha_2) \to \alpha_5, \alpha_3 \Rightarrow (\alpha_5 \to \alpha_2) \to \alpha_5,$$
$$\alpha_4 \Rightarrow \alpha_5 \to \alpha_2, \alpha_6 \Rightarrow \alpha_5 \to \alpha_2, \alpha_7 \Rightarrow \alpha_2]$$
$$\sigma(\alpha_1) = \alpha_2 \to (\alpha_5 \to \alpha_2) \to \alpha_5$$

Let-Polymorphismus

- Typvariablen wie α_1 können jeden beliebigen Typ annehmen...
- ... aber innerhalb einer Typinferenz keine unterschiedlichen Typen
- Lösung: Let-Ausdruck **let** $\texttt{f} = t_1$ **in** t_2 mit Typschema $\forall \alpha_1, \ldots, \alpha_n . \tau$
- Instanziieren eines Typschemas ($\succeq$):
  Instanziieren aller $\forall$-gebundenen Variablen
- mehrfach (auch unterschiedlich) instanziierbar

Beispiele:

$\forall \alpha . \alpha \succeq \texttt{int}$	$\forall \alpha . \alpha \succeq \texttt{bool}$
$\forall \alpha . \alpha \succeq \alpha_1$	$\forall \alpha . \alpha \succeq \alpha_1 \to \alpha_2$
$\forall \alpha, \beta . \alpha \to \beta \succeq \texttt{int} \to \alpha_1$	$\forall \alpha, \beta . \alpha \to \beta \succeq \texttt{int} \to \texttt{int}$
$\forall \alpha . \alpha \to \beta \succeq \alpha_1 \to \beta$	$\alpha_1 \succeq \alpha_1$
$\texttt{int} \not\succeq \texttt{bool}$	$\alpha_1 \not\succeq \alpha_2$
$\forall \alpha . \alpha \to \alpha \not\succeq \alpha_1 \to \alpha_2$	$\forall \alpha, \beta . \alpha \to \beta \not\succeq \forall \beta . \alpha_1 \to \beta$

Typabstraktion

Freie Variablen $\equiv$ nicht durch ein $\forall$ gebundenen

Typabstraktion *ta*(τ, Γ) entsteht aus τ, indem all jene Typvariablen allquantifiziert werden, die frei in τ, aber nicht frei in Γ vorkommen.

Beispiele:

$\mathit{ta}(\alpha_1 \to \alpha_2, \emptyset) = \forall \alpha_1, \alpha_2 . \alpha_1 \to \alpha_2$

$\mathit{ta}(\mathtt{int} \to \alpha_1, \mathtt{x} : \alpha_2) = \forall \alpha_1 . \mathtt{int} \to \alpha_1$

$\mathit{ta}(\alpha_1, \mathtt{f} : \alpha_2 \to \alpha_1) = \alpha_1$

$\mathit{ta}(\alpha_1 \to \alpha_2, (\mathtt{f} : \forall \alpha_1 . \alpha_1, \mathtt{x} : \alpha_2)) = \forall \alpha_1 . \alpha_1 \to \alpha_2$

Typregeln für Let-Polymorphismus

Var-Regel (angepasst):

$$\textsc{Var}\ \frac{\Gamma(\mathtt{x}) = \tau' \qquad \tau' \succeq \tau}{\Gamma \vdash \mathtt{x} : \tau}$$

In Typinferenz: τ' mit neuen Typvariablen instanziieren

Abs-Regel (angepasst):

$$\textsc{Abs}\ \frac{\Gamma, \mathtt{x} : \tau_1 \vdash t : \tau_2 \qquad \tau_1\ \texttt{kein Typschema}}{\Gamma \vdash \lambda \mathtt{x}.\ t : \tau_1 \to \tau_2}$$

In Typinferenz: Algorithmus erfüllt zusätzliche Bedingung automatisch

Let-Regel:

$$\frac{\Gamma \vdash t_1 : \tau_1 \qquad \Gamma, \mathtt{x} : ta(\tau_1, \Gamma) \vdash t_2 : \tau_2}{\Gamma \vdash \textbf{let}\ \mathtt{x} = t_1\ \textbf{in}\ t_2 : \tau_2}$$

Typinferenz mit Let-Polymorphismus

$$\cfrac{\cfrac{\cfrac{\dots}{\Gamma \vdash t_1 : \alpha_2} \qquad \cfrac{\dots}{\Gamma' \vdash t_2 : \alpha_3}}{\Gamma \vdash \textbf{let}\ \mathtt{x} = t_1\ \textbf{in}\ t_2 : \alpha_1}}{\dots}$$

Allgemeine Vorgehensweise:

1. Sei C_0 die bisherige Constraintmenge, inkl. $\{\alpha_1 = \alpha_3\}$
2. Sammle Constraints aus linkem Teilbaum in C_{let}
3. Berechne den allgemeinsten Unifikator σ_{let} von C_{let}
4. Berechne $\Gamma' := \sigma_{let}(\Gamma), \mathtt{x} : ta(\sigma_{let}(\alpha_2), \sigma_{let}(\Gamma))$
5. Benutze Γ' in rechtem Teilbaum, sammle Constraints in C_1
6. Ergebnisconstraints sind $C_0 \cup C'_{let} \cup C_1$ mit
   $C'_{let} := \{\alpha_i = \sigma_{let}(\alpha_i) \mid \sigma_{let}\ \mathtt{definiert\ für}\ \alpha_i\}$