Polymorpher Lambda-Kalkül [Girard 1972, Reynolds 1974]

Typen: $\tau := \alpha \mid \tau \to \tau \mid \forall\alpha.\tau$

Terme: $t := x \mid \lambda x : \tau.t \mid t\ t \mid \Lambda\alpha.t \mid t\ \tau$

$$\Gamma, x : \tau \vdash x : \tau \qquad [\![x]\!]_{\theta,\sigma} = \sigma(x)$$

$$\frac{\Gamma, x : \tau_1 \vdash t : \tau_2}{\Gamma \vdash (\lambda x : \tau_1.t) : \tau_1 \to \tau_2} \qquad [\![\lambda x : \tau_1.t]\!]_{\theta,\sigma}\ a = [\![t]\!]_{\theta,\sigma[x \mapsto a]}$$

$$\frac{\Gamma \vdash t : \tau_1 \to \tau_2 \qquad \Gamma \vdash u : \tau_1}{\Gamma \vdash (t\ u) : \tau_2} \qquad [\![t\ u]\!]_{\theta,\sigma} = [\![t]\!]_{\theta,\sigma}\ [\![u]\!]_{\theta,\sigma}$$

$$\frac{\alpha, \Gamma \vdash t : \tau}{\Gamma \vdash (\Lambda\alpha.t) : \forall\alpha.\tau} \qquad [\![\Lambda\alpha.t]\!]_{\theta,\sigma}\ S = [\![t]\!]_{\theta[\alpha \mapsto S],\sigma}$$

$$\frac{\Gamma \vdash t : \forall\alpha.\tau}{\Gamma \vdash (t\ \tau') : \tau[\tau'/\alpha]} \qquad [\![t\ \tau']\!]_{\theta,\sigma} = [\![t]\!]_{\theta,\sigma}\ [\![\tau']\!]_{\theta}$$

Rekursion im λ-Kalkül (3)

- Die Transformation von fak zu fak‘ war systematisch – zu jeder rekursiven Funktion H kann ein entsprechendes (nicht-rekursives) Funktional H‘ gefunden werden.

- Wir suchen jetzt eine (nicht-rekursive) Funktion **Y**, für die gilt:

$$Y\,(H') = H$$

(H rekursiv, Y und H‘ nicht-rekursiv)

- Da generell H‘(H) = H gilt, ist H ein Fixpunkt der Funktion H‘.

- Weil für die gesuchte Funktion Y(H‘) → H gilt, wird sie auch als Fixpunktkombinator bezeichnet:

$$H = Y\,(H') = H'\,(H) = H'\,(Y\,H')$$

insbes. gilt:

$$Y\,(H') = H'\,(Y\,H')$$

Rekursion im λ-Kalkül (4)

- Die Funktion **Y** wird definiert als:

$$\lambda\, h\, x\, .\, h\, (x\, x)\;\; (\lambda\, x\, .\, h\, (x\, x))$$

- Dass diese Funktion das gesuchte Y ist, wird durch folgenden Zusammenhang belegt:

$$\begin{aligned} \mathbf{Y}\;\; \mathbf{H'} \;\; &= (\lambda\, h\, x\, .\, h\, (x\, x)\; (\lambda\, x\, .\, h\, (x\, x))\;)\;\; H' \\ &\rightarrow (\lambda\, x\, .\, H'\, (x\, x))\; (\lambda\, x\, .\, H'\, (x\, x)) \\ &\rightarrow H'\; ((\lambda\, x\, .\, H'\, (x\, x))\; (\lambda\, x\, .\, H'\, (x\, x))) \\ &= \mathbf{H'\; (\; Y\; H')} \end{aligned}$$

Ein vollständiger Beweis ist das aber nicht – dieser kann z.B. bei Meyer oder Stoy nachgelesen werden.

Allgemeine Rekursion im Polymorphen Lambda-Kalkül

Terme: $t := \cdots \mid \mathbf{fix}\ t$

$$\frac{\Gamma \vdash t : \tau \to \tau}{\Gamma \vdash (\mathbf{fix}\ t) : \tau}$$

Um Semantik anzugeben, werden Typen als vollständige Halbordnungen mit kleinstem Element interpretiert, und:

$$[\![\mathbf{fix}\ t]\!]_{\theta,\sigma} = \bigsqcup_{i \geq 0} ([\![t]\!]^i_{\theta,\sigma}\ \bot).$$

Verwendung im Beispiel

Aus:

$$
\begin{aligned}
&\texttt{filter} :: (\alpha \to \mathsf{Bool}) \to [\alpha] \to [\alpha] \\
&\texttt{filter}\ p\ [\,] \qquad\quad = [\,] \\
&\texttt{filter}\ p\ (a : as) = \textbf{if}\ p\ a\ \textbf{then}\ a : \texttt{filter}\ p\ as \\
&\qquad\qquad\qquad\qquad\qquad\quad \textbf{else}\ \ \texttt{filter}\ p\ as
\end{aligned}
$$

wird:

$$
\begin{aligned}
&\textbf{fix}\ (\lambda f : (\forall\alpha.(\alpha \to \mathsf{Bool}) \to [\alpha] \to [\alpha]). \\
&\quad \Lambda\alpha.\lambda p : (\alpha \to \mathsf{Bool}).\lambda l : [\alpha]. \\
&\quad \textbf{case}\ l\ \textbf{of}\ \{[\,] \qquad\quad \to [\,]_\alpha\,; \\
&\qquad\qquad\qquad (a : as) \to \textbf{case}\ p\ a\ \textbf{of} \\
&\qquad\qquad\qquad\qquad\qquad \{\mathsf{True} \to a : (f\ \alpha\ p\ as)\,; \\
&\qquad\qquad\qquad\qquad\qquad\ \ \mathsf{False} \to f\ \alpha\ p\ as\}\})
\end{aligned}
$$

Allgemeine Rekursion im Polymorphen Lambda-Kalkül

Terme: $t := \cdots \mid \mathbf{fix}\ t$

$$\frac{\Gamma \vdash t : \tau \to \tau}{\Gamma \vdash (\mathbf{fix}\ t) : \tau}$$

Um Semantik anzugeben, werden Typen als vollständige Halbordnungen mit kleinstem Element interpretiert, und:

$$[\![\mathbf{fix}\ t]\!]_{\theta,\sigma} = \bigsqcup_{i \geq 0} ([\![t]\!]^i_{\theta,\sigma}\ \bot).$$

Und was ist mit dem Parametrizitäts-Theorem?

Der relevante Induktions-Fall ist:

$$\frac{\forall (a_1, a_2) \in \Delta_{\tau,\rho}.\ ([\![t]\!]_{\theta_1,\sigma_1}\ a_1, [\![t]\!]_{\theta_2,\sigma_2}\ a_2) \in \Delta_{\tau,\rho}}{([\![\mathbf{fix}\ t]\!]_{\theta_1,\sigma_1}, [\![\mathbf{fix}\ t]\!]_{\theta_2,\sigma_2}) \in \Delta_{\tau,\rho}}$$

Das Parametrizitäts-Theorem gilt weiter, vorausgesetzt alle Relationen sind strikt und stetig.

Anwendung dieser Erkenntnis

Für jedes $\texttt{g} :: (a \to \text{Bool}) \to [a] \to [a]$,

$$\texttt{g}\ p\ (\texttt{map}\ h\ l) \quad = \quad \texttt{map}\ h\ (\texttt{g}\ (p \circ h)\ l)$$

wenn h strikt ($h\ \bot = \bot$).

Analoge Auswirkungen ergeben sich für andere freie Theoreme.

Übrigens sind solche Nebenbedingungen im Allgemeinen nur hinreichend, nicht unbedingt notwendig.

Hinzufügen Selektiver Striktheit

Terme: $t := \quad \cdots \mid \mathbf{seq}\ t\ t$

$$\frac{\Gamma \vdash t_1 : \tau_1 \qquad \Gamma \vdash t_2 : \tau_2}{\Gamma \vdash (\mathbf{seq}\ t_1\ t_2) : \tau_2}$$

Semantik:

$$[\![\mathbf{seq}\ t_1\ t_2]\!]_{\theta,\sigma} = \begin{cases} \bot & \text{wenn } [\![t_1]\!]_{\theta,\sigma} = \bot \\ [\![t_2]\!]_{\theta,\sigma} & \text{wenn } [\![t_1]\!]_{\theta,\sigma} \neq \bot. \end{cases}$$

Das Parametrizitäts-Theorem steht wieder in Zweifel!

Ohne **seq**, $\mathtt{g}\ p\ (\mathtt{map}\ h\ l) = \mathtt{map}\ h\ (\mathtt{g}\ (p \circ h)\ l)$

- $\mathtt{g} :: (a \to \mathrm{Bool}) \to [a] \to [a]$ muss einheitlich arbeiten.
- Die Ausgabeliste kann nur Elemente aus l, sowie $\bot$ enthalten.
- Welche, und in welcher Reihenfolge/Vielfachheit, kann lediglich von l und dem Eingabeprädikat p abhängen.
- Einzig mögliche Grundlagen zur Entscheidung sind die Länge von l und Ergebnisse von p auf Elementen von l und auf $\bot$.
- Aber, die Listen $(\mathtt{map}\ h\ l)$ und l haben stets die selbe Länge.
- Und Anwendung von p auf ein Element von $(\mathtt{map}\ h\ l)$ hat stets das selbe Ergebnis wie Anwendung von $(p \circ h)$ auf das entsprechende Element von l.
- Und Anwendung von p auf $\bot$ hat das selbe Ergebnis wie Anwendung von $(p \circ h)$, vorausgesetzt h ist strikt.
- Also wählt $\mathtt{g}$ mit p stets „die selben“ Elemente aus $(\mathtt{map}\ h\ l)$ wie es $\mathtt{g}$ mit $(p \circ h)$ aus l tut, außer dass im ersten Fall die entsprechenden Abbilder unter h ausgegeben werden, und dass sie auch, an gleichen Positionen, $\bot$ ausgeben können.
- Also $(\mathtt{g}\ p\ (\mathtt{map}\ h\ l)) = (\mathtt{map}\ h\ (\mathtt{g}\ (p \circ h)\ l))$, wenn h strikt ist.

Mit **seq**, $\mathtt{g}\ p\ (\mathtt{map}\ h\ l) = \mathtt{map}\ h\ (\mathtt{g}\ (p \circ h)\ l)$?

- $\mathtt{g} :: (a \to \mathrm{Bool}) \to [a] \to [a]$ muss einheitlich arbeiten.
- Die Ausgabeliste kann nur Elemente aus l, sowie $\bot$ enthalten.
- Welche, und in welcher Reihenfolge/Vielfachheit, kann lediglich von l und dem Eingabeprädikat p abhängen.
- Einzig mögliche Grundlagen zur Entscheidung sind die Länge von l und Ergebnisse von p auf Elementen von l und auf $\bot$.

↯ Falsch! Auch möglich:

- Elemente von l auf $\bot$ prüfen
- p auf $\bot$ prüfen

... ???

Anpassungen Freier Theoreme

[Wadler 1989] : für jedes g :: $(a \to \text{Bool}) \to [a] \to [a]$,

$$\texttt{g}\ p\ (\texttt{map}\ h\ l) \quad = \quad \texttt{map}\ h\ (\texttt{g}\ (p \circ h)\ l)$$

- wenn h strikt.

[Johann & V. 2004] : in Gegenwart von **seq**, wenn zusätzlich:

- $p \neq \bot$ und
- h total ($\forall x \neq \bot.\ h\ x \neq \bot$).

[Johann & V. 2009] : Betrachtung von Laufzeitfehlern

[Stenger & V. 2009] : Betrachtung von „impräziser“ Fehlersemantik