### Beispiele

#### Beispiel 9.4 Betrachte

►  $f(f(\underline{x},\underline{y}),z) \rightarrow f(x,f(y,z))$   $f(f(x',y'),\underline{z'}) \rightarrow f(x',f(y',z'))$ Unifizierbar mit  $x \leftarrow f(x',y'), y \leftarrow z'$ 

$$f(f(f(x',y'),z'),z)$$

$$t_1 = f(f(x',y'),f(z',z)) \qquad f(f(x',f(y',z')),z) = t_2$$

►  $t = f(x, g(x, a)) \rightarrow h(x)$   $h(x') \rightarrow g(x', x'), t|_1 = t|_{21} = x$ Keine kritischen Paare. Betrachte Variablenüberlappung:

$$f(h(z), g(h(z), a)))$$

$$t_1 = h(h(z))$$

$$f(g(z, z), g(h(z), a)) = t_2$$

$$f(g(z, z), g(g(z, z), a))$$

$$h(g(z, z))$$

## Eigenschaften

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

$$\sigma(t) \stackrel{*}{\rightarrow}_R \tau(t)$$

Seien  $l_1 \to r_1, l_2 \to r_2$  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

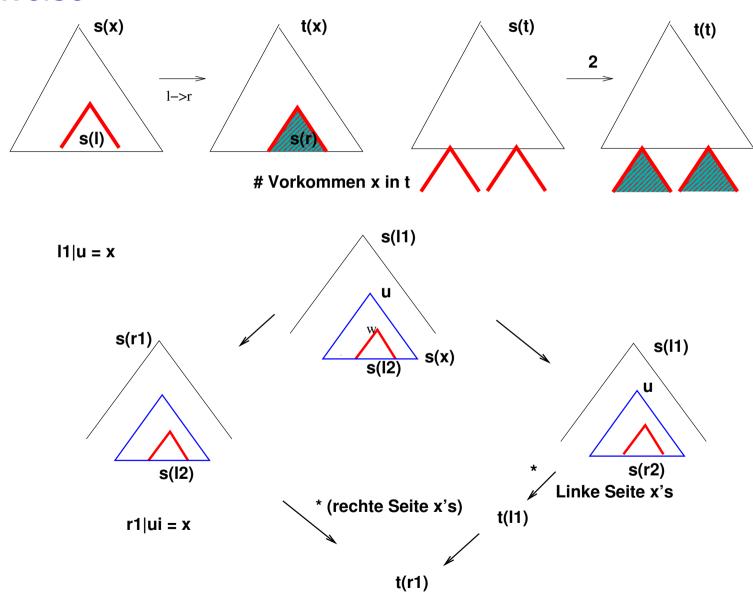
$$t_1 := \sigma(r_1) \leftarrow \sigma(l_1) \rightarrow \sigma(l_1)[\sigma(r_2)]_{uw} =: t_2$$

**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  $t_1 \leftrightarrow_{CP(R)} t_2$ .

Kritische Paare, Unifikation

### Beweise



### Konfluenztest

Satz 9.1 Hauptergebniss: Sei R ein Regelsystem.

- ▶ R ist genau dann lokal konfluent, wenn alle Paare  $(t_1, t_2) \in CP(R)$  zusammenführbar sind.
- ▶ Ist R terminierend, so gilt: R konfluent gdw.  $(t_1, t_2) \in CP(R) \rightsquigarrow t_1 \downarrow t_2$ .
- ▶ Sei R linear (d.h. für  $I, r \in I \rightarrow r \in R$  kommen Variablen höchstens einmal vor). Gilt  $CP(R) = \emptyset$ , so ist R konfluent.

**Beispiel 9.5** Sei  $R = \{f(x,x) \rightarrow a, f(x,s(x)) \rightarrow b, a \rightarrow s(a)\}.$  R ist lokal konfluent, aber nicht konfluent:

$$a \leftarrow f(a, a) \rightarrow f(a, s(a)) \rightarrow b$$

aber nicht a  $\downarrow$  b. R ist weder terminierend noch links-linear.

# Beispiel (Fort.)

$$R = \{f(f(x)) \rightarrow g(x)\}$$

$$t_1 = g(f(x)) \leftarrow f(f(f(x))) \rightarrow f(g(x)) = t_2$$

Es gilt nicht,  $t_1 \downarrow_R t_2 \rightsquigarrow R$  nicht konfluent.

Füge Regel  $t_1 \rightarrow t_2$  zu R hinzu.  $R_1$  ist äquivalent zu R, terminierend und konfluent.

$$g(f(f(x)))$$

$$f(g(f(x)))$$

$$g(g(x))$$

$$f(f(g(x)))$$

- ►  $R = \{x + 0 \rightarrow x, x + s(y) \rightarrow s(x + y)\}$ , linear ohne krit. Paare  $\rightsquigarrow$  confluent.
- ▶  $R = \{f(x) \rightarrow a, f(x) \rightarrow g(f(x)), g(f(x)) \rightarrow f(h(x)), g(f(x)) \rightarrow b\}$  ist lokal konfluent aber nicht konfluent.

### Konfluenz ohne Terminierung

**Definition 9.10**  $\epsilon - \epsilon$  - *Eigenschaften. Sei*  $\stackrel{\epsilon}{\rightarrow} = \stackrel{0}{\rightarrow} \cup \stackrel{1}{\rightarrow}$ .

- ▶ R heißt  $\epsilon \epsilon$  abgeschlossen , falls für jedes kritische Paar  $(t_1, t_2) \in CP(R)$  es ein t gibt mit  $t_1 \xrightarrow{\epsilon}_{R} t \xleftarrow{\epsilon}_{R} t_2$  .
- ▶  $R \text{ heißt } \epsilon \epsilon \text{ konfluent } gdw. \leftarrow \circ \rightarrow R \subseteq R \rightarrow R \circ \leftarrow R$

**Folgerung 9.2**  $\rightarrow$   $\epsilon - \epsilon$  konfluent  $\rightsquigarrow$   $\rightarrow$  streng konfluent.

- ▶ R  $\epsilon \epsilon$  abgeschlossen  $\Rightarrow$  R  $\epsilon \epsilon$  konfluent  $R = \{f(x,x) \rightarrow a, f(x,g(x)) \rightarrow b, c \rightarrow g(c)\}$ .  $CP(R) = \emptyset$ , d.h. R  $\epsilon \epsilon$  abgeschlossen aber  $a \leftarrow f(c,c) \rightarrow f(c,g(c)) \rightarrow b$ , d.h. R nicht konfluent  $\xi$ .
- ▶ Ist R linear und  $\epsilon \epsilon$  abgeschlossen , dann ist R streng konfluent also konfluent (Zeige R ist  $\epsilon \epsilon$  konfluent).

Diese Bedingungen sind leider zu einschränkend für die Programmierung.

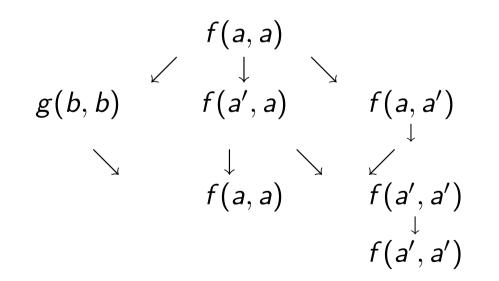
## Beispiel

**Beispiel 9.6** R links linear  $\epsilon - \epsilon$  abgeschlossen reicht nicht aus:

$$R = \{f(a,a) \rightarrow g(b,b), a \rightarrow a', f(a',x) \rightarrow f(x,x), f(x,a') \rightarrow f(x,x), g(b,b) \rightarrow f(a,a), b \rightarrow b', g(b',x) \rightarrow g(x,x), g(x,b') \rightarrow g(x,x)\}$$

Es gilt  $f(a', a') \stackrel{*}{\underset{R}{\longleftrightarrow}} g(b', b')$  jedoch nicht  $f(a', a') \downarrow_R g(b', b')$ .

R links linear  $\epsilon - \epsilon$  abgeschlossen :



### Parallel Reduktion

Beachte: Seien  $\rightarrow$ ,  $\Rightarrow$  mit  $\stackrel{*}{\rightarrow} = \stackrel{*}{\Rightarrow}$ . (Oft:  $\rightarrow \subseteq \Rightarrow \subseteq \stackrel{*}{\rightarrow}$ ). Dann ist  $\rightarrow$  konfluent gdw.  $\Rightarrow$  konfluent.

#### **Definition 9.11** Sei R Regelsystem.

- ▶ Die Parallelreduktion,  $\mapsto_R$ , ist definiert durch  $t \mapsto_R t'$  gdw.  $\exists U \subset O(t) : \forall u_i, u_j (u_i \neq u_j \rightsquigarrow u_i | u_j) \ \exists I_i \to r_i \in R, \sigma_i \text{ mit } t|_{u_i} = \sigma_i(I_i) :: t' = t[\sigma_i(r_i)]_{u_i} (u_i \in U) \ (t[u_1 \leftarrow \sigma_1(r_1)]...t[u_n \leftarrow \sigma_1(r_n)]).$
- ► Ein kritisches Paar von  $R: (\sigma(r_1), \sigma(l_1[r_2]_u)$  ist parallel 0-zusammenführbar falls  $\sigma(l_1[r_2]_u) \mapsto_R \sigma(r_1)$ .
- R ist parallel 0-abgeschlossen falls jedes kritische Paar von R parallel 0-zusammenführbar ist.

Eigenschaften:  $\mapsto_R$  ist stabil und monoton. Es gilt  $\mapsto_R^* = \xrightarrow{*}_R$  und somit ist  $\mapsto_R$  konfluent, so auch  $\to_R$ .



### Parallel Reduktion

**Satz 9.2** Ist R links-linear und parallel 0-abgeschlossen, so ist  $\mapsto_R$  streng konfluent, also konfluent, und somit ist auch R konfluent.

- **Folgerung 9.3** Erfüllt R die O'Donnel Bedingungen, so ist R konfluent. O'Donnel Bedingungen: R links-linear,  $CP(R) = \emptyset$ , R links-sequentiell (Redexe sind beim Lesen der Terme von links nach rechts eindeutig:  $f(g(x,a),y) \rightarrow 0$ ,  $g(b,c) \rightarrow 1$  hat diese Eigenschaft nicht. Durch Umgruppieren der Argumente kann oft die Eigenschaft erreicht werden, etwa  $f(g(a,x),y) \rightarrow 0$ ,  $g(b,c) \rightarrow 1$ 
  - ▶ Orthogonale Systeme:: R links-linear und  $CP(R) = \emptyset$ , so R konfluent. (In Lit. auch als reguläre Systeme bezeichnet).
  - ▶ Variationen: R ist stark-abgeschlossen, falls für jedes kritische Paar (s,t) is Terme u,v gibt mit  $s \stackrel{*}{\to} u \stackrel{\leq 1}{\longleftarrow} t$  und  $s \stackrel{\leq 1}{\to} v \stackrel{*}{\longleftarrow} t$ . R linear und stark-abgeschlossen, so R streng-konfluent.



## Fogerungen

- Folgt aus  $CP(R) = \emptyset$  die Konfluenz? Nein.  $R = \{f(x,x) \to a, g(x) \to f(x,g(x)), b \to g(b)\}$ . Betrachte  $g(b) \to f(b,g(b)) \to f(g(b),g(b)) \to a$  "Outermost" Reduktion.  $g(b) \to g(g(b)) \stackrel{*}{\to} g(a) \to f(a,g(a))$  nicht zusammenführbar.
- Reguläre Systeme können nicht terminierend sein:  $\{f(x,b) \rightarrow d, a \rightarrow b, c \rightarrow c\}$ . Offenbar  $CP = \emptyset$ .  $f(c,a) \rightarrow f(c,b) \rightarrow d$   $\downarrow^*$   $f(c,a) \rightarrow f(c,b)$ . Beachte f(c,a) hat eine Normalform.  $\rightsquigarrow$  Reduktionsstrategien die Normalformberechnend sind oder kürzeste Berechnungen liefern.
- ▶ Ein Kontext ist ein Term mit "Lücken"  $\square$ , z.B.  $f(g(\square, s(0)), \square, h(\square))$  als "Baummuster" (pattern) zu Regel  $f(g(x, s(0)), y, h(z)) \rightarrow x$ . Lücken dürfen beliebig gefüllt werden.

### Terminierungs-Kriterien

**Satz 9.3** R ist genau dann terminierend, wenn es eine noethersche Partialordnung  $\succ$  auf den Grundtermen Term(F) gibt die monoton ist, so dass gilt  $\sigma(I) \succ \sigma(r)$  für jede Regel  $I \rightarrow r \in R$  und Grundsubstitution  $\sigma$ .

**Beweis:**  $\curvearrowright$  Definiere  $s \succ t$  gdw.  $s \stackrel{+}{\rightarrow} t$   $(s, t \in Term(F))$   $\curvearrowright$  Ang.  $\rightarrow_R$  nicht terminierend,  $t_0 \rightarrow t_1 \rightarrow ...(V(t_i) \subseteq V(t_0))$ . Sei  $\sigma$  eine Grundsubstitution mit  $V(t_0) \subset D(\sigma)$ , dann  $\sigma(t_0) \succ \sigma(t_1) \succ ... \not\downarrow$ . **Problem:** Unendlicher Test.

**Definition 9.12** Eine Reduktionsordnung ist Partialordnung  $\succ$  auf Term(F, V) mit  $(i) \succ ist$  Noethersch  $(ii) \succ ist$  stabil und  $(iii) \succ ist$  monoton.

**Satz 9.4** R ist genau dann Noethersch, wenn es eine Reduktionsordnung  $\succ$  gibt mit  $I \succ r$  für alle  $I \rightarrow r \in R$ 

## Terminierungs-Kriterien

Beachte: Es gibt keine totalen Reduktionsordnungen.

$$x \succ y? \rightsquigarrow \sigma(x) \succ \sigma(y)$$

f(x,y) > f(y,x)? Kommutativität kann nicht gerichtet werden.

Beispiele für Reduktionsordnungen:

Knuth-Bendix Ordnungen: Gewicht für jedes Funktionssymbol und Präzedenz auf F.

Rekursive Pfad Ordnungen (RPO): Präzedenz auf F wird rekursiv auf Pfade (Wörter) in den zu vergleichenden Termen fortgesetzt.

Lexikographische Pfad Ordnungen (LPO), Polynominterpretationen, usw.

$$f(f(g(x))) = f(h(x)) \quad f(f(x)) = g(h(g(x))) \quad f(h(x)) = h(g(x))$$
 $\downarrow KB \quad \rightarrow \quad l(f) = 3 \quad l(g) = 2 \quad \rightarrow \quad l(h) = 1 \quad \rightarrow \quad \leftarrow$ 
 $\downarrow KPO \quad \leftarrow \quad g > h \quad > f \quad \leftarrow \quad \leftarrow$ 

Konfluenz Modulo Aquivalenzrelation (z.B. AC):

 $R:: f(x,x) \to g(x)$   $G:: \{(a,b)\}$   $g(a) \leftarrow f(a,a) \sim f(a,b)$  jedoch nicht  $g(a) \downarrow_{\sim} f(a,b)$ .

## Knuth-Bendix Vervollständigungsverfahren

**Eingabe:** E Gleichungsmenge,  $\succ$  Reduktionsordnung,  $R = \emptyset$ .

#### **Repeat** while *E* nicht leer

- (1) Entferne t = s aus E mit  $t \succ s$ ,  $R := R \cup \{t \rightarrow s\}$  sonst abort
- (2) Bringe rechte Seite der Regeln in Normalform mit R
- (3) Erweitere E um alle mit R normalisierten kritischen Paare die  $t \rightarrow s$  mit R bildet
- (4) Entferne alle Regeln aus R, deren linke Seite eine echte Instanz von t enthalten.
- (5) Verwende R um beide Seiten von Gleichungen aus E zu Normalisieren. Entferne Identitäten.

Ausgang: Terminierung mit R konvergent, äquivalent zu E. Abbruch (abort), Nicht Terminierung (läuft unendlich lange).

### Beispiele für Knuth-Bendix-V

**Beispiel 9.7**  $\blacktriangleright$  *WES*::  $\Sigma = \{a, b, c\}, E = \{a^2 = \lambda, b^2 = \lambda, ab = c\}$  $u < v \text{ gdw. } |u| < |v| \text{ oder } |u| = |v| \text{ und } u <_{lex} v \text{ mit } a <_{lex} b <_{lex} c$  $E_0 = \{a^2 = \lambda, b^2 = \lambda, ab = c\}, R_0 = \emptyset$  $E_1 = \{b^2 = \lambda, ab = c\}, R_1 = \{a^2 \to \lambda\}, CP_1 = \emptyset$  $E_2 = \{ab = c\}, R_2 = \{a^2 \to \lambda, b^2 \to \lambda\}, CP_2 = \emptyset$  $R_3 = \{a^2 \rightarrow \lambda, b^2 \rightarrow \lambda, ab \rightarrow c\}, NCP_3 = \{(b, ac), (a, cb)\}$  $E_3 = \{b = ac, a = cb\}$  $R_4 = \{a^2 \rightarrow \lambda, b^2 \rightarrow \lambda, ab \rightarrow c, ac \rightarrow b\}, NCP_4 = \emptyset, E_4 = \{a = cb\}$  $R_5 = \{a^2 \rightarrow \lambda, b^2 \rightarrow \lambda, ab \rightarrow c, ac \rightarrow b, cb \rightarrow a\}, NCP_5 = \emptyset, E_5 = \emptyset$