01.12.05

Theorie der Programmierung I

(Mitschrift von Simon Schöling und Lars Friedrich email@lars-friedrich-home.de)

Beispiel: $\alpha \to \alpha' = \alpha' \to \alpha$ hat allgemeinste Lösung s = $[\alpha/\alpha']$ denn:

- s ist eine Lösung
- Wenn s' eine Lösung ist, dann muss $s'(\alpha) = s'(\alpha')$ gelten. Dann gilt s'=ss', denn:
  - Es gilt: $(ss')(\alpha') =_{def.} (s(\alpha'))s' = \alpha s' = s'(\alpha) = s'(\alpha')$
  - Und für alle $\alpha'' \neq \alpha'$: $(ss')(\alpha'') =_{def.} (s(\alpha''))s' = \alpha''s' = s'(\alpha')$

Damit ist gezeigt: s ist allgemeiner als s'

Der Unifikationsalgorithmus (Robinson 1971)

Schreibweisen:

- $s \sqsubseteq s'$ bedeutet: s ist allgemeiner als s'
- Für $\tau \in$ Type sei $|\tau|$ die Länge der Zeichenreihe τ, tvar(τ) = Menge aller in τ vorkommenden Typvariablen, $Es =_{def} \{\tau_1 s = \tau_1's, \ldots, \tau_n s = \tau_n's\}$ (Anwendung einer Substitution s auf die Gleichungsmenge E)
- Für eine Menge von Typgleichungen $E = \{\tau_1 = \tau_1', \ldots, \tau_n = \tau_n'\}$ sei:
  - $size(E) =_{def} \sum^n_{i=1}(|\tau_i| + |\tau_i'|)$
  - $tvar(E) =_{def} \bigcup^n_{i=1}(tvar(\tau i) \cup tvar(\tau_i'))$ (die MEnge aller in E vorkommenden typvariablen)
  - $|tvar(E)| =_{def}$ Größe der Menge tvar(E) (d.h. die Anzahl der unterschieldichen Typvariablen in E)

Bemerkung: s ist Lösung von E $\Leftrightarrow$ Es enthält nur „triviale" Gleichungen, d.h. Gleichungen der Form $\tau=\tau$

Eingabe für den Unifikationsalgorithmus: Eine unendliche Menge E von Typgleichungen

Gewünschte Ausgabe:
- eine allgemeinste Lösung s von E, falls E lösbar
- „nicht unifizierbar", falls E nicht lösbar

Der Algorithmus wird als rekursive Funktion „unify" dargestellt.

Leere Gleichungsmenge:
(1) unify($\emptyset$) = [] (= die Substitution s mit $s(\alpha)=\alpha$ für alle $\alpha \in$ TVar)

Nichtleere Gleichungsmenge:
(2) unify($\tau=\tau \cup E$) = unify(E)
(3) unify($\{\tau_1 \to \tau_1' = \tau_2 \to \tau_2'\} \cup E$) = unify($\{\tau_1=\tau_2, \tau_1'=\tau_2'\}$)
(4) unify($\{\alpha=\tau\} \cup E$)[=unfify($\{\tau=\alpha\} \cup E$)] = { „nicht unifizierbar", wenn $\alpha \in tvar(E)$ und $\alpha \neq \tau$; $s_1 s_2$ mit $s_1=[\tau/\alpha]$ und s_2=unify(Es_1), wenn $\alpha \notin tvar(\tau)$ und $\alpha \neq \tau$
(5) unfiy($\{\tau_1=\tau_2\} \cup E$) = „nicht unifizierbar" sonst

Beispiel: Die Typgleichungen für $\lambda f.\ \lambda x.\ f(fx)$ waren:

(1) $\alpha_1=\alpha_5$
(2) $\alpha_1=\alpha_7$
(3) $\alpha_3=\alpha_8$
(4) $\alpha_7=\alpha_8 \to \alpha_6$
(5) $\alpha_5=\alpha_6 \to \alpha_4$
(6) $\alpha_2=\alpha_3 \to \alpha_4$
(7) $\alpha_0=\alpha_1 \to \alpha_2$

(1) ➔ $s_1=[\alpha_1/\alpha_5]$ ➔ aus Gleichung (5) wird (5') $\alpha_1=\alpha_6 \to \alpha_4$
(2) ➔ $s_2=[\alpha_1/\alpha_7]$ ➔ aus Gleichung (4) wird (4') $\alpha_1=\alpha_8 \to \alpha_6$
(3) ➔ $s_3=[\alpha_3/\alpha_8]$ ➔ aus Gleichung (4) wird (4'') $\alpha_1=\alpha_3 \to \alpha_6$
(4'') ➔ $s_4=[\alpha_3 \to \alpha_6/\alpha_1]$ ➔ aus Gleichung (5') wird (5'') $\alpha_3 \to \alpha_6=\alpha_6 \to \alpha_4$
➔ aus Gleichung (7) wird (7') $\alpha_0= \alpha_3 \to \alpha_6 \to \alpha_2$
(5'') ➔ wird zerlegt in (5a) $\alpha_3=\alpha_6$ und (5b) $\alpha_6=\alpha_4$
(5a) ➔ $s_5=[\alpha_6/\alpha_3]$ ➔ aus Gleichung (6) wird (6') $\alpha_2=\alpha_6 \to \alpha_4$
➔ aus Gleichung (7') wird (7'') $\alpha_0= \alpha_6 \to \alpha_6 \to \alpha_2$

(5b) ➔ $s_6=[\alpha_6/\alpha_4]$ ➔ aus Gleichung (6') wird (6'') $\alpha_2=\alpha_6\rightarrow\alpha_6$
(6'') ➔ $s_7=[\alpha_6\rightarrow\alpha_6/\alpha_2]$ ➔ aus Gleichung (7'') wird (7''') $\alpha_0= (\alpha_6\rightarrow\alpha_6)\rightarrow(\alpha_6\rightarrow\alpha_6)$
(7''') ➔ $s_8=[(\alpha_6\rightarrow\alpha_6)\rightarrow(\alpha_6\rightarrow\alpha_6)/\alpha_0]$ Es bleibt die leere Gleichungsmenge zu lösen ➔ $s_9[]$

Gesamtlösung $s=s_1\ldots s_9$ (Komposition)

Es gilt: $s = [(\alpha_6\rightarrow\alpha_6)\rightarrow(\alpha_6\rightarrow\alpha_6)/\alpha_0, \alpha_6\rightarrow\alpha_6/\alpha_1, \alpha_6\rightarrow\alpha_6/\alpha_2, \alpha_6/\alpha_3, \alpha_6/\alpha_4, \alpha_6\rightarrow\alpha_6/\alpha_5, \alpha_6\rightarrow\alpha_6/\alpha_7, \alpha_6/\alpha_8]$

Das eigentliche Ergebnis ist $s(\alpha_0) = (\alpha_6\rightarrow\alpha_6)\rightarrow(\alpha_6\rightarrow\alpha_6)$

$s(\alpha_0)$ ist allgemeinster Typ von $\lambda f.\ \lambda x.\ f(fx)$

Satz 14 **(Korrektheit des Unifikationsalgorithmus)**

unify(E) terminiert immer mit dem gewünschten Ergebnis, d.h.:

- mit einer allgemeinsten Löguns von E, falls E lösbar
- mit enicht unifizierbar sonst

Beweis: Es genügt folgendes zu zeigen

(a) unify(E) terminiert immer
(b) wenn unify(E) eine Substitution s liefert, dann ist s eine Lösung von E
(c) wenn E eine Lösung s' hat, dann liefert unify(E) eine Substitution sEs'

(a) Induktion über (|tvar(E)|, size(E)) bezüglich der lexikographischen Ordnung auf IN^2 (=Noethersche Ordnung)
Es ist zu zeigen, dass (|tvar(E)|, size(E)) bei jedem rekursiven Aufruf abnimmt.
Rekursive Aufrufe kommen nur in Fällen (2), (3), (4) des Algorithmus vor
(2) $|tvar(E)| \leq |tvar(\{\alpha=\tau\}\cup E)|$, $size(E) < size(\{\tau=\tau\}\cup E)$
(3) $|tvar(\{\tau_1=\tau_2, \tau_1'=\tau_2'\}\cup E)| = |tvar(\{\tau_1=\tau_1', \tau_2=\tau_2'\}\cup E)|$,
$size(\{\tau_1=\tau_2, \tau_1'=\tau_2'\}\cup E) < size(\{\tau_1=\tau_1', \tau_2=\tau_2'\}\cup E)$ (denn die „$\rightarrow$" entfallen)
(4) $|tvar(Es_1)| < |tvar(\{\alpha=\tau\}\cup E)|$ bzw. $\{\tau=\alpha\}$ denn: $s_1=[\tau/\alpha]$ und **$\alpha\notin tvar(\tau)$** ➔ deshalb kommt α in Es_1 nicht mehr vor
(Bem.: Der Test $\alpha\notin tvar(\tau)$ wist als „occurcheck" bekannt. Es garantiert die Terminierung.) (vgl. Prolog: Unifikation zur Laufzeit ohne occurcheck ➔ kann zur Endlosschleife führen.)

Wegen (a) wissen wir, dass die Rekursionstiefe für jeden Aufruf unify(E) endlich ist, also können wir (b) und (c) durch Induktion über die Rekursionstiefe beweisen.
Anders formuliert: Es ist nur noch zu zeigen: unify(E) liefert das richtige Ergebnis unter der Annahme der Terminierung („partielle Korrektheit").