Merkzettel für „Theorie der Programmierung“

Marco Ammon

4. Oktober 2018

Termersetzungssysteme

Terminierung

Polynomordnungen

Konfluenz

Critical Pairs

λ-Kalkül

Ungetypt

Rekursion

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 $t \rightarrow_n t'$
  - $ts \rightarrow_n t's$m 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*

- **TODO**

- 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 **TODO** *Lem. 3.29 (41)*
- 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 $\sigma_2 \Leftrightarrow \exists \tau . \sigma_1 \tau = \sigma_2$ *GLoIn, S. 38*
  - Prinzipaltyp von Γ, t ist $\sigma(a)$ für allgemeinste Lösung σ von $\Gamma \vdash t : a$ (a frisch)
  - Algorithmus W (Hindley/Milner) *Alg. 3.31 (42)*
    - Menge PT von Typgleichungen

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

    - Typinferenz des Terms u mit leerem Kontext:

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

    $\Rightarrow$ Prinzipaltyp von u: $\mathrm{mgu}(\varepsilon)(a)$
- Subjektreduktion: Wenn $\Gamma \vdash t : \alpha$ und $t \to_\beta^* s$, 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)*

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

- 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)*

$$\alpha, \beta \coloneqq a \mid \alpha \to \beta \mid \forall a.\alpha \qquad\qquad (a \in V)$$

- Typisierung: **TODO:** $\lambda \to$, **TODO** *Def. 5.1 (84)*

Inhalt...

Church-Kodierung

S. 84

- Natürliche Zahlen

Todo

- Paare

Inhalt...

- Summen

Inhalt...

- Listen

Inhalt...

ML-Polymorphie

- Einschränkung von System F durch $\forall$ nur auf oberster Ebene sowie Mehrfachinstanziierung polymorpher Funktionen nur in *let*-Konstrukt *S. 88*
- Typen

$$\alpha, \beta \coloneqq a \mid \alpha \to \beta$$

- Typschemata

$$S \coloneqq \forall a_1, \ldots, a_k.\alpha \qquad\qquad (k \geq 0)$$

- Terme

$$t, s \coloneqq x \mid t\ s \mid \lambda x.t \mid let\ x = t\ in\ s$$

- Kontexte

$$\begin{aligned} \Gamma &= (x_1 : S_1, \ldots, x_n : S_n) \\ Cl(\Gamma, \alpha) &= \forall a_1, \ldots, a_k.\alpha \qquad\qquad \text{für } FV(\alpha) \backslash FV(\Gamma) = \{a_1, \ldots, a_k\} \end{aligned}$$

- Typisierungsregeln **TODO**
- Inversionslemma **TODO**
- erweiterter Algorithmus W **TODO**

Reguläre Ausdrücke

Unifikationsalgorithmus (Martelli/Montanari)

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