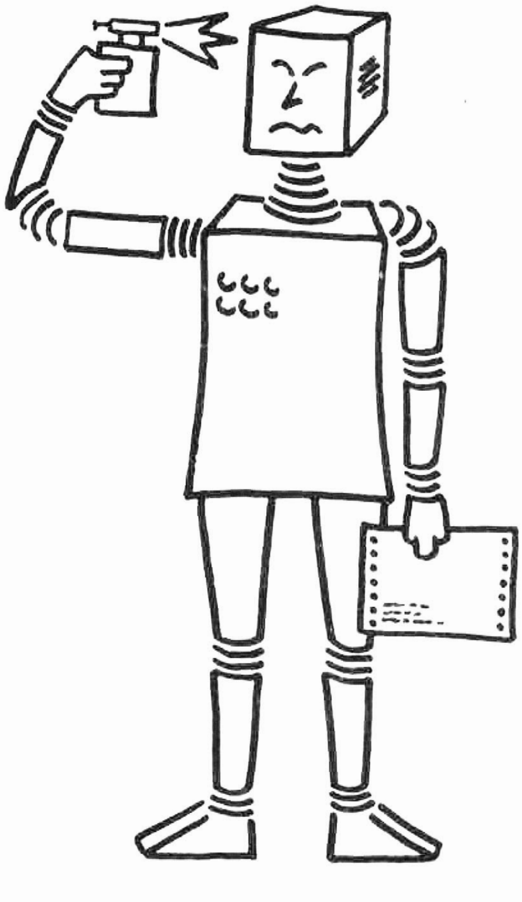
SEKI-PROJEKT

SEKI MEMO

Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern 1, W. Germany



A Completion Procedure
for Globally Finite
Term Rewriting Systems

Richard Göbel

Memo SEKI-83-12

A COMPLETION PROCEDURE FOR GLOBALLY FINITE TERM REWRITING SYSTEMS

Richard Göbel

Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern
West-Germany

November 1983

This research was supported by the Bundesministerium für Forschung und Technologie under contract IT.8302363.

Abstract

The Knuth-Bendix Algorithm is not able to complete term rewriting systems (TRS) with cyclic rules such as commutativity, associativity etc. A solution for this problem is to separate the cyclic rules from the TRS and incorporate them into the unification algorithm. This requires a unification algorithm for the specific theory. Usually it is very difficult to develop such algorithms, and we therefore discuss another way to solve this problem.

We propose an extension of the completion procedure for globally finite TRS. For a globally finite TRS cycles may occur in a reduction chain, but for each term there is only a finite set of reductions. A confluent and globally finite TRS R provides a decision procedure for the equality induced by R:

Two terms are equal iff there is a common term in their reduction sets.

This extension requires new methods for testing whether a given TRS is confluent and globally finite. For proving that a TRS is globally finite appropriate ordering relations are required, which are usually reflexive. Proving the confluence of a globally finite TRS is more difficult than for nötherian TRS and the set of critical pairs which has to be tested is in general infinite. Therefore a completion procedure with this test may not terminate. Further work has to be done to decrease this critical pair set.

Contents

1. A decision procedure for the equalitiy based on globally finite and confluent TRS 1

2. Ordering relations for proving a TRS globally finite 4

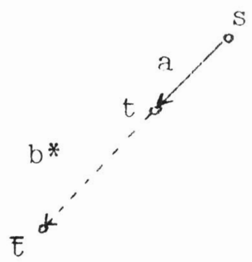
3. A confluence test for globally finite TRS 15

Technical Remarks

We assume familiarity of the reader with the basic proofs and results of the Knuth-Bendix Algorithm (e.g. [HUE 77], [HUE 80], [KB 70]).
We denote by:

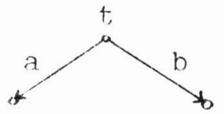
s, t	terms
u, v, w	occurrences in terms
a, b, c, d $(\alpha, \beta), (\gamma, \delta)$	} rewriting rules
σ	substitutions
R	TRS (always finite in this paper)
$s \rightarrow_R t$	one step reductions
$s \stackrel{*}{\rightarrow}_R t$	reduction in a finite number of steps we omit R, if we consider only one TRS
$\{t \mid s \stackrel{*}{\rightarrow} t\}$	reduction set of s
$\Sigma = \Sigma_0 \cup \ldots \cup \Sigma_n$	set of operators

diagrams:

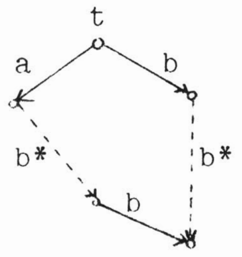


s is reduced to t by a and t is reduced by multiple application of b to $\bar{t}$.

For example:



If a is applied at a variable occurrence of b (the occurrence where a is applied is matched by a variable of b), then we can find a term $\bar{t}$ and reductions to $\bar{t}$ (result of Knuth and Bendix in [KB 70]):



1. A decision procedure for the equality based on globally finite and confluent TRS

First, we define the globally finiteness property for TRS. This definition is equivalent to the definition given in [HUE 77].

Definition 1.1

A TRS R is globally finite iff for each term s the reduction set is finite:

$$\forall s: \exists n: |\{t|\ s \overset{*}{\rightarrow} t\}| < n$$

Given a confluent TRS R, the test for the equality induced by R for two terms s,t is equivalent to the test whether there is a common term in their reduction sets. If R is also globally finite, each reduction set is finite and the equality is decidable:

Decision Procedure 1

(1) $i := 0,\ M_1(0) := \{s\}$

(2) $i := i + 1,\ M_1(i) := \{t|s \rightarrow t \wedge s \in M_1(i-1)\} \cup M_1(i-1)$

$M_1(i) = M_1(i-1)$? true: → 3

false: → 2

(3) $j := 0,\ M_2(0) := \{t\}$

(4) $j := j + 1,\ M_2(j) := \{t|s \rightarrow t \wedge s \in M_2(j-1)\} \cup M_2(j-1)$

$M_2(j) = M_2(j-1)$? true: → 5

false: → 4

(5) $M_1(i) \quad M_2(j) = 0$?

true: s,t not equal

false: s,t equal

This test is rather inefficient as it enumerates the complete reduction sets for both terms. For a more efficient decision procedure, we will distinguish between cyclic and reduction

rules. Cyclic rules always create terms with the same reduction set as the original term.
After applying a reduction rule, the original term does not belong to the reduction set of the resulting term.

Definition 1.2

Let R be a globally finite TRS.
A rule $(\alpha,\beta) \in R$ is a cyclic rule iff:
$\forall s: s/u = \sigma(\alpha) \Rightarrow s[u \leftarrow \sigma(\beta)] \overset{*}{\rightarrow} s$

Definition 1.3

Let R be a globally finite TRS.
A rule $(\alpha,\beta) \in R$ is a reduction rule iff:
$\forall s: s/u = \sigma(\alpha) \Rightarrow s \notin \{t \mid s[u \leftarrow \sigma(\beta)] \overset{*}{\rightarrow} t\}$

Note:
There might be rules which are neither cyclic nor reduction rules:

Example:

$R = \{(f(x,g(y,z)),f(g(y,z),x))\}$

$f(x,g(y,z)) \rightarrow f(g(y,z),x) \overset{*}{\not\rightarrow} f(x,g(y,z))$
$f(g(c,c),g(y,z)) \rightarrow f(g(y,z),g(c,c)) \rightarrow f(g(c,c),g(y,z))$

A test for checking whether a rule is a cyclic or reduction rule is given in chapter 2.

If we are able to split a TRS R into cyclic rules CR and reduction rules RR, we can define a more efficient decision procedure for the equality.
First, we note that there is a least cycle for each reduction set, since otherwise R can not be confluent.

Second, if there is a common term in two different reduction sets, this term has a unique least cycle, which is the least cycle of these two sets. Thus it is sufficient for testing the equality of two terms s,t to compare the two least cycles of the reduction sets of s and t:

<u>Decision Procedure 2</u>

(1) $i := 0$, $s(i) := s$, $M_1(i) := \{s(i)\}$

(2) is there any reduction rule which reduces s(i) to a term s(i+1)?
true: → 4
false: → 3

(3) is there any cyclic rule which reduces a term in $M_1(i)$ to a <u>new</u> term $s(i+1) \notin M_1(i)$?
true: $M_1(i+1) := M_1(i) \cup \{s(i+1)\}$, $i := i+1$, → 2
false: → 5

(4) $M_1(i+1) := \{s(i+1)\}$, $i := i+1$, → 2

(5) $j := 0$, $t(j) := t$, $M_2(0) := \{t(i)\}$

(6) is there any reduction rule which reduces t(j) to a term t(j+1)?
true: → 8
false: → 7

(7) is there any cyclic rule which reduces a term in $M_2(j)$ to a <u>new</u> term $t(j+1) \notin M_2$?
true: $M_2(j+1) := M_2(j) \cup \{t(j+1)\}$, $j := j+1$, → 6
false: → 5

(8) $M_2(j+1) := \{t(j+1)\}$, $j := j+1$, → 6

(9) $\bar{s}$ is an arbitrary term in $M_1(i)$
$\bar{s} \in M_2(j)$?
true: s,t equal
false: s,t not equal

2. Ordering relations for proving a TRS globally finite

In this chapter we show how to prove a TRS to be globally finite by some simple tests. These tests are very similar to the usual tests for nötherian TRS where the only difference between the used ordering relations is, that ours are usually reflexive. Thus we are able to extend many of the existing orderings for our purposes.

Definition 2.1

$\leq$ is a partial ordering relation for terms iff:

- $s \leq s$ — reflexive
- $s \leq t \wedge t \leq \bar{t} \Rightarrow s \leq \bar{t}$ — transitive
- $\{t \mid t \leq s\}$ finite for all s

Note:

For two different terms s,t on the same cycle of a reduction chain we have both $s \leq t$ and $t \leq s$. Thus the antisymmetric axiom for partial ordering relations $s \leq t \wedge t \leq s \Rightarrow s = t$ is missing here.

For proving a TRS globally finite, we have to show that each reduction is not greater than the original term:

Lemma 2.2

$\forall (\alpha,\beta) \in R: \forall s: s/u = \sigma(\alpha) \Rightarrow s[u \leftarrow \sigma(\beta)] \leq s$

$\Rightarrow$ R is globally finite

Proof

Assume R is not globally finite.

Then there is an infinite reduction chain:

$s_1 \to s_2 \to \ldots \to s_n \to \ldots$

with $s_i = s_j \Rightarrow i = j$

and:

$s_1 \geq s_2 \geq \ldots \geq s_n \geq \ldots$ premise

$\Rightarrow \{t \mid t \leq s_1\}$ is infinte (contradiction) •

Another restriction for orderings seems to be natural. For comparing two terms which differ only at a certain subtermposition, we should compare only these subterms. With this restriction, we get a lemma with a weaker premise:

Lemma 2.3

Let $\leq$ be a partial ordering for terms with:

$\forall\, s,t,\bar{t}: \;\forall u \in O(\bar{t}): \; s \leq t \;\; \Rightarrow \;\; \bar{t}[u \leftarrow s] \leq \bar{t}[u \leftarrow t]$

Then R is globally finite if for every rule $(\alpha,\beta) \in R : \sigma(\beta) \leq \sigma(\alpha)$ holds.

Proof

$s/u = \sigma(\alpha)$

$\Rightarrow \;\; s[u \leftarrow \sigma(\beta)]/u \leq s/u$

$\Rightarrow \;\; s[u \leftarrow \sigma(\beta)] \leq s$ premise. ●

Some simple orderings are extendable now:

The depth of a term is the maximal length of a path in its tree representation. A term s is not greater than a term t if its depth is less or equal than the depth of t.

Definition 2.4

The depth of a term t is defined as:

$$D(t) := \begin{cases} 0 & t \in V \wedge t \in \Sigma_0 \\ MAX(\{D(t_1),\dots,D(t_n)\})+1 & t=f(t_1,\dots,t_n) \wedge f \in \Sigma \end{cases}$$

Lemma 2.5

The relation $\leq$:

$s \leq t \;\; \Leftrightarrow \;\; D(s) \leq D(t)$

is a partial ordering relation in the sense of definition 2.1

The reflexivity and the transitivity of this relation is obvious.
The set of terms of a limited depth is finite, thus we can prove the third property of definition 2.1 too.
For testing whether a rule does not increase the depth of a term, we have to check whether the right hand side is not greater than the left hand side of the rule and the occurrences of variables of the right hand side are not deeper than on the left hand side.

Definition 2.6

The depth of an occurrence is defined as:

$$d(u) := \begin{cases} 0 & u = \varepsilon \\ d(\bar{u})+1 & u = i.\bar{u} \wedge i \in \mathbf{N} \end{cases}$$

Note:
The depth of a term is equal to the maximal depth of its occurrences:

$$D(s) = \mathrm{Max}\{d(u) \mid u \in O(s)\}$$

Lemma 2.7

Let (α,β) be a rewrite rule:

$\forall\, x \in O(\alpha)$: $s/u = x \wedge t/v = x \;\Rightarrow\; d(u) \geq d(v) \wedge D(\beta) \leq D(\alpha)$

$\Rightarrow\; \forall\, s$: $s/u = \sigma(\alpha) \;\Rightarrow\; D(s[u \leftarrow \sigma(\beta)]) \leq D(s)$

Proof

First we show for $\leq$:

(i) $\forall\, s,t,\bar{t}$: $\forall\, u \in O(\bar{t})$: $s \leq t \;\Rightarrow\; \bar{t}[u \leftarrow s] \leq \bar{t}[u \leftarrow t]$

Assume: v is the occurrence in $\bar{t}[u \leftarrow s]$ with the maximal depth.

We obtain two cases:

- v overlaps with $u \Rightarrow v = u.\bar{v}$
  $\Rightarrow \bar{v}$ is a deepest occurrence in s
  $\Rightarrow \exists\, \bar{w} \in O(t)\colon d(\bar{w}) > d(\bar{v})$
  $\Rightarrow u.\bar{w} \in O(\bar{t}[u \leftarrow t]) \wedge d(u.\bar{w}) > d(u.\bar{v})$
  $\Rightarrow D(\bar{t}[u \leftarrow t]) > D(\bar{t}[u \leftarrow s])$

- v does not overlap with u
  $\Rightarrow v \in O(\bar{t}[u \leftarrow t])$
  $\Rightarrow D(\bar{t}[u \leftarrow t]) > D(\bar{t}[u \leftarrow s])$

and the proof for (i) is complete.
Now it is sufficient to prove:
$\forall\, \sigma\colon \sigma(\beta) \leq \sigma(\alpha)$ (by Lemma 2.3)

Assume: $D(\alpha) > D(\beta)$

$\exists\, w \in O(\sigma(\beta))\colon D(\sigma(\beta)) = d(w)$

<u>$w \in O(\beta)$</u>
$\Rightarrow D(\beta) = D(\sigma(\beta))$
$\Rightarrow D(\sigma(\beta)) \leq D(\alpha)$ $\qquad D(\alpha) > D(\beta)$
$\Rightarrow D(\sigma(\beta)) \leq D(\sigma(\alpha))$

<u>$w \notin O(\beta)$</u>
$\Rightarrow w = w_1.w_2 \wedge \beta/w_1 \in V$
$\Rightarrow \exists\, \bar{w}_1\colon \alpha/\bar{w}_1 = \beta/w_1 \in V$ $\qquad V(\beta) \subset V(\alpha)$
$\Rightarrow d(\bar{w}_1) > d(w_1)$ $\qquad$ premise of Lemma
$\Rightarrow d(\bar{w}_1.w_2) > d(w_1.w_2)$
$\Rightarrow d(\bar{w}_1.w_2) > D(\sigma(\beta))$
$\Rightarrow D(\sigma(\alpha)) > D(\sigma(\beta))$

We get another simple ordering relation by comparing the number of operators (length) of terms:

Definition 2.8

We define the number of operators of a term t as:

$$\#(t) = \begin{cases} 0 & t \in V \\ 1 & t \in \Sigma_0 \\ \#(t_1)+\ldots+\#(t_n)+1 & t=f(t_1,\ldots,t_n) \wedge f \in \Sigma \end{cases}$$

The ordering relation $t<s \Leftrightarrow \#(t) < \#(s)$ can be proved to be a partial ordering relation as in Definition 2.1. This proof is very simple and omitted.

For proving a TRS to be globally finite with this ordering, the number of operators in the left hand side has to be not less than those in the right hand side of the rule and for any variable, there are not more occurrences in the right hand side than in the left hand side.

Definition 2.9

The number of variable occurrences for a variable x in a term t is defined as:

$$\#(x,t) = \begin{cases} 0 & (t \in V \wedge t \neq x) \vee t \in \Sigma_0 \\ 1 & t = x \\ \#(x,t_1)+\ldots+\#(x,t_n) & s=f(t_1,\ldots,t_n) \wedge f \in \Sigma_n \end{cases}$$

Lemma 2.10

Let (α,β) be a rewrite rule:

$(\forall\ x\colon \#(x,\alpha) \geq \#(x,\beta)) \wedge \#(\alpha) \geq \#(\beta)$

$\Rightarrow \forall s\colon s/u = \sigma(\alpha) \Rightarrow \#(s[u \leftarrow \sigma(\beta)]) \leq \#(s)$

Proof

$\#(\bar{t}[u \leftarrow s]) = \#(\bar{t}) - \#(\bar{t}/u) + \#(s)$

$\#(\bar{t}[u \leftarrow t]) = \#(\bar{t}) - \#(\bar{t}/u) + \#(t)$

$\#(s) < \#(t)$

$\Rightarrow \#(\bar{t}) - \#(\bar{t}/u) + \#(s) < \#(\bar{t}) - \#(\bar{t}/u) + \#(t)$

$\Rightarrow \#(\bar{t}[u \leftarrow s]) < \#(\bar{t}[u \leftarrow t])$

$\Rightarrow$ prove: $\forall \sigma: \sigma(\beta) < \sigma(\alpha)$

$\left.\begin{array}{l} \#(\alpha) > \#(\beta) \\ \#(x,\alpha) > \#(x,\beta) \end{array}\right\}$ premises

$\Rightarrow \#(x,\alpha) * \#(\sigma(x)) > \#(x,\beta) * \#(\sigma(x))$

$$\Rightarrow \#(\alpha) + \sum_{x \in V(\alpha)} \#(x,\alpha) * \#(\sigma(x))$$

$$> \#(\beta) + \sum_{x \in V(\beta)} \#(x,\beta) * \#(\sigma(x)) \qquad V(\beta) \subseteq V(\alpha)$$

with:

$$\#(\sigma(s)) = \#(s) + \sum_{x \in V(s)} \#(x,s) * \#(\sigma(x))$$

$\Rightarrow \#(\sigma(\alpha)) > \#(\sigma(\beta))$

Although these orderings are much simpler than many of the existing orderings for nötherian TRS, we can for example prove that a TRS with abelian group axioms is globally finite:

<u>Example</u>

(1) $f(x,y) \rightarrow f(y,x)$

(2) $f(f(x,y),z) \rightarrow f(x,f(y,z))$

(3) $f(x,0) \rightarrow x$

(4) $f(x,I(x)) \rightarrow 0$

$\#(f(x,y)) = 1 = \#(f(y,x))$

$\#(f(f(x,y),z)) = 2 = \#(f(x,f(y,z)))$

$\#(f(x,0)) = 2 \qquad \#(x) = 0$

$\#(f(x,I(x))) = 2 \qquad \#(0) = 1$

For each variable the number of occurrences on the left hand side is not less than on the right hand side of these rules. Hence, this TRS is globally finite.

In fact it might be "easier" to prove a TRS globally finite than proving it nötherian, because global finiteness is the weaker property:

R nötherian <=> R globally finite
$\wedge\ s \ddagger t \Rightarrow t \overset{*}{\not\to} s$

Although the proof for global finiteness might be "easier", there is no way to find a general ordering relation, for this problem is still undecidable.

We get another ordering relation by combining the depth and length ordering. First we split the set of operators Σ into two disjoint sets F and G. The computation of the depth is restricted to operators of F and only operators from G are counted to get the number of operators. The definition of the depth of an occurrence is different too, only operators from F are considered for computing the depth.

<u>Definition 2.11</u>

Split Σ into two disjoint sets F and G:

$\Sigma = F \cup G \ \wedge\ F \cap G = 0$

We define the depth of a term as:

$$D(t) := \begin{cases} 1 & t \in V \cup (t \in F \wedge t \in \Sigma_0) \\ 0 & t \in G \wedge t \in \Sigma_0 \\ MAX(\{D(t_1),\ldots,D(t_n)\}) & t = g(t_1,\ldots,t_n) \wedge g \in G \\ MAX(\{D(t_1),\ldots,D(t_n)\})+1 & t = f(t_1,\ldots,t_m) \wedge f \in F \end{cases}$$

$\#(t)$ is the number of operations in t:

$$\#(t) := \begin{cases} 1 & t\varepsilon G \wedge t\varepsilon\Sigma_0 \\ 0 & (t\varepsilon\Sigma_0 \wedge t\varepsilon F) \vee t\varepsilon V \\ \#(t_1)+\ldots+\#(t_n)+1 & t = g(t_1,\ldots,t_n) \wedge g\varepsilon G \\ \#(t_1)+\ldots+\#(t_n) & t = f(t_1,\ldots,t_m) \wedge f\varepsilon F \end{cases}$$

d(u,t) is the depth of an occurrence u in a term t:

$$d(u,t) := \begin{cases} 0 & u = \varepsilon \\ d(\bar{u},t_i) & u = i.\bar{u} \wedge s=g(t_1,\ldots,t_m) \wedge g\varepsilon G \\ & \quad \wedge 1\leq i\leq m \\ d(\bar{u},t_i)+1 & u = i.\bar{u} \wedge s=f(t_1,\ldots,t_m) \wedge f\varepsilon F \\ & \quad \wedge 1\leq i\leq m \end{cases}$$

A term t is not greater than s iff the depth of t and the number of operators in t is not greater than in s. For proving a TRS globally finite the two variable conditions for the depth and length ordering have to hold too:

<u>Lemma 2.12</u>

Let (α,β) be a rewrite rule:

$$D(\beta) \leq D(\alpha) \quad \wedge \quad \#(\beta) \leq \#(\alpha)$$
$$\wedge \quad \forall x\varepsilon V(\alpha): \alpha/u = x \quad \wedge \beta/v = x \quad \Rightarrow \quad \bar{d}(u,\alpha) \leq \bar{d}(v,\beta)$$
$$\wedge \quad \forall x\varepsilon V(\alpha): \#(x,\alpha) \geq \#(x,\beta)$$

$$\Rightarrow \quad \forall s: s/u = \sigma(\alpha) \quad \Rightarrow \quad D(s[u \leftarrow \sigma(\beta)]) \leq D(s)$$
$$\wedge \quad \#(s[u \leftarrow \sigma(\beta)]) \leq \#(s)$$

The proof of lemma 2.12 is similar to the proofs of lemma 2.10 and 2.7, thus we skip it.

These ordering relations are easily extendable for sorted algebras. Also weights for operators might be added. More work has to be done to find more general ordering relations, but

for first tests with the extended completion procedure, these orderings might be sufficient.
With these ordering relations we can split a globally finite TRS into cyclic and reduction rules. The test for cyclic rules is rather simple. The left hand side has to be in the reduction set of the right hand side of the rule:

Lemma 2.13
$(\alpha,\beta) \in R$ is a cyclic rule iff $\beta \overset{*}{\rightarrow} \alpha$.

Proof
(=>)
(α,β) is cyclic
$\alpha \rightarrow \beta$ with (α,β)
=> $\beta \overset{*}{\rightarrow} \alpha$

(<=)
$s/u = \sigma(\alpha)$
$s[u \leftarrow \sigma(\beta)]/u = \sigma(\beta)$
$\sigma(\beta) \overset{*}{\rightarrow} \sigma(\alpha)$ with $\beta \overset{*}{\rightarrow} \alpha$
$s[u \leftarrow \sigma(\beta)]/u \rightarrow s/u$
=> $s[u \leftarrow \sigma(\beta)] \rightarrow s$
=> (α,β) is cyclic rule ●

For testing whether a rule is a reduction rule, we need the same ordering relation which was used to prove that the TRS is globally finite. After applying a reduction rule to a term, the resulting term has to be less than the original term.

Lemma 2.14
Let $<$ be an ordering relation in the sense of Definition 2.1 and R is a TRS with:
$s \rightarrow t \Rightarrow t < s$
Then $(\alpha,\beta) \in R$ is a reduction rule if:

$\forall s: s/u = \sigma(\alpha) \Rightarrow s \nless s[u \leftarrow \sigma(\beta)]$

Proof

Assume: $s \nless s[u\leftarrow \sigma(\beta)] \wedge s[u \leftarrow \sigma(\beta)] \overset{*}{\rightarrow} s$

$\Rightarrow s < s[u \leftarrow \sigma(\beta)]$ transivity of $<$

(contradiction) ●

For the depth, length and combined ordering, the test of Lemma 2.14 is rather simple, and it is sufficient to prove $s \nless t$ for a reduction rule.

Lemma 2.15

R is a rewriting system:

$s \rightarrow t \Rightarrow t < s$

- $t < s \iff D(t) < D(s)$ (depth)
  $(\alpha,\beta) \in R \wedge \alpha \nless \beta$
  $\Rightarrow (\alpha,\beta)$ is a reduction rule

- $t < s \iff \#(t) < \#(s)$ (number of operators)
  $(\alpha,\beta) \in R \wedge \alpha \nless \beta \Rightarrow (\alpha,\beta)$ is a reduction rule

- $t < s \iff \#(t) < \#(s) \wedge D(t) < D(s)$ (combination)
  $(\alpha,\beta) \in R \wedge \alpha \nless \beta \Rightarrow (\alpha,\beta)$ is a reduction rule

Proof

We prove this lemma only for the depth ordering because the ideas of these proofs are very similar.

We prove:

$\forall s: s/u = \sigma(\alpha) \Rightarrow s \nless s[u \leftarrow \sigma(\beta)]$

Because of:

$s < t \iff \bar{t}[u \leftarrow s] < \bar{t}[u \leftarrow t]$

it is sufficient to prove:
$\forall\sigma: \sigma(\alpha) \nless \sigma(\beta)$

$\alpha \nless \beta \iff D(\alpha) \nless D(\beta) \iff D(\alpha) > D(\beta)$

We prove now:
$\forall\sigma: D(\sigma(s)) > D(\sigma(t))$

This proof is equivalent to the proof of Lemma 2.7. In the same way we can prove:
$\#(\sigma(s)) > \#(\sigma(t))$
$\#(\sigma(s)) > \#(\sigma(t)) \lor D(\sigma(s)) > D(\sigma(t))$

●

Note:
A set of reduction rules is a nötherian TRS.

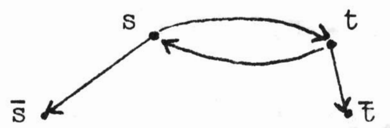
Now we are able to give a procedure that attempts to split a TRS into cyclic rules and reduction rules:

(1) prove R globally finite by an ordering relation <.
(2) test for each rule whether the left hand side of the rule is in the reduction set of the right hand side.
→ CR
(3) test with this ordering whether each rule in R - CR is a reduction rule:
successful?
true: stop with RR = R - CR
false: stop with failure

3. A confluence test for globally finite TRS

Unfortunately there seems to be no simple extension for the confluence test as for the test for global finiteness. The local confluence is not equivalent to the confluence of these systems.

For example:



This globally finite relation is locally confluent, but not confluent. Thus, the local confluence has to be extended for globally finite TRS. If we consider equivalence classes of terms, we are able to define a similar property.

Lemma 3.1

Let R be a globally finite TRS. The equivalence class $[s]$ is defined as: $[s] = \{t \mid s \overset{*}{\rightarrow} t \wedge t \overset{*}{\rightarrow} s\}$

Then the relation $\rightarrow\!>$:

$[s] \rightarrow\!> [t] \Leftrightarrow \exists \bar{s} \in [s]: \exists \bar{t} \in [t]: \bar{s} \rightarrow \bar{t}$

is nötherian.

Proof

Assume there is an infinite chain:

$[s_1] \rightarrow\!> [s_2] \rightarrow\!> [s_3] \rightarrow\!> \ldots$

All s_i are different, for if they were not there would be a cyle in this chain, and since all elements on a cycle have to be in the same equivalence class, this chain would be finite.

Lemma 3.2

Let R be a globally finite TRS. R is confluent iff:

$\forall s_1, s_2, t_1, t_2: s_1 \overset{*}{\rightarrow} s_2 \wedge s_2 \overset{*}{\rightarrow} s_1 \wedge s_1 \rightarrow t_1 \wedge s_2 \rightarrow t_2$
$\Rightarrow \exists \bar{t}: t_1 \overset{*}{\rightarrow} \bar{t} \wedge t_2 \overset{*}{\rightarrow} \bar{t}$

Proof

$(\Leftarrow)$ obvious

$(\Rightarrow)$ $\twoheadrightarrow$ is nötherian

$\quad \Rightarrow [s] \twoheadrightarrow [t_1] \wedge [s] \twoheadrightarrow [t_2]$

$\qquad \Rightarrow \exists[\bar{t}]: [t_1] \overset{*}{\to} [\bar{t}] \wedge [t_2] \overset{*}{\to} [\bar{t}]$

$\qquad \Leftrightarrow \twoheadrightarrow$ confluent.

$\quad [s] \twoheadrightarrow [t_1]$

$\Leftrightarrow \exists \bar{s}, \bar{t}_1: \bar{s} \varepsilon [s] \wedge \bar{t}_1 \varepsilon [t_1] \wedge \bar{s} \to \bar{t}_1$

$\quad [s] \twoheadrightarrow [t_2]$

$\Leftrightarrow \exists \bar{\bar{s}}, \bar{t}_2: \bar{\bar{s}} \varepsilon [s] \wedge \bar{t}_2 \varepsilon [t_2] \wedge \bar{\bar{s}} \to \bar{t}_2$

$\Rightarrow \bar{s} \overset{*}{\to} \bar{\bar{s}} \wedge \bar{\bar{s}} \overset{*}{\to} \bar{s}$

$\Rightarrow \exists \bar{t} \quad \bar{t}_1 \overset{*}{\to} \bar{t} \wedge \bar{t}_2 \overset{*}{\to} \bar{t}$ premise

$\Rightarrow \exists[\bar{t}]:[\bar{t}_1] \overset{*}{\twoheadrightarrow} [\bar{t}] \wedge [\bar{t}_2] \overset{*}{\twoheadrightarrow} [\bar{t}]$

$\Rightarrow \exists[\bar{t}]:[t_1] \overset{*}{\twoheadrightarrow} [\bar{t}] \wedge [t_2] \overset{*}{\twoheadrightarrow} [\bar{t}]$

$\Rightarrow \twoheadrightarrow$ confluent

$\quad s \overset{*}{\to} t_1 \wedge s \overset{*}{\to} t_2$

$\Rightarrow [s] \overset{*}{\twoheadrightarrow} [t_1] \wedge [s] \overset{*}{\twoheadrightarrow} [t_2]$

$\Rightarrow \exists [\bar{t}]: [t_1] \overset{*}{\twoheadrightarrow} [\bar{t}] \wedge [t_2] \overset{*}{\twoheadrightarrow} [\bar{t}]$

$\Rightarrow t_1 \overset{*}{\to} \bar{t} \wedge t_2 \overset{*}{\to} \bar{t}$

$\Rightarrow \to$ confluent

●

Lemma 3.2 does not give a simple test of critical pairs between rules.
In the next theorem we prove a test where we have to consider also critical pairs between critical pairs.

Theorem 3.3

Let R be a globally finite TRS, consisting of cyclic and reduction rules:

$R = CR \cup RR$

C(R), C(CR) and C(RR) are the smallest sets with:

- $(\alpha,\beta) \in CR \Rightarrow (\alpha,\beta) \in C(CR)$
  $(\beta,\alpha) \in C(CR)$

$(\alpha_1,\beta_1),(\alpha_2,\beta_2) \in C(CR) \wedge \sigma(\alpha_1)/u = \sigma(\alpha_2)$

$\Rightarrow (\sigma(\alpha_1)[u \leftarrow \sigma(\beta_2)],\sigma(\beta_1)) \in C(CR)$

$(\sigma(\beta_1),\sigma(\alpha_1)[u \leftarrow \sigma(\beta_2)]) \in C(CR)$

- $RR \subset C(RR)$

$(\alpha_1,\beta_1) \in C(RR) \wedge (\alpha_2,\beta_2) \in C(CR) \wedge \sigma(\alpha_1)/u = \sigma(\alpha_2)$

$\Rightarrow (\sigma(\alpha_1)[u \leftarrow \sigma(\beta_2)],\sigma(\beta_1)) \in C(RR)$

$(\alpha_1,\beta_1) \in C(RR) \wedge (\alpha_2,\beta_2) \in C(CR) \wedge \sigma(\alpha_1) = \sigma(\alpha_2)/u$

$\Rightarrow (\sigma(\beta_2),\sigma(\alpha_2)[u \leftarrow \sigma(\beta_1)]) \in C(RR)$

- $(\alpha_1,\beta_1),(\alpha_2,\beta_2) \in C(RR) \wedge \sigma(\alpha_1)/u = \sigma(\alpha_2)$

$\Rightarrow (\sigma(\alpha_1)[u \leftarrow \sigma(\beta_2)],\sigma(\beta_1)) \in C(R)$

with:

$(u \in O(\alpha_1) \wedge \alpha_1/u \notin V) \vee (u \in O(\alpha_2) \wedge \alpha_2/u \notin V)$

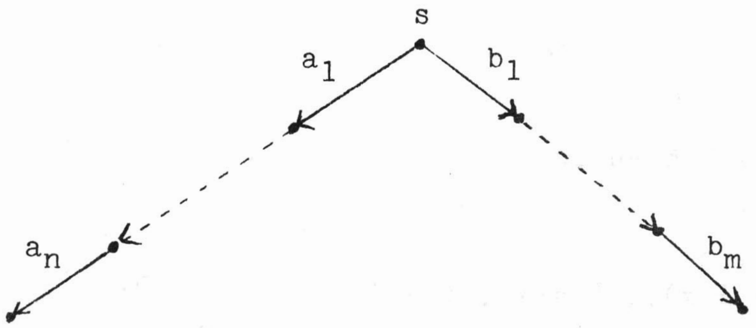
for C(CR), C(RR) and C(R).

R is confluent iff:

$\forall(\alpha,\beta) \in C(R): \exists t: \alpha \overset{*}{\rightarrow} t \wedge \beta \overset{*}{\rightarrow} t$

The proof theorem 3.3 is rather long, thus we will describe the ideas of the proof first.

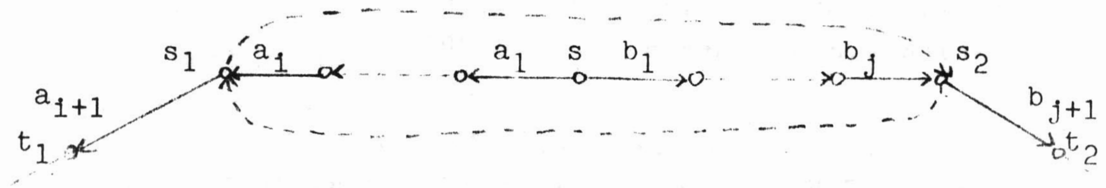
Consider two different reduction chains of a term s:



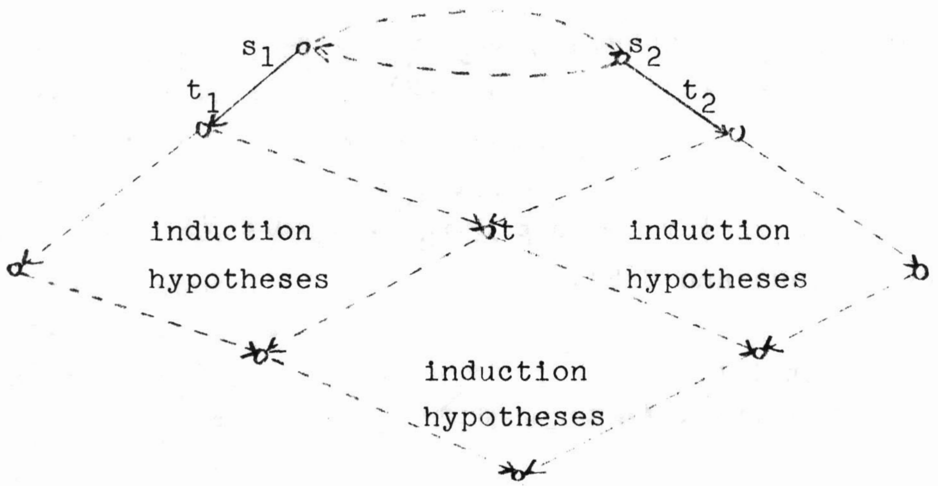
$a_1,\ldots,a_n,b_1,\ldots,b_m \in R$

$a_1,\ldots,a_i$ and $b_1,\ldots,b_j$ might be cyclic rules and a_{i+1},b_{j+1}

reduction rules:



Thus, there are reduction chains from s_1 to s_2 and vice versa. It is sufficient to prove the existence of a common term in the reduction sets of t_1 and t_2:



This proof is done by induction on the number of reduction rules in a chain.
In the following example we show how critical pairs have to be created for proving a TRS confluent.

<u>Example</u>

R is a TRS with 6 rules.

(a) $(f(x,x),c_1)$

(b) $(\bar{f}(x,x),c_2)$

(c) $(f(g(x),\bar{g}(x)),\ \bar{f}(g(x),g(x)))$

(d) $(\bar{f}(g(x),\bar{g}(x)),f(g(x),g(x)))$

(e) $(g(x),\bar{g}(x))$

(f) $(\bar{g}(x),g(x))$

(c),(d),(e) and (f) are cyclic rules, (a) and (b) are reduction rules.

We want to reduce the term: $f(g(x)\ g(x))$

$$\begin{array}{ccccc} & f(g(x),g(x)) \xrightarrow{e} f(g(x),\bar{g}(x)) \xrightarrow{c} \bar{f}(g(x),g(x)) & \\ \swarrow^{a} & & \searrow^{b} \\ c_1 & & c_2 \end{array}$$

=> R is not confluent.

There are no critical pairs between the reduction rules. Even between a and e there is no critical pair. Thus, we try to find critical pairs between cyclic rules first. We get a critical pair between rules (c) and (e):

($\bar{c}$) $(f(g(x),g(x)),\bar{f}(g(x),g(x)))$

If we use this pair as a cyclic rule, we can replace the rules (c) and (e) by ($\bar{c}$) in the reduction chain:

$$\begin{array}{ccccc} & f(g(x),g(x)) \xrightarrow{\bar{c}} \bar{f}(g(x),g(x)) & \\ \swarrow^{a} & & \searrow^{b} \\ c_1 & & c_2 \end{array}$$

Now we get another critical pair between a and $\bar{c}$:

($\bar{a}$) $(\bar{f}(g(x)\ g(x)),c_1)$

We replace (a) and ($\bar{c}$) by ($\bar{a}$):

$$\begin{array}{ccc} & \bar{f}(g(x),g(x)) & \\ \swarrow^{\bar{a}} & & \searrow^{b} \\ c_1 & & c_2 \end{array}$$

At last we get a critical pair between $\bar{a}$ and b

(c_1, c_2)

There is no way to reduce this critical pair to a common term, thus R is not confluent.

As a conclusion of this example, we have to create three kinds of critical pairs:

- between cyclic rules C(CR)
- between cyclic and reduction rule C(RR)
- between reduction rules C(R)

Critical pairs between cyclic rules and between reduction and cyclic rules are also considered as cyclic and reduction rules. Thus we have to create critical pairs between critical pairs too.

Before we start to prove the theorem we want to know whether it makes sense to consider critical pairs as reduction or cyclic rules (Lemma 3.5). In Lemma 3.4 we prove that we need not consider cyclic rules with left hand sides consisting of a single variable.

Lemma 3.4

Let R be a globally finite TRS consisting of cyclic rules CR and reduction rules RR:

$(\forall x : (x,x) \notin R) \Rightarrow \forall y,\beta : (y,\beta) \notin CR$

Proof

Assume: $(x,\beta) \in CR$

(i) $x \notin V(\beta)$

$\beta \overset{*}{\rightarrow} x$ $\quad (x,\beta)$ cyclic

This contradicts the variable condition for rewrite rules: $(\alpha,\beta) \in R \Rightarrow V(\beta) \subset V(\alpha)$

(ii) $\beta = f(t_1,\ldots,t_n) \wedge \exists\, u \in O(\beta)$: $\beta/u = x$

$\Rightarrow f(t_1,\ldots,t_n) \rightarrow f(t_1,\ldots,t_n)[u \leftarrow f(t_1,\ldots,t_n)] \quad (x,\beta)$

$\overset{*}{\rightarrow} f(t_1,\ldots,t_n)[u \leftarrow f(t_1,\ldots,t_n)[u \leftarrow f(t_1,\ldots t_n)]]$

.

.

.

$\Rightarrow$ R is not globally finite (contradiction)

Lemma 3.5

R is a globally finite TRS.

C(CR) and C(RR) are the sets of theorem 3.3:

$(\alpha,\beta) \in C(CR) \Rightarrow (\alpha,\beta)$ is a cyclic rule in C(CR)

$(\alpha,\beta) \in C(RR) \Rightarrow (\alpha,\beta)$ is a reduction rule in $C(RR) \cup C(CR)$

Proof

First we check the variable condition for these rules:

- cyclic rules

  $(\alpha,\beta) \in CR$

  Assume $\exists\, x$: $x \in V(\alpha) \wedge x \notin V(\beta)$

  $\Rightarrow \beta \overset{*}{\not\rightarrow} \alpha$ $\quad$ variable condition

  $\Rightarrow (\alpha,\beta)$ is not cyclic $\quad$ (contradiction)

  $\Rightarrow V(\alpha)=V(\beta)$

  $(\alpha_1,\beta_1), (\alpha_2,\beta_2) \in C(CR) \wedge \sigma(\alpha_1)/u = \sigma(\alpha_2)$

  $\Rightarrow V(\sigma(\alpha_1)) = V(\sigma(\beta_1))$

  $\Rightarrow V(\sigma(\alpha_2)) = V(\sigma(\beta_2))$

$\Rightarrow V(\sigma(\alpha_1)[u \leftarrow \sigma(\beta_2)]) = V(\sigma(\alpha_1))$
$\Rightarrow V(\sigma(\alpha_1)[u \leftarrow \sigma(\beta_2)]) = V(\sigma(\beta_1))$

- reduction rules
  $(\alpha_1,\beta_1) \in C(CR)$ is a cyclic rule
  $(\alpha_2,\beta_2) \in C(RR)$ is a reduction rule

  (i) $\sigma(\alpha_1) = \sigma(\alpha_2)/u$
  $\Rightarrow V(\sigma(\beta_2)) \subset V(\sigma(\alpha_2))$
  $\Rightarrow V(\sigma(\alpha_1)) = V(\sigma(\beta_1))$
  $\Rightarrow V(\sigma(\alpha_2)[u \leftarrow \sigma(\beta_1)]) = V(\sigma(\alpha_2))$
  $\Rightarrow V(\sigma(\beta_2)) \subset V(\sigma(\alpha_2)[u \leftarrow \sigma(\beta_1)])$

  (ii) $\sigma(\alpha_1)/u = \sigma(\alpha_2)$
  $\Rightarrow V(\sigma(\beta_2)) \subset V(\sigma(\alpha_2))$
  $\Rightarrow V(\sigma(\alpha_1)) = V(\sigma(\beta_1))$
  $\Rightarrow V(\sigma(\alpha_1)[u \leftarrow \sigma(\beta_2)]) \subset V(\sigma(\alpha_1))$
  $\Rightarrow V(\sigma(\alpha_1)[u \leftarrow \sigma(\beta_2)]) \subset V(\sigma(\beta_1))$

Now we check whether C(CR) are cyclic rules and C(RR) are reduction rules:

- $(\alpha,\beta) \in C(CR)$
  $\Rightarrow \exists (\alpha_1,\beta_1),(\alpha_2,\beta_2) \in C(CR) \wedge \sigma(\alpha_1)/u=\sigma(\alpha_2)$

Assume $(\alpha_1,\beta_1),(\alpha_2,\beta_2)$ are cyclic rules

(i) $\alpha = \sigma(\alpha_1)[u \leftarrow \sigma(\beta_2)] \wedge \beta=\sigma(\beta_1)$

$\sigma(\beta_1) \overset{*}{\rightarrow} \sigma(\alpha_1)$	(α_1,β_1) cyclic
$\sigma(\alpha_2) \rightarrow \sigma(\beta_2)$	(α_2,β_2)
$\Rightarrow \sigma(\alpha_1)/u \rightarrow \sigma(\beta_2)$	$\sigma(\alpha_1)/u=\sigma(\alpha_2)$
$\Rightarrow \sigma(\alpha_1) \overset{*}{\rightarrow} \sigma(\alpha_1)[u \leftarrow \sigma(\beta_2)]$	
$\Rightarrow \sigma(\beta_1) \overset{*}{\rightarrow} \sigma(\alpha_1)[u \leftarrow \sigma(\beta_2)]$	

(ii) $\alpha = \sigma(\beta_1) \wedge \beta=\sigma(\alpha_1)[u \leftarrow \sigma(\beta_2)]$

$\sigma(\beta_2) \overset{*}{\rightarrow} \sigma(\alpha_2)$	(α_2,β_2) cyclic

$\Rightarrow \sigma(\alpha_1)[u \leftarrow \sigma(\beta_2)]/u \overset{*}{\rightarrow} \sigma(\alpha_2)$

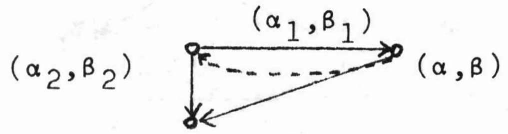
$\Rightarrow \sigma(\alpha_1)[u \leftarrow \sigma(\beta_2)] \overset{*}{\rightarrow} \sigma(\alpha_1)$ $\quad \sigma(\alpha_1)/u = \sigma(\beta_2)$

$\sigma(\alpha_1) \rightarrow \sigma(\beta_1)$ $\quad (\alpha_1, \beta_1)$

$\Rightarrow \sigma(\alpha_1)[u \leftarrow \sigma(\beta_2)] \overset{*}{\rightarrow} \sigma(\beta_1)$

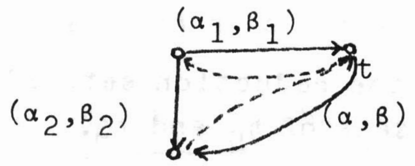
- $(\alpha,\beta) \in C(RR)$

$\Rightarrow \exists (\alpha_1, \beta_1) \in C(CR), (\alpha_2, \beta_2) \in C(CR)$:



Assume: (α_2,β_2) is a reduction rule.

If (α,β) is no reduction rule, there is a term t:



(α_2,β_2) would not be a reduction rule (induction hypothesis)

$\Rightarrow$ (α,β) is a reduction rule.

●

Proof of Theorem 3.3

$b_1,\ldots,b_n$ are cyclic rules applied at $u_1,\ldots,u_n$.

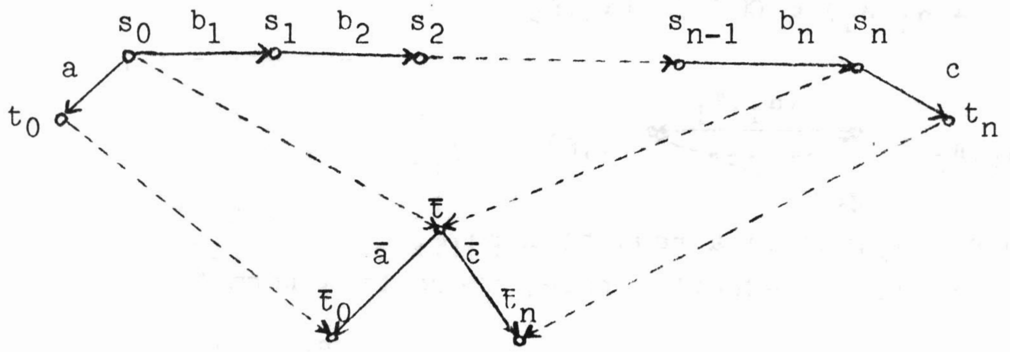
a,c are reduction rules applied at v,w.

$b_i = (\alpha_i,\beta_i)$, $s_{i-1}/u_i = \sigma_i(\alpha_i)$, $s_i = s_{i-1}[u_i \leftarrow \sigma_i(\beta_i)]$

$a = (\alpha,\beta)$, $s_0/v = \sigma(\alpha)$, $t_0 = s_0[v \leftarrow \sigma(\beta)]$

$c = (\gamma,\delta)$, $s_n/w = \bar{\sigma}(\gamma)$, $t_n = s_n[w \leftarrow \bar{\sigma}(\delta)]$

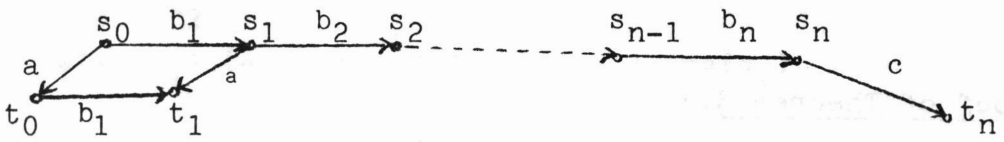
We want to prove the existence of an equivalent problem where we have to test only two reduction rules $\bar{a}$, $\bar{c}$ from $\mathfrak{a}$(RR) which are applied to a single term $\bar{t}$:



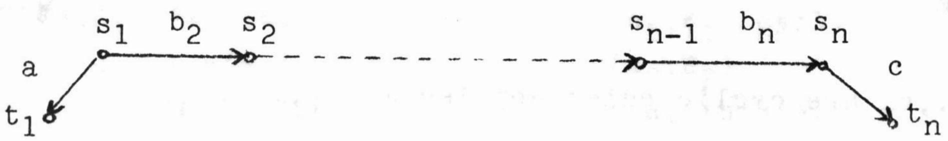
If there is a common term $\tilde{t}$ in the reduction sets of $\bar{t}_0$ and $\bar{t}_n$, $\tilde{t}$ is also in the reduction sets of t_0 and t_n.

There are seven cases:

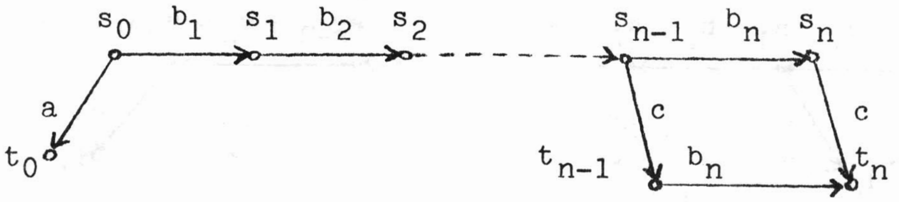
(1) b_1 and a do not overlap



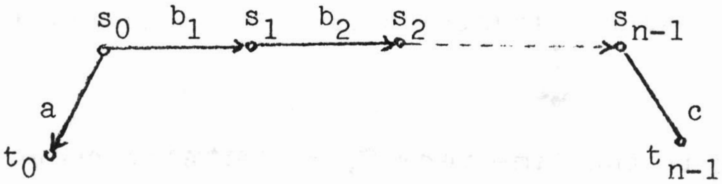
=> consider the subproblem



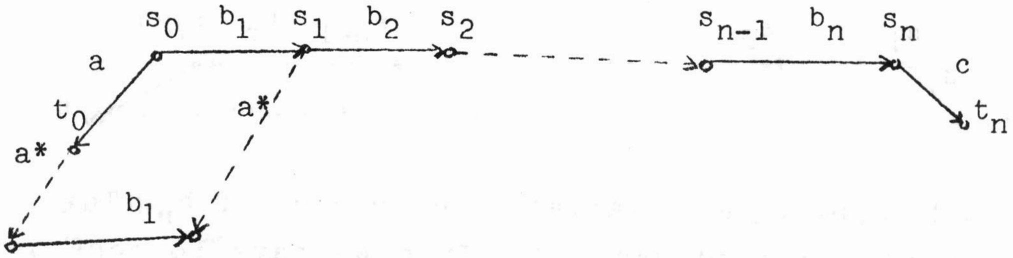
(2) b_n and c do not overlap



=> consider the subproblem

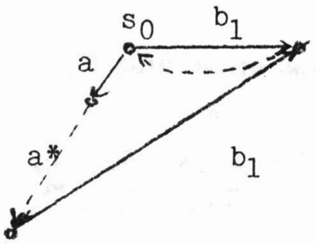


(3) a is applied at a variable occurrence of b_1

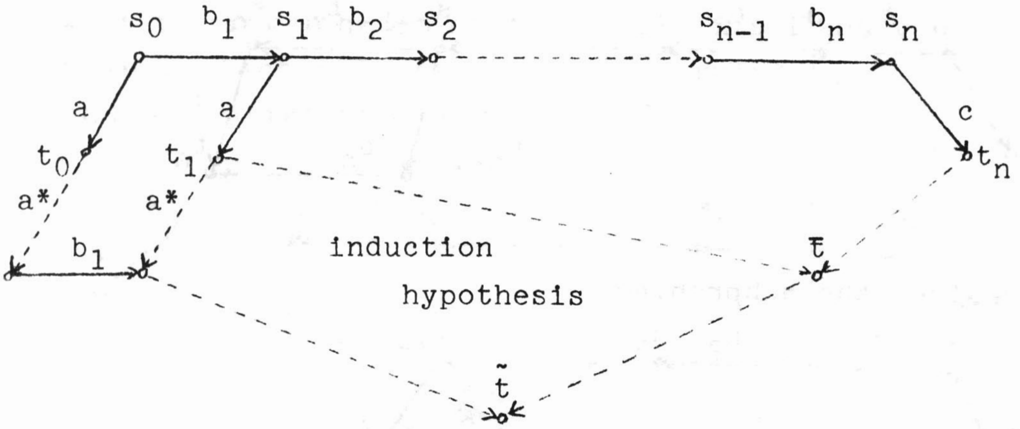


This is the classical way to reduce t_0 and s_1 to the same term.

There has to be at least one application of a at s_1, otherwise a would not be a reduction rule:

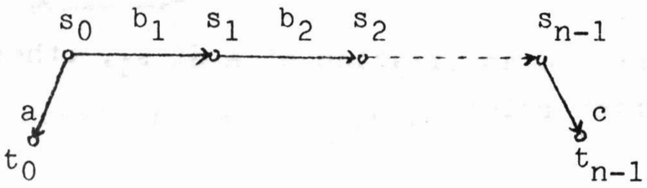


Therefore, we get:

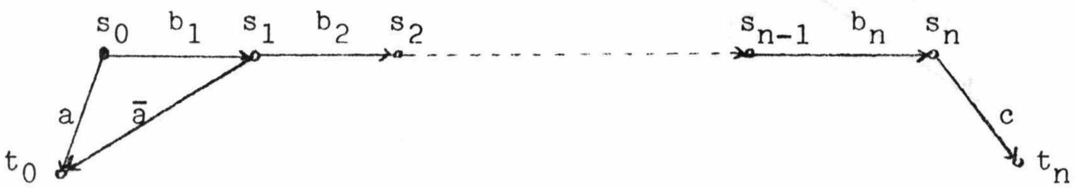


If t_1 and t_n reduce to the same term $\bar{t}$, we can also reduce t_0 and t_n to a common term. Thus, we have to consider the subproblem:

(4) c is applied at a variable occurrence of b_n. This case is equivalent to case (3). Thus we have to consider the subproblem:



(5) there is a critical overlapping between a and b_1:



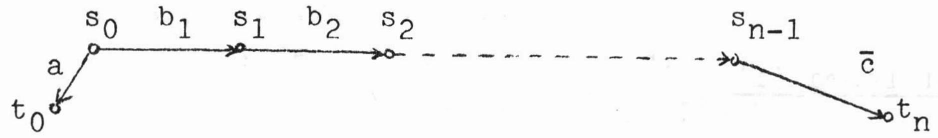
$\bar{a} \in C(RR)$

=> consider the subproblem:

(6) there is a critical overlapping between c and b_n

$b_n = (\alpha_n,\beta_n)$ => $(\beta_n,\alpha_n) \in C(CR)$
=> $\bar{c} \in C(RR)$

=> consider the subproblem:



(7) The problem cannot be reduced by any of the cases (1) - (6)

=> - b_1 is applied at a variable occurrence of a
- b_n is applied at a variable occurrence of c

=> $\sigma_1(\alpha_1)$ is a subterm of $\sigma(\alpha)$
$\sigma_n(\beta_n)$ is a subterm of $\bar{\sigma}(\gamma)$

First we want to find a "minimal" rule. We consider the rules with their substitutions:

$(\sigma(\alpha),(\sigma(\beta)),(\sigma_1(\alpha_1),\sigma_1(\beta_1)),\ldots,(\sigma_n(\alpha_n),\sigma_n(\beta_n)),(\bar{\sigma}(\gamma),\bar{\sigma}(\delta))$

A rule is "minimal" if no left or right side of other rules occures at a variable occurrence in the left side of this rule.

We define an ordering relation < for these rules as:

Definition (i)

$(\sigma_i(\alpha_i),\sigma_i(\beta_i)) > (\sigma_j(\alpha_j),\sigma_j(\beta_j))$

$\Leftrightarrow \exists u \in O(\sigma_i(\alpha_i))$: $u=u_1.u_2 \wedge \alpha_i/u_1 \in V$

$\wedge (\sigma_i(\alpha_i)/u = \sigma_j(\alpha_j) \vee \sigma_i(\alpha_i)/u = \sigma_j(\beta_j))$

We prove for >:

Lemma (ii)

- > is antisymmetric:

$(\sigma_i(\alpha_i),\sigma_i(\beta_i)) > (\sigma_j(\alpha_j),\sigma_j(\beta_j))$

$\Rightarrow (\sigma_j(\alpha_j),\sigma_j(\beta_j)) \not> (\sigma_i(\alpha_i),\sigma_i(\beta_i))$

- > is transitive:

$(\sigma_i(\alpha_i).\sigma_i(\beta_i)) > (\sigma_j(\alpha_j),\sigma_j(\beta_j))$

$\wedge (\sigma_j(\alpha_j),\sigma_j(\beta_j)) > (\sigma_k(\alpha_k),\sigma_k(\beta_k))$

$\Rightarrow (\sigma_i(\alpha_i),\sigma_i(\beta_i)) > (\sigma_k(\alpha_k),\sigma_k(\beta_k))$

Proof of Lemma (i)

Assume:

$(\sigma_i(\alpha_i),\sigma_i(\beta_i)) > (\sigma_j(\alpha_j),\sigma_j(\beta_j))$

$\wedge (\sigma_j(\alpha_j),\sigma_j(\beta_j)) > (\sigma_i(\alpha_i),\sigma_i(\beta_i))$

$u \in O(\sigma_i(\alpha_i))$, $u=u_1.u_2$, $\alpha_i/u_1 \in V$

$v \in O(\sigma_j(\alpha_j))$, $v=v_1.v_2$, $\alpha_j/v_1 \in V$

(i) $\sigma_i(\alpha_i)/u = \sigma_j(\alpha_j) \wedge \sigma_j(\alpha_j)/v = \sigma_i(\alpha_i)$

$\Rightarrow \sigma_i(\alpha_i)/u.v = \sigma_i(\alpha_i)$

$\Rightarrow u=\varepsilon \wedge v=\varepsilon$

$\Rightarrow \alpha_i, \alpha_j \in V$ (contradiction)

(ii) $\sigma_i(\alpha_i)/u = \sigma_j(\alpha_j) \wedge \sigma_j(\alpha_j)/v = \sigma_i(\beta_i)$

$\sigma_i(\alpha_i)/u.v = \sigma_i(\beta_i)$

$\sigma_i(\beta_i) \overset{*}{\to} \sigma_i(\alpha_i)$ (α_i,β_i) cyclic

$\overset{*}{\to} \sigma_i(\alpha_i)[u.v \leftarrow \sigma_i(\alpha_i)]$

=> R is not globally finite (contradiction)

(iii) $\sigma_i(\alpha_i)/u = \sigma_j(\beta_j) \wedge \sigma_j(\alpha_j)/v = \sigma_i(\alpha_i)$

$v=v_1.v_2 \wedge \alpha_j/v_1 \in V$

$\Rightarrow \exists\, \bar{v}_1 : \beta_j/\bar{v}_1 = \alpha_j/v_1$ $V(\alpha_j) = V(\beta_j)$

$\Rightarrow \sigma_j(\beta_j)/\bar{v}_1 = \sigma_j(\alpha_j)/v_1$

$\Rightarrow \sigma_j(\beta_j)/\bar{v}_1.v_2 = \sigma_j(\alpha_j)/v_1.v_2$

$\Rightarrow \sigma_i(\alpha_i)/u.\bar{v}_1.v_2 = \sigma_j(\alpha_j)/v$ $\sigma_i(\alpha_i)/u = \sigma_j(\beta_j)$

$\Rightarrow \sigma_i(\alpha_i)/u.\bar{v}_1.v_2 = \sigma_i(\alpha_i)$

$\Rightarrow u.\bar{v}_1.v_2 = \varepsilon$

$\Rightarrow \alpha_i \in V$ (contradiction)

(iv) $\sigma_i(\alpha_i)/u = \sigma_j(\beta_j) \wedge \sigma_j(\alpha_j)/v = \sigma_i(\beta_i)$

$v=v_1.v_2 \wedge \alpha_j/v_1 \in V$

$\Rightarrow \exists\, \bar{v}_1 : \beta_j/\bar{v}_1 = \alpha_j/v_1$

$\Rightarrow \sigma_j(\beta_j)/\bar{v}_1 = \sigma_j(\alpha_j)/v_1$

$\Rightarrow \sigma_j(\beta_j)/\bar{v}_1.v_2 = \sigma_j(\alpha_j)/v_1.v_2$

$\Rightarrow \sigma_i(\alpha_i)/u.\bar{v}_1.v_2 = \sigma_j(\alpha_j)/v$ $\sigma_i(\alpha_i)/u = \sigma_j(\beta_j)$

$\Rightarrow \sigma_i(\alpha_i)/u.\bar{v}_1.v_2 = \sigma_i(\beta_i)$

$\sigma_i(\beta_i) \overset{*}{\to} \sigma_i(\alpha_i)$ (α_i,β_i) cyclic

$\overset{*}{\to} \sigma_i(\alpha_i)[u.\bar{v}_1.v_2 \leftarrow \sigma_i(\alpha_i)]$

=> R is not globally finite (contradiction)

Assume:

$(\sigma_i(\alpha_i),\sigma_i(\beta_i)) > (\sigma_j(\alpha_j),\sigma_j(\beta_j))$

$\wedge\ (\sigma_j(\alpha_j),\sigma_j(\beta_j)) > (\sigma_k(\alpha_k),\sigma_k(\beta_k))$

$u \in O(\sigma_i(\alpha_i))$, $u=u_1.u_2$, $\alpha_i/u_1 \in V$

$v \in O(\sigma_j(\alpha_j))$, $v=v_1.v_2$, $\alpha_j/v_1 \in V$

(i) $\sigma_i(\alpha_i)/u = \sigma_j(\alpha_j) \wedge \sigma_j(\alpha_j)/v = \sigma_k(\alpha_k)$

$\Rightarrow \sigma_i(\alpha_i)/u.v = \sigma_k(\alpha_k)$

$\Rightarrow (\sigma_i(\alpha_i),\sigma_i(\beta_i)) > (\sigma_k(\alpha_k),\sigma_k(\beta_k))$

(ii) $\sigma_i(\alpha_i)/u = \sigma_j(\alpha_j) \wedge \sigma_j(\alpha_j)/v = \sigma_k(\beta_k)$

$\Rightarrow \sigma_i(\alpha_i)/u.v = \sigma_k(\beta_k)$

$\Rightarrow (\sigma_i(\alpha_i),\sigma_i(\beta_i)) > (\sigma_k(\alpha_k),\sigma_k(\beta_k))$

(iii) $\sigma_i(\alpha_i)/u = \sigma_j(\beta_j) \wedge \sigma_j(\alpha_j)/v = \sigma_k(\alpha_k)$

$v=v_1.v_2 \wedge \alpha_j/v_1 \in V$

$\Rightarrow \exists\, \bar{v}_1: \beta_j/\bar{v}_1 = \alpha_j/v_1$ $\quad V(\alpha_j) = V(\beta_j)$

$\Rightarrow \sigma_j(\beta_j)/\bar{v}_1 = \sigma_j(\alpha_j)/v_1$

$\Rightarrow \sigma_j(\beta_j)/\bar{v}_1.v_2 = \sigma_j(\alpha_j)/v_1.v_2$

$\Rightarrow \sigma_i(\alpha_i)/u.\bar{v}_1.v_2 = \sigma_k(\alpha_k)$

$\Rightarrow (\sigma_i(\alpha_i),\sigma_i(\beta_i)) > (\sigma_k(\alpha_k),\sigma_k(\beta_k))$

(iv) $\sigma_i(\alpha_i)/u = \sigma_j(\beta_j) \wedge \sigma_j(\alpha_j)/v = \sigma_k(\beta_k)$

$v=v_1.v_2 \wedge \alpha_j/v_1 \in V$

$\Rightarrow \exists\, \bar{v}_1: \beta_j/\bar{v}_1 = \alpha_j/v_1$ $\quad V(\alpha_j) = V(\beta_j)$

$\Rightarrow \sigma_j(\beta_j)/\bar{v}_1 = \sigma_j(\alpha_j)/v_1$

$\Rightarrow \sigma_j(\beta_j)/\bar{v}_1.v_2 = \sigma_j(\alpha_j)/v_1.v_2$

$\Rightarrow \sigma_i(\alpha_i)/u.\bar{v}_1.v_2 = \sigma_k(\beta_k)$

$\Rightarrow (\sigma_i(\alpha_i),\sigma_i(\beta_i)) > (\sigma_k(\alpha_k),\sigma_k(\beta_k))$

•

Thus we can find a minimal rule $(\sigma_k(\alpha_k),\sigma_k(\beta_k))$ by this ordering.

Assume:

- $\exists\, w: w \in \sigma_k(\alpha_k) \wedge w=w_1.w_2 \wedge \alpha_k/w_1 \in V \wedge \sigma_k(\alpha_k)/w = \sigma(\alpha)$

  $\exists\, \bar{w}: \sigma(\alpha)/\bar{w} = \sigma_1(\alpha_1)$ $\quad b_1$ is applied at a variable occurrence of a

  $\Rightarrow \sigma_k(\alpha_k)/w.\bar{w} = \sigma_1(\alpha_1)$

  $\Rightarrow (\sigma_k(\alpha_k),\sigma_k(\beta_k))$ is not a minimal rule

- $\exists\, w: w \in \sigma_k(\alpha_k) \wedge w=w_1.w_2 \wedge \alpha_k/w_1 \in V \wedge \sigma_k(\alpha_k)/w = \bar{\sigma}(\gamma)$

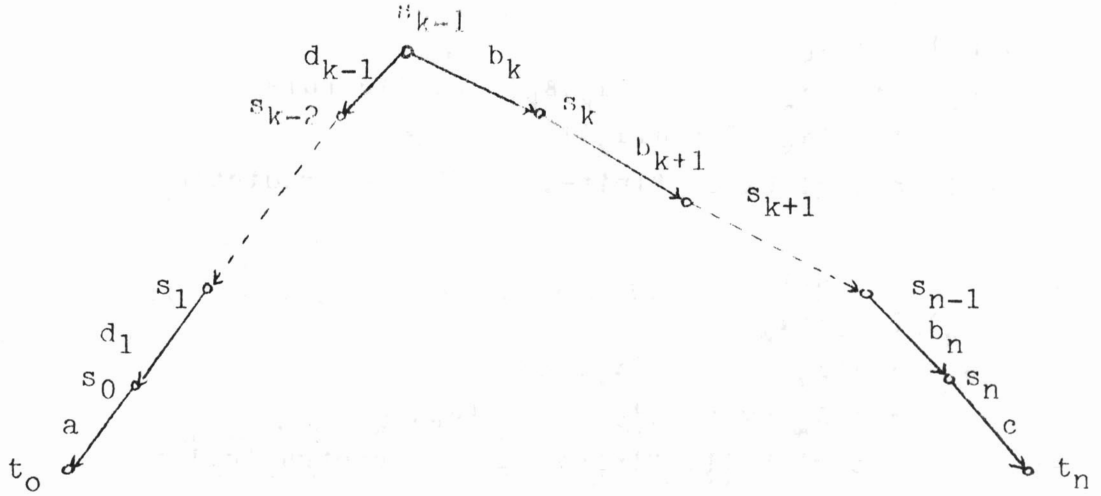
  $\exists\, \bar{w}: \bar{\sigma}(\gamma)/\bar{w} = \sigma_n(\beta_n)$ $\quad b_n$ is applied at avariable occurrence of c

  $\Rightarrow \sigma_k(\alpha_k)/w.\bar{w} = \sigma_n(\beta_n)$

$\Rightarrow (\sigma_k(\alpha_k), \sigma_k(\beta_k))$ is not a minimal rule

Therefore, any left or right side of other rules does not occur as a subterm at a variable position of rule $(\sigma_k(\alpha_k), \sigma_k(\beta_k))$.

Consider now:



with $d_i = (\beta_i, \alpha_i) \in C(CR)$ $i=1,\ldots,k-1$

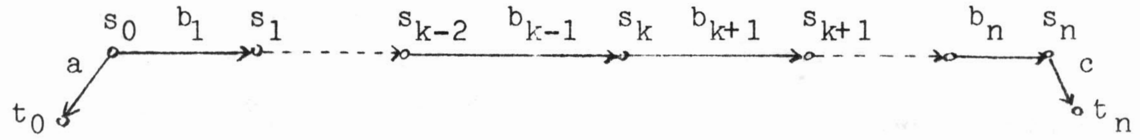
Now we replace each subterm $\sigma_k(\alpha_k)$ in s_k by $\sigma_k(\beta_k)$ and continue until no subterm $\sigma_k(\alpha_k)$ is left.
We will prove the termination of this procedure:

Assume:
this procedure does not terminate
$\Rightarrow O(s_k)$ is finite, thus there are overlappings between the application of this rule.
u first application of rule b_k
v second application of rule b_k

u=v

$\Rightarrow \sigma(\beta_k) = \sigma(\alpha_k)$

this rule does not change the term, so we can skip it.
Consider the subproblem:



u>v $\Rightarrow$ $u = v.\bar{u}$

$\Rightarrow \sigma(\beta_k) = \sigma(\alpha_k)/u$

$\sigma(\beta_k) \overset{*}{\rightarrow} \sigma(\alpha_k)$ (α_k, β_k) cyclic rule

$\overset{*}{\rightarrow} \sigma(\alpha_k)[\bar{u} \leftarrow \sigma(\alpha_k)]$

$\Rightarrow$ R is not globally finite. (contradiction)

u<v $\Rightarrow$ $v = u.\bar{v}$

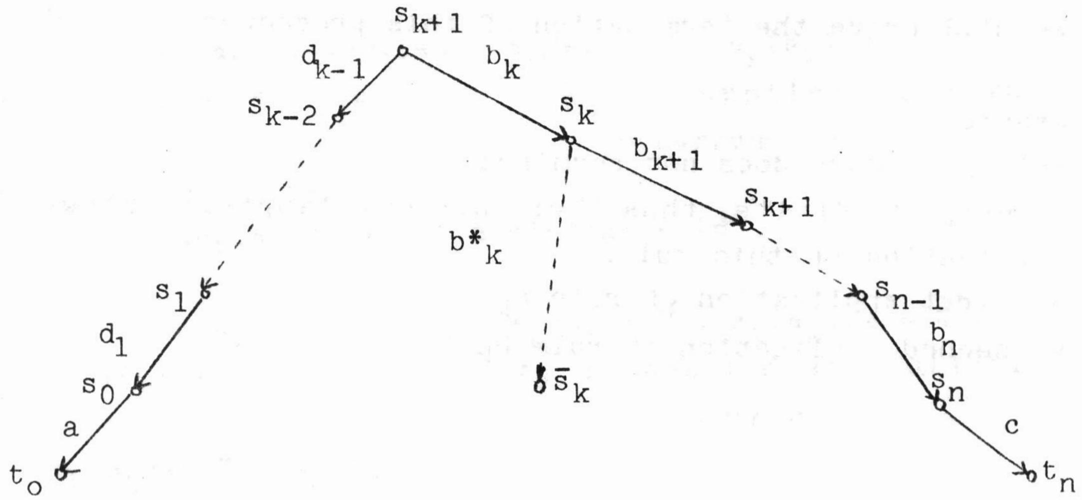
$\Rightarrow \sigma(\beta_k)/\bar{v} = \sigma(\alpha_k)$

$\sigma(\alpha_k) \rightarrow \sigma(\beta_k)$ (α_k, β_k)

$\rightarrow \sigma(\beta_k)[\bar{v} \leftarrow \sigma(\beta_k)]$ (α_k, β_k)

$\Rightarrow$ R is not globally finite. (contradiction)

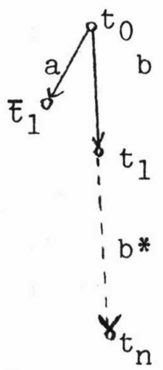
Thus, the occurrences where we replace $\sigma_k(\alpha_k)$ by $\sigma_k(\beta_k)$ do not overlap.



Before we continue with the proof, we show Lemma (iii) first:

<u>Lemma (iii)</u>

Let $a = (\alpha,\beta)$ and $b = (\gamma,\delta)$ be two rules, t a term and $\sigma,\bar{\sigma}$ two substitutions. a is applied at occurrence v and b is applied at $u_1,\ldots,u_n$:



$t_0/v = \sigma(\alpha) \wedge \bar{t}_1 = t_0[v \leftarrow \sigma(\beta)]$

$t_i/u_{i+1} = \bar{\sigma}(\gamma) \wedge t_{i+1} = t_i[u_{i+1} \leftarrow \bar{\sigma}(\delta)]$ $i=0,1,\ldots,n$

$i \neq j \Rightarrow u_i \not\leq u_j \wedge u_i \not\geq u_j$ no overlappings

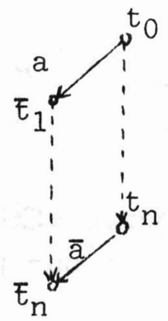
$\sigma(\alpha)$ is not a subterm at a variable occurrence of γ:

$\forall u: (u \in O(\gamma) \Rightarrow \gamma/u \in V) \Rightarrow \bar{\sigma}(\gamma)/u \neq \sigma(\alpha)$

There is no subterm $\bar{\sigma}(\gamma)$ in t_n:

$\forall u: t_n/u \neq \bar{\sigma}(\gamma)$

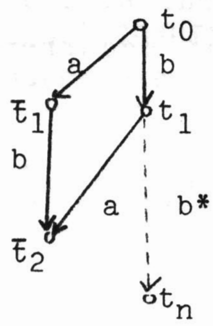
$\Rightarrow$



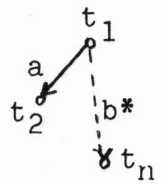
$\bar{a}$ is constructed by critical pairs between a and b.

Proof of Lemma (iii)

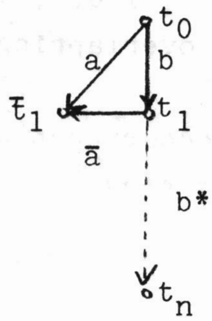
- v and u_1 do not overlap



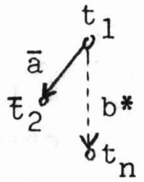
=> consider:



- there is a critical overlapping between a,b



=> consider:

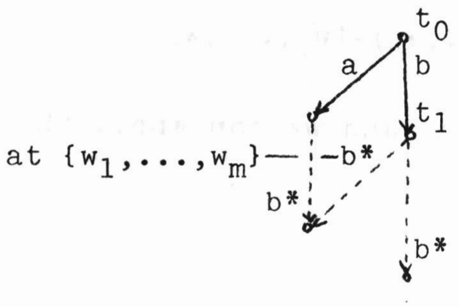


- a is applied at a variable occurrence of b

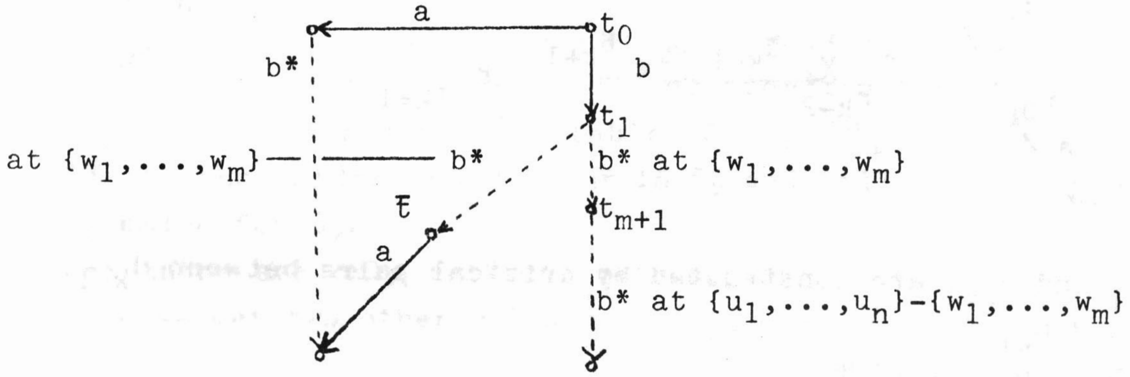
  This case does not occure because of our premise:

$$\forall w: (u \in O(\gamma) \Rightarrow \gamma/u \in V) \Rightarrow \bar{\sigma}(\gamma)/w \neq \sigma(\alpha)$$

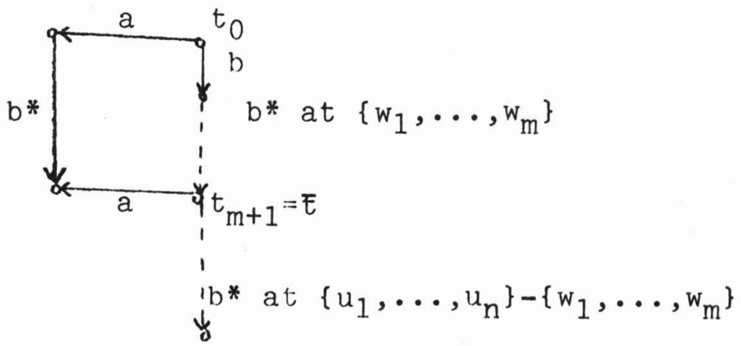
- b is applied at a variable occurrence of a



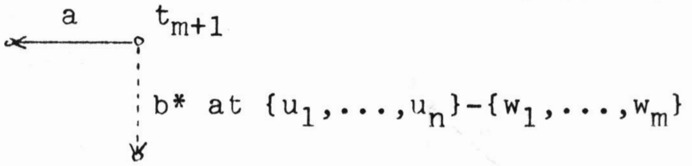
$\{w_1,\ldots,w_m\}$ is a subset of $\{u_1,\ldots,u_n\}$, because every subterm $\sigma(\gamma)$ in t_0 is replaced by $\sigma(\delta)$.
There is no overlapping between the w_i, thus we can apply b in an arbitrary sequence:



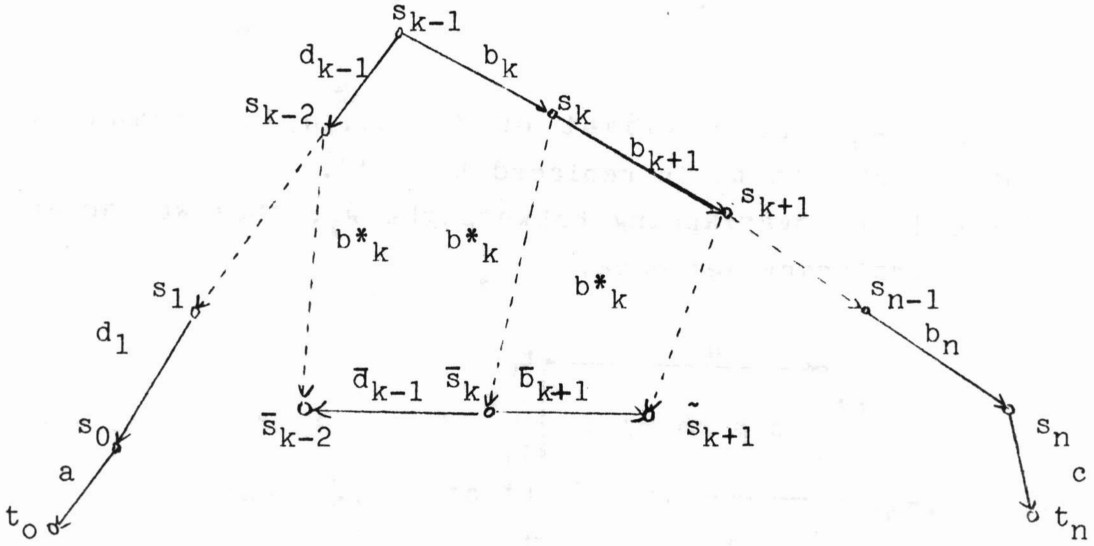
=> t_{m+1} and $\bar{t}$ are the same terms:



=> consider the subproblem:



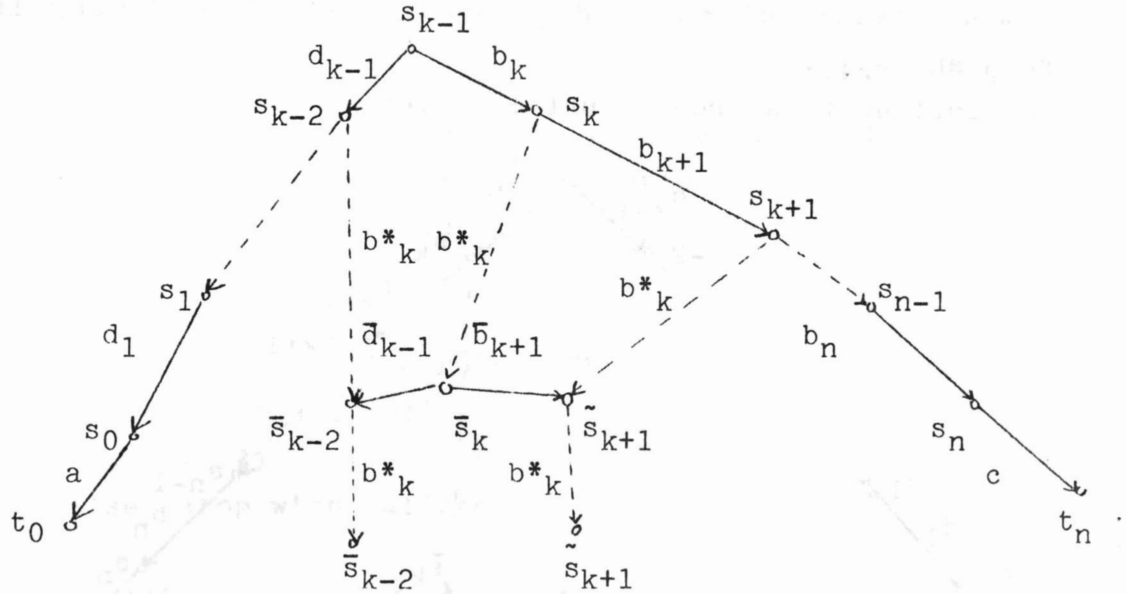
This completes the proof of Lemma (iii) and we can apply the result to our main proof:



$\bar{d}_{k-1}$ and $\bar{b}_{k+1}$ are constructed by critical pairs between d_{k-1}, b_k and b_{k+1}

$\Rightarrow \bar{d}_{k-1}, \bar{b}_{k+1} \in C(CR)$

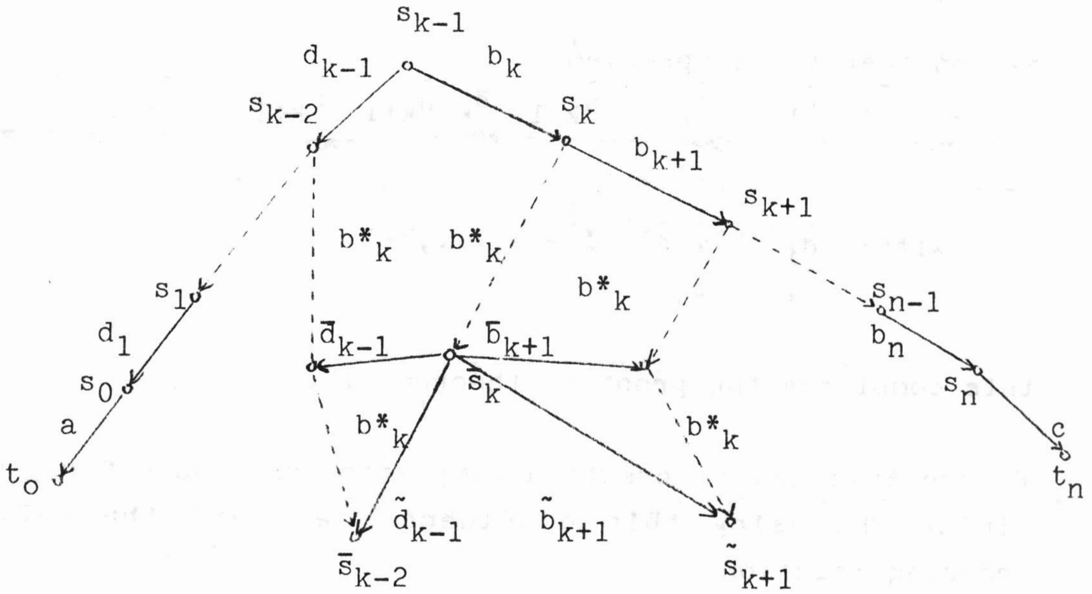
We replace every subterm in $\sigma_k(\alpha_k)$ in $\bar{s}_{k-2}$ and $\bar{s}_{k+1}$ by $\sigma_k(\beta_k)$:



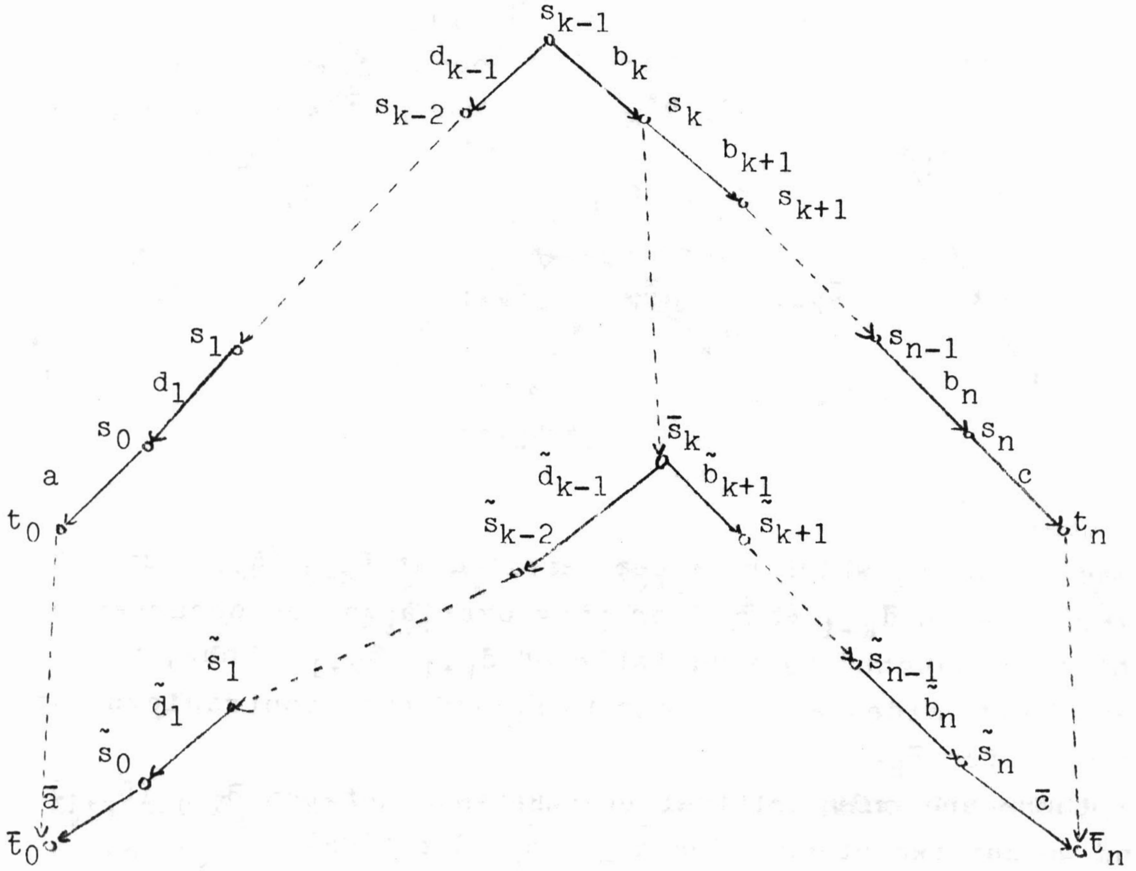
Assume:

Some of the b_k which have been applied at $\bar{s}_{k-2}$ ($\bar{s}_{k+1}$) do not overlap with $\bar{d}_{k-1}$ ($\bar{b}_{k+1}$) or they overlap at an occurrence which is matched by a variable of $\bar{d}_{k+1}$ ($\bar{b}_{k+1}$). Thus, their left hand sides would occur in $\bar{s}_k$ and this contradicts our premise for $\bar{s}_k$.

=> there are only critical overlappings between $\bar{d}_{k-1}$ ($\bar{b}_{k+1}$) and we get two other rules $\tilde{d}_{k-1}$ ($\tilde{b}_{k+1}$) ε C(CR):

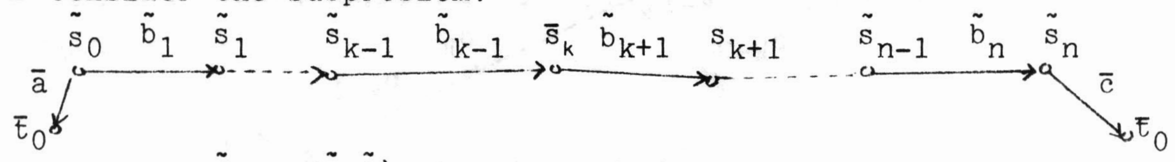


Now we create rules from d_{k-2} and b_{k+2} which can be applied at $\tilde{s}_{k-2}$ and $\tilde{s}_{k+1}$.
We continue this process until we get:



with: $\bar{a},\bar{b} \in C(RR)$

=> consider the subproblem:



with: $\tilde{d}_i = (\tilde{\gamma},\tilde{\delta}) \quad i = 1,\ldots,k-1$
$\tilde{b}_i = (\tilde{\delta},\tilde{\gamma})$

this completes the proof of theorem 3.3.

We are able now to create a completion procedure for globally finite TRS using this confluence test and the reflexive ordering relations:

Completion Algorithm

(1) $j=0$
transform the equations into a globally finite TRS $R(0)$,
no direction found for some equations?
true: stop with failure
false: → (2)

(2) $i=0$, $CC(0)$:= cyclic rules, $CR(0)$:= reduction rules
$R(j) = CR(0) \cup CC(0)$?
true: → (3)
false: stop with failure

(3) $i = i+1$

$CR(i) :=$

$$\{(\sigma(\alpha_1)[u \leftarrow \sigma(\beta_2)], \sigma(\alpha_1)) \mid (\alpha_1,\beta_1) \in CC(i-1) \wedge (\alpha_2,\beta_2) \in CR(i-1) \wedge \sigma(\alpha_1)/u = \sigma(\alpha_2)\}$$

$$\cup \{(\sigma(\beta_2), \sigma(\alpha_2)[u \leftarrow \sigma(\beta_1)]) \mid (\alpha_1,\beta_1) \in CC(i-1) \wedge (\alpha_2,\beta_2) \in CR(i-1) \wedge \sigma(\alpha_1) = \sigma(\alpha_2)/u\}$$

$CC(i) :=$

$$\{(\sigma(\alpha_1)[u \leftarrow \sigma(\beta_2)], \sigma(\beta_1)) \mid (\alpha_1,\beta_1),(\alpha_2,\beta_2) \in CC(i-1) \wedge \sigma(\alpha_1)/u = \sigma(\alpha_2)\}$$

$$\cup \{(\sigma(\beta_1), \sigma(\alpha_1)[u \leftarrow \sigma(\beta_2)]) \mid (\alpha_1,\beta_1),(\alpha_2,\beta_2) \in CC(i-1) \wedge \sigma(\alpha_1)/u = \sigma(\alpha_2)\}$$

$C(i) :=$

$$\{(\sigma(\alpha_1)[u \leftarrow \sigma(\beta_2)], \sigma(\beta_1)) \mid (\alpha_1,\beta_1),(\alpha_2,\beta_2) \in CR(i-1) \wedge \sigma(\alpha_1)/u = \sigma(\alpha_2)\}$$

with:
$(u \in O(\alpha_1) \wedge \alpha_1/u \notin V) \vee (u \in O(\alpha_2) \wedge \alpha_2/u \notin V)$
for $CR(i)$, $CC(i)$, $C(i)$

Is there a critical pair (α,β) in $C(i)$ which cannot be reduced to a common term?

```
true:  j = j+1
  R(j-1) u {(α,β)} globally finite?
  true:  R(j) := R(j-1) u {(α,β)} → (2)
  false: R(j-1) u {(β,α)} globally finite?
         true:  R(j) := R(j-1) u {(β,α)} → (2)
         false: stop with failure
false: → (4)
```

(4) $CR(i)=CR(i-1) \wedge CC(i) = CC(i-1)$?

true: stop R complete

false: → (3)

This completion procedure will find every critical pair which has to be added as a new rule to the TRS. If there are no cyclic rules, the set of critical pairs is the same as in the classical procedure. Therefore, it stops if R is complete. But if there are cyclic rules in R the procedure does not stop for every complete TRS. For example if we want to complete the abelian group axioms the associativity axiom produces an infinite set of cyclic critical pairs (C(CR)). In fact the confluence of globally finite TRS might be undecidable. We are currently implementing this procedure on a Symbolics 3600 Lispmachine, for testing some examples and finding stop criteria for this procedure.

References

[HUE 77] Huet, G.:
Confluent reductions:
Abstract properties and applications to term rewriting systems.
18th IEEE Symposium on Foundations of Computer Science 1977, p. 30-45.

[HUE 80] Huet, G., Oppen, D.:
Equations and rewrite rules:
A survey
Technical Report CSL 111, SRI International 1980

[JOU 83] Jouannaud, J.P.:
Confluent and coherent equational TRS, application to proofs in abstract data types.
Centre de Recherche en Informatique de Nancy, 1983.

[KB 70] Knuth, D., Bendix, P.:
Simple word problems in universal algebras.
Computational probelms in abstract algebra.
Ed. Leech I., Pergamon Press

11/22/1983

SEKI Memos

The following memos are available free of charge from

Mrs. Dorothea Kilgore
Universität Kaiserslautern
Fachbereich Informatik
Postfach 3049
D-6750 Kaiserslautern
West Germany

MEMO SEKI-81-01 U. Bartels, W. Olthoff and P. Raulefs: APE: An Expert System for Automatic Programming from Abstract Specifications of Data Types and Algorithms.

MEMO SEKI-81-03 Peter Raulefs: Expert Systems: State of the Art and Future Prospects.

MEMO SEKI-81-04 Christoph Beierle: Programmsynthese aus Beispielsfolgen.

MEMO SEKI-81-05 Erich Rome: Implementierungen Abstrakter Datentypen in terminaler Algebrasemantik.

MEMO SEKI-81-06 Christoph Beierle: Synthesizing Minimal Programs from Traces of Observable Behaviour.

MEMO SEKI-81-07 Dieter Wybranietz: Ein verteiltes Betriebssystem für CSSA.

MEMO SEKI-81-08 Ulrich Bartels and Walter Olthoff: APE - Benutzerbeschreibung.

MEMO SEKI-82-01 Hans Voß: Programming in a Distributed Environment: A Collection of CSSA Examples.

MEMO SEKI-82-02 Hartmut Grieneisen: Eine algebraische Spezifikation des Software-Produkts INTAKT.

MEMO SEKI-82-03 Christian Beilken, Friedemann Mattern and Michael Spenke: Entwurf und Implementierung von CSSA - Beschreibung der Sprache, des Compilers und des Mehrrechnersimulationssystems.
Printed in 6 volumes, which can be ordered individually:
Vol-A: Konzepte
Vol-B: CSSA-Sprachbeschreibung
Vol-C: CSSA-Systembenutzung
Vol-D: CSSA-Programmbeispiele

Vol-E1: Programmdokumentation Teil I
Vol-E2: Programmdokumentation Teil II

MEMO SEKI-83-01	Wilfried Schrupp and Johann Tamme: Spezifikation und abstrakte Implementierung des Aufbereitungsteils von INTAKT.
MEMO SEKI-83-02	Frank Puppe and Bernd Puppe: Overview on MED1: A Heuristic Diagnostic System with an Efficient Control-Structure.
MEMO SEKI-83-03	Elisabeth Hülsmann: LISP-SP : A portable INTERLISP Subset Interpreter for Mini-Computers.
MEMO SEKI-83-04	FrankPuppe: MED1 - Ein heuristisches Diagnosesystem mit effizienter Kontrollstruktur.
MEMO SEKI-83-05	Horst Peter Borrmann: MODIS - Ein Expertensystem zur Erstellung vonReparaturdiagnosen für den Ottomotor und seineAggregate.
MEMO SEKI-83-06	Harold Boley: From Pattern-Directed to Adapter-Driven Computation via Function-Applying Matching.
MEMO SEKI-83-07	Christoph Beierle and Angi Voß: Canonical Term Functors and Parameterization-by-use for for the Specification of Abstract Data Types.
MEMO SEKI-83-08	Christoph Beierle and Angi Voß: Parameterization-by-use for hierarchically structured objects.
MEMO SEKI-83-09	Christoph Beierle, Michael Gerlach and Angi Voß: Parameterization without parameters in : The History of a Hierarchy of Specifications.
MEMO SEKI-83-10	Ulrike Petersen: Elimination von Rekursionen.
MEMO SEKI-83-11	Gerd Krützer: An Approach to Parameterized Continuous Data Types.
MEMO SEKI-83-12	Richard Göbel: A Completion Procedure for Globally Finite Term Rewriting Systems.
MEMO SEKI-83-13	Michael Gerlach: A Second-Order Matching Procedure for the Practical Use in a Program Transformation System.