Grundlagen der Systementwicklung

Martin Wirsing

in Zusammenarbeit mit
Axel Rauschmayer

Ziele

1) Zusammenhang zwischen Konfluenz und
Church-Rosser-Eigenschaft verstehen,

2) Terminierungsbeweis durch
lexikographische Pfadordnung,

3) Zusammenhang zwischen
Konfluenz und lokaler Konfluenz verstehen,

4) Beweis lokaler Konfluenz durch
Untersuchung der kritischen Paare.

Gleichungsbeweise durch Normalisierung

In manchen Fällen läßt sich der Beweis einer Gleichung u=v durch

reines Anwenden von Termersetzungsregeln

durchführen, und zwar durch

Vorwärtsanwendungen auf u und durch Vorwärtsanwendungen auf v,
bis jeweils eine Normalform (kanonische Form) auftritt,
bei der keine Anwendung der Termersetzungsregeln mehr möglich ist.

Sind die beiden Normalformen syntaktisch gleich, so ist die Gleichheit bewiesen, andernfalls ist u=v nicht allgemeingültig.

Dazu betrachten wir zunächst allgemeine binäre Relationen $\rightarrow \subseteq M \times M$ auf einer Menge M. Wir nennen das Paar $<M, \rightarrow>$ ein *Reduktionssystem*.

Beispiel: Sei $M = N \times N\backslash\{0\}$ und $P \subseteq N\backslash\{0\}$ die Menge der Primzahlen.
Definiere $< n*p, d*p> \rightarrow <n,d>$. Dann definiert
$M = < M, \rightarrow >$ die Reduktion zu relativ primen Zahlenpaaren.

Definition

Sei $M = < M, \rightarrow >$ ein Reduktionssystem; $u,v \in M$.

(1a) $u \rightarrow^0 v$ gdw. $u \equiv v$;

(1b) $u \rightarrow^{i+1} v$ gdw. $\exists z \in M: u \rightarrow^i z \rightarrow v$;

(1c) $u \rightarrow^* v$ gdw. $\exists i \in N: u \rightarrow^i v$
("reflexiv-transitive Hülle");

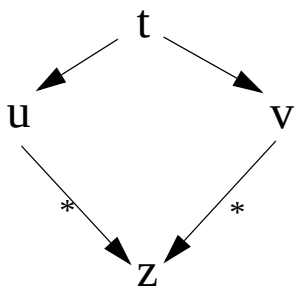
(1d) $u \rightarrow^+ v$ gdw. $\exists i \in N: u \rightarrow^{i+1} v$ ("transitive Hülle");

(2) v heißt *Redukt* von u, falls $u \rightarrow^* v$;

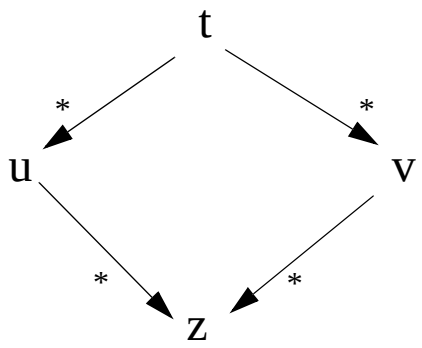
(3) $u \downarrow v$ gdw. "u und v *haben ein gemeinsames Redukt*", d.h. $\exists z \in M: u \rightarrow^* z$ und $v \rightarrow^* z$;

(4) M heißt *lokal konfluent*, wenn je zwei beliebige "Einschritt"-Redukte eines Elementes ein gemeinsames Redukt besitzen, d.h. gdw. $\forall t,u,v \in M: t \rightarrow u$ und $t \rightarrow v$ impliziert $u \downarrow v$,

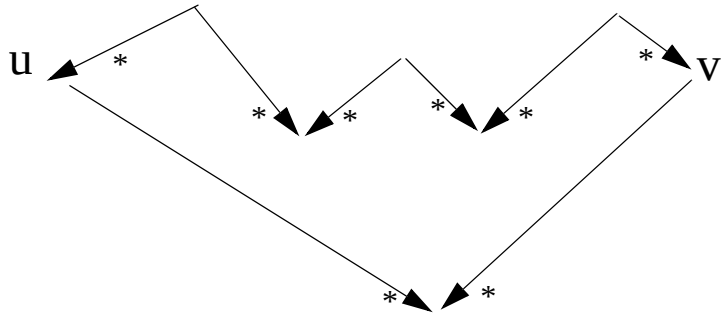
d.h.



(5) M heißt *konfluent*, wenn je zwei beliebige Redukte eines Elementes ein gemeinsames Redukt besitzen, d.h.
gdw. $\forall$ t,u,v $\in$ M: t $\rightarrow$* u und t $\rightarrow$* v impliziert u$\downarrow$v



(6) M heißt *Church-Rosser*, wenn zwei gleichungsäquivalente Elemente ein gemeinsames Redukt besitzen, d.h.
wenn $\forall\ t_1,t_2 \in M: t_1 <->^* t_2 \quad => \quad t_1 \downarrow t_2$



Satz (Satz von Church-Rosser)

Ein Reduktionssystem ist Church-Rosser genau dann, wenn es konfluent ist, d.h.

$\forall a,b \in M: a <\!\!\rightarrow^* b \Rightarrow$

$\exists c \in M: a \rightarrow^* c$ und $b \rightarrow^* c$ (“Church-Rosser”)

gdw.

$\forall a,b,d \in M: a \rightarrow^* b \wedge a \rightarrow^* d$)

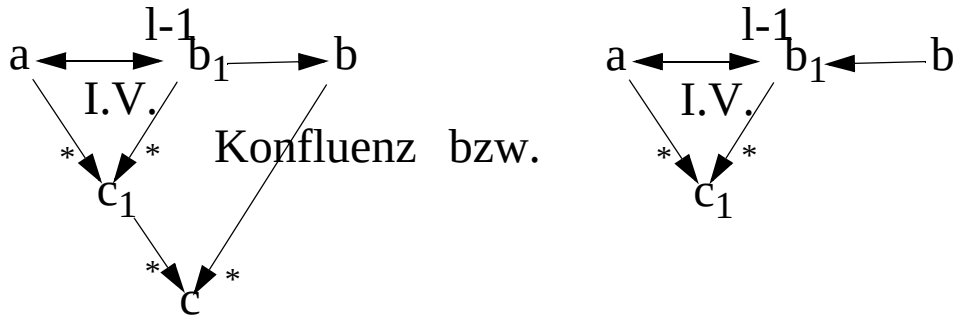
$\exists c \in M: b \rightarrow^* c$ und $d \rightarrow^* c$ (“konfluent”).

Beweis: “CR => konfluent”: Sei a,b,d mit $a \rightarrow^* b \wedge a \rightarrow^* d$. Also gilt $b <\!\!\rightarrow^* d$. Wegen CR gibt es $c \in M$ mit $b \rightarrow^* c$ und $d \rightarrow^* c$.

“konfluent => CR”: Beweis durch Induktion nach der Länge l der Ableitung:

“l=0”: Offensichtlich ist dann $a \equiv b$ und c kann als a gewählt werden.

"l-1) l": Betrachte folgende Bilder:



Formal schließt man so:

Wir bezeichnen eine Ableitung in l Schritten mit $<\!\!-\!\!>^l$ bzw. mit $\rightarrow^l$.

Sei $a <\!\!-\!\!>^l b$. Dann gibt es $b_1 \in M$ mit $a <\!\!-\!\!>^{l-1} b_1$ und $b_1 \rightarrow b$ oder $b \rightarrow b_1$.

Nach Induktionsvoraussetzung gibt es $c_1 \in M$ mit $a \rightarrow^* c_1$ und $b_1 \rightarrow^* c_1$.

1) $b_1 \rightarrow b$:

Es gilt $b_1 \rightarrow^* c_1$ und $b_1 \rightarrow^* b$. Also gibt es wegen der Konfluenz von $\rightarrow$ ein $c \in M$ mit $c_1 \rightarrow^* c$ und $b \rightarrow^* c$. Wegen $a \rightarrow^* c_1$ gilt auch $a \rightarrow^* c$.

2) $b \rightarrow b_1$:

Dann gilt $b \rightarrow b_1 \rightarrow^* c_1$, also $b \rightarrow^* c_1$, d.h. c_1 ist das gesuchte Element.

q.e.d.

Zum Beweis der obigen Äquivalenz wird nur die Eigenschaft der “semi-lokalen Konfluenz” verwendet, d.h. diese ist äquivalent zur Konfluenz.

Definition

Ein Reduktionssystem M heißt *semi-lokal konfluent* gdw.

$$\forall\ t,u,v \in M: t \rightarrow u \text{ und } t \rightarrow^* v, \text{ dann } u \downarrow v.$$

Satz (Konfluenz ist äquivalent zur semi-lokalen Konfluenz)

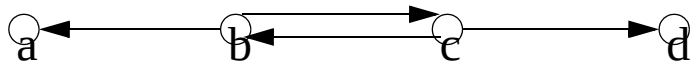
Ein Reduktionssystem $M = \langle M, \rightarrow \rangle$ ist konfluent genau dann, wenn es semi-lokal konfluent ist.

Beweis durch Induktion über die Länge der Ableitung.

Damit sind die Begriffe “Konfluenz” und “semi-lokale Konfluenz” äquivalent.

Jedes konfluente Reduktionssystem ist auch lokal konfluent, aber die Umkehrung gilt nicht. Betrachte das folgende Gegenbeispiel:

Beispiel: :



Es gilt: $c \rightarrow^* d$ und $c \rightarrow^* a$, aber nicht $d \downarrow a$.

Um aus der lokalen Konfluenz die Konfluenz folgern zu können, benötigt man daher weitere Eigenschaften.

Definition Sei $M = < M, \rightarrow >$ ein Reduktionssystem; $u,v \in M$

(1) M heißt *terminierend* (oder streng normalisierend), falls jede Reduktion terminiert, d.h. es existiert keine unendliche Sequenz $(u_n)_{n \in \mathbf{N}}$ mit $u_n \rightarrow u_{n+1}$ für alle $n \in N$.

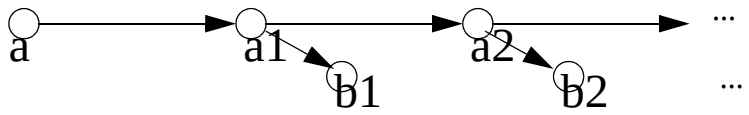
(2a) u ist *in Normalform (kanonischer Form)*, wenn kein v existiert mit $u \rightarrow v$;

(2b) v ist *Normalform* von u, falls
v in Normalform ist und $u \rightarrow^* v$ gilt;

(2c) u *hat eine Normalform*, wenn es zu einer Normalform reduziert.

Beispiel:

Das Reduktionssystem aus dem vorhergehenden Beispiel ist **nicht terminierend**. Die Elemente a,d sind in Normalform, b und c haben beide a und d als Normalformen.
Das Reduktionssystem mit den relativen Primzahlpaaren ist terminierend .



Dieses Reduktionssystem ist nicht terminierend, aber jedes Element hat eine Normalform.

Reduktionsordnungen

Definition:

Eine Noethersche Ordnung > über $T_\Sigma(V)$ heißt Reduktionsordnung gdw. die folgenden zwei Bedingungen erfüllt sind:

1) **Strikte Monotonie**: Für jedes $f \in \Sigma$, und alle
   $f(t_1, \ldots, t_n), f(t_1, \ldots, t_{i-1}, u, t_{i+1}, \ldots, t_1) \in T_\Sigma(V)$ gilt, dass
   $t_i > u$ impliziert $f(t_1, \ldots, t_n) > f(t_1, \ldots, t_{i-1}, u, t_{i+1}, \ldots, t_n)$
2) **Abschluss unter Substitution**:
   Wenn $t > t'$,
   dann gilt $\overline{\sigma}(t) > \overline{\sigma}(t')$ für jede Substitution $\overline{\sigma}: V \rightarrow T_\Sigma(V)$.

Theorem:
Sei (Σ,E) eine Gleichungstheorie (ohne bedingte Gleichungen).
Dann gilt:

E terminiert gdw
es gibt eine Reduktionsordnung > so dass
für jede Gleichung u=v in E gilt: u > v.

Beweis:

=>) Wenn E terminiert, dann ist $\rightarrow_E^*$ (wegen der Kongruenz- und Substitutionseigenschaft) offensichtlich eine Reduktionsordnung.

<=) Sei > eine Reduktionsordnung mit der Eigenschaft u>v für alle $u=v \in E$. Es gelte $t \rightarrow_E t'$, d.h. es gibt eine Position p in t, eine Gleichung u = v in E und eine Substitution σ so dass $t = t[p <- \overline{\sigma}(u)]$ und $t' = t[p <- \overline{\sigma}(v)]$.
Wegen des Abschlusses unter Substitution gilt $\overline{\sigma}(u) > \overline{\sigma}(v)$. Daraus folgt t > t' mit mehrfacher Anwendung der strikten Monotonie. qed

Lexikographische Pfadordnung

Es gibt mehrere Ordnungen, die ausgehend von einer minimalen Information eine Ordnung auf allen Termen induzieren. Eine solche Ordnung ist die lexikographische Pfadordnung, die auf einer Ordnung der Funktionssymbole aufbaut.

Idee: Komplexere Funktionen sind größer in derr Ordnung als einfache Funktionen.

Beispiel: Natürliche Zahlen * > + > s

Die lexikographische Pfadordnung wird vom „Termination Chekker“ von Maude unterstützt. Der TC überprüft all möglichen Ordnungen von Funktionssymbolen auf ihre Reduktionseigenschaft.

Lexikographische Pfadordnung

Definition:

Sei S eine endliche Signatur und > eine Ordnung ihrer Funktionssymbole.

Dann ist die lexikographische Pfadordnung $>_{lpo}$ über $T_\Sigma(V)$ rekursiv folgendermaßen definiert:

1) $x \in Var(t)$ und x verschieden von t, dann $t >_{lpo} x$.

2) $f(t_1, \ldots, t_n) >_{lpo} g(t'_1, \ldots, t'_m)$ wenn

entweder 2.1 $t_i >_{lpo} g(t'_1, \ldots, t'_m)$ für ein i,

oder 2.2 $f > g$ and $f(t_1, \ldots, t_n) >_{lpo} t'_j$, für alle j,

oder 2.3 $f = g$, $f(t_1, \ldots, t_n) >_{lpo} t'_j$, für alle j und
es gibt ein i mit $t_j = t'_j$, für alle $j < i$, und $t_i >_{lpo} t'_i$.

Äquivalent zu 2.3 ist folgende etwas einfachere Bedingung

2.3' $f = g$ und es gibt ein i mit
$t_j = t'_j$, für alle $j < i$, und
$t_i >_{lpo} t'_i$ und
$f(t_1, \ldots, t_n) >_{lpo} t'_j$, für alle $j > i$

Beispiel Ackermann-Funktion:

(a) ack(0, y) = s(y)
(b) ack(s(x), 0) = ack(x, s(0))
(c) ack(s(x), s(y)) = ack(x, ack(s(x), y)

Zum **Beweis der Terminierung** von ack wählt man ack > s > 0.
Dann gilt für die Termersetzungsregeln von ack:

(a) ack(0, y) $>_{lpo}$ s(y)

wegen 2.2: Es gilt ack > s und ack(0,y) $>_{lpo}$ y (wg. 1).

(b) ack(s(x), 0) $>_{lpo}$ ack(x, s(0))

wegen 2.3':
Es gilt s(x) $>_{lpo}$ x (wg. 1) und
ack(s(x), 0) $>_{lpo}$ s(0) (wg. 2.2 ack > s und ack(s(x), 0) $>_{lpo}$ 0 (wg. ack>0))

(c) ack(s(x), s(y)) $>_{lpo}$ ack(x, ack(s(x), y))

wegen 2.3':

Es gilt (i) s(x) $>_{lpo}$ x wg. 1.

(ii) ack(s(x), s(y)) $>_{lpo}$ ack(s(x), y)

da der linke Term lexikographisch größer ist als der rechte;

formal zeigt man, dass 2.3' gilt:

s(x) = s(x) und s(y) $>_{lpo}$ x (wg. 1).

Für Beweise bei terminierenden Reduktionssystemen verwendet man auch gerne das wichtige Prinzip der Noetherschen Induktion.

Noethersche Induktion:

Gegeben $M = < M, \rightarrow >$ terminierendes Reduktionssystem.
Sei P eine Aussage über M.
Wenn für alle $x \in M$

aus der Gültigkeit von P(y) für alle $y \in M$ mit $x \rightarrow^+ y$ auch P(x) folgt, d.h.

$$\forall x \in M\ [\ \forall y \in M: x \rightarrow^+ y \Rightarrow P(y)] \Rightarrow P(x),$$

dann gilt P(x) für alle $x \in M$, d.h. $\forall x \in M: P(x)$.

Satz (Newman's Lemma)

Sei M = < M, $\rightarrow$ > terminierendes Reduktionssystem. Dann gilt:

M ist konfluent gdw. M ist lokal konfluent**.**

Beweis: "=>" trivial.

"<=" durch Noethersche Induktion.

Sei $P(x) \equiv \forall y,z \in M: x \rightarrow^* y \wedge x \rightarrow^* z \Rightarrow y \downarrow z$.

Sei $x \rightarrow^m y$ und $x \rightarrow^n z$ für gewisse $n,m > 0$.

Wir zeigen:

$\exists r \in M: y \rightarrow^* r$ und $z \rightarrow^* r$:

1) m=0: Wähle $r \equiv z$.

2) n=0: Wähle $r \equiv y$.

3) m>0, n>0:

Dann gibt es $y_1, z_1 \in M$ mit

$x \rightarrow y_1 \rightarrow^* y$ und $x \rightarrow z_1 \rightarrow^* z$.

Aufgrund der lokalen Konfluenz existiert ein $u \in M$ mit

$y_1 \rightarrow^* u$ und $z_1 \rightarrow^* u$.

Die Induktionsvoraussetzung garantiert für y_1 und z_1 die Existenz von v,w $\in$ M mit

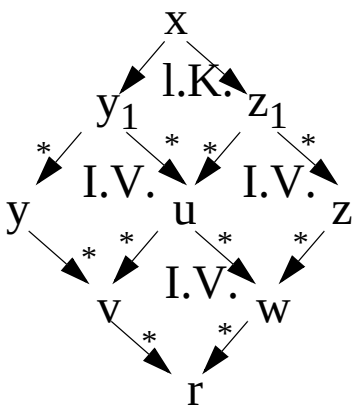
$y \rightarrow^* v$, $u \rightarrow^* v$, $z \rightarrow^* w$ und $u \rightarrow^* w$.

Schließlich erhält man aufgrund der Induktionsvoraussetzung für u ein r $\in$ M mit

$v \rightarrow^* r$ und $w \rightarrow^* r$,

d.h. $y \rightarrow^* v \rightarrow^* r$ und $z \rightarrow^* w \rightarrow^* r$, womit P(x) bewiesen ist.

Bildlich läßt sich der Induktionsschluss folgendermaßen darstellen:



lokale Konfluenz

zweimal Ind.-vorauss.

Ind.vorauss.

qed

Satz Seien Σ, X, E gegeben.

Sei $<T(\Sigma,X), \rightarrow_E>$ ein konfluentes Termersetzungssystem, wobei $E=\{l_i=r_i\}$ die Termersetzungsrelation $\rightarrow_E$ definiert durch $l_i \rightarrow r_i$ für $l_i=r_i \in E$.

Dann gilt für alle $t_1,t_2 \in T(\Sigma,X)$:

$<\Sigma,E> \models t_1=t_2$ gdw. $t_1 \downarrow t_2$.

Beweis:"<=" trivial.

"=>" $Mod(<\Sigma,E>) \models t_1=t_2$

=> [Satz Vollständigkeit Birkhoff]

$E \vdash_B t_1 = t_2$

=> [Satz Vollständigkeit Paramodulation]

$t_1 <\!\!\rightarrow_E^* t_2$

=> [Satz , Church-Rosser]

$t_1 \downarrow t_2$.

qed

Satz Seien Σ, X, E gegeben. Sei $T=<T(\Sigma,X), \rightarrow_E>$ ein lokal konfluentes und terminierendes Termersetzungssystem. Σ sei bewohnt.
Dann ist entscheidbar, ob eine Gleichung in allen Modellen von $<\Sigma,E>$ gilt oder nicht.

Beweis: Angabe eines Algorithmus.

Für je zwei Terme $t_1,t_2 \in T(\Sigma,X)$ bilde die kanonischen Normalformen $NF(t_1)$ und $NF(t_2)$. Diese existieren, da T terminierend ist und sind eindeutig bestimmt, da T lokal konfluent ist. Sind $NF(t_1)$ und $NF(t_2)$ syntaktisch gleich, so gilt $t_1=t_2$ in $Mod(<\Sigma,E>)$.
Ansonsten gibt es ein Modell $A \in Mod(<\Sigma,E>)$ mit
$A \models t_1=t_2$, das folgendermaßen definiert werden kann:

$A_s = \{NF(t) \mid t \in T(\Sigma,X)_s\}$ für alle $s \in S$,

$f^A(x_1,...,x_n) = NF(f(x_1,...,x_n))$,

für alle $f \in F_{<<s1,...,sn> s>}$, $x_i \in A_{si}$, i=1,...,n.

A ist wohldefiniert, da NF(t) eindeutig bestimmt für alle $t \in T(\Sigma,X)$.

A erfüllt die Axiome $u=v \in E$: T enthält die Regel $u \rightarrow_E v$, also gilt für die Instanz mit einer beliebigen Substitution σ: $\overline{\sigma}u \rightarrow_E \overline{\sigma}v$. Wegen der Konfluenz gilt $NF(\overline{\sigma}u) = NF(\overline{\sigma}v)$, d.h. $\overline{\sigma}u^A = \overline{\sigma}v^A$ und somit gilt $A \models u=v$.

qed

Der folgende Begriff der Unifikation wird benötigt, um die lokale Konfluenz mit Hilfe sogenannter “kritischer Paare” testen zu können. Wir suchen jetzt die “allgemeinste” Substitution, die zwei Terme t und t’ syntaktisch gleich macht (“unifiziert”).

Definition (Unifikation)

1) Eine Substitution σ: X $\rightarrow$ T (Σ, X) heißt *Unifikator* von t und t' (t, t' $\in$ T_Σ (X)), falls $\overline{\sigma}t = \overline{\sigma}t'$.

2) σ ist *allgemeiner* als σ' ($\sigma' < \sigma$), falls $\exists\sigma_1$: $\sigma_1 \circ \sigma = \sigma'$

3) Mit *mgu (t, t')* wird der *allgemeinste Unifikator* von t und t' bezeichnet.

Satz 11 Der allgemeinste Unifikator ist bis auf Umbenennung der Variablen eindeutig bestimmt.

Beweis:

Seien σ, τ zwei mgu von t, t'.

Dann $\exists$ σ_1, τ_1: $\sigma_1 \circ \sigma = \tau$ und $\tau_1 \circ \tau = \sigma$.

Also $(\sigma_1 \circ \tau_1) \circ \tau = \sigma_1 \circ (\tau_1 \circ \tau) = \sigma_1 \circ \sigma = \tau$,

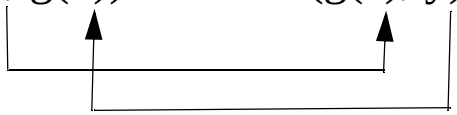
d.h. $(\sigma_1 \circ \tau_1)|_{Var(\tau)} = id$.

Ebenso $(\tau_1 \circ \sigma_1) \circ \sigma = \sigma$, d. h. $(\tau_1 \circ \sigma_1)|_{Var(\sigma)} = id$.

Betrachte $Var(t) \cup Var(t')$:
Für diese gilt $\sigma_1 \circ \tau_1(x) = \tau_1 \circ \sigma_1(x) = x$,
d.h. σ, τ gehen durch Umbenennung ineinander über.

Beispiel:

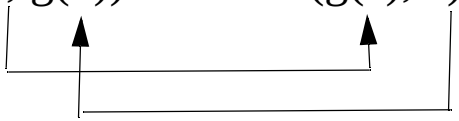
1) t = f(x, g(x)) t’ = f(g(z), y)



$\sigma_1 = [g(z)/x]$
$\sigma_2 = [\ g(x)\sigma_1/y] = [\ g(g(z))/y]$
$mgu(t, t') = \sigma_2 \circ \sigma_1 = [\ g(z)/x, g(g(z))/y]$,
ergibt t = t’ = f(g(z), g(g(z)))

Beispiel:

2) t = f(x, g(x)) t’ = f(g(z), z)



σ_1 = [g(z)/x]
ergibt t = f(g(z), g(g(z))), t’ = f(g(z),z).
Also ist g(g(z)) mit z zu unifizieren,
das ist aber für endliche Terme unmöglich.

Bemerkung:
Unifizieren wird u.a. gebraucht für:

- Resolution logischer Programme,
- Typinferenz bei funktionalen Programmen.

Algorithmus von Robinson zur Bestimmung des mgu

I) mgu (t, t'):

Seien x, y Variablen:

1) t=x, t' = y : mgu (t, t') = [y/x]

2a) t=x, t'=f(t_1,...,t_n) mit x ∈ Var(t_1) ∪ ... ∪ Var(t_n):
mgu existiert nicht.

2b) t=x, t'=f(t_1,...,t_n) mit x ∉ Var(t_1) ∪ ... ∪ Var(t_n):
mgu (t, t') = [f(t_1,...,t_n)/x]

3) t=f(t_1,...,t_n), t'=x: wie 2a), 2b)

4) t=f(t_1,...,t_n), t'=g(t'_1,...,t'_m)

a) f ≡ g. Dann ist n = m,
mgu(t, t') = mgu(<t_1,...,t_n>, <t'_1,...,t'_n>)

b) Nicht f ≡ g: mgu (t, t') existiert nicht.

II) mgu($\langle t_1, t_2 \rangle$, $\langle t'_1, t'_2 \rangle$)

$= \sigma_2 \circ \sigma_1$

wobei $\sigma_1 = mgu(t_1, t'_1)$, $\sigma_2 = mgu(\overline{\sigma}_1 t_2, \overline{\sigma}_1 t'_2)$

III) **mgu($\langle t_1,...,t_n \rangle$, $\langle t'_1,...,t'_n \rangle$)**

$= mgu(\langle\langle t_1,..., t_{n-1}\rangle, t_n\rangle, \langle\langle t'_1,...,t'_{n-1}\rangle, t'_n\rangle) =$

$= mgu(mgu(\langle t_1,..., t_{n-1}\rangle, \langle t'_1,...,t'_{n-1}\rangle), \langle t_n, t'_n\rangle)$

Bemerkung:
Die Abfrage in 2a) und 2b), bzw. in 3), ob x in $f(t_1,...,t_n)$ vorkommt, wird auch als **"Occur-Check"** bezeichnet.

Als Nächstes suchen wir nach einem Test für lokale Konfluenz für ein Termersetzungssystem R. Dazu betrachten wir die Überlappungen von linken Seiten von Regeln aus R.

Definition (Varianten, Kritische Paare)

Sei R ein Termersetzungssystem, so daß jede Regel $l \to r$ aus R folgende einschränkende Bedingung erfüllt: $FV(r) \subseteq FV(l)$ und $l \notin X$.
Seien $l_1 \to r_1$ und $l_2 \to r_2$ zwei Termersetzungsregeln mit $FV(l_1) \cap FV(l_2) = \varnothing$.

1) Die beiden Regeln heißt **Varianten** voneinander, falls sie durch Umbenennung von Variablen unifiziert werden können.

2) Es gebe $p \in O(l_1)$, so daß $l_1|_p \notin X$ und $\sigma = mgu(l_1|_p, l_2)$ existiert (d.h. l_1 und l_2 haben eine echte Überlappung). Weiter sei entweder p ungleich ε oder die eine Regel nicht Variante der anderen. Dann heißt

$$(\overline{\sigma} r_1, \overline{\sigma} l_1[p <- \overline{\sigma} r_2])$$

kritisches Paar von $l_2 \to r_2$ auf $l_1 \to r_1$.

Mit CP(R) bezeichnet man die Menge der kritischen Paare, die man mit (variablendisjunkten Varianten von) Regeln aus R bilden kann.

Bemerkung:
1) Die Forderung $FV(l_1) \cap FV(l_2) = \varnothing$ läßt sich durch Anwendung einer Variablenumbenennung stets erreichen.

2) Es kann eine Überlappung einer Regel mit sich selbst geben. Eine Regel überlappt sich stets mit sich selbst bei $p = \varepsilon$. Da dies aber nur die triviale Divergenz $r \leftarrow l \rightarrow r$ erzeugt, wird diese Situation in obiger Definition ausgeschlossen.

3) Hier ist die Variablenbedingung $FV(r) \subseteq FV(l)$ wichtig. Würde man auch Regeln der Form $f(x) \rightarrow g(x, y)$ zulassen, so müßte man auch die lokale Divergenz $g(x, z) \leftarrow f(x) \rightarrow g(x, y)$ betrachten.

Beispiel:

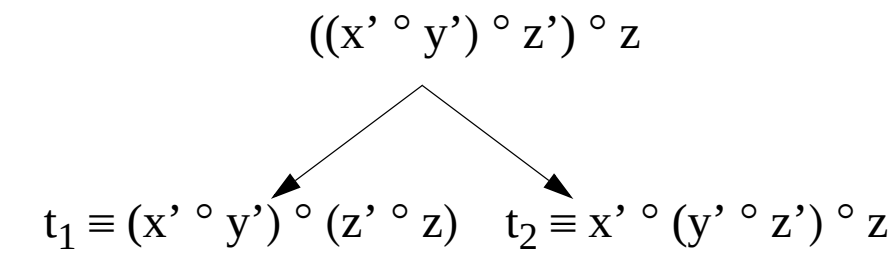
1) $(x' \circ y') \circ z' \rightarrow x' \circ (y' \circ z')$, $\quad x \circ x^{-1} \rightarrow n$
liefert die kritischen Paare

$(x \circ (x^{-1} \circ z), n \circ z,)$ und $(n, x \circ (y \circ (x \circ y)^{-1}))$

2) Die Assoziativität $(x \circ y) \circ z \rightarrow x \circ (y \circ z)$.
hat ein kritisches Paar aus einer Überlappung mit sich selbst:
Eine Variablenumbenennung liefert

$$(x' \circ y') \circ z' \rightarrow x' \circ (y' \circ z').$$

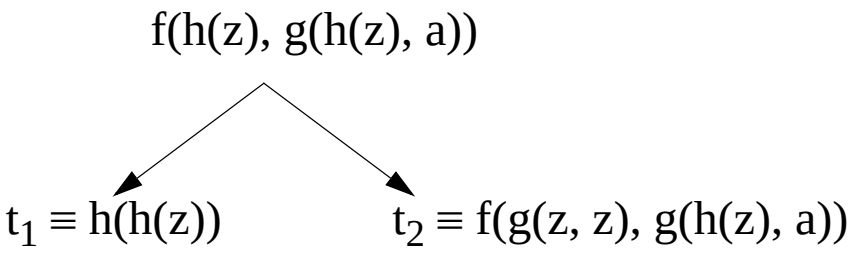
Es ist $x \circ y$ ist mit $(x' \circ y') \circ z'$ unifizierbar, denn



Also ist (t_1, t_2) ein kritisches Paar der Regel mit sich selbst.

3) $f(x, g(x, a)) \rightarrow h(x)$, $h(x') \rightarrow g(x', x')$.
Die einzigen Teilterme von $t \equiv f(x, g(x, a))$, die mit $h(x')$ unifizierbar sind, sind $t|_1 \equiv x$ und $t|_{2.1} \equiv x$.
Es gibt also kein kritisches Paar.

Diese Regeln führen aber doch zu einer lokalen Divergenz. Es gibt z.B. folgende Variablenüberlappung:



Es gilt:
$t_1 \rightarrow h(g(z, z))$ mit der 2. Regel

$t_2 \rightarrow f(g(z, z), g(g(z, z), a)) \rightarrow h(g(z, z))$mit 2. u. 1. Regel

Also sind t_1 und t_2 auf den gleichen Term reduzierbar.
Somit ist das Paar (t_1, t_2) unkritisch.

Satz [Knuth-Bendix 70]

Sei R ein Termersetzungssystem, bei dem für alle Regeln $l \to r$ aus R $FV(r) \subseteq FV(l)$ und $l \notin X$ gilt.

R ist genau dann lokal konfluent, wenn für alle kritischen Paare (t_1, t_2) in R $t_1 \downarrow t_2$ gilt.

Bemerkung:

Diese Ergebnisse können auf Termersetzungssysteme modulo Assoziativität, Kommutativität und neutralem Element und mit bedingten Gleichungen verallgemeinert werden; die Verallgemeinerungen erfordern aber spezielle Überlegungen; z.B. fordert man bei einem Terminierungsbeweis, dass die rechte Seite der Konklusion und alle Terme in den Bedingungen kleiner als die linke Seite der Konklusion sind.

Zusammenfassung

1) Konfluenz und Church-Rosser-Eigenschaft sind äquivalent.

2) Jedes terminierende und lokal-konfluente Reduktions
system ist konfluent (Newman’s Lemma).

3) Terminierungsbeweise können mit Hilfe der
lexikographischen Pfadordnung oder der
Noetherschen Induktion
geführt werden.

4) Ein Termersetzungssystem ist lokal konfluent, wenn alle kritischen Paare ein gemeinsames Redukt besitzen.