

3.2 Termersetzungssysteme

Eine Motivation für die Entwicklung der Theorie der Termersetzungssysteme war die Präzisierung des Rechnens mit Gleichungsspezifikationen.

Beispiel 3.1 (Gruppenaxiome) *Betrachten wir die aus der Mathematik bekannten Gruppenaxiome:*

$$\begin{aligned} x * 1 &= x & (1) \\ x * x^{-1} &= 1 & (2) \\ (x * y) * z &= x * (y * z) & (3) \end{aligned}$$

Rechnen in der Mathematik bedeutet ja, Gleiches durch Gleiches zu ersetzen (auch in beliebiger Richtung!). Dies ist häufig notwendig, um bestimmte Eigenschaften zu zeigen.

*Als Beispiel wollen wir zeigen, dass für alle Gruppen gilt: $1 * x = x$*

$$\begin{aligned} 1 * x &= (x * x^{-1}) * x & (2) \\ &= x * (x^{-1} * x) & (3) \\ &= x * (x^{-1} * (x * 1)) & (1) \\ &= x * (x^{-1} * (x * (x^{-1} * (x^{-1})^{-1}))) & (2) \text{ für } x^{-1} \text{ statt } x \\ &= x * (x^{-1} * ((x * x^{-1}) * (x^{-1})^{-1})) & (3) \\ &= x * (x^{-1} * (1 * (x^{-1})^{-1})) & (2) \\ &= x * ((x^{-1} * 1) * (x^{-1})^{-1}) & (3) \\ &= x * (x^{-1} * (x^{-1})^{-1}) & (1) \\ &= x * 1 & (2) \\ &= x & (1) \end{aligned}$$

Die Aspekte dieser Berechnung, die wir nachfolgend präzisieren wollen, sind:

- Termstrukturen, Anwendungen von Gleichungen in Teiltermen, um diese umzuformen
- „Pattern matching“: Ersetzen von Variablen in Gleichungen durch andere Terme
- Gleichungen in beide Richtungen anwenden

Nachfolgend wollen wir daher **Termersetzung als Modell zum Rechnen mit Funktionen** genauer einführen. Im Gegensatz zu allgemeinen Berechnungen in mathematischen Beweisen werden hier

- Gleichungen nur von links nach rechts angewendet, und
- linke Seiten so eingeschränkt, damit effektive Auswertungsstrategien definiert werden können.

Nachfolgend werden wir daher die notwendigen Details genau definieren.

Definition 3.5 Wir legen die folgenden Grundbegriffe für Termersetzungssysteme fest:

- **Signatur:** $\Sigma = (S, F)$ mit
 - S : Menge von **Sorten** (\approx Basistypen $\text{Nat}, \text{Bool}, \dots$)
 - F : Menge von **Funktionssymbolen** $f :: s_1, \dots, s_n \rightarrow s, n \geq 0$
($c :: \rightarrow s$ heißen auch **Konstanten**)

- **Variablen** haben auch Sorten, d.h. eine Variablenmenge

$$V = \{x :: s \mid x \text{ Variablensymbol}, s \in S\}$$

hat die Eigenschaft $x :: s, x :: s' \in V \Rightarrow s = s'$
(wir erlauben also kein Überladen/Overloading bei Variablen)

- Die Menge der **Terme** über Σ und V der Sorte s wird mit $T(\Sigma, V)_s$ bezeichnet und ist wie folgt definiert:

- $x \in T(\Sigma, V)_s$ falls $x :: s \in V$
- $f(t_1, \dots, t_n) \in T(\Sigma, V)_s$ falls $f :: s_1, \dots, s_n \rightarrow s \in F$ (wobei $\Sigma = (S, F)$) und $t_i \in T(\Sigma, V)_{s_i}$ ($i = 1, \dots, n$)

Notation: Schreibe c statt $c()$ für Konstanten

- Die Menge aller Variablen in einem Term t wird mit $\text{Var}(t)$ bezeichnet und ist definiert durch

$$\begin{aligned} \text{Var}(x) &= \{x\} \\ \text{Var}(f(t_1, \dots, t_n)) &= \text{Var}(t_1) \cup \dots \cup \text{Var}(t_n) \end{aligned}$$

- t heißt **Grundterm** falls $\text{Var}(t) = \emptyset$
- t heißt **linear** falls keine Variable in t mehrfach vorkommt.

Beispiele: $f(x, y)$ ist linear, $f(x, x)$ ist nicht linear.

- Ein **Termersetzungssystem (TES)** ist eine Menge von **Regeln** der Form $l \rightarrow r$ mit $l, r \in T(\Sigma, V)_s$ (für ein $s \in S$) und $\text{Var}(r) \subseteq \text{Var}(l)$.

Hierbei wird auch l als **linke Seite** und r als **rechte Seite** der Regel bezeichnet.

Als Beispiel wollen wir ein Termersetzungssystem definieren, das die Addition auf natürlichen Zahlen darstellt (wobei natürliche Zahlen aus 0 und s (uccessor) aufgebaut sind). Die Signatur ist:

$$S = \text{Nat}, F = \{0 :: \rightarrow \text{Nat}, s :: \text{Nat} \rightarrow \text{Nat}, + :: \text{Nat}, \text{Nat} \rightarrow \text{Nat}\}$$

Die Regeln des Termersetzungssystems sind:

$$\begin{aligned} 0 + n &\rightarrow n \\ s(m) + n &\rightarrow s(m + n) \end{aligned}$$

Dieses Termersetzungssystem hat die folgende intuitive Semantik: Regeln sind Gleichungen, die jedoch nur von links nach rechts angewendet werden.

Unser Ziel ist die Beschreibung des Rechnens mit Termersetzungssystemen, wie z.B. ein Reduktionsschritt:

$$s(0) + s(0) \rightarrow s(s(0))$$

Zur präzisen Definition eines solchen Reduktionsschrittes ist es notwendig, Begriffe wie die Anwendung einer Regel (*Substitution*) an einer beliebigen *Position* zu beschreiben. Aus diesem Grund definieren wir:

Definition 3.6 (Substitution, Position, Teilterm, Ersetzung)

- Eine **Substitution** σ ist eine Abbildung $\sigma : V \rightarrow T(\Sigma, V)$ mit der Eigenschaft $\sigma(x) \in T(\Sigma, V)_s$ für alle $x :: s \in V$, wobei der **Definitionsbereich**

$$\text{Dom}(\sigma) = \{x \mid \sigma(x) \neq x\}$$

endlich ist. Wir notieren Substitutionen in der Form:

$$\sigma = \{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$$

Dies bezeichnet die Substitution

$$\sigma(x) = \begin{cases} t_i & \text{falls } x = x_i \\ x & \text{sonst} \end{cases}$$

Wir können eine Substitution einfach auf Termen erweitern:

Die Abbildung $\sigma : T(\Sigma, V) \rightarrow T(\Sigma, V)$ wird definiert durch

$$\sigma(f(t_1, \dots, t_n)) = f(\sigma(t_1), \dots, \sigma(t_n))$$

- **Positionen** in einem Term sind Bezeichner für Teilterme, ausgedrückt durch Zahlenfolgen.

$\mathcal{P}os(t)$ bezeichnet die Menge aller Positionen in einem Term t :

- $\epsilon \in \mathcal{P}os(t)$ (Wurzelposition, wobei ϵ : leere Folge)
- Falls $p \in \mathcal{P}os(t_i)$, dann ist auch $i \cdot p \in \mathcal{P}os(f(t_1, \dots, t_n))$

- Der **Teilterm** bzw. **Unterterm** von t an der Position $p \in \text{Pos}(t)$ wird mit $t|_p$ bezeichnet und ist wie folgt definiert:

$$\begin{aligned} t|_\epsilon &= t \\ f(t_1, \dots, t_n)|_{i \cdot p} &= t_i|_p \end{aligned}$$

- Die **Ersetzung** des Teilterms an der Position p im Term t durch s wird mit $t[s]_p$ bezeichnet und ist wie folgt definiert:

$$\begin{aligned} t[s]_\epsilon &= s \\ f(t_1, \dots, t_n)[s]_{i \cdot p} &= f(t_1, \dots, t_{i-1}, t_i[s]_p, t_{i+1}, \dots, t_n) \end{aligned}$$

- Zum Vergleich von Positionen verwenden wir die folgenden Begriffe:
 - $p \leq q$ (p ist **über** q) falls p Präfix von q , d.h. $\exists r$ mit $p \cdot r = q$
 - p und q sind **disjunkt** $:\Leftrightarrow$ es gilt weder $p \leq q$ noch $q \leq p$
 - p heißt **links** von q $:\Leftrightarrow p = p_0 \cdot i \cdot p_1$ und $q = p_0 \cdot j \cdot q_1$ mit $i < j$

Nun können wir die Ersetzungsrelation bzgl. eines Termersetzungssystems R definieren:

Definition 3.7 (Termersetzung) Sei R ein TES. Dann ist die Reduktionsrelation (bzgl. R)

$$\rightarrow_R \subseteq T(\Sigma, V) \times T(\Sigma, V)$$

wie folgt definiert:

$$t_1 \rightarrow_R t_2 \quad :\Leftrightarrow \quad \exists l \rightarrow r \in R, \text{ Position } p \in \text{Pos}(t_1) \text{ und Substitution } \sigma \text{ mit } t_1|_p = \sigma(l) \text{ und } t_2 = t_1[\sigma(r)]_p$$

$t_1 \rightarrow_R t_2$ wird auch **Reduktionsschritt** oder **Ersetzungsschritt** genannt. $\sigma(l)$ ist eine „passende“ linke Regelseite. In diesem Fall wird $t_1|_p$ auch **Redex** genannt.

Notation:

$$\begin{aligned} t_1 \rightarrow_{p,l \rightarrow r} t_2 & \quad \text{falls Position und Regel wichtig} \\ t_1 \rightarrow t_2 & \quad \text{falls } R \text{ fest gewählt ist} \end{aligned}$$

Beispiel:

$$R: \quad 0 + n \rightarrow n \quad (1)$$

$$s(m) + n \rightarrow s(m + n) \quad (2)$$

Für dieses Termersetzungssystem gibt es folgende Reduktionsschritte:

$$\begin{aligned} s(s(0)) + s(0) & \xrightarrow{\epsilon, (2)} s(s(0) + s(0)) \\ & \xrightarrow{1, (2)} s(s(0 + s(0))) \\ & \xrightarrow{1 \cdot 1, (1)} s(s(s(0))) \end{aligned}$$

Der letzte Term $s(s(s(0)))$ ist in Normalform, d.h. nicht weiter reduzierbar.

Für sinnvoll definierte funktionale Programme ist es wünschenswert, dass Normalformen *eindeutig* sind. In Kapitel 3.1 haben wir gesehen, dass die Konfluenz bzw. Church-Rosser-Eigenschaft hinreichend für eindeutige Normalformen ist.

Tatsächlich ist das obige +-Beispiel konfluent, wohingegen das Gruppenbeispiel nicht konfluent ist. Um letzteres zu sehen, betrachte die beiden Regeln:

$$\begin{aligned} x * x^{-1} &\rightarrow 1 & (1) \\ (x * y) * z &\rightarrow x * (y * z) & (2) \end{aligned}$$

Dann gibt es für den Term $(a * a^{-1}) * b$ die folgenden Reduktionsschritte:

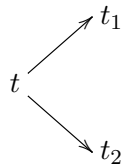
$$\begin{aligned} (a * a^{-1}) * b &\rightarrow_{\epsilon, (2)} a * (a^{-1} * b) \\ (a * a^{-1}) * b &\rightarrow_{1, (1)} 1 * b \end{aligned}$$

Die jeweils letzten Terme sind verschiedene Normalformen des Terms $(a * a^{-1}) * b$. Daher sind die Regeln nicht konfluent.

Diese Beispiele zeigen, dass Konfluenz stark abhängig vom jeweiligen Termersetzungssystem ist. Gibt es also hinreichende Kriterien, die z.B. durch einen Compiler überprüft werden könnten?

Hierzu haben Knuth und Bendix gezeigt ([Knuth/Bendix 70]), dass es ausreicht, sogenannte **kritische Paare** (d.h. Überlappungen linker Seiten) zu betrachten. Dies basiert auf folgender Idee:

Falls es eine lokale Divergenz

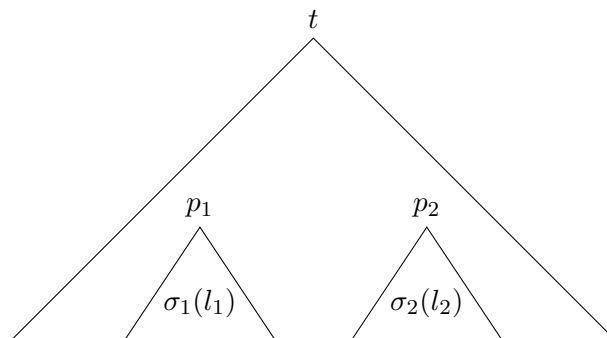


gibt, dann muss auf Grund der Definition von Termersetzungsschritten gelten:

$$\exists p_i, \sigma_i, l_i \rightarrow r_i \in R \text{ mit } t|_{p_i} = \sigma_i(l_i) \text{ und } t_i = t[\sigma_i(r_i)]p_i \text{ (für } i = 1, 2)$$

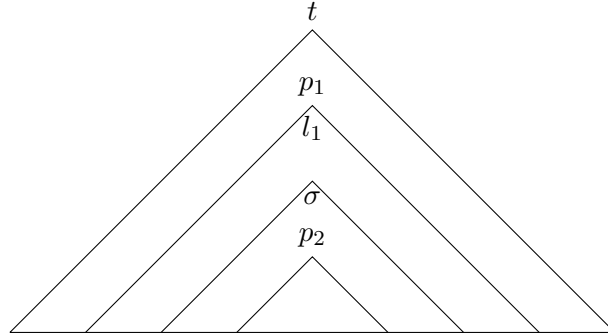
Wir können dann die folgenden Fälle unterscheiden:

- p_1 und p_2 sind disjunkt:



Führe t_1 und t_2 zusammen durch Anwendung von $l_i \rightarrow r_i$ an Stelle p_i in t_{3-i} ($i = 1, 2$)

- p_1 über p_2 , aber nicht innerhalb von l_1 :



Dann sind t_1 und t_2 ebenfalls zusammenführbar (beachte: p_2 liegt innerhalb einer Variablen in l_1 ; somit ist die Regel $l_2 \rightarrow r_2$ auch anwendbar in allen diesen Variablen in r_2).

- p_2 über p_1 , aber nicht innerhalb von l_2 : analog
- Falls keiner der vorigen Fälle zutrifft, haben wir eine „echte“ Überlappung von l_1 und l_2 , die wir nicht einfach zusammenführen können. Daher spricht man in diesem Fall von einem *kritischen Paar*.

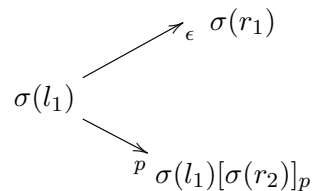
Definition 3.8 Seien $l_1 \rightarrow r_1, l_2 \rightarrow r_2 \in R$ mit $\text{Var}(l_1) \cap \text{Var}(l_2) = \emptyset$, $p \in \text{Pos}(l_1)$ mit $l_1|_p \notin V$ und σ mgu (vgl. Kapitel 2.5.2, Typinferenz) für $l_1|_p$ und l_2 . Falls $p = \epsilon$, dann sei auch $l_1 \rightarrow r_1$ keine Variante (gleich bis auf Variablenumbenennung) von $l_2 \rightarrow r_2$. Dann heißt

$$\langle \sigma(r_1), \sigma(l_1)[\sigma(r_2)]_p \rangle$$

kritisches Paar von $l_1 \rightarrow r_1$ und $l_2 \rightarrow r_2$.

Weiterhin bezeichnen wir mit $CP(R)$ die Menge aller kritischer Paare aus R .

Erklärung:



ist eine „kritische“ Divergenz (da $\sigma(l_1)|_p = \sigma(l_2)$).

Satz 3.3 (Kritisches-Paar-Lemma [Knuth/Bendix 70]) Sei R ein TES.
 Falls $t \rightarrow t_1$ und $t \rightarrow t_2$, dann ist $t_1 \downarrow_R t_2$ oder $t_1 \leftrightarrow_{CP(R)} t_2$.

D.h. alle echten lokalen Divergenzen resultieren aus kritischen Paaren. Somit können wir die lokale Konfluenz eines TES mittels Zusammenführung kritischer Paare charakterisieren:

Satz 3.4 Sei R TES. Dann gilt:

$$R \text{ ist lokal konfluent} \iff t_1 \downarrow_R t_2 \quad \forall \langle t_1, t_2 \rangle \in CP(R)$$

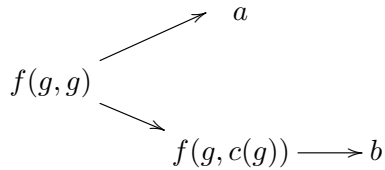
Falls also R endlich ist und \rightarrow_R terminierend, so können wir die Konfluenz durch Zusammenführung aller kritischen Paare entscheiden.

Kann man dieses Kriterium auch bei nichtterminierenden TES verwenden?

Leider nicht, denn dann ist die Untersuchung kritischer Paare nicht ausreichend. Betrachte hierzu das folgende Termersetzungssystem R :

$$\begin{aligned} f(x, x) &\rightarrow a \\ f(x, c(x)) &\rightarrow b \\ g &\rightarrow c(g) \end{aligned}$$

Offensichtlich gilt hier $CP(R) = \emptyset$, d.h. es existieren keine kritischen Paare. Somit ist R lokal konfluent, aber nicht konfluent:



Beachte: Die linke Seite der ersten Regel ist nicht-linear!

Aus dieser Beobachtung kann man eine andere hinreichende Charakterisierung von Konfluenz für nicht-terminierende TES erhalten:

Definition 3.9 Sei R ein TES wobei l linear ist für alle Regeln $l \rightarrow r \in R$. Ein solches TES nennen wir auch **links-linear**. Dann heißt R

- **orthogonal** falls $CP(R) = \emptyset$
- **schwach orthogonal** falls für alle $\langle t, t' \rangle \in CP(R)$ gilt: $t = t'$
 (d.h. es existieren nur triviale kritische Paare)

Es gilt:

Satz 3.5 (Schwach) orthogonale TES sind konfluent.

Somit verfügen wir nun über folgende hinreichende Kriterien für Konfluenz:

- bei terminierenden *TES*: Zusammenführung kritischer Paare
(die Eigenschaft „terminierend“ ist teilweise prüfbar mit sog. *Terminierungsordnungen*)
- im Allg.: Prüfe Linkslinearität und Trivialität von Überlappungen

Wenn wir nun ein konfluentes TES haben, muss noch folgendes Problem gelöst werden:

Wie findet man die Normalform (falls sie existiert) eines Terms bzgl. eines konfluenten *TES*?

- Bei terminierenden *TES*: durch eine beliebige Folge von Reduktionsschritten
- Im Allgemeinen: durch eine geeignete *Reduktionsstrategie*

Im Folgenden sei nun immer R ein konfluentes Termersetzungssystem, falls dies nicht explizit erwähnt ist.