

Merkzettel für „Theorie der Programmierung“

Marco Ammon

5. Oktober 2018

Termersetzungssysteme

Terminierung

Polynomordnungen

Konfluenz

Critical Pairs

λ -Kalkül

Ungetypt

- β -Reduktion

$$(\lambda x.t)s \rightarrow_{\beta} t[s/x]$$

- η -Reduktion

$$\lambda x.yx \rightarrow_{\eta} y$$

Auswertungsstrategien

- applikativ (*leftmost-innermost*) \rightarrow_a

Def. 3.13 (33)

- $\lambda x.t \rightarrow_a \lambda x.t'$, wenn $t \rightarrow_a t'$
- $ts \rightarrow_a t's$, wenn $t \rightarrow_a t'$
- $ts \rightarrow_a ts'$, wenn $s \rightarrow_a s'$ und t normal ist
- $(\lambda x.t)s \rightarrow_a t[s/x]$, wenn t und s normal sind
- effizient

- normal (*leftmost-outermost*) \rightarrow_n

Def. 3.14 (34)

- $(\lambda x.t)s \rightarrow_n t[s/x]$
- $\lambda x.t \rightarrow_n \lambda x.t'$, wenn $t \rightarrow_n t'$
- $ts \rightarrow_n t's$ wenn $t \rightarrow_n t'$ und t keine λ -Abstraktion ist
- $ts \rightarrow_n ts'$, wenn $s \rightarrow_n s'$ und t normal und keine λ -Abstraktion ist
- terminiert immer, falls Normalform existiert (nach Standardisierungssatz)

Satz 3.17 (35)

Einfach getypt ($\lambda \rightarrow$)

S. 39

- Church: Annotation der Variablen mit Typen, nur herleitbare Terme hinschreibbar
- Curry: Alle Terme hinschreibbar, dann Aussondern der nicht typisierbaren
- Typregeln:

S. 39

$$\begin{aligned}
 (Ax) \quad & \frac{}{\Gamma \vdash x : \alpha} x : \alpha \in \Gamma \\
 (\rightarrow_e) \quad & \frac{\Gamma \vdash t : \alpha \rightarrow \beta \quad \Gamma \vdash s : \alpha}{\Gamma \vdash ts : \beta} \\
 (\rightarrow_i) \quad & \frac{\Gamma, x : \alpha \vdash t : \beta}{\Gamma \vdash \lambda x. t : \alpha \rightarrow \beta}
 \end{aligned}$$

- Typisierungsprobleme

S. 40

- Typcheck: „Gilt $\Gamma \vdash t : \alpha$?“
- Typinferenz: „Was ist das beste α / Existiert α mit $\Gamma \vdash t : \alpha$?“
- Type inhabitation: „Existiert t mit $\Gamma t : \alpha$?“

- Inversionslemma

Lem. 3.29 (41)

1. $\Gamma \vdash x : \alpha \Rightarrow x : \alpha \in \Gamma$
2. $\Gamma \vdash ts : \beta \Rightarrow \exists \alpha$ mit $\Gamma \vdash t : \alpha \rightarrow \beta$ und $\Gamma \vdash s : \alpha$
3. $\Gamma \vdash \lambda x. t : \gamma \Rightarrow \gamma = \alpha \rightarrow \beta$ mit $\Gamma, x : \alpha \vdash t : \beta$

- Typinferenz

S. 41

- Typsubstitution σ ist Lösung von $\Gamma \vdash t : \alpha$, wenn $\Gamma \sigma \vdash t : \alpha \sigma$ herleitbar
- Substitutionen: σ_1 allgemeiner als $\sigma_2 \Leftrightarrow \exists \tau. \sigma_1 \tau = \sigma_2$
- Prinzipaltyp von Γ, t ist $\sigma(a)$ für allgemeinste Lösung σ von $\Gamma \vdash t : a$ (a frisch)
- Algorithmus W (Hindley/Milner)
- * Menge PT von Typgleichungen

GLoIn, S. 38

Alg. 3.31 (42)

$$\begin{aligned}
 PT(\Gamma; x; \alpha) &= \{a \doteq b \mid x : \beta \in \Gamma\} \\
 PT(\Gamma; \lambda x. t; \alpha) &= PT((\Gamma; x : a); t; b) \cup \{a \rightarrow b \doteq \alpha\} \text{ mit } a, b \text{ frisch} \\
 PT(\Gamma; ts; \alpha) &= PT(\Gamma; t; a \rightarrow \alpha) \cup PT(\Gamma; s; a) \text{ mit } a \text{ frisch}
 \end{aligned}$$

- * Typinferenz des Terms u mit leerem Kontext:

$$\varepsilon := PT(\emptyset; u; a)$$

$$\Rightarrow \text{Prinzipaltyp von } u: \text{mgu}(\varepsilon)(a)$$

- Subjektreduktion: Wenn $\Gamma \vdash t : \alpha$ und $t \rightarrow_{\beta}^* s$, dann auch $\Gamma \vdash s : \alpha$, aber nicht umgekehrt!

Satz 3.38 (45)

Induktive Datentypen

Mengenkonstruktionen

Mehrsortigkeit

Strukturelle Induktion

- über einsortige Datentypen S. 63
 - Induktionsanfang: „Anfangs“-Konstruktor (etwa *Nil*)
 - Induktionsschritt: alle anderen Konstruktoren (etwa *cons*)
- über mehrsortige Datentypen S. 64
 - Funktionen müssen immer auf allen Datentypen definiert werden

Kodatentypen

Koinduktion

- Bisimulation $R \subseteq A^\omega \times A^\omega$, wenn für alle $(s, t) \in R$ gilt: Def. 4.39 (74)

$$\begin{aligned} hd\ s &= hd\ t \\ (tl\ s) &R\ (tl\ t) \end{aligned}$$

- Wenn R eine Bisimulation ist, gilt $sRt \Rightarrow s = t$ Satz 4.40 (74)

Kodatentypen mit Alternativen

System F

Curry

- Typen: Def. 5.1 (84)

$$\alpha, \beta := a \mid \alpha \rightarrow \beta \mid \forall a. \alpha \quad (a \in V)$$

- Typisierung: Def. 5.1 (84)

$$\begin{aligned} (Ax) \quad & \frac{}{\Gamma \vdash x : \alpha} (x : \alpha \in \Gamma) \\ (\rightarrow_e) \quad & \frac{\Gamma \vdash t : \alpha \rightarrow \beta \quad \Gamma \vdash s : \alpha}{\Gamma \vdash ts : \beta} \\ (\rightarrow_i) \quad & \frac{\Gamma, x : \alpha \vdash t : \beta}{\Gamma \vdash \lambda x. t : \alpha \rightarrow \beta} \\ (\forall_i) \quad & \frac{\Gamma \vdash s : \alpha \quad a \notin FV(\Gamma)}{\Gamma \vdash s : \forall a. \alpha} \\ (\forall_e) \quad & \frac{\Gamma \vdash s : \forall a. \alpha}{\Gamma \vdash s : (\alpha[\beta/a])} \end{aligned}$$

Church-Kodierung

S. 84

- Natürliche Zahlen

$$\begin{aligned}\mathbb{N} &:= \forall a.(a \rightarrow a) \rightarrow a \rightarrow a \\ \text{zero} &: \mathbb{N} \\ \text{zero} &= \lambda f x.x \\ \text{suc} &: \mathbb{N} \rightarrow \mathbb{N} \\ \text{suc} &= \lambda n f x.f(n f x) \\ \text{fold} &: \forall a.(a \rightarrow a) \rightarrow a \rightarrow \mathbb{N} \rightarrow a \\ \text{fold} &= \lambda f x n.n f x \\ \text{add} &: \mathbb{N} \rightarrow \mathbb{N} \rightarrow \mathbb{N} \\ \text{add} &= \lambda n.\text{fold suc } n\end{aligned}$$

- Paare

$$\begin{aligned}(a \times b) &:= \forall r.(a \rightarrow b \rightarrow r) \rightarrow r \\ \text{pair} &: \forall ab.a \rightarrow b \rightarrow (a \times b) \\ \text{pair} &= \lambda x y f.f x y \\ \text{fst} &: \forall ab.(a \times b) \rightarrow a \\ \text{fst} &= \lambda p.p(\lambda x y.x) \\ \text{snd} &: \forall ab.(a \times b) \rightarrow b \\ \text{snd} &= \lambda p.p(\lambda x y.y)\end{aligned}$$

- Summen

$$\begin{aligned}(a + b) &:= \forall r.(a \rightarrow r) \rightarrow (b \rightarrow r) \rightarrow r \\ \text{inl} &: \forall ab.a \rightarrow (a + b) \\ \text{inl} &= \lambda x f g.f x \\ \text{inr} &: \forall ab.b \rightarrow (a + b) \\ \text{inr} &= \lambda y f g.g y \\ \text{case} &: \forall abs.(a \rightarrow s) \rightarrow (b \rightarrow s) \rightarrow (a + b) \rightarrow s \\ \text{case} &= \lambda f g s.s f g\end{aligned}$$

- Listen

$$\begin{aligned}\text{List } a &:= \forall r.r \rightarrow (a \rightarrow r \rightarrow r) \rightarrow r \\ \text{Nil} &: \forall a.\text{List } a \\ \text{Nil} &= \lambda u f.u \\ \text{Cons} &: \forall a.a \rightarrow \text{List } a \rightarrow \text{List } a \\ \text{Cons} &= \lambda x l u f.f x(l u f) \\ \text{len} &: \forall a.\text{List } a \rightarrow \mathbb{N} \\ \text{len} &= \lambda l.l \text{ zero } (\lambda x r.\text{suc } r)\end{aligned}$$

ML-Polymorphie

- Einschränkung von System F durch \forall nur auf oberster Ebene sowie Mehrfachinstanziierung polymorpher Funktionen nur in *let*-Konstrukt

S. 88

- Typen

$$\alpha, \beta := a \mid \alpha \rightarrow \beta$$

- Typschemata

$$S := \forall a_1, \dots, a_k. \alpha \quad (k \geq 0)$$

- Terme

$$t, s := x \mid t \ s \mid \lambda x. t \mid \text{let } x = t \text{ in } s$$

- Kontexte

$$\Gamma = (x_1 : S_1, \dots, x_n : S_n) \\ Cl(\Gamma, \alpha) = \forall a_1, \dots, a_k. \alpha \quad \text{für } FV(\alpha) \setminus FV(\Gamma) = \{a_1, \dots, a_k\}$$

- Typisierungsregeln

S. 88

$$(\forall_e) \frac{}{\Gamma \vdash x : \alpha[\beta_1/a_1, \dots, \beta_k/a_k]} (x : \forall a_1, \dots, \forall a_k. \alpha) \in \Gamma$$

$$(\rightarrow_i) \frac{\Gamma, x : \alpha \vdash t : \beta}{\Gamma \vdash \lambda x. t : \alpha \rightarrow \beta}$$

$$(\rightarrow_e) \frac{\Gamma \vdash t : \alpha \rightarrow \beta \quad \Gamma \vdash s : \alpha}{\Gamma \vdash ts : \beta}$$

$$(\text{let}) \frac{\Gamma \vdash t : \alpha \quad \Gamma, x : Cl(\Gamma, \alpha) \vdash s : \beta}{\Gamma \vdash \text{let } x = t \text{ in } s : \beta}$$

- Inversionslemma

S. 89

1. Wenn $\Gamma \vdash x : \alpha$, dann existieren Typen β_i und ein Typschema $S = \forall a_1 \dots \forall a_k. \gamma$, so dass $(x : S) \in \Gamma$ und $\alpha = \gamma[\beta_1/a_1, \dots, \beta_k/a_k]$
2. Wenn $\Gamma \vdash (\text{let } x = s \text{ in } t) : \alpha$, dann existiert ein Typ β mit $\Gamma \vdash s : \beta$ und $\Gamma, x : Cl(\Gamma, \beta) \vdash t : \alpha$

- erweiterter Algorithmus W mit Menge PT von Typgleichungen

$$PT(\Gamma; x; \alpha) = \{a \doteq \gamma[a'_1/a_1, \dots, a'_k/a_k] \mid (x : \forall a_1, \dots, \forall a_k. \gamma) \in \Gamma\}$$

$$PT(\Gamma; \lambda x. t; \alpha) = PT((\Gamma; x : a); t; b) \cup \{a \rightarrow b \doteq \alpha\} \text{ mit } a, b \text{ frisch}$$

$$PT(\Gamma; ts; \alpha) = PT(\Gamma; t; a \rightarrow \alpha) \cup PT(\Gamma; s; a) \text{ mit } a \text{ frisch}$$

$$PT(\Gamma; (\text{let } x = s \text{ in } t); \alpha) = PT(\Gamma\sigma, x : Cl(\Gamma\sigma, \sigma(b))); t; \alpha\sigma)$$

wobei $\sigma = mgu(PT(\Gamma; s; b))$ mit b frisch

Minimierung von deterministischen endlichen Automaten

1. Entferne aus Q alle nicht erreichbaren Zustände
2. Initialisiere R auf $\{(q_1, q_2) \mid q_1 \in F \Leftrightarrow q_2 \in F\}$
3. Suche ein Paar $(q_1, q_2) \in R$ und einen Buchstaben $a \in \Sigma$ mit

$$(\delta(a, q_1), \delta(a, q_2)) \notin R$$

Wenn kein solches Paar gefunden wird, gehe zu Schritt 4. Andernfalls entferne (q_1, q_2) aus R und fahre bei 3. fort.

4. Identifiziere alle Zustandspaare in R .

Unifikationsalgorithmus (Martelli/Montanari)

$S \cup \{x \doteq x\}$	$\rightarrow S$	(delete)
$S \cup \{f(E_1, \dots, E_n) \doteq f(D_1, \dots, D_n)\}$	$\rightarrow S \cup \{E_1 \doteq D_1, \dots, E_n \doteq D_n\}$	(decomp)
$S \cup \{f(E_1, \dots, E_n) \doteq g(D_1, \dots, D_k)\}$	$\rightarrow \perp$ (für $f \neq g$)	(conflict)
$S \cup \{E \doteq x\}$	$\rightarrow S \cup \{x \doteq E\}$ (für E keine Variable)	(orient)
$S \cup \{x \doteq E\}$	$\rightarrow \begin{cases} \perp & (\text{für } x \in FV(E), x \neq E) \\ S[E/x] \cup \{x \doteq E\} & (\text{für } x \notin FV(E), x \in FV(S)) \end{cases}$	(occurs)/(eli)