Termersetzungssysteme

Ziel: Operationalisierung von Spezifikationen und Implementierung funktionaler Programmiersprachen

$spec = (sig, E)$ Wann ist T_{spec} berechenbare Algebra?

$(T_{spec})_s = \{[t] : t \in Term(sig)_s\}$
T_{spec} ist berechenbare Algebra wenn es eine berechenbare Funktion $rep : Term(sig) \rightarrow Term(sig)$ gibt, mit $rep(t) \in [t]$ eindeutiger Repräsentant in seiner Äquivalenzklasse.

Paradigma: Wähle als Repräsentanten minimale Objekte in der Äquivalenzklasse bezüglich einer Ordnung.

$f(x_1, ..., x_n) : ((T_{spec})_{s_1} \times ...(T_{spec})_{s_n}) \rightarrow (T_{spec})_s$
$f([r_1], ..., [r_n]) := [rep(f(rep(r_1), ..., (rep(r_n))]$

Termersetzungssysteme

Definition 9.1 *Regeln, Regelmengen, Reduktionsrelation*

- *Mengen von Variablen: Für $t \in Term_s(F, V)$ sei $V(t)$ die Menge der Variablen in t (Rek. Definition, immer endlich)*
  Beachte: $V(t) = \emptyset$ gdw t ist Grundterm.
- *Eine Regel ist ein Paar $(l, r), l, r \in Term_s(F, V)(s \in S)$ mit $Var(r) \subseteq Var(l)$*
  Schreibe: $l \to r$
- *Ein Regelsystem R ist eine Menge von Regeln.*
  R definiert eine Reduktionsrelation $\to_R$ auf $Term(F, V)$ durch:
  $t_1 \to_R t_2 \quad \text{gdw } \exists\, l \to r \in R, p \in O(t_1), \sigma \text{ Substitution } :$
  $$t_1|_p = \sigma(l) \wedge t_2 = t_1[\sigma(r)]_p$$
- *$(Term(F, V), \to_R)$ sei das von R definierte Reduktionssystem (Termersetzungssystem).*

Termersetzungssysteme

Ziel: Transformiere E in R, so dass $=_E = \overset{*}{\longleftrightarrow}_R$ gilt und $\rightarrow_R$ gutartige Terminierungs- und Konfluenzeigenschaften hat.
Etwa konvergent oder konfluent. Oft genügt es wenn diese Eigenschaften "nur" auf der Menge der Grundterme gelten.

Beachte:

- Die Bedingung $V(r) \subseteq V(l)$ in Regel $l \rightarrow r$ ist notwendig für die Terminierung.
  Gilt weder $V(r) \subseteq V(l)$ noch $V(l) \subseteq V(r)$ in einer Gleichung $l = r$ einer Spezifikation, so hat man wohl überflüssige Variablen bei der Funktionsdefinition verwendet.
- $\rightarrow_R$ ist verträglich mit der Substitution und der Termersetzung. D.h. Aus $s \rightarrow_R t$ folgt, $\sigma(s) \rightarrow_R \sigma(t)$ und $u[s]_p \rightarrow_R u[t]_p$
- Insbesondere $=_R = \overset{*}{\longleftrightarrow}_R$

Matching oder Treffende Substitution

Definition 9.2 *Seien $l, t \in Term_s(F, V)$. Eine Substitution σ heißt Match (treffende Substitution) von l auf t, falls $\sigma(l) = t$*

Folgerung 9.1 *Es gilt:*

- $\forall\ \sigma$ *Substitution* $O(l) \subseteq O(\sigma(l))$.
- $\exists \sigma : \sigma(l) = t$ *gdw es gilt für* σ *definiert durch*
  $\forall u\ O(l) : l|_u = x \in V \rightsquigarrow u \in O(t) \wedge \sigma(x) = t|_u$
  σ *ist Substitution* $\wedge\ \sigma(l) = t$.
- *Gibt es eine solche Substitution, so ist sie eindeutig auf* $V(l)$. *Existenz und gegebenfalls Berechnung sind effektiv.*
- *Es ist entscheidbar ob* t *mit Regel* $l \rightarrow r$ *reduzierbar.*
- *Ist* R *endlich, so* $\Delta(s) = \{t : s \rightarrow_R t\}$ *endlich un berechenbar.*

Beispiele

Beispiel 9.1 *Ganze Zahlen*

$$\begin{aligned} sig &: 0 :\longrightarrow int \\ & s, p : int \longrightarrow int \\ if0 &: int, int, int \longrightarrow int \\ & F : int, int \longrightarrow int \end{aligned}$$

$$\begin{aligned} eqns &: 1 :: p(0) = 0 \\ & 2 :: p(s(x)) = x \\ & 3 :: if0(0, x, y) = x \\ & 4 :: if0(s(z), x, y) = y \\ & 5 :: F(x, y) = if0(x, 0, F(p(x), F(x, y))) \end{aligned}$$

Interpretation: $\langle \mathbb{N}, ..., \rangle$ *spec- Algebra mit Funktionen*
$O_{\mathbb{N}} = 0, s_{\mathbb{N}} = \lambda n.\ n + 1,$
$p_{\mathbb{N}} = \lambda n.\ if\ \ n = 0\ \ then\ \ 0\ \ else\ \ n - 1\ \ fi$
$if0_{\mathbb{N}} = \lambda i, j, k.\ if\ \ i = 0\ then\ \ j\ else\ \ k\ \ fi$
$F_{\mathbb{N}} = \lambda m, n.\ 0$

Orientiere Gleichungen von links nach rechts $\rightsquigarrow$ *Regeln R (Variablenbedingung ist erfüllt). Ist R terminierend? Nicht mit syntaktischer Ordnung, da linke Seite in rechter Seite vorkommt.*

Beispiel (Fort.)

Reduktionsfolge:

$$F(s(0),0) \to_5 \underbrace{if0(s(0),0,\underbrace{F(\underbrace{p(s(0))}_{2},\underbrace{F(s(0),0)}_{5})}_{4})}_{}$$

$$\to_4 \underbrace{F(\underbrace{p(s(0))}_{},\underbrace{F(s(0),0)}_{})}_{5}$$

$$\to_2 \underbrace{F(0,\underbrace{F(s(0),0)}_{})}_{5}$$

$$\to_5 \underbrace{if0(0,0,\underbrace{F(\underbrace{p(0)}_{},\underbrace{F(0,\underbrace{F(s(0),0)}_{})}_{})}_{})}_{3} \to_3 0$$

Äquivalenz

Definition 9.3 *Seien* $spec = (sig, E), spec' = (sig, E')$ *Spezifikationen. Sie sind äquivalent falls* $=_E == =_{E'}$, *d.h.* $T_{spec} = T_{spec'}$.
Ein Regelsystem R *über* sig *ist äquivalent zu* E, *falls* $=_E = \overset{*}{\longleftrightarrow}_R$.

Beachte: Ist R endlich, konvegent, äquivalent zu E so ist die E Gleichheit entscheidbar

$$s =_E t \quad gdw \; s \downarrow = t \downarrow \text{ d.h. identische NF}$$

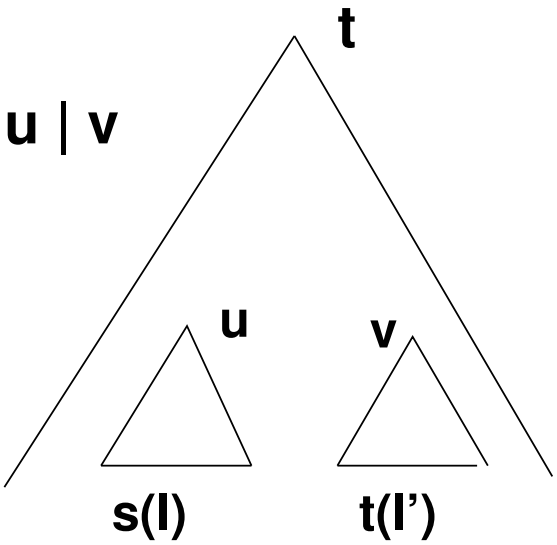
Für funktionale Programme und Berechnungen in T_{spec} genügt die Grundkonvergenz, d.h. Konvergenz auf Grundtermen.
Probleme: Entscheide ob

- R noethersch (grundnoethersch)
- R konfluent (grundkonfluent)
- R Wie transformiert man E in äquivalentes R mit diesen Eigenschaften?

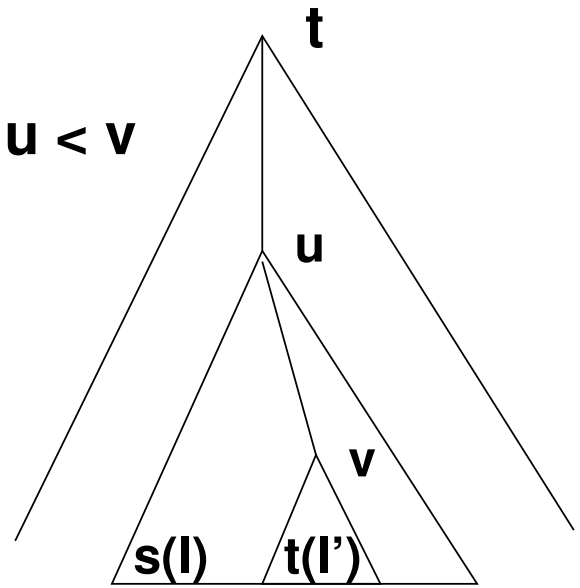
Entscheidbarkeitsfragen

Für endliche Grundtermersetzungssysteme sind die Probleme entscheidbar.

Für terminierende Systeme genügt es lokale Konfluenz zu entscheiden, d.h. aus $t_1 \leftarrow t \rightarrow t_2$ folgt $t_1 \downarrow t_2$.



Zusammenführbar



Kritische Paare

Kritische Paare

Betrachte Gruppenaxiome:

$$\underbrace{(x' \cdot y') \cdot z}_{l_1} \to x' \cdot (y' \cdot z) \quad \text{und} \quad \underbrace{x \cdot x^{-1}}_{l_2} \to 1.$$

"Überlappungen" (Superpositionen)

$$\begin{array}{ccc} & (x \cdot x^{-1}) \cdot z & \\ & \swarrow l_2 \qquad \searrow l_1 & \\ 1 \cdot z & & x \cdot (x^{-1} \cdot z) \end{array} \qquad \begin{array}{ccc} & (x \cdot y) \cdot (x \cdot y)^{-1} & \\ & \swarrow l_2 \qquad \searrow l_1 & \\ 1 & & x \cdot (y \cdot (x \cdot y)^{-1}) \end{array}$$

- $l_1|_1$ ist "unifizierbar" mit l_2 mit Substitution
  $\sigma :: \{x' \leftarrow x, y' \leftarrow x^{-1}, x \leftarrow x\} \rightsquigarrow \sigma(l_1|_1) = \sigma(l_2)$
- l_1 "unifizierbar" mit l_2 mit Substitution
  $\sigma :: \{x' \leftarrow x, y' \leftarrow y, z \leftarrow (x \cdot y)^{-1}, x \leftarrow x \cdot y\} \rightsquigarrow \sigma(l_1) = \sigma(l_2)$

Subsumption, Unifikation

Definition 9.4 *Subsumptionsordnung auf Terme:*
$s \preceq t$ *gdw* $\exists\sigma$ *Substitution* $: \sigma(s)$ *Teilterm von* t
$s \approx t$ *gdw* $(s \preceq t \wedge t \preceq s)$
$s \succ t$ *gdw* $(t \preceq s \wedge \neg(s \preceq t))$
$\succeq$ *ist noethersche Partialordnung auf* $Term(F, V)$ *Beweis!.*

Beachte:
$O(\sigma(t)) = O(t) \cup \bigcup_{w \in O(t): t|_w = x \in V} \{wv : v \in O(\sigma(x))\}$

Verträglichkeitseigenschaften:
$t|_u = t' \rightsquigarrow \sigma(t)|_u = \sigma(t')$
$t|_u = x \in V \rightsquigarrow \sigma(t)|_{uv} = \sigma(x)|_v \quad (v \in O(\sigma(x)))$
$\sigma(t)[\sigma(t')]_u = \sigma(t[t']_u)$ für $u \in O(t)$

Definition 9.5 $s, t \in Term(F, V)$ *sind gdw unifizierbar wenn es Substitution* σ *gibt mit* $\sigma(s) = \sigma(t)$. σ *heißt Unifikator von* s *und* t.

Unifikation, Allgemeinster Unifikator

Definition 9.6 *Sei* $V' \subseteq V, \sigma, \tau$ *Substitutionen.*

- $\sigma \preceq \tau(V')$ *gdw* $\exists \rho$ *Substitution* : $\rho \circ \sigma|_{V'} = \tau|_{V'}$
  Sprechweise: σ *ist allgemeiner als* τ *auf* V'
- $\sigma \approx \tau(V')$ *gdw* $\sigma \preceq \tau(V') \wedge \tau \preceq \sigma(V')$
- $\sigma \prec \tau(V')$ *gdw* $\tau \preceq \sigma(V') \wedge \neg(\sigma \preceq \tau(V'))$
- *Beachte:* $\prec$ *ist noethersche Partialordnung auf den Substitutionen.*

Frage: *Seien* s, t *unifizierbar. Gibt es einen allgemeinsten Unifikator* $mgu(s, t)$ *auf* $V = Var(s) \cup Var(t)$*. D.h. für Unifikator* σ *für* s, t *gilt stets* $mgu(s, t) \preceq \sigma(V)$*?*
Ist $mgu(s, t)$ *eindeutig? (bis auf Umbenennung von Variablen).*

Unifikationsproblem und seine Lösung

Definition 9.7

- *Ein Unifikationsproblem besteht aus einer Menge* $E = \{s_i \stackrel{?}{=} t_i : i = 1, ..., n\}$ *von Gleichungen.*
- σ *heißt Lösung (oder Unifikator) falls* $\sigma(s_i) = \sigma(t_i)$ *für* $i = 1, ..., n$.
- *Gilt* $\tau \succeq \sigma(Var(E))$ *für jede Lösung* τ *von* E, *so* $mgu(E) := \sigma$ *allgemeinste Lösung oder allgemeinster Unifikator.*
- $Sol(E)$ *sei die Menge der Lösungen von* E.
  E *und* E' *sind äquivalent, falls* $Sol(E) = Sol(E')$.
- E' *ist in gelöster Form, falls*
  $E' = \{x_j \stackrel{?}{=} t_j : x_i \neq x_j \ (i \neq j),\ x_i \notin Var(t_j) \ (1 \leq i \leq j \leq m)\}$
- E' *ist gelöste Form zu* E, *falls* E' *in gelöster Form und äquivalent zu* E *mit* $Var(E') \subseteq Var(E)$.

Beispiele

Beispiel 9.2 *Betrachte*

- $s = f(x, g(x, a)) \quad \stackrel{?}{=} \quad f(g(y, y), z) = t$
  $\rightsquigarrow x \stackrel{?}{=} g(y, y) \qquad g(x, a) \stackrel{?}{=} z \qquad$ *split*
  $\rightsquigarrow x \stackrel{?}{=} g(y, y) \qquad g(g(y, y), a) \stackrel{?}{=} z \qquad$ *merge*
  $\rightsquigarrow \sigma :: x \leftarrow g(y, y) \qquad z \leftarrow g(g(y, y), a) \qquad y \leftarrow y$
- $f(x, a) \stackrel{?}{=} g(a, z)$ *Unlösbar (nicht unifizierbar).*
- $x \stackrel{?}{=} f(x, y)$ *Unlösbar da $f(x, y)$ nicht x frei.*
- $x \stackrel{?}{=} f(a, y) \rightsquigarrow$ *Lösung $\sigma :: x \leftarrow f(a, y)$ ist allgemeinste Lösung.*

Inferenzsystem für die Unifikation

Definition 9.8 *Kalkül* ***UNIFY****. Sei* $\sigma =$ *Bindungsmenge.*

(1) Löschen

$$\frac{(E \cup \{s \stackrel{?}{=} s\}, \sigma)}{(E, \sigma)}$$

(2) Split (Zerlegen)

$$\frac{(E \cup \{f(s_1, \ldots, s_m) \stackrel{?}{=} g(t_1, \ldots, t_n)\}, \sigma)}{\lightning \text{ (Unlösbar)}} \quad \text{falls } f \neq g$$

$$\frac{(E \cup \{f(s_1, \ldots, s_m) \stackrel{?}{=} f(t_1, \ldots, t_m)\}, \sigma)}{(E \cup \{s_i \stackrel{?}{=} t_i : i = 1, \ldots, m\}, \sigma)}$$

(3) Merge (Lösen)

$$\frac{(E \cup \{x = t\}, \sigma)}{(\tau(E), \sigma \cup \tau)} \quad \text{falls } x \notin Var(t), \tau = \{x \stackrel{?}{=} t\}$$

"occur check"

$$\frac{(E \cup \{x = t\}, \sigma)}{\lightning \text{ (Unlösbar)}} \quad \text{falls } x \in Var(t) \wedge x \neq t$$

Unifikationsalgorithmen

Unifikationsalgorithmen basierend auf diesem Kalkül starten stets mit $(E_0, S_0) := (E, \emptyset)$ liefern eine Folge $(E_0, S_0) \vdash_{UNIFY} \ldots \vdash_{UNIFY} (E_n, S_n)$ Sie sind erfolgreich falls sie mit $E_n = \emptyset$, unerfolgreich falls sie mit $S_n = \lightning$ enden. S_n beschreibt eine Substitution σ die $Sol(S_n)$ und somit auch $Sol(E)$ erfasst.

Lemma 9.1 *Korrektheit.*
Jede Folge $(E_0, S_0) \vdash_{UNIFY} \ldots \vdash_{UNIFY} (E_n, S_n)$ bricht ab: Entweder mit $\lightning$ (Unlösbar, nicht unifizierbar) oder mit $(\emptyset, S)$ und S ist eine gelöste Form zu E.

Beachte: Darstellungen der Substitution in gelöster Form können recht unterschiedlich sein (Aufwand!!):

$s \stackrel{?}{=} f(x_1, \ldots, x_n) \qquad t \stackrel{?}{=} f(g(x_0, x_0), \ldots, g(x_{n-1}))$

$S = \{x_i \stackrel{?}{=} g(x_{i-1}, x_{i-1}) : i = 1, \ldots, n\}$ und

$S_1 = \{x_{i+1} \stackrel{?}{=} t_i : t_0 = g(x_0, x_0), t_{i+1} = g(t_i, t_i)\ i = 0, \ldots, n-1\}$

sind beide in gelöster Form. Die Länge der t_i wächst exponentiell mit i.

Beispiel

Beispiel 9.3 *Ausführung:*

$$f(x, g(a, b)) \stackrel{?}{=} f(g(y, b), x)$$

E_i	S_i	*Regel*
$f(x, g(a, b)) \stackrel{?}{=} f(g(y, b), x)$	$\emptyset$	
$x \stackrel{?}{=} g(y, b), x \stackrel{?}{=} g(a, b)$	$\emptyset$	*split*
$g(y, b) \stackrel{?}{=} g(a, b)$	$x \stackrel{?}{=} g(a, b)$	*lösen*
$y \stackrel{?}{=} a, b \stackrel{?}{=} b$	$x \stackrel{?}{=} g(a, b)$	*split*
$b \stackrel{?}{=} b$	$x \stackrel{?}{=} g(a, b), y \stackrel{?}{=} a$	*lösen*
	$x \stackrel{?}{=} g(a, b), y \stackrel{?}{=} a$	*löschen*

Lösung: $mgu = \sigma = \{x \leftarrow g(a, b), y \leftarrow a\}$

Kritische Paare - Lokale Konfluenz

Definition 9.9 *Sei R Regelsystem und $l_1 \to r_1, l_2 \to r_2 \in R$ mit $V(l_1) \cap V(l_2) = \emptyset$ (Umbenennung von Variablen falls nötig, u.U. $l_1 \approx l_2$ bzw. $l_1 \to r_1 \approx l_2 \to r_2$).*
Sei $u \in O(l_1)$ mit $l_1|_u \notin V, \sigma = mgu(l_1|_u, l_2)$ existiere.
$\sigma(l_1)$ heißt dann eine Überlappung (Superposition) von $l_2 \to r_2$ in $l_1 \to r_1$ und $(\sigma(r_1), \sigma(l_1[r_2]_u))$ ist das zugehörige kritische Paar zu $l_1 \to r_1, l_2 \to r_2, u \in O(l_1)$, sofern $\sigma(r_1) \neq \sigma(l_1[r_2]_u)$.

$CP(R)$ sei die Menge aller kritischen Paare die man mit Regeln aus R bilden kann.

Beachte: Die Überlappung und somit die Menge der kritischen Paare ist bis auf Umbenennung der Variablen eindeutig.

Eigenschaften

- Seien σ, τ Substitutionen, $x \in V$, $\sigma(y) = \tau(y)$ für $y \neq x$ und $\sigma(x) \rightarrow_R \tau(x)$. Dann gilt für jeden Term t:

$$\sigma(t) \xrightarrow{*}_R \tau(t)$$

- Seien $l_1 \rightarrow r_1, l_2 \rightarrow r_2$ Regeln, $u \in O(l_1), l|_u = x \in V$. Sei $\sigma(x)|_w = \sigma(l_2)$, d.h. $\sigma(l_2)$ wird durch $\sigma(x)$ eingeführt. Dann gilt $t_1 \downarrow_R t_2$ für

$$t_1 := \sigma(r_1) \leftarrow \sigma(l_1) \rightarrow \sigma(l_1)[\sigma(r_2)]_{uw} =: t_2$$

Lemma 9.2 *Critical-Pair Lemma von Knuth/Bendix*
Sei R ein Regelsystem. Dann gilt:

Aus $t_1 \leftarrow_R t \rightarrow_R t_2$ folgt $t_1 \downarrow_R t_2$ oder $t_1 \leftrightarrow_{CP(R)} t_2$.