Operationen mit (Teil)Termen

$t[p]$: Teilterm von t an Position p

Beispiele:

$$\blacktriangleright f(g(f(f(c)), c))[0, 0] = f(f(c))$$

$$\blacktriangleright f(g(f(f(c)), c))[0, 1] = \dots$$

(induktive) Definition (über die Länge von p):

$$\text{IA } p = [] : t[] = t$$

$$\text{IS } p = ip' : f(t_1, \dots, t_n)[p] = t_i[p']$$

$t[p := s]$: in t Teilterm $t[p]$ an Position p durch Term s ersetzt

Beispiele:

$$\blacktriangleright f(g(f(f(c)), c))[[0, 0] := c] = f(g(c, c))$$

$$\blacktriangleright f(g(f(f(c)), c))[[0, 1] := f(c)] = \dots$$

(induktive) Definition (über die Länge von p):

$$\text{IA } p = [] : t[] := s = s$$

$$\text{IS } p = ip' : f(t_1, \dots, t_n)[p := s] = f(t_1, \dots, t_i[p' := s], \dots, t_n)$$

Operationen mit Variablen in Termen

Menge $\text{Term}(\Sigma, \mathbb{X})$ aller Terme über Signatur Σ mit Variablen aus $\mathbb{X}$
Beispiel: $\Sigma = \{(Z, 0), (S, 1), (f, 2)\}$, $\mathbb{X} = \{x, y\}$, $f(Z, y) \in \text{Term}(\Sigma, \mathbb{X})$.

Substitution σ : partielle Abbildung $\mathbb{X} \rightarrow \text{Term}(\Sigma, \mathbb{X})$

Beispiel: $\sigma_1 = \{(y, S(Z))\}$ ($\sigma_1(x)$ ist nicht definiert)

Anwendung einer Substitution auf einen Term: $t\sigma$

jedes Vorkommen von $x \in \mathbb{X}$ in t ersetzt durch $\sigma(x)$

Beispiel: $f(Z, y)\sigma_1 = f(Z, S(Z))$

Definition von $t\sigma$ durch Induktion über t :

IA $t = x \in \mathbb{X}$:

$t\sigma = \sigma(x)$, falls definiert, sonst x

IS $t = f(t_1, \dots, t_n) : t\sigma = f(t_1\sigma, \dots, t_n\sigma)$

Termersetzungssysteme

Daten : Terme (ohne Variablen)

Regel : Paar (l, r) von Termen mit Variablen
(häufig notiert als $l \rightarrow r$)

Programm R : Menge von Regeln

Bsp: $R = \{f(Z, y) \rightarrow y, f(S(x), y) \rightarrow S(f(x, y))\}$

Relation $\rightarrow_R$: Menge aller Paare $t \rightarrow t'$ mit

- ▶ es existiert $(l, r) \in R$
- ▶ es existiert Position p in t
- ▶ es existiert Substitution
 $\sigma : (\text{var}(l) \cup \text{var}(r)) \rightarrow \text{Term}(\Sigma)$
- ▶ so dass $t[p] = l\sigma$ und $t' = t[p := r\sigma]$.

Termersetzungssysteme als Programme

- ▶ $\rightarrow_R$ beschreibt **einen** Schritt der Rechnung von R ,
- ▶ transitive Hülle $\rightarrow_R^*$ beschreibt **Folge** von Schritten.
- ▶ **Resultat** einer Rechnung ist Term in R -Normalform (ohne $\rightarrow_R$ -Nachfolger)

Dieses Berechnungsmodell ist im allgemeinen

nichtdeterministisch $R_1 = \{C(x, y) \rightarrow x, C(x, y) \rightarrow y\}$
(ein Term kann mehrere $\rightarrow_R$ -Nachfolger haben,
ein Term kann mehrere Normalformen erreichen)

nicht terminierend $R_2 = \{p(x, y) \rightarrow p(y, x)\}$
(es gibt eine unendliche Folge von $\rightarrow_R$ -Schritten,
es kann Terme ohne Normalform geben)

Konstruktor-Systeme

Für TRS R über Signatur Σ :

Symbol $s \in \Sigma$ heißt

definiert, wenn $\exists(l, r) \in R : l[] = s(\dots)$

Konstruktor, sonst

Das TRS R heißt **Konstruktor-TRS**, falls die definierten Symbole links **nur** in den Wurzeln vorkommen (rechts egal)

Übung: diese Eigenschaft formal spezifizieren

Beispiele:

- ▶ $R_1 = \{a(b(x)) \rightarrow b(a(x))\}$ über $\Sigma_1 = \{(a, 1), (b, 1)\}$,
- ▶ $R_2 = \{f(f(x, y), z) \rightarrow f(x, f(y, z))\}$ über $\Sigma_2 = \{(f, 2)\}$:

definierte Symbole? Konstruktoren? Konstruktor-System?

Funktionale Programme sind ähnlich zu Konstruktor-TRS.

Funktionale Programme

... sind spezielle Term-Ersetzungssysteme.

Ausdruck = Term über einer Signatur (Baumstruktur)

Jeder Ausdruck hat

- ▶ einen Typ und
- ▶ einen Wert

Beispiel: $\Sigma = \{+ :: (\mathbb{N} \times \mathbb{N}) \rightarrow \mathbb{N}, 0 :: \mathbb{N}, 1 :: \mathbb{N}, 2 :: \mathbb{N}, \dots\}$

Ausdruck 2 hat Typ $\mathbb{N}$ und Wert 2

Ausdruck $2 + ((1 + 3) + 2)$ hat Typ $\mathbb{N}$ und Wert 8

(rekursive) Berechnung des Wertes = Ersetzung von Teiltermen

$$2 + ((\underline{1 + 3}) + 2) \rightarrow 2 + (\underline{4 + 2}) \rightarrow (\underline{2 + 6}) \rightarrow 8$$

heißt **Reduktion** des Ausdrucks

Funktionale Programme

Funktionsdeklarationen

`double :: Int -> Int` (Typdeklaration)

`double x = x + x` (Funktionsdefinition)

definiert das Termersetzungssystem mit einer Regel

$$R = \{d(x) \rightarrow x + x\}$$

Reduktion des Ausdrucks `double 3`:

$$d(3) \rightarrow_R 3 + 3 \rightarrow 6$$

Reduktion des Ausdrucks `double (double 3)`:

$$\underline{d}(d(3)) \rightarrow_R \underline{d}(3) + d(3) \rightarrow_R (3+3) + \underline{d}(3) \rightarrow_R (3+3) + (3+3) \rightarrow 12$$

oder

$$d(\underline{d}(3)) \rightarrow_R d(3 + 3) \rightarrow_R (3 + 3) + (3 + 3) \rightarrow 12$$

Auswertung = Folge von Ersetzungsschritten $\rightarrow_R^*$

Resultat = Normalform (hat keine $\rightarrow_R$ -Nachfolger)