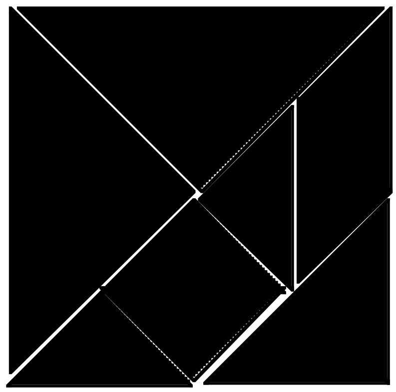
Ten Topics in Term Rewriting

Jan Willem Klop

Logic Colloquium
August 1999
Utrecht



1. knots, braids and abstract reduction systems
2. decreasing diagrams
3. infinite diagrams
4. first-order term rewriting systems
5. critical pair completion
6. orthogonal term rewriting systems
7. transfinite rewriting
8. recursive path orders with stars
9. higher-order rewriting
10. references

Knots, Braids and Abstract Reduction Systems

In our first topic we start with introducing Abstract Reduction Systems or Abstract Rewrite Systems, consisting of just a set of objects together with one or more reduction or rewrite relations on them, to be perceived as 'transformation' or 'computation' relations. Before giving several formal definitions of the relevant notions, we consider two examples from topology, knots and braids. In this way already several rewrite notions will be encountered.

1.1. Knots

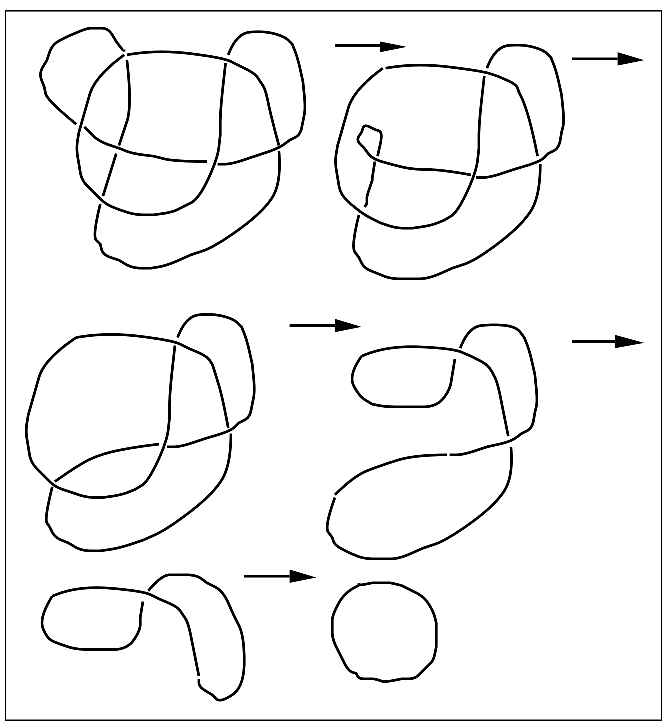


Figure 1.1. *un-knotting*

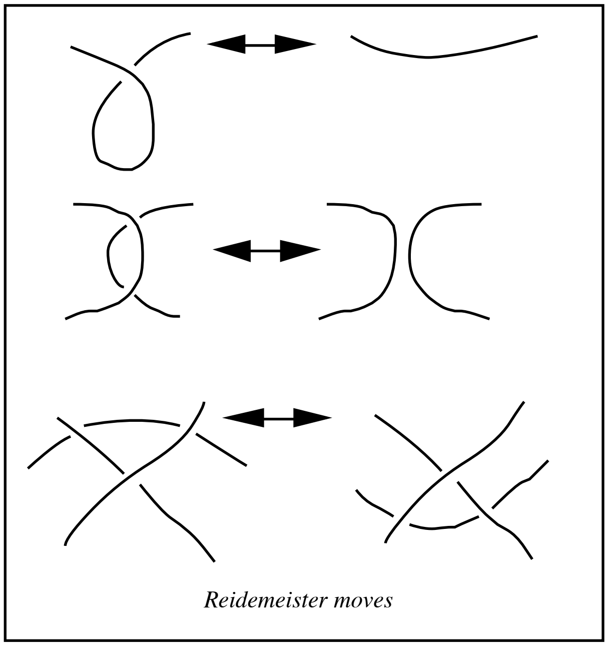


Figure 1.2

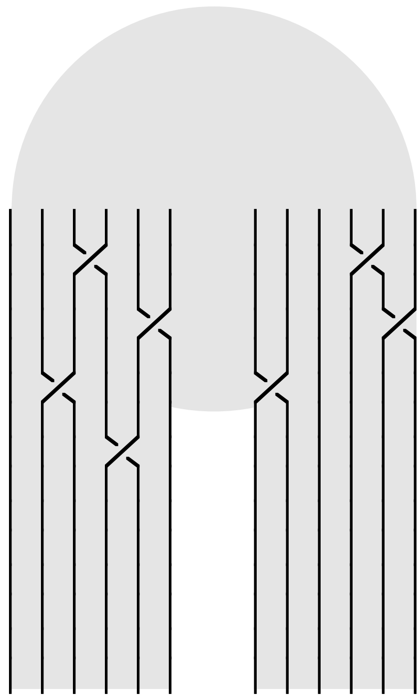
Discussion. Important is the idea that transformations on the objects of 'interest' (representations of knots) are made 'locally', by changing some small part of the object. In this case the transformation or 'rewrite' or 'reduction' relation as given by the Reidemeister moves is symmetric; there is no direction of preference.

1.2. Braids

The next example, braids, has much to do with the previous one but will also present some new concepts. It is about the 'confluence'problem for braids, as described on p.132-134 in Schmidt and Ströhlein [91], in the following anthropomorphic terms. The study of braids goes back to Artin [26, 47, 47a].

1.2.1. The semi-group of braids

A girl has two braids consisting of, say, 6 strings (see Figure 1.3). The father starts braiding the left braid, the mother of the girl starts braiding the right braid. After some initial 'twists' as indicated in the figure, they notice that they do it in a different way. But they want to arrive, eventually, at two identical braids. *Question: can they go on and still arrive at identical braids?*



Girl with two braids

Fig. 1.3

Note that braids are subject to a topological equivalence, which will be explained now. First we need a notation for braids. Consider Figure 1.4. The openings between the strings are numbered 1,2,3,... . A twist or crossing in which the upper string moves over the lower is denoted 'positively', just as the corresponding opening: if this is i, the positive twist twist at this position is also denoted by i. Otherwise we have a 'negative' crossing, denoted by i^{-1} if it is in the i-th opening. Thus the braid in Figure 1.4a is $1.2^{-1}.1.2^{-1}.1.2$.

Now we restrict attention to *positive crossings only*. E.g. in Figure 1.4b we have the braid 1.2.4.1.3.1.4.3 . The restriction means that we work in the semi-group generated by 1,2,3,4 (if there are five strings).

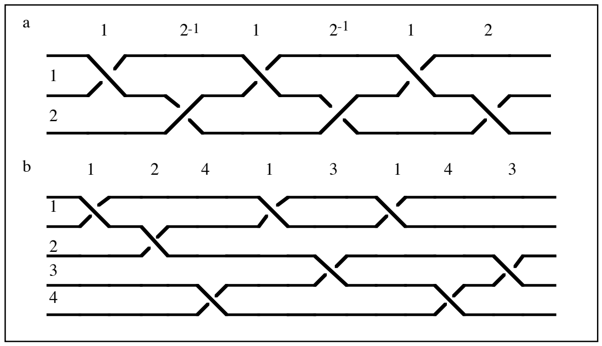


Figure 1.5

Not all these braids are really different. See Figure 1.5a. The braid 1.3 is 'the same', topologically viewed, as 3.1, just by shifting the crossings in the other order. Also 1.4 is equivalent with 4.1. We will write 1.3 = 3.1, and 1.4 = 4.1. In general we have:

$$i.j = j.i \quad \text{if } |i-j| \geq 2$$

For consecutive openings like 1 and 2, respective crossings do not commute. But it is not hard to see that starting with 1.2 and 2.1, we can make them (topologically) equal by continuing 1.2 with 1 and 2.1 with 2. So 1.2.1 = 2.1.2. See Figure 1.5b. Note that 1.2.1 and 2.1.2 are indeed topologically the same; an experiment with actual strings of wire will demonstrate this. In general we have for all i:

$$i.(i+1).i = (i+1).i.(i+1)$$

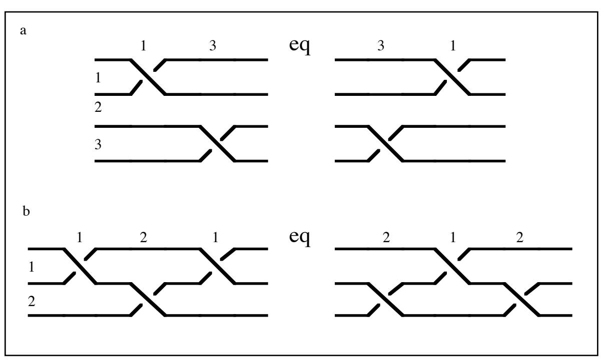
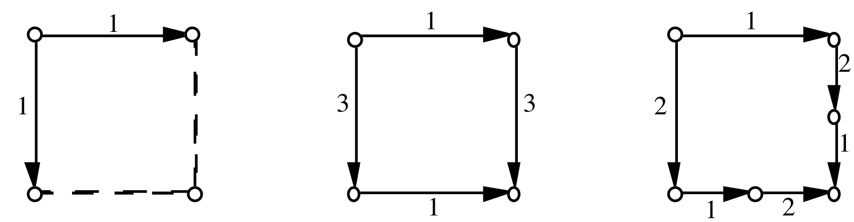


Figure 1.5

The equations above completely define the topological equivalence considered (see Artin [47]). The confluence problem is now: given two elements u, v of this braid semi-group, can we always find elements x, y such that ux = vy? The problem can be approached by means of an abstract rewriting analysis using the "elementary diagrams" as in Figure 1.6. (Only some of the generators 1,2,3,... are mentioned in the figure.) These diagrams are just another way of phrasing the equations above; e.g. the second e.d. states that 1.3 = 3.1 the last one states that 1.2.1 = 2.1.2. The first e.d is trivial, it states that 1 = 1; but such trivial e.d's still are useful as will be apparent in the next example. The question is now whether 'tiling' with these diagrams always succeeds in a confluent reduction diagram.



elementary diagrams for confluence problem in braid semi-group

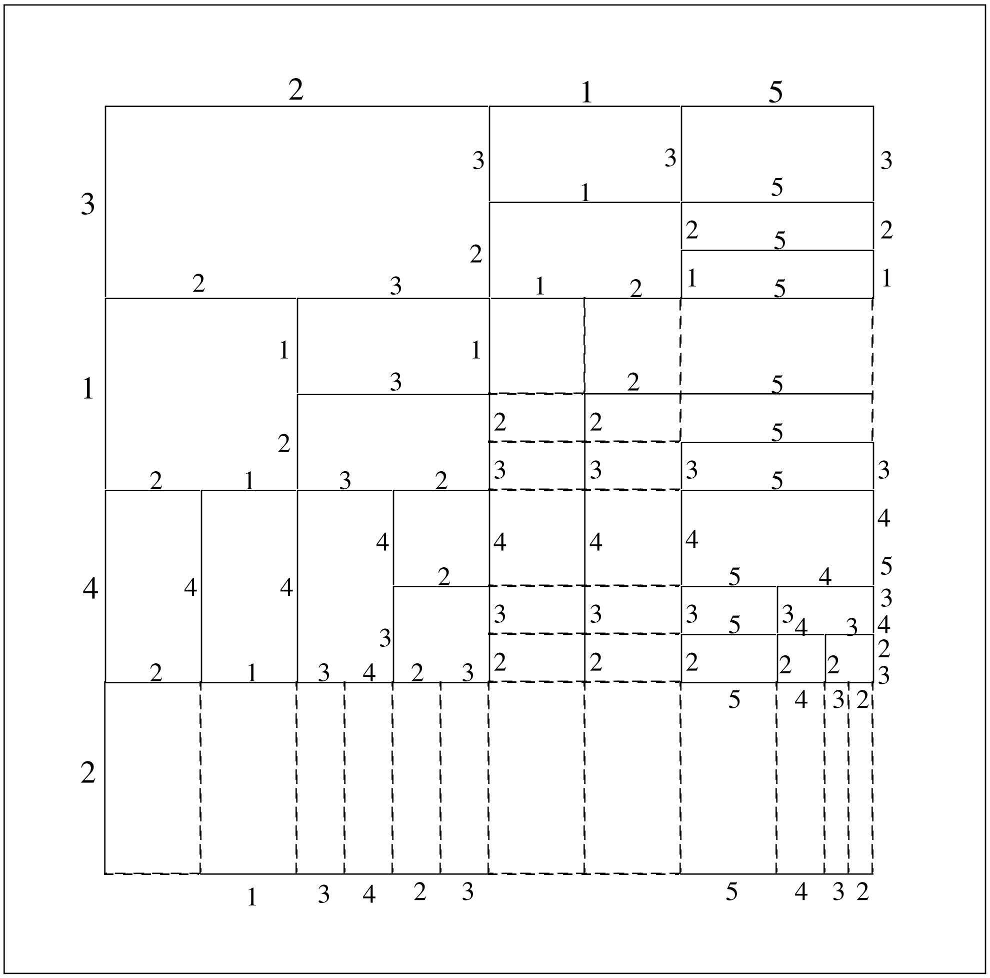
Figure 1.6

1.2.1.1. EXAMPLE. We complete in a diagram the braidings started by the father and the mother as in Figure 1.3: the braids there are 3142 and 215 (counting the openings from right to left). See Figure 1.7.

As it turns out, we are lucky in this example; the tiling procedure terminates succesfully in a completed "reduction diagram", whose lower and right sides yield the (or rather, an) answer to our question.

1.2.1.2. THEOREM (Garside [69]). *Braids are confluent. That is: For all* u, v *of the braid semi-group, there exist elements* x, y *such that* ux = vy.

1.2.1.3. REMARK.(i) Actually, braids are confluent in a canonical way, namely by the tiling procedure as demonstrated in the example. This was proved recently by Melliès and van Oostrom.
(ii) Note that 'empty steps', as introduced in the trivial e.d.'s, propagate 'through' steps in the obvious way; see the reduction diagram in Figure 1.7.



2 1 5
3 3 3
3 1 5
2 2 5 2
2 3 1 2 1 5 1
1 1
1 3 2 5
2 2 2 5
2 3 3 3 3 5 3
2 1 3 2
4
4 4 4 4 4
4 4 4 2 5 4 5
3 3 3 5 3 4 3 3 4
2 1 3 4 2 3 2 2 2 2 2 2 3
5 4 3 2
2
1 3 4 2 3 5 4 3 2

Figure 1.7

1.3. Abstract Rewrite Systems

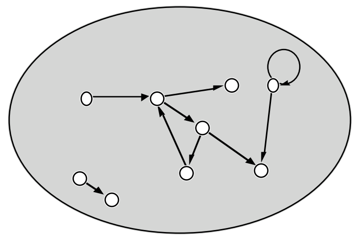


Figure 1.8

1.3.1. DEFINITION.

(i) An *Abstract Reduction System* (ARS) is a structure $\mathcal{A} = \langle A, (\to_\alpha)_{\alpha \in I} \rangle$ consisting of a set A and a sequence of binary relations $\to_\alpha$ on A, also called reduction or rewrite relations. Sometimes we will refer to $\to_\alpha$ as α. In the case of just one reduction relation (see Figure 1.8), we also use $\to$ without more. (An ARS with just one reduction relation is called 'replacement system' in Staples [75], and a 'reduction system' in Jantzen [88].) If for a,b $\in$ A we have (a,b) $\in \to_\alpha$, we write a $\to_\alpha$ b and call b a one-step (α-)reduct of a.

(ii) The transitive reflexive closure of $\to_\alpha$ is written as $\twoheadrightarrow_\alpha$. (More customary is the notation $\to_\alpha{}^*$, but we prefer the double arrow notation as we find it more convenient in diagrams.) So a $\twoheadrightarrow_\alpha$ b if there is a possibly empty, finite sequence of 'reduction steps' $a \equiv a_0 \to_\alpha a_1 \to_\alpha \ldots \to_\alpha a_n \equiv b$. Here $\equiv$ denotes identity of elements of A. The element b is called an (α-)*reduct* of a. The equivalence relation generated by $\to_\alpha$ is $=_\alpha$, also called the *convertibility* relation generated by $\to_\alpha$. The reflexive closure of $\to_\alpha$ is $\to_{\alpha\equiv}$. The transitive closure of $\to_\alpha$ is $\to_\alpha{}^+$. The converse relation of $\to_\alpha$ is $\leftarrow_\alpha$ or $\to_{\alpha^{-1}}$. The union $\to_\alpha \cup \to_\beta$ is denoted by $\to_{\alpha\beta}$. The composition $\to_\alpha \circ \to_\beta$ is defined by: a $\to_\alpha \circ \to_\beta$ b if a $\to_\alpha$ c $\to_\beta$ b for some c $\in$ A.

(iii) If α, β are reduction relations on A, we say that α *commutes weakly* with β if the following diagram (see Figure 1.1a) holds, i.e. if $\forall$a,b,c$\in$A $\exists$d$\in$A (c $\leftarrow_\beta$ a $\to_\alpha$ b $\Rightarrow$ c $\twoheadrightarrow_\alpha$ d $\twoheadleftarrow_\beta$ b), or in a shorter notation: $\leftarrow_\beta \circ \to_\alpha$ ™ $\twoheadrightarrow_\alpha \circ \twoheadleftarrow_\beta$.

Further, α *commutes* with β if $\twoheadrightarrow_\alpha$ and $\twoheadrightarrow_\beta$ commute weakly. (This terminology differs from that of Bachmair & Dershowitz [86], where α commutes with β if $\alpha^{-1} \circ \beta \subseteq \beta^{-1} \circ \alpha$.)

(iv) The reduction relation $\to$ is called *weakly confluent* or *weakly Church-Rosser* (WCR) if it is weakly self-commuting (see Figure 1.1b), i.e. if $\forall$a, b, c $\in$ A $\exists$d $\in$ A (c $\leftarrow$ a $\to$ b $\Rightarrow$ c $\twoheadrightarrow$ d $\twoheadleftarrow$ b).
(The property WCR is also often called 'local confluence', e.g. in Jantzen [86].)

(v) $\to$ is *subcommutative* (notation $WCR^{\leq 1}$) if the diagram in Figure 1.1c holds, i.e. if

$\forall a,b,c \in A\ \exists d \in A\ (c \leftarrow a \rightarrow b \Rightarrow c \rightarrow_{\equiv} d \leftarrow_{\equiv} b)$.

(vi) $\rightarrow$ is *confluent* or is *Church-Rosser*, has the Church-Rosser property (CR) if it is self-commuting (see Figure 1.1d), i.e. $\forall a,b,c \in A\ \exists d \in A\ (c \twoheadleftarrow a \twoheadrightarrow b \Rightarrow c \twoheadrightarrow d \twoheadleftarrow b)$.

In the sequel we will use the terms ‘confluent’ and ‘Church-Rosser’ or ‘CR’ without preference. Likewise for weakly confluent and WCR, etc. The following proposition follows immediately from the definitions. Note especially the equivalence of (i) and (vi); sometimes (vi) is called ‘Church-Rosser’ and the situation as in Definition 1.1(vi) ‘confluent’.

1.3.2. PROPOSITION. *The following are equivalent:*
(i) $\rightarrow$ *is confluent*
(ii) $\twoheadrightarrow$ *is weakly confluent*
(iii) $\twoheadrightarrow$ *is self-commuting*
(iv) $\twoheadrightarrow$ *is subcommutative*
(v) *the diagram in Figure 1.1e holds, i.e.*

$$\forall a, b, c \in A\ \exists d \in A\ (c \leftarrow a \twoheadrightarrow b \Rightarrow c \twoheadrightarrow d \twoheadleftarrow b)$$

(vi) $\forall a, b \in A\ \exists c \in A\ (a = b \Rightarrow a \twoheadrightarrow c \twoheadleftarrow b)$
(*Here* ‘=’ *is the convertibility relation generated by* $\rightarrow$. *See diagram in Figure 1.1f.*) □

1.3.3. DEFINITION. Let $\mathcal{A} = \langle A, \rightarrow \rangle$ be an ARS.
(i) We say that $a \in A$ is a *normal form* if there is no $b \in A$ such that $a \rightarrow b$. Further, $b \in A$ *has a normal form* if $b \twoheadrightarrow a$ for some normal form $a \in A$.
(ii) The reduction relation $\rightarrow$ is *weakly normalizing* (WN) if every $a \in A$ has a normal form. In this case we also say that A is WN.
(iii) $\mathcal{A}$ (or $\rightarrow$) is *strongly normalizing* (SN) if every reduction sequence $a_0 \rightarrow a_1 \rightarrow \ldots$ eventually must terminate. (Other terminology: $\rightarrow$ is *terminating*, or *noetherian*.) If the converse reduction relation $\leftarrow$ is SN, we say that A (or $\rightarrow$) is SN^{-1}.
(iv) $\mathcal{A}$ (or $\rightarrow$) has the *unique normal form property* (UN) if
$\forall a, b \in A\ (a = b\ \&\ a,b \text{ are normal forms} \Rightarrow a \equiv b)$.
(v) $\mathcal{A}$ (or $\rightarrow$) has the *normal form property* (NF) if
$\forall a, b \in A\ (a \text{ is a normal form}\ \&\ a = b \Rightarrow b \twoheadrightarrow a)$.
(vi) $\mathcal{A}$ (or $\rightarrow$) is *inductive* (Ind) if for every reduction sequence (possibly infinite) $a_0 \rightarrow a_1 \rightarrow \ldots$ there is an $a \in A$ such that $a_n \twoheadrightarrow a$ for all n.
(vii) A (or $\rightarrow$) is *increasing* (Inc) if there is a map $|\ |: A \rightarrow \mathbb{N}$ such that
$\forall a, b \in A\ (a \rightarrow b \Rightarrow |a| < |b|)$.
Here $\mathbb{N}$ is the set of natural numbers with the usual ordering $<$.
(viii) $\mathcal{A}$ (or $\rightarrow$) is *finitely branching* (FB) if for all $a \in A$ the set of one step reducts of a,
$\{b \in A \mid a \rightarrow b\}$, is finite. If the converse reduction relation $\leftarrow$ is FB, we say that A (or $\rightarrow$)

is FB^{-1}. (In Huet [78], FB is called 'locally finite'.)

An ARS which is confluent and terminating (CR & SN) is also called *complete* (other terminology: 'canonical' or 'uniquely terminating').

Before exhibiting several facts about all these notions, let us first introduce some more concepts.

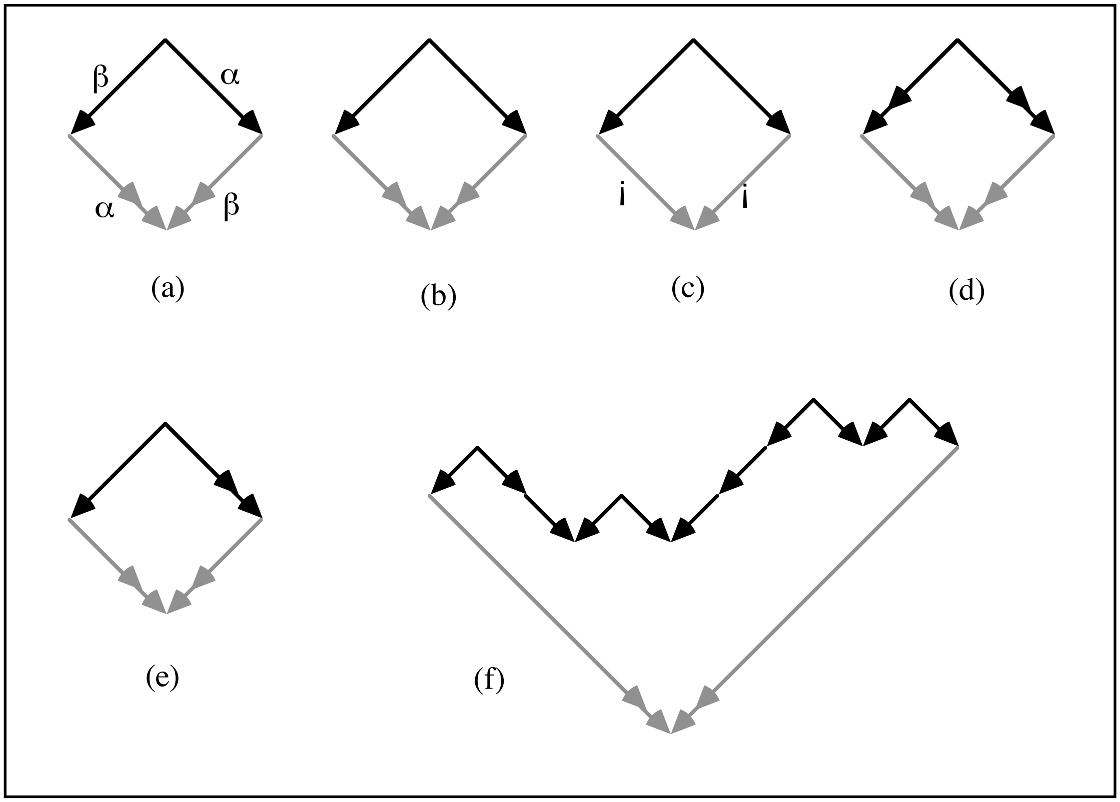


Figure 1.9

font correction: in (c) the inverted exclamation symbol should be ≡

1.3.4. DEFINITION. Let $\mathcal{A} = \langle A, \to_\alpha \rangle$ and $\mathcal{B} = \langle B, \to_\beta \rangle$ be two ARSs. Then $\mathcal{A}$ is a *sub-ARS* of $\mathcal{B}$, notation $\mathcal{A} \subseteq \mathcal{B}$, if:

(i) $\mathcal{A} \subseteq \mathcal{B}$

(ii) α is the restriction of β to A, i.e. $\forall a, a' \in A\ (a \to_\beta a' \Leftrightarrow a \to_\alpha a')$

(iii) A is closed under β, i.e. $\forall a \in A\ (a \to_\beta b \Rightarrow b \in A)$.

The ARS $\mathcal{B}$ is also called an *extension* of $\mathcal{A}$.

Note that all properties introduced so far (CR, WCR, $WCR^{\leq 1}$, WN, SN, UN, NF, Ind, Inc, FB) are preserved downwards: e.g. if $\mathcal{A} \subseteq \mathcal{B}$ and $\mathcal{B}$ is CR, then also $\mathcal{A}$ is so.

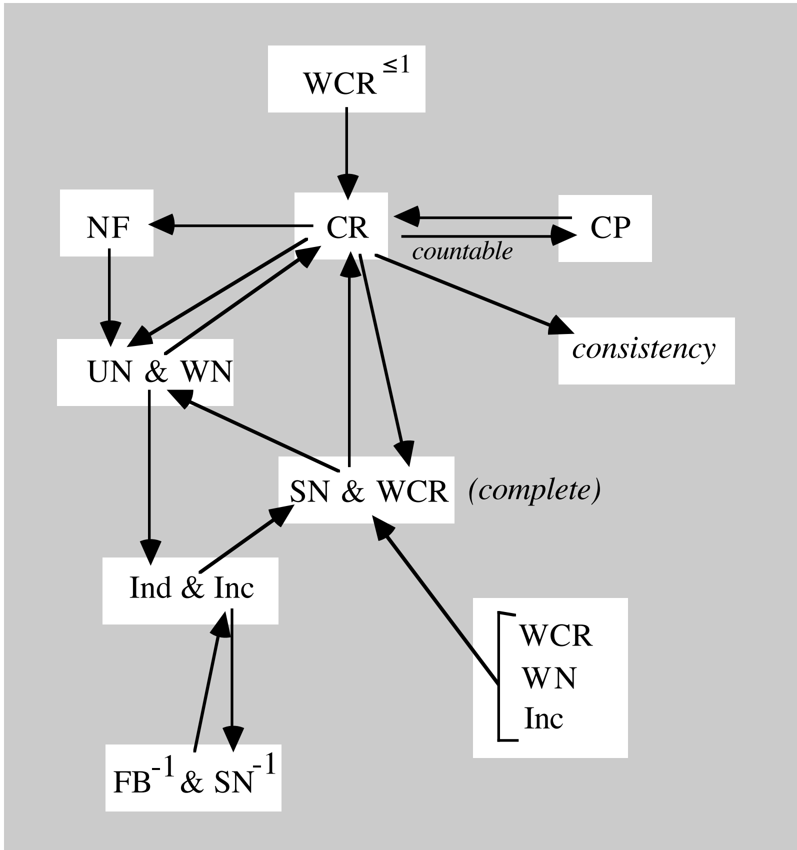
Of particular interest is the sub-ARS determined by an element a in an ARS:

1.3.5. DEFINITION. Let $\mathcal{A} = \langle A, \rightarrow \rangle$ be an ARS, and $a \in A$. Then $\mathcal{G}(a)$, the *reduction graph* of a, is the smallest sub-ARS of $\mathcal{A}$ containing a. So $\mathcal{G}(a)$ has as elements all reducts of a (including a itself) and is structured by the relation $\rightarrow$ restricted to this set of reducts.

We will now collect in one theorem several implications between the various properties of ARSs. The first part (i) is actually the main motivation for the concept of confluence: it guarantees unique normal forms, which is of course a desirable state of affairs in (implementations of) algebraic data type specifications. Apart from the fundamental implication CR $\Rightarrow$ UN, the most important fact is (ii), also known as Newman's Lemma. The property CP ('cofinality property') is... see Exercise 1.7.13 below,

1.3.6. THEOREM.

(i) CR $\Rightarrow$ NF $\Rightarrow$ UN
(ii) SN & WCR $\Rightarrow$ CR *(Newman's Lemma)*
(iii) UN & WN $\Rightarrow$ CR
(iv) UN & WN $\Rightarrow$ Ind
(v) Ind & Inc $\Rightarrow$ SN
(vi) WCR & WN & Inc $\Rightarrow$ SN
(vii) CR $\Leftrightarrow$ CP *for countable ARSs.* □



relations between properties of Abstract Reduction Systems

Figure 1.10

PROOF. (i) Immediate from the definitions.
(ii) Short proofs of Newman's Lemma are given in Huet [78] and in Barendregt [84]. An alternative proof, illustrating the notion of 'proof ordering', is given in Chapter 6 (Exercise 6.10.1). Another proof, using König's Lemma, is given in the solution to Exercise xxx. The proof in Barendregt [84] is very simple:

Call an element in the ARS under consideration *good* if it has exactly one normal form, *bad* if it has two or more. Now we claim that a bad point has a one step reduct which is again bad. To this end, let a be bad, reducing to different normal forms n_1, n_2: $a \to b \to \ldots n_1$ and $a \to c \to \ldots n_2$. If $b \equiv c$ the claim is proved: b is bad. Otherwise, apply WCR on the diverging steps $a \to b$, $a \to c$ to yield a common reduct d such that $b \twoheadrightarrow d$ and $c \twoheadrightarrow d$. If d is bad, also b, c are bad. If d is good, it reduces to n_1, n_2 or some other normal form n_3. In all cases b or c is bad. This proves the claim. However, the claim contradicts with the assumption of SN.

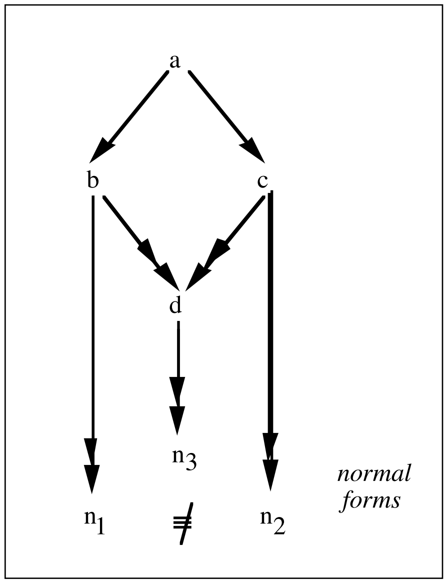


Figure xx

(v) is from Nederpelt [73]; the proof is obvious.
(vi) is proved in Klop [80].
(vii) see Exercise 1.7.13 below, where it is noted that the condition 'countable' cannot be missed.

The propositions in the statement of the theorem (and some more) are displayed also in Figure 1.2; here it is important whether an implication arrow points to the conjunction sign &, or to one of the conjuncts. Likewise for the tail of an implication arrow. (E.g. UN & WN ⇒ Ind, SN & WCR ⇒ UN & WN, Inc ⇒ SN^{-1}, FB^{-1} & SN^{-1} ⇒ Inc, CR ⇒ UN but not CR ⇒ UN & WN.)

It is not possible to reverse any of the arrows in this diagram of implications. An instructive counterexample to WCR ⇒ CR is the TRS in Figure 1.3 (given by R. Hindley, see also Huet [78]).

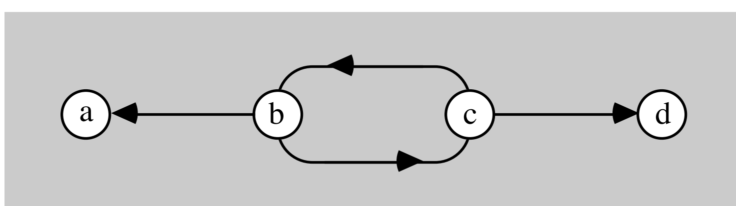


Figure 1.3

References

ARTIN, E.(1926). *Theorie der Zöpfe*. Abh. math. Semin. Hamburg Univ. 4 (1926), 47-72.

ARTIN, E.(1947). *Theory of braids*. Ann. of Math. (2) 48 (1947) 101-126

ARTIN, E.(1947a). *Braids and permutations*. Ann. of Math. vol. 48 (1947), 643-649

GARSIDE, F.A. (1969). *The braid group and other groups*. Quart. J. Math. 20 (1969) 235-254

MAKANIN, G.S. (1968). *The conjugacy problem in the braid group*. Soviet Math. Dokl. 9 (1968) 1156-1157

HINDLEY, J.R. (1964). *The Church-Rosser property and a result in combinatory logic*. Ph.D. Thesis, Univ. Newcastle-upon-Tyne, 1964.

HUET, G. (1980). *Confluent reductions: Abstract properties and applications to term rewriting systems*. JACM, Vol.27, No.4 (1980), 797-821.

KLOP, J.W. (1980). *Combinatory Reduction Systems*. Mathematical Centre Tracts 127, Amsterdam 1980.

KLOP, J.W. (1992). *Term rewriting systems*. In Vol.2 of Handbook of Logic in Computer Science (eds. S. Abramsky, D. Gabbay & T. Maibaum), Oxford University Press 1992, p.1-116

NEWMAN, M.H.A. (1942). *On theories with a combinatorial definition of* "*equivalence*". Annals of Mathematics, 43(2):223-243, April 1942.

SCHMIDT, G. & STRÖHLEIN, T. (1991). *Relations and Graphs - Discrete Mathematics for Computer Scientists*. Springer-Verlag, EATCS Monographs on Theoretical Computer Science, 1991.

1.7.13. EXERCISE (Klop [80]). Let $\mathcal{A} = \langle A, \rightarrow \rangle$ be an ARS. Let $\mathcal{B} \subseteq \mathcal{A}$. Then $\mathcal{B}$ is *cofinal* in $\mathcal{A}$ if $\forall a \in A \ \exists b \in B \ \ a \twoheadrightarrow b$. Furthermore, $\mathcal{A}$ is said to have the *cofinality property* (CP) if in every reduction graph $\mathcal{G}(a)$, $a \in A$, there is a (possibly infinite) reduction sequence $a \equiv a_0 \rightarrow a_1 \rightarrow \ldots$ such that $\{a_n \mid n \geq 0\}$ is cofinal in $\mathcal{G}(a)$.

(i) Prove that for *countable* ARSs: $\mathcal{A}$ is CR $\Leftrightarrow$ $\mathcal{A}$ has CP.

(ii) Show that the condition of countability cannot be missed. (See Solutions.)

1.3. Confluence by decreasing diagrams

In this section we present a recently found quite powerful criterion for confluence of abstract rewriting. The method, developed by van Oostrom [94, 94a] and called ‘confluence by decreasing diagrams’, generalizes several well-known confluence criteria for abstract rewriting such as the Lemma of Hindley and Rosen, Huet’s Strong Confluence lemma , Newman’s Lemma, the ‘request’ lemma’s of Staples, the relative termination lemma of Geser (see the exercises at the end of this chapter). For these applications we refer to van Oostrom [94, 94a]. Actually, the way to van Oostrom's method was prepared by an unpublished note of De Bruijn [78], containing a slightly weaker form of van Oostrom's theorem with a complicated inductive proof. The notion of decreasing diagrams was not yet present in that note.

We will consider ARSs, indexed by some set I: $\mathcal{A} = \langle A, (\rightarrow_\alpha)_{\alpha \in I} \rangle$, which in this section is a well-founded partial order. In examples, we will use the set of natural numbers with the usual ordering as index set.

1.2.1. Reduction diagrams

An important ingredient in finding a common reduct of the end points of two diverging reduction sequences consists of the *elementary diagrams*, see the examples in Figure 1.1. They are the ‘atomic’ or basic building blocks for constructing reduction diagrams. An non-trivial elementary diagram consists of two diverging steps (arrows), joined by two sequences of steps of arbitrary length. Note that in the e.d.’s we may use empty sides (the dashed sides, in some figures shaded), to keep matters orthogonal. This gives rise to some trivial e.d.’s as in the lower part of Figure 1.1. The e.d.’s are used as ‘tiles’ with the intention to obtain a completed reduction diagram as in Figure 1.2. Usually we will forget the direction of the arrows (second picture in Figure 1.2): they always are from left to right, or downwards (except the empty ‘steps’ that have no direction).

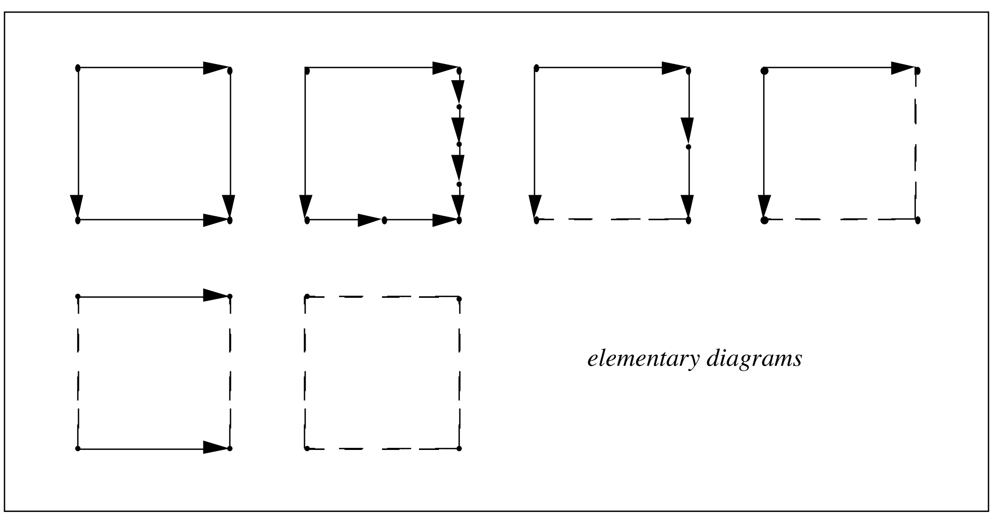
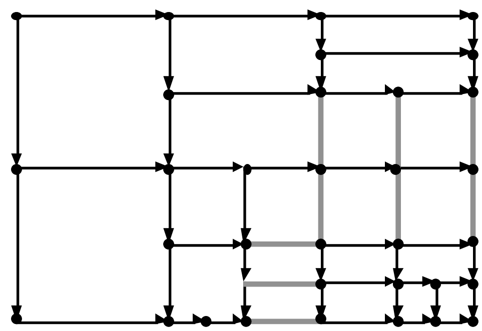


Figure 1.4



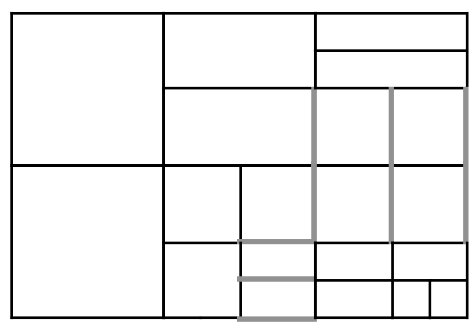


Figure 1.5

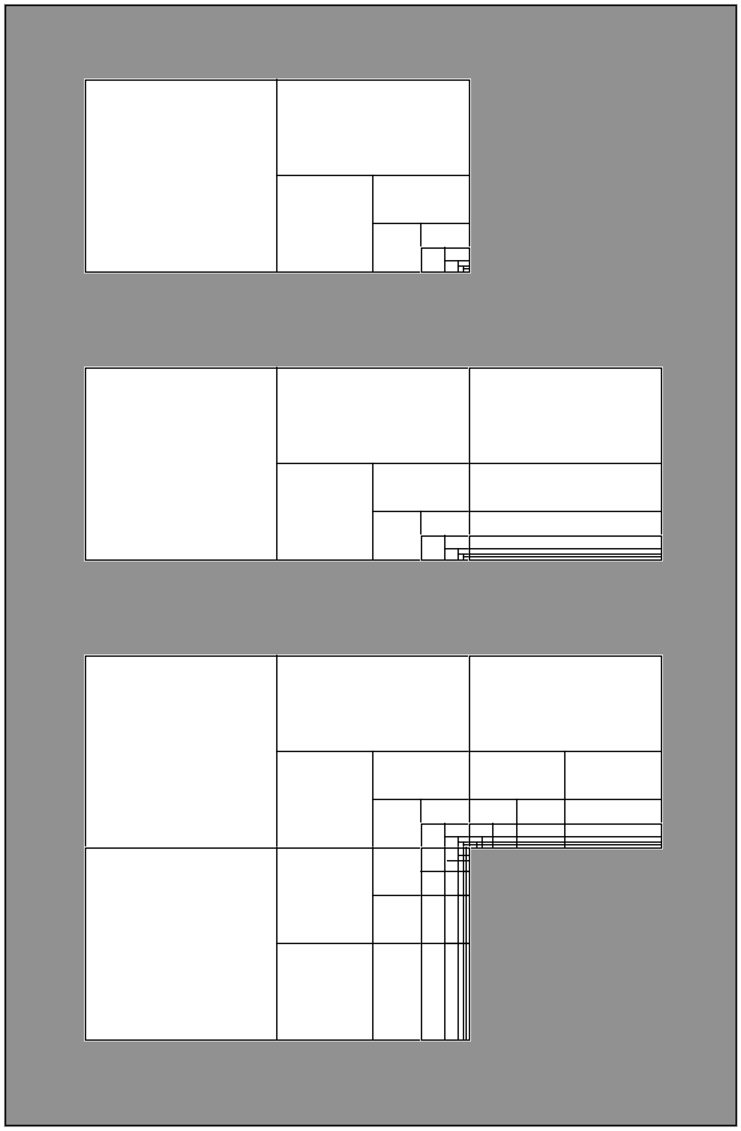


Figure x

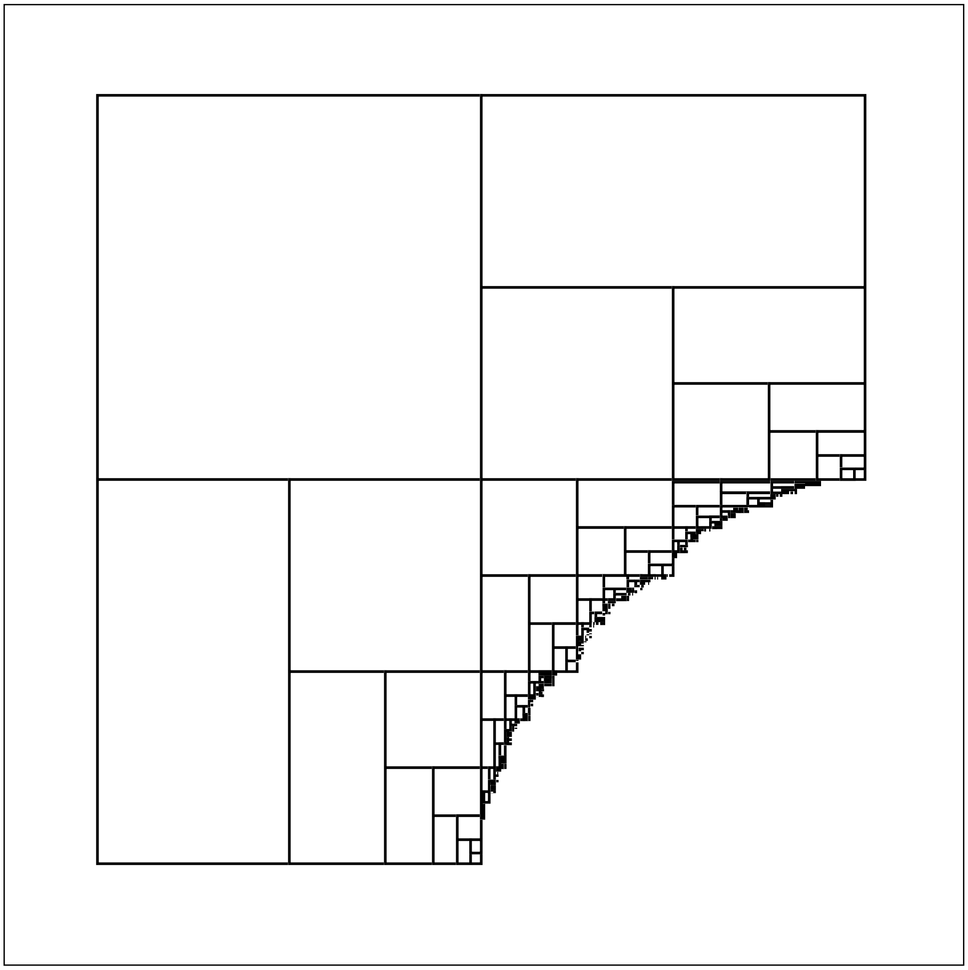


Figure x

EXERCISE. (i) Prove that an infinite reduction diagram must possess an infinite proper reduction (i.e. one without empty 'steps'. (Solution included.)
(ii) Obtain Newman's Lemma as a corollary of (i).

SOLUTION. Consider a construction of the infinite diagram in stages, by repeatedly adjoining an e.d. After each finite stage the diagram contains finitely many proper reductions starting from the initial term of the two diverging reductions that constitute stage 0 of the construction. Also, after each finite stage there must be eventually an adjunction of an e.d. with splitting converging sides; otherwise the construction would terminate. But such an adjunction will prolong, with at least one proper step, at least one of the finite proper reductions from the initial term that are present in the diagram at that stage. Now apply König's Lemma: in the limit an infinite proper reduction must arise.

We will need somewhat more structure on the multiset p.o. If we have X $\geq_{\mu}$ Y, there is a 'descendant' relation between the elements of X and Y. Some elements of X are 'preserved' in Y: this is indicated by heavy arrows (see Figure 1.3). Some elements of X will be replaced by some elements in Y that are strictly smaller (in the p.o. I); this is indicated by light arrows. Heavy arrows cannot split, light arrows can. From an element of X also zero light arrows can exit: that element just disappears. (E.g. the '1' in Figure 1.3.) A descendant relation for X $\geq_{\mu}$ Y by means of 'multiset arrows' need not be unique, e.g. the pair of multisets in Figure 1.3 admits several other descendant relations.

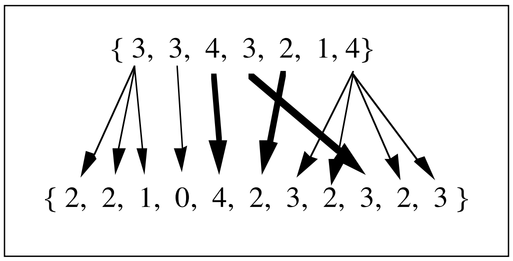


Figure 1.6

1.2.2. Monotonic filtering

We start with an important definition. Given a tuple σ of natural numbers (we will use natural numbers as running example, but everything below is in fact intended for a well-founded p.o. I), *filter*(σ) is the tuple obtained by 'reading' σ from left-to-right, removing the elements that are less than what was already encountered, and taking the tuple of the remaining elements. See example in Figure 2.1. Another operation on tuples is *multiset*; it yields the corresponding multiset. In the sequel we will be especially interested in *multiset* (*filter*(σ)).

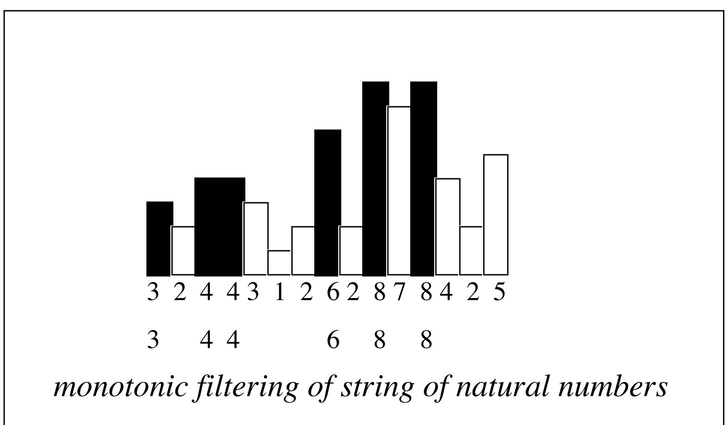


Figure 1.7

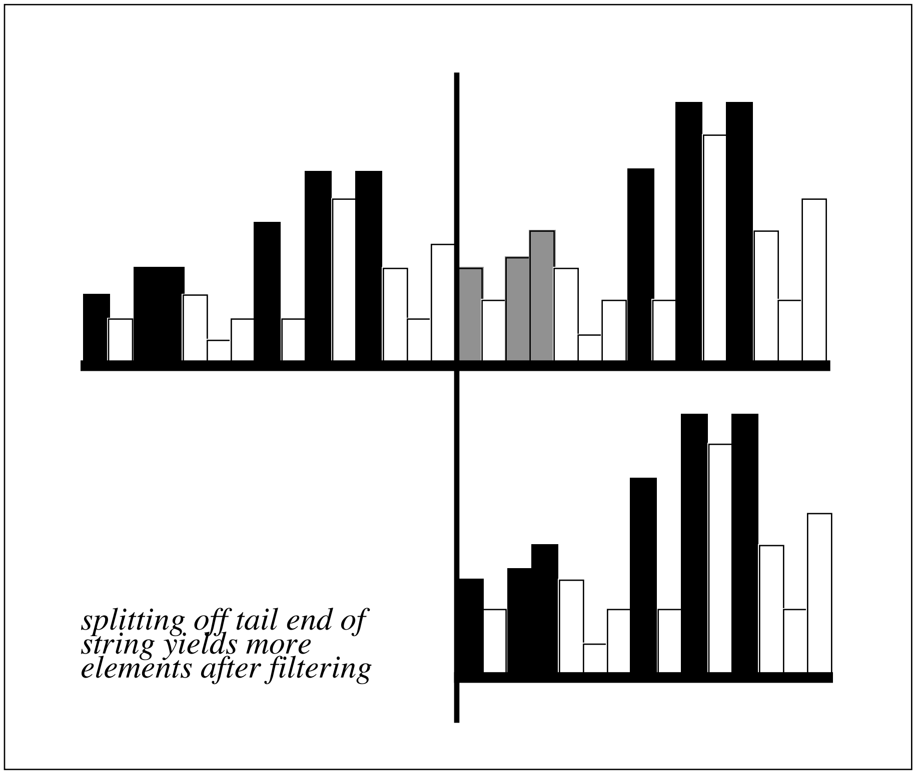


Figure 1.8

In the sequel the following phenomenon will appear: if σ, τ are tuples (strings), then we do not simply have *multiset*(*filter*(σ·τ)) = *multiset*(*filter*(σ)) ∪ *multiset*(*filter*(τ)). Figure 2.2 explains what happens, some extra elements may crop up after splitting.

1.2.3. Decreasing diagrams

Before defining what a decreasing diagram is, we need the notion of 'norm of a reduction sequence' in the ARS with indexed rewrite relations. This will be a tuple of natural numbers (in general, elements of I). Par abus de langage, we will also denote reduction sequences with σ, τ. If σ is a reduction sequence, *label*(σ) is the string of indexes of consecutive reduction steps in σ. Single steps will be denoted by α, β. So *label*(α) is the index of the step α. If σ = ...α...β...we say that α is *before* β in σ.

1.2.3.1. DEFINITION.

(i) Let σ be a reduction sequence. Then |σ|, the *norm* of σ, is *multiset*(*filter*(*label*(σ)))

(ii) The norm of two diverging reductions σ, τ is $|\sigma| \cup |\tau|$. (Figure 2.3.)

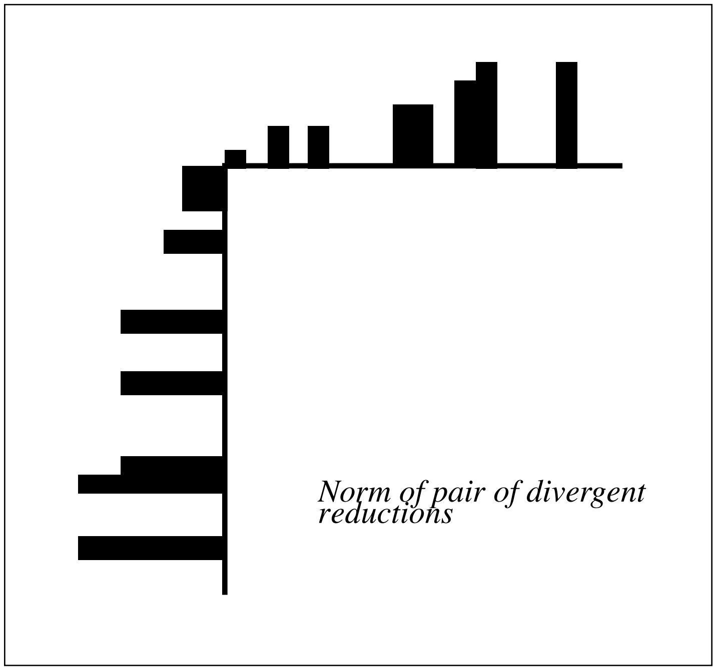


Figure 1.9

1.2.3.2. DEFINITION. Let $\sigma: a \twoheadrightarrow b$, $\tau: a \twoheadrightarrow c$, $\sigma': c \twoheadrightarrow d$, $\tau': b \twoheadrightarrow d$ be reductions forming the reduction diagram $\mathcal{D}$ with corners a, b, c, d. (Figure 2.4.) Then $\mathcal{D}$ is a *decreasing diagram*, if

$|\sigma| \cup |\tau| \geq_{\mu} |\sigma{\cdot}\tau'|$ and
$|\sigma| \cup |\tau| \geq_{\mu} |\tau{\cdot}\sigma'|$

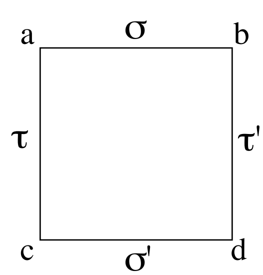


Figure 1.10

We will now give a slightly different definition of 'decreasing diagram', which we will call 'trace-decreasing diagram'. It will turn out that trace-decreasing implies decreasing, but not vice versa. However, in case the index set I is a total well-founded order, then the two notions coincide. Moreover, also for partially ordered I the two notions coincide for elementary diagrams. The notion of trace-decreasing is more cumbersome to formulate than van Oostrom's definition above, but it helps in visualizing this rather intricate concept of decreasing diagram.

1.2.3.3. DEFINITION. A diagram $\mathcal{D}$ as in the preceding definition is *trace-decreasing* if:

(i) Every step α' in τ' traces back to a unique step α in σ or τ, called its ancestor, such that *label*(α) ≥ *label*(α'). If *label*(α) > *label*(α') we say that the tracing relation from α to α' is given by a light arrow, if *label*(α) = *label*(α') by a heavy arrow. Likewise dually.

(ii) Heavy arrows can only go from τ to τ' and from σ to σ'. The first are called horizontal, the second are vertical. Horizontal heavy arrows do not cross, likewise dually. Heavy arrows cannot split (i.e. no two heavy arrows have the same begin). See Figure 2.5a.

(iii) If α' in τ' traces back by a heavy arrow to α in τ, and if β' is a step in τ' before α' then β' traces back to a step β (by a light arrow) in σ or a step β (by a light or heavy arrow) in τ before α. Likewise dually. See Figure 2.5b.

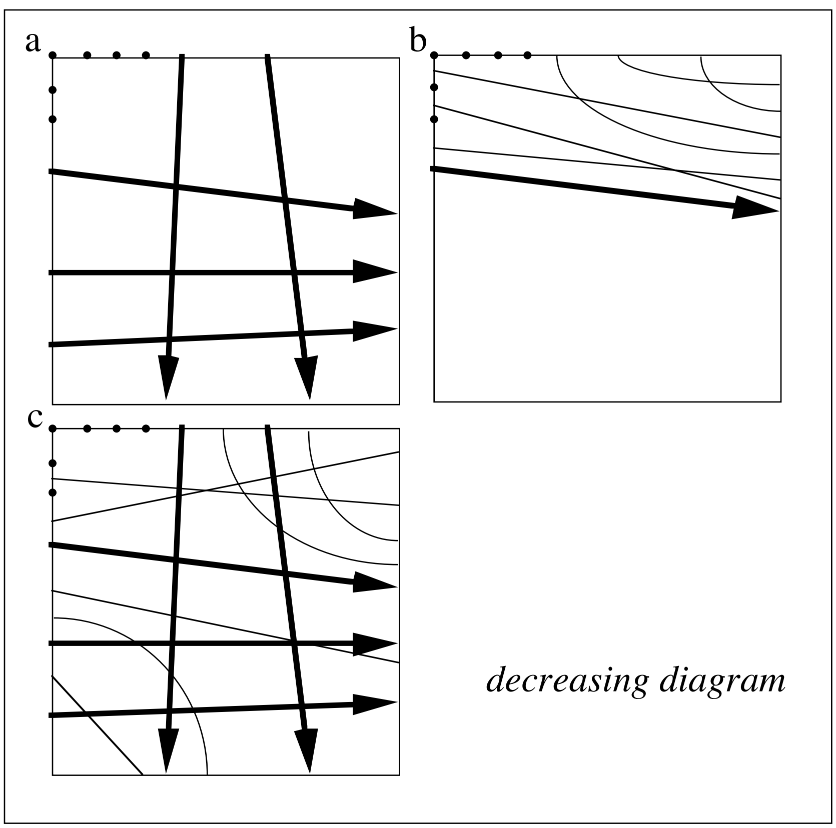


Figure 1.11

The notion of trace-decreasing can conveniently be described by forbidding certain configurations of the tracing arrows, as in Figure 2.6.

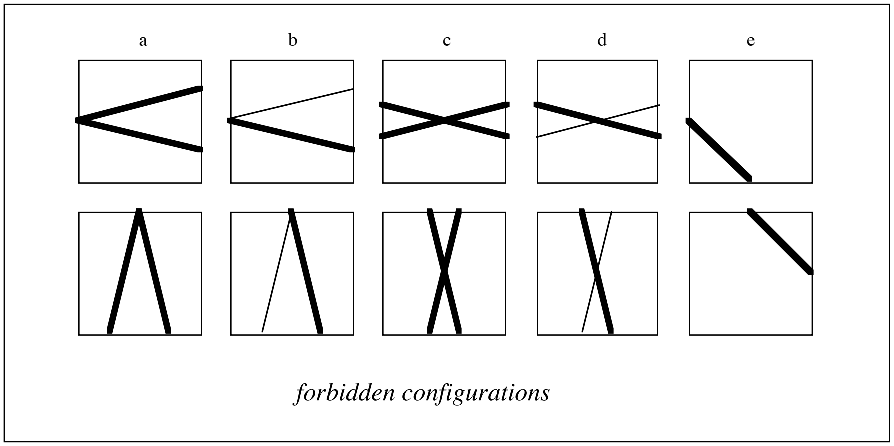


Figure 1.12

1.2.3.4. REMARK. Note that the situation as in Figure 2.7 (and also its dual) is allowed. This means that the tracing arrows are not 'multiset arrows' as in Figure 1.3; a heavy multiset arrow cannot be co-initial with any arrow.

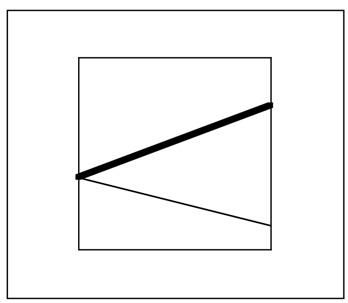


Figure 1.13

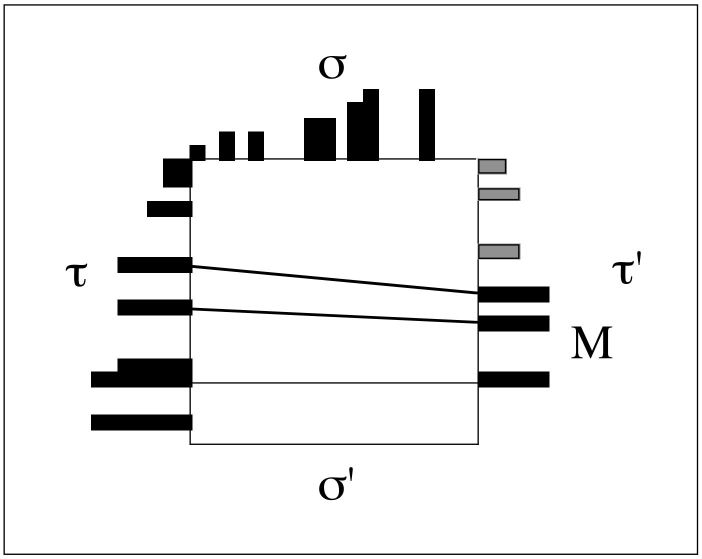


Figure 1.14

1.2.3.5. PROPOSITION. *A trace-decreasing diagram is decreasing*.

PROOF. Suppose the diagram as in Definition 2.2.2 (see Figure 2.8) is trace-decreasing. We have to prove that $|\sigma| \cup |\tau| \geq_\mu |\sigma{\cdot}\tau'|$. According to Proposition 1.3.1, the elements in $|\sigma|$ 'take care for themselves', so we can leave them out and prove that $|\tau| \geq_\mu |\tau'|$ *minus* the elements majorized by some element in σ. (The deleted elements are shaded in Figure 2.8.) Call the latter multiset M. Now consider the given tracing relation. Each element in M traces back to the opposite τ (not to σ since elements in M are not majorized by those in σ, and heavy arrows cannot go from σ to τ').

However, we are not yet done, since (1) the tracing arrows need not be multiset trace arrows and (2) since the ancestors are in τ, not the filtered $|\tau|$.

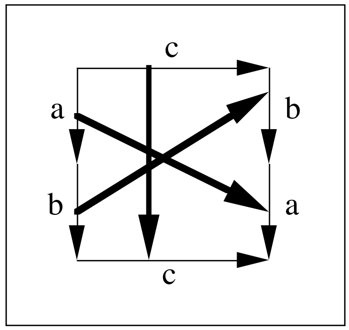
Ad (1). This obstacle is in fact absent: situations as in Figure 2.6a, b (top row) are forbidden by definition of trace-decreasing, and the situation as in Figure 2.7 does not occur because M (or rather the tuple underlying the multiset M) is monotone.

Ad (2). Obstacle (2) is overcome as follows. When an element α' in M has as ancestor a 'white' α in τ (that is therefore filtered out in $|\tau|$) then we take as new ancestor of α', the element α* in τ that is the nearest black step in τ before α. It is not hard to check (using the fact that M is monotone, and properties of trace-decreasing) that this redirection of arrow-roots does the job: all ancestors of elements in M are now in $|\tau|$, and the arrows are multiset arrows, i.e. $|\tau| \geq_\mu$ M. □

1.2.3.6. REMARK. (i) The reverse: *decreasing* ⇒ *trace-decreasing,* does not hold. A counterexample is given in Figure 2.9. Here a, b, c are incomparable elements in the p.o. I. This diagram is decreasing:
$|\sigma| \cup |\tau| = |\sigma{\cdot}\tau'| = |\tau{\cdot}\sigma'| = \{a, b, c\}$, but not trace-decreasing since there is a crossing of heavy

arrows.

However, in case the order I is total, the reverse does hold. The proof is routine (just 'uncross' the crossed heavy arrows) and omitted; we do not need it in the sequel.



(ii) Note why crossing heavy arrows are harmful (see Figure 2.10): an allowed configuration as in Figure 2.7 could turn after appending a pair of crossing heavy arrows into a forbidden situation (as in Figure 2.6b).

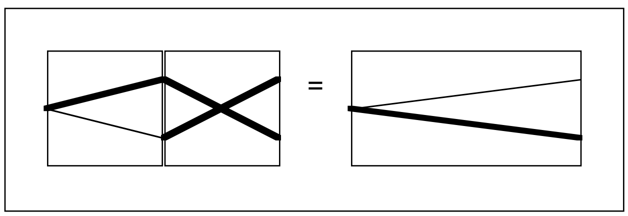
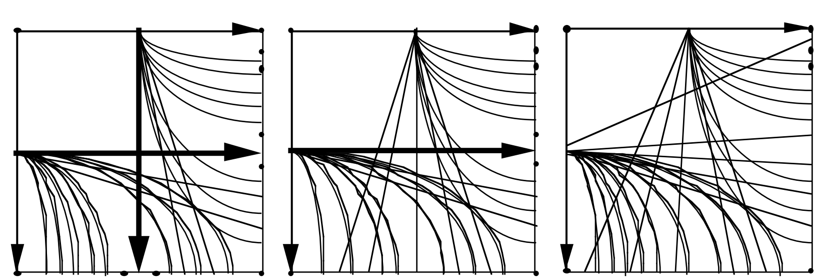


Figure 1.16

For elementary diagrams, the property of being decreasing amounts to the situation as in Figure 2.11. The second and third e.d. in this figure are degenerated cases, where only one or zero heavy arrows are present.



decreasing elementary diagram

Figure 1.17

1.2.3.7. EXAMPLE. So, we have examples as in Figure 2.12 of some decreasing and non-decreasing e.d.'s. (We will come back to the set of non-decreasing tiles in Figure 2.12(a) in Section 4, about confluence of braids.)

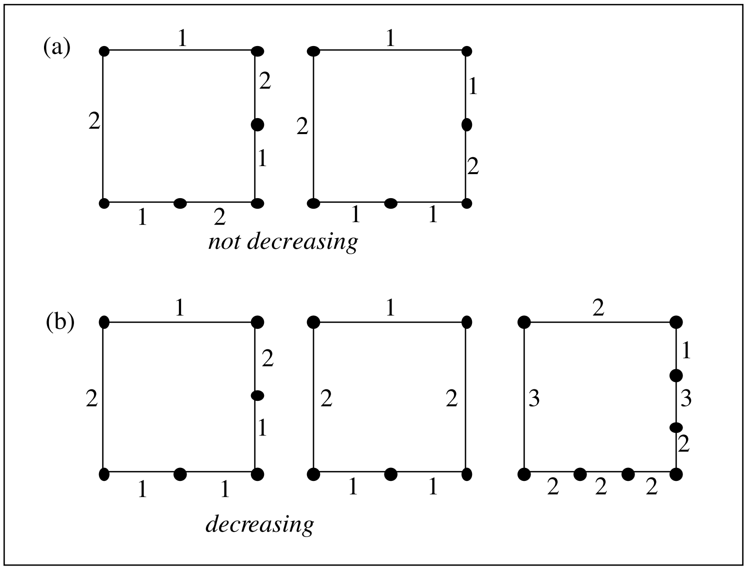


Figure 1.18

Now we will establish the two important properties of trace-decreasing diagrams that give confluence. The first is indicated in Figure 2.13: pasting preserves trace-decreasingness.

1.2.3.8. PROPOSITION. *Let two trace-decreasing diagrams be joined as in Figure 2.13. Then the resulting diagram is again trace-decreasing.*

PROOF. The proof is simply by checking that no forbidden trace configurations arise by joining two trace-decreasing diagrams as indicated. See also Figure 2.14, indicating how new trace configurations arise after pasting two trace-decreasing diagrams together to yield one. (In particular, concatenating a light arrow with a heavy arrow yields a light one, two heavy arrows concatenated yield again a heavy one, etc.) ∎

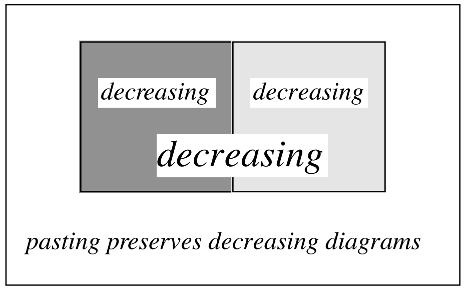


Figure 1.19

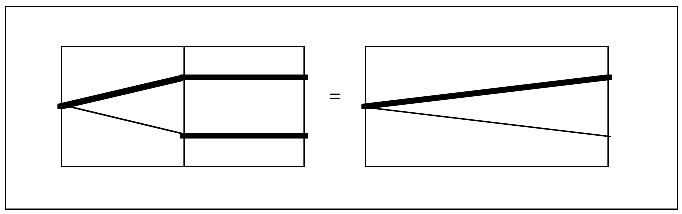


Figure 1.20

The second important property is indicated in Figure 2.15: inserting a decreasing diagram in a pair of co-initial reductions reduces the norm of the resulting pair of co-initial reductions. The proof uses Figure 2.16.

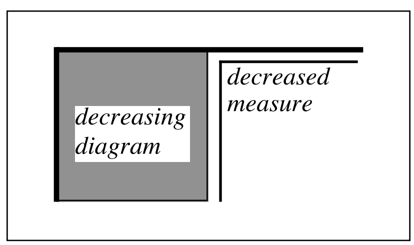


Figure 1.21

1.2.3.9. PROPOSITION. *Let a trace-decreasing diagram be inserted as in Figure 2.15 into a pair of diverging reductions. Then the resulting pair of diverging reductions has a smaller norm.*

PROOF. See Figure 2.16, where 1,2,...,6 denote the multisets of the indicated elements. In particular, 3 and 5 denote those elements (shaded in the figure) that are majorized by (an element in) 2.

Because the inserted diagram is trace-decreasing, it is also decreasing (Proposition 2.2.5); so we have

$1 \cup 2 \geq_\mu 2 \cup 6.$

By Proposition 1.3.1 therefore

$1 \geq_\mu 6.$

Furthermore

$2 \geq_\mu 3 \cup 5.$

Hence

$1 \cup 2 \geq_\mu 3 \cup 5 \cup 6.$

So

$1 \cup 2 \cup 4 \geq_\mu 3 \cup 4 \cup 5 \cup 6.$

Finally,

$1 \cup 2 \cup 4 >_\mu 3 \cup 4 \cup 5 \cup 6,$

since the first element in the horizontal reduction of the inserted decreasing diagram has been 'used' (is not preserved in $3 \cup 4 \cup 5 \cup 6$) either by disappearing or by yielding some lesser elements. □

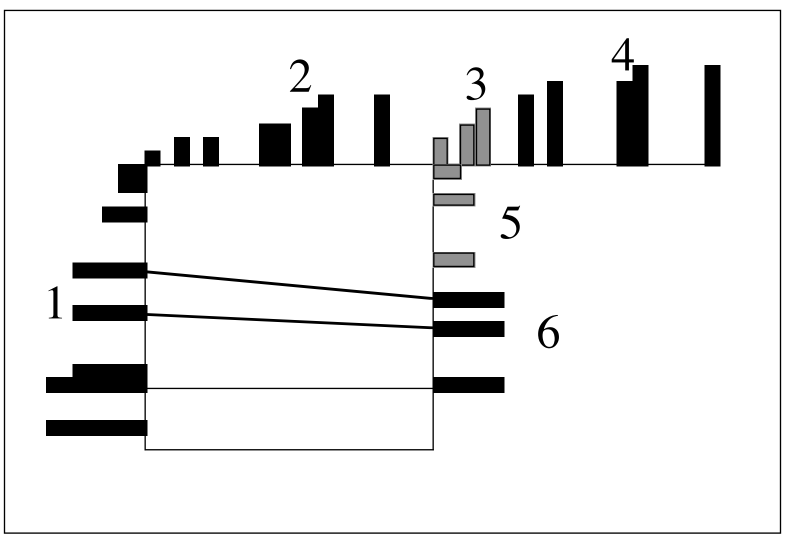


Figure 1.22

Finally, we can combine the two important properties to yield a proof of confluence, based on well-founded induction. See Figure 2.17.

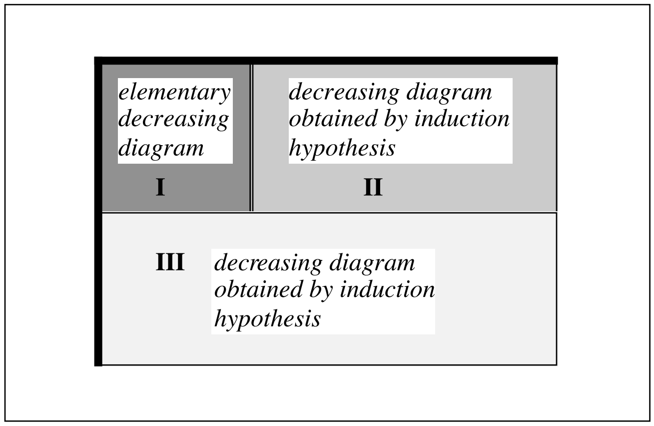


Figure 1.23

We now arrive at the conclusion: ‘confluence by decreasing diagrams’.

1.2.3.10. THEOREM (Van Oostrom [94])
Every ARS with reduction relations indexed by a well-founded partial order I, and satisfying the decreasing criterion for its e.d.’s, is confluent.

PROOF. Immediate, by combining Propositions 2.2.7 and 2.2.8. (See Figure 2.17.) ∎

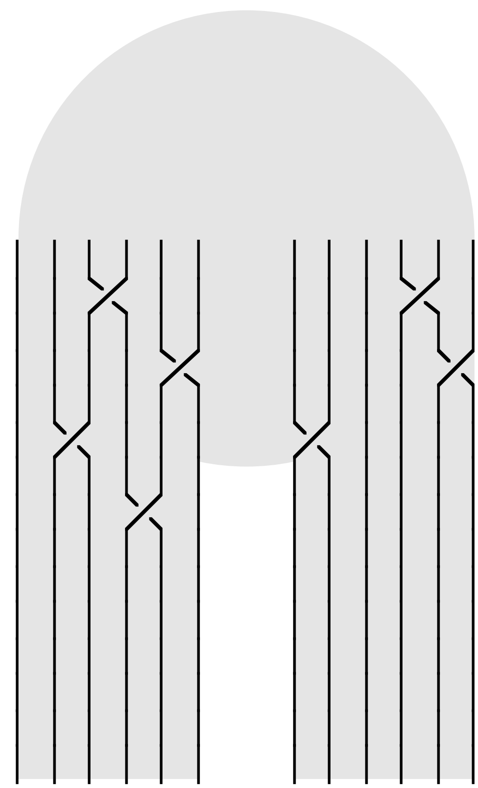
1.2.3.11. REMARK. H. Zantema (personal communication) remarked that in fact we needed to prove this theorem only for the restricted case of well-founded totally ordered I. The theorem above then follows by this reasoning: every well-founded p.o. I can be extended by Zorn’s Lemma to a well-founded total order I^*. If an elementary diagram is decreasing with respect to I, it is also decreasing with respect to I^*. An appeal to the theorem above for the restricted case of total orders, then yields confluence. Hence we have the unrestricted theorem above. We did not follow this route since it would entail no significant reduction in the burden of proof.

1.3. Confluence of braids

As a contrast, however, to the applicability of the decreasing diagram method, we discuss in the final section of this chapter a situation where confluence is obtained while the diagrams involved are not decreasing. The example concerns the group and the semi-group of 'braids', studied already more than half a century ago (see Artin [26, 47, 47a]). The braid confluence problem is described on p.132-134 in Schmidt and Ströhlein [91], in the following anthropomorphic terms.

1.3.1. The semi-group of braids

A girl has two braids consisting of, say, 6 strings (see Figure 4.1). The father starts braiding the left braid, the mother of the girl starts braiding the right braid. After some initial 'twists' as indicated in the figure, they notice that they do it in a different way. But they want to arrive, eventually, at two identical braids. *Question:* can they go on and still arrive at identical braids?



Girl with two braids

Figure 1.24

Note that braids are subject to a topological equivalence, which will be explained now. First we need a notation for braids. Consider Figure 4.2. The openings between the strings are numbered

1, 2, 3, ... A twist or crossing in which the upper string moves over the lower is denoted 'positively', just as the corresponding opening: if this is i, the positive twist at this position is also denoted by i. Otherwise we have a 'negative' crossing, denoted by i^{-1} if it is in the i-th opening. Thus the braid in Figure 4.2a is $1.2^{-1}.1.2^{-1}.1.2$.

Now we restrict attention to *positive crossings only*. E.g. in Figure 4.2b we have the braid 1.2.4.1.3.1.4.3 . The restriction means that we work in the semi-group generated by 1, 2, 3, 4 (if there are five strings).

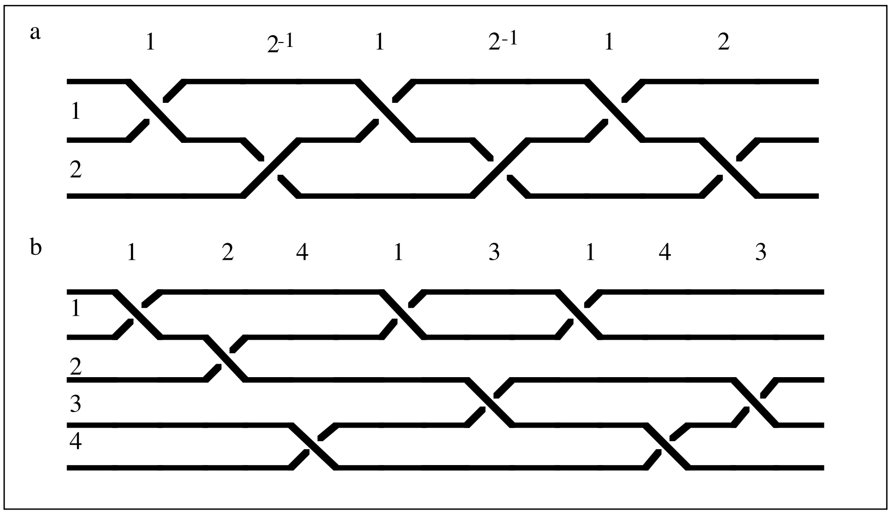


Figure 1.25

Not all these braids are really different. See Figure 4.3a. The braid 1.3 is 'the same', topologically viewed, as 3.1, just by shifting the crossings in the other order. Also 1.4 is equivalent with 4.1. We will write 1.3 = 3.1, and 1.4 = 4.1. In general we have:

$$i.j = j.i \quad \text{if } |i-j| \geq 2$$

For consecutive openings like 1 and 2, respective crossings do not commute. But it is not hard to see that starting with 1.2 and 2.1, we can make them (topologically) equal by continuing 1.2 with 1 and 2.1 with 2. So 1.2.1 = 2.1.2. See Figure 4.3b. Note that 1.2.1 and 2.1.2 are indeed topologically the same; an experiment with actual strings of wire will demonstrate this. In general we have for all i:

$$i.(i+1).i = (i+1).i.(i+1)$$

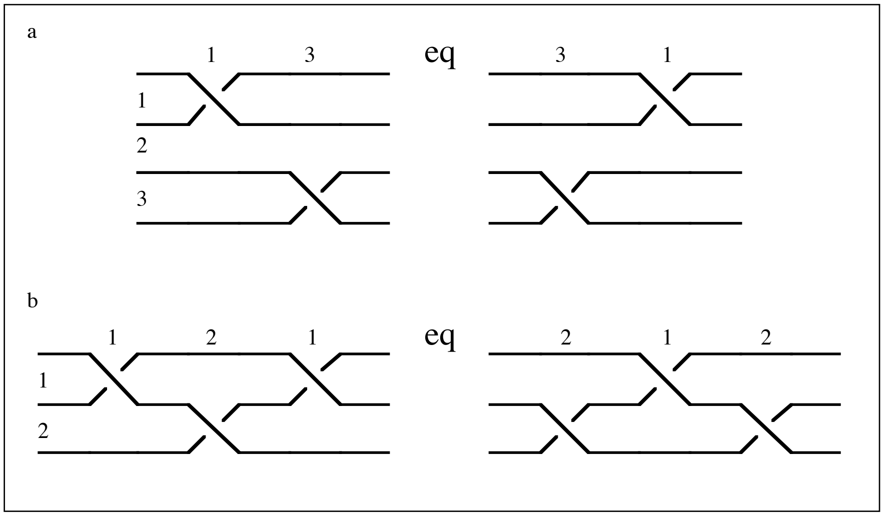
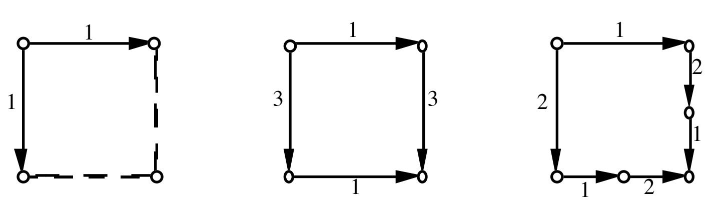


Figure 1.26

The equations above completely define the topological equivalence considered (see Artin [47]). The confluence problem is now: given two elements u, v of this braid semi-group, can we always find elements x, y such that ux = vy? The problem can be approached by means of an abstract rewriting analysis using the elementary diagrams as in Figure 4.4. (Only some of the generators 1, 2, 3,... are mentioned in the figure.) The question is now whether tiling with these diagrams always succeeds in a confluent reduction diagram. Note that these e.d.'s are *not* decreasing (that is, the third one in Figure 4.4 is not).



elementary diagrams for confluence problem in braid semi-group

Figure 1.27

1.3.2. EXAMPLE. We complete in a diagram the braidings started by the father and the mother as in Figure 4.1: the braids there are 3142 and 215 (counting the openings from right to left). See Figure 4.5.

1.3.3. REMARK. If we admit also negative crossings, the confluence problem trivializes: given the braids 3142 and 215 the continuations as in $3142(3142)^{-1} = 31422^{-1}4^{-1}1^{-1}3^{-1}$ and $215(215)^{-1} = 2155^{-1}1^{-1}2^{-1}$ yield the same result, ε. We are now in the braid *group*; it has the same defining relations as above for the semi-group.

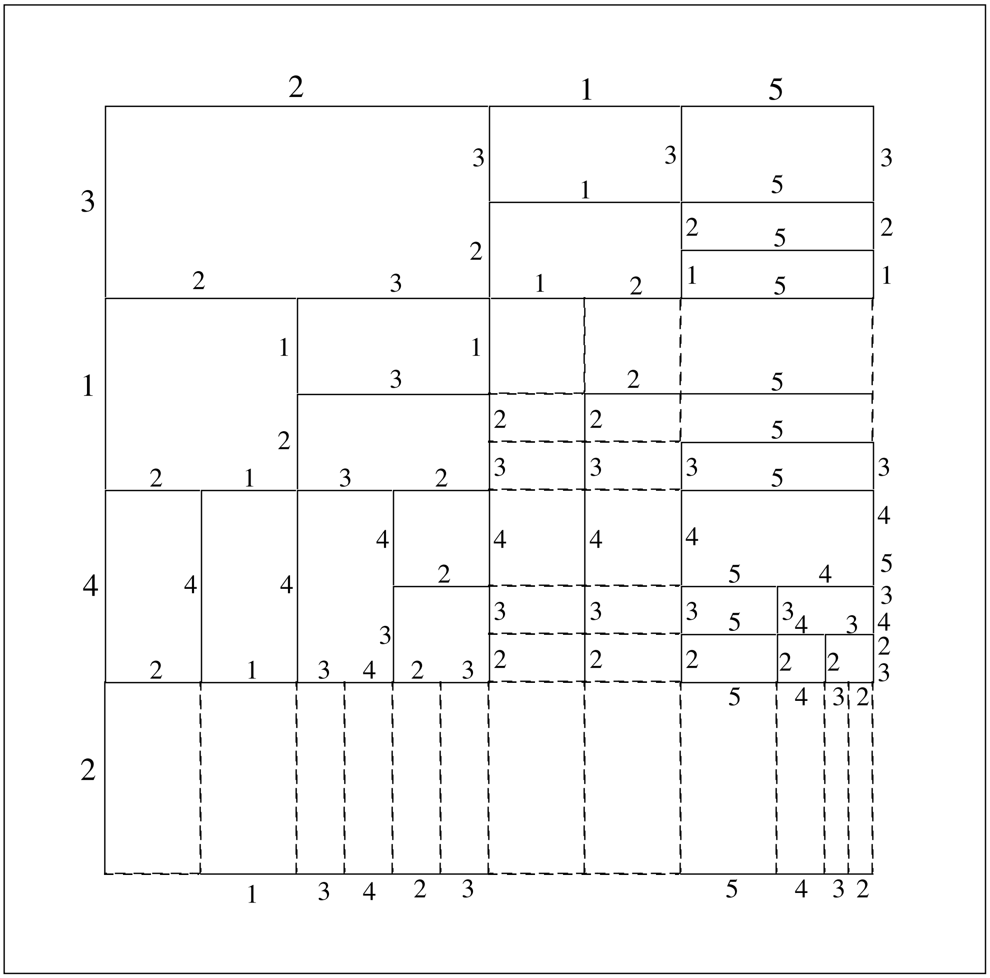


Figure 1.28

1.4. Confluence and termination

The confluence problem for the braid semi-group can be reformulated as a termination problem in the braid group, as follows. To prove: in every braid, i.e. tuple of consecutive positive and negative crossings, we can ‘postpone’ (= push to the right) the negative crossings, using the rewrite rules (writing 2' instead of 2^{-1}, etc.):

$$i'\cdot j \rightarrow j\cdot i\cdot j'\cdot i' \text{ if } |i-j| = 1$$

$i' \cdot j \rightarrow j \cdot i'$ if $|i-j| \geq 2$
$i' \cdot i \rightarrow \varepsilon$, the empty string

Example (see Figures 4.6 and 4.7)
3' 2 2 1 3 → 2 3 2' 3' 2 1 3 → 2 3 2' 2 3 2' 3' 1 3 → 2 3 3 2' 3' 1 3 →
2 3 3 2' 1 3' 3 → 2 3 3 2' 1 → 2 3 3 1 2 1' 2'

Observe that this termination problem is equivalent to the preceding confluence problem.

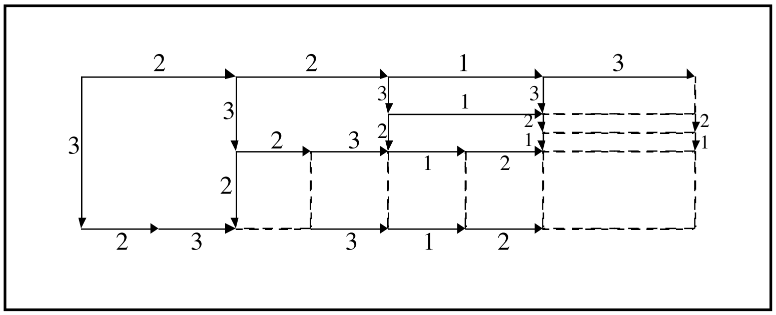


Figure 1.29

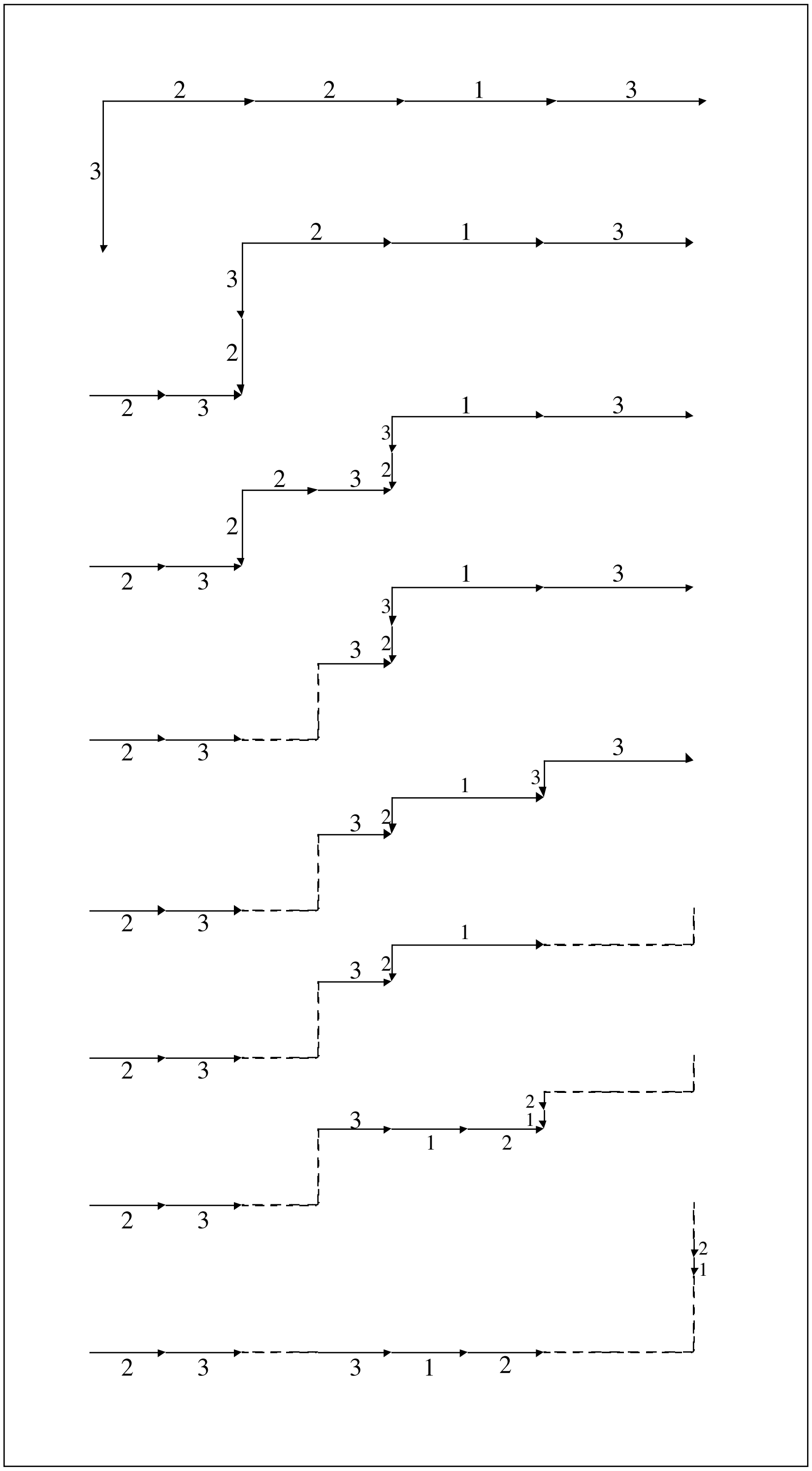


Figure 1.30

1.4.1. REMARK. Also in the case of confluence by decreasing diagrams we have termination of the corresponding rewrite rules, using in addition to the labels in the set I the inverses of the labels, corresponding to reverse rewrite steps. One can ask whether the decreasing hypothesis says something about the way of termination. H. Zantema observed (personal communication) that in this case we have in fact *simple* termination. For this notion, see Middeldorp and Zantema [94]. He also observed that in the present case of the non-decreasing tiles for braids, the corresponding termination is not simple termination.

A quite different way to reformulate an abstract confluence problem into an equivalent termination problem for a first-order TRS (modulo associativity of some operator) is as follows. Let x, y, z,... denote reduction sequences; the empty reduction sequence (containing zero steps) is ε. Concatenating x, y yields x·y, written as xy. Concatenation is associative. Furthermore, x/y denotes the *projection of* x *over* y; this is the lower side of the reduction diagram with upper side x and left side y (see Figure 4.8). This operation and the corresponding equations are originally due to J.-J. Lévy in the setting of λ-calculus and orthogonal term rewriting (see Barendregt [84]). See Table 2, where n, m = 1, 2, 3,... The first six rules are general and are just Lévy's equations with an orientation. The last three rule schemes are specific for the braid problem. To understand the rules, see Figure 4.8.

$(xy)/z \to (x/z)(y/(z/x))$
$z/(xy) \to (z/x)/y$
$\varepsilon/x \to \varepsilon$
$x/\varepsilon \to x$
$\varepsilon x \to x$
$x\varepsilon \to x$
$n/n \to \varepsilon$
$n/m \to n$ *if* $\lvert n-m \rvert \geq 2$
$n/m \to nm$ *if* $\lvert n-m \rvert = 1$
projection rules for confluence of braids

Table 2

1.4.2. EXAMPLE. (i) Consider the first six rules together with $1/2 \to \varepsilon$, $2/1 \to 1\cdot2$. Now 2/12 gives rise to an infinite reduction diagram as the first in Figure 3.1. Indeed, the 'projection rules' are non-terminating:

$$2/12 \to (2/1)/2 \to 12/2 \to (1/2)(2/(2/1)) \to\!\!\to \varepsilon(2/12) \to 2/12 \to \ldots$$

(ii) It is easy to show that the six general rules are terminating (by lexicographic path ordering), and that the critical pairs are convergent. Hence these six rules are also confluent.

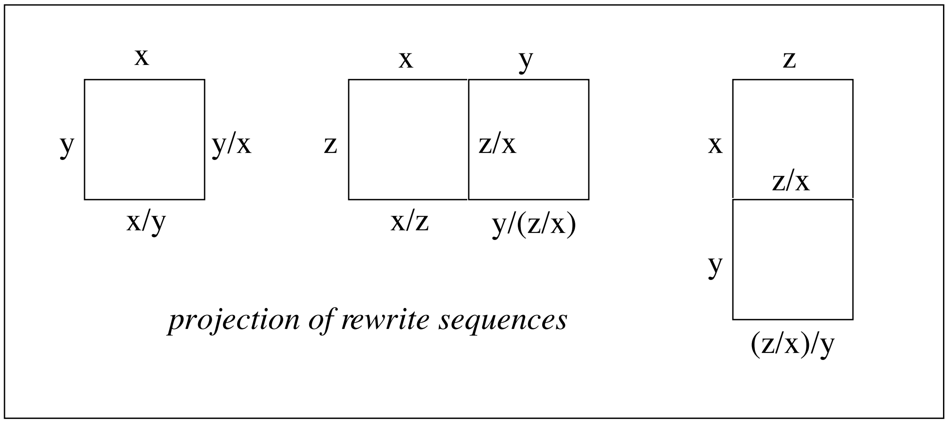


Figure 1.31

We now have three different but equivalent versions of the braid confluence problem:
(1) as confluence problem in the braid semi-group,
(2) as termination problem in the braid group,
(3) as termination problem of the projection rules in Table 2.
As to (1): here we mean the strong version (called CR^+ in Klop [80]) stating that the tiling procedure yields a completed, finite diagram.

As far as we know, there is no *direct* proof for any of the three versions. An indirect proof can be obtained as follows. In Garside [69] it is proved by lengthy combinatory manipulations that in the braid semi-group (so for positive braids) we have: $\forall u, v\ \exists x, y\ \ xu = yv$. Here xu = yv means that xu can be converted into yv by the defining equations 13 = 31, 232 = 323 etc. By a simple appeal to symmetry we therefore also have $\forall u,v\ \exists x, y\ \ ux = vy$. That is, we have confluence in the general sense, but not yet in the strong sense by the tiling procedure. Indeed, the converging x, y constructed in Garside [69] would be relatively long, and not the 'best possible' as would be obtained by the tiling procedure. However, this is easy to obtain now. With some effort, we can prove that if p = q in the braid semi-group, the diagram with diverging sides p, q will be completed by tiling into one

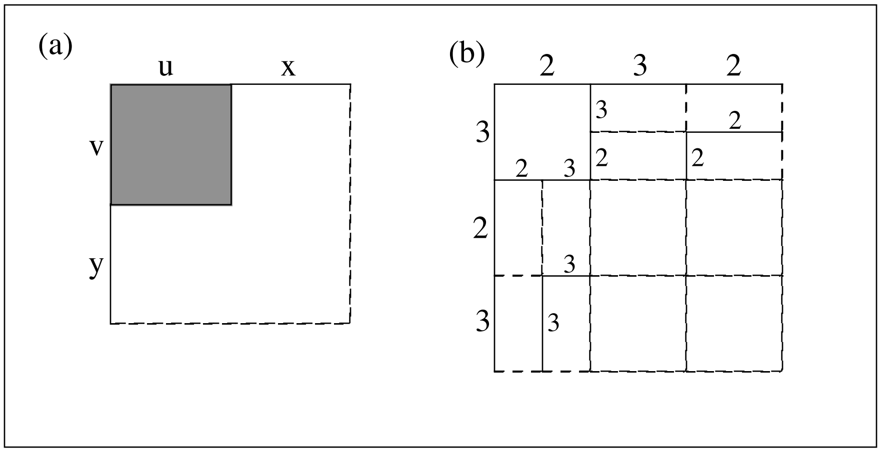


Figure 1.32

with empty converging sides. In fact we have that the following are equivalent:

(i) $p = q$ in the braid semi-group,
(ii) $q^{-1}p = \varepsilon$ in the braid group,
(iii) $p/q \twoheadrightarrow \varepsilon$ & $q/p \twoheadrightarrow \varepsilon$ in the projection TRS of Table 2.

Applying this on ux, vy with ux = vy we find a diagram as in Figure 4.9(a) with empty converging sides; and this diagram contains a subdiagram (shaded) with diverging sides u, v. This proves the strong, tiling version of confluence referred to in (1) above, and hence also termination as in (2), (3) above.

1.5. Exercises: Criteria for confluence.

1.5.1. EXERCISE (Rosen [73]). If $\langle A, \rightarrow_1, \rightarrow_2 \rangle$ is an ARS such that $\twoheadrightarrow_1 = \twoheadrightarrow_2$ and $\rightarrow_1$ is subcommutative, then $\rightarrow_2$ is confluent.

1.6.2. EXERCISE (Hindley [64]). Let $\langle A, (\rightarrow_\alpha)_{\alpha \in I} \rangle$ be an ARS such that for all $\alpha, \beta \in I$, $\rightarrow_\alpha$ commutes with $\rightarrow_\beta$. (In particular, $\rightarrow_\alpha$ commutes with itself.) Then the union $\rightarrow = \bigcup_{\alpha \in I} \rightarrow_\alpha$ is confluent.

(This proposition is sometimes referred to as the Lemma of Hindley-Rosen; see e.g. Barendregt [84], Proposition 3.3.5. For an application, see Exercise 2.2.10 and 2.2.11.)

1.6.3. EXERCISE (Hindley [64]). Let $\langle A, \rightarrow_1, \rightarrow_2 \rangle$ be an ARS. Suppose:
$\forall a, b, c \in A\ \exists d \in A\ (a \rightarrow_1 b\ \&\ a \rightarrow_2 c \Rightarrow b \twoheadrightarrow_2 d\ \&\ c \rightarrow_1^{\equiv} d)$. (See Figure 1.4a.) Then $\rightarrow_1, \rightarrow_2$ commute.

1.6.4. EXERCISE (Staples [75]). Let $\langle A, \rightarrow_1, \rightarrow_2 \rangle$ be an ARS. Suppose: $\forall a, b, c \in A\ \exists d \in A\ (a \rightarrow_1 b\ \&\ a \twoheadrightarrow_2 c \ \Rightarrow\ b \twoheadrightarrow_2 d\ \&\ c \twoheadrightarrow_1 d)$. (See Figure 1.4b.) Then $\rightarrow_1, \rightarrow_2$ commute.

1.6.5. EXERCISE (Rosen [73]). Let $\langle A, \rightarrow_1, \rightarrow_2 \rangle$ be an ARS.

DEFINITION: $\rightarrow_1$ *requests* $\rightarrow_2$ if $\forall a, b, c \in A\ \exists d, e \in A\ (a \twoheadrightarrow_1 b\ \&\ a \twoheadrightarrow_2 c \ \Rightarrow\ b \twoheadrightarrow_2 d\ \&\ c \twoheadrightarrow_1 e \twoheadrightarrow_2 d))$.

(See Figure 1.4c.) To prove: if $\rightarrow_1, \rightarrow_2$ are confluent and if $\rightarrow_1$ requests $\rightarrow_2$, then $\rightarrow_{12}$ is confluent.

1.6.6. EXERCISE (Rosen [73]). Let $\langle A, \rightarrow_1, \rightarrow_2 \rangle$ be an ARS such that $\rightarrow_2$ is confluent and: $\forall a, b, c \in A\ \exists d, e \in A\ (a \twoheadrightarrow_1 b\ \&\ a \rightarrow_2 c \ \Rightarrow\ b \twoheadrightarrow_2 d\ \&\ c \twoheadrightarrow_1 e \twoheadrightarrow_2 d)$. (See Figure 1.4d.) Then $\rightarrow_1$ requests $\rightarrow_2$.

1.6.7. EXERCISE (Staples [75]). Let $\langle A, \rightarrow_1, \rightarrow_2 \rangle$ be an ARS such that $\rightarrow_2$ is confluent and $\rightarrow_1$ requests $\rightarrow_2$. Let $\rightarrow_3$ be the composition of $\twoheadrightarrow_1$ and $\twoheadrightarrow_2$, i.e. $a \rightarrow_3 b$ iff $\exists c\ \ a \twoheadrightarrow_1 c \twoheadrightarrow_2 b$. Suppose moreover that $\forall a, b, c \in A\ \exists d \in A\ (a \twoheadrightarrow_1 b\ \&\ a \twoheadrightarrow_1 c \ \Rightarrow\ b \rightarrow_3 d\ \&\ c \rightarrow_3 d)$. Then $\rightarrow_{12}$ is confluent.

1.6.8. EXERCISE (Staples [75]) .

DEFINITION: In the ARS $\langle A, \rightarrow_1, \rightarrow_2 \rangle$ the reduction relation $\rightarrow_2$ is called a *refinement* of $\rightarrow_1$ if $\rightarrow_1 \subseteq \twoheadrightarrow_2$. If moreover $\forall a, b \in A\ \exists c \in A\ (a \twoheadrightarrow_2 b \ \Rightarrow\ a \twoheadrightarrow_1 c\ \&\ b \twoheadrightarrow_1 c)$, then $\rightarrow_2$ is a *compatible* refinement of $\rightarrow_1$.

Let in the ARS $\langle A, \rightarrow_1, \rightarrow_2 \rangle$ the reduction relation $\rightarrow_2$ be a refinement of $\rightarrow_1$. Prove that $\rightarrow_2$ is a compatible refinement of $\rightarrow_1$ iff $\forall a, b, c \in A\ \exists d \in A\ (a \rightarrow_2 b\ \&\ b \twoheadrightarrow_1 c \ \Rightarrow\ c \twoheadrightarrow_1 d\ \&\ a \twoheadrightarrow_1 d)$.

1.7.9. EXERCISE (Staples [75]). Let $\langle A, \rightarrow_1, \rightarrow_2 \rangle$ be an ARS where $\rightarrow_2$ is a compatible refinement of $\rightarrow_1$. Then: $\rightarrow_1$ is confluent iff $\rightarrow_2$ is confluent.

1.7.10. EXERCISE (Huet [80]). DEFINITION: Let $\langle A, \rightarrow \rangle$ be an ARS. Then $\rightarrow$ is called *strongly confluent* (see Figure 1.4e) if: $\forall a, b, c \in A\ \exists d \in A\ (a \rightarrow b\ \&\ a \rightarrow c \ \Rightarrow\ b \twoheadrightarrow d\ \&\ c \rightarrow^{\equiv} d)$. Prove that strong confluence implies confluence.

1.7.11. EXERCISE. Let $\langle A, (\rightarrow_\alpha)_{\alpha \in I} \rangle$ be an ARS such that for all $\alpha, \beta \in I$, $\rightarrow_\alpha$ commutes weakly with $\rightarrow_\beta$.

DEFINITION: (a) $\to_\alpha$ is *relatively terminating* if no reduction $a_0 \to a_1 \to a_2 \to \ldots$ (where $\to = \bigcup_{\alpha \in I} \to_\alpha$) contains infinitely many α-steps.

(b) $\to_\alpha$ *has splitting effect* if there are a, b, c, $\in$ A such that for every d $\in$ A and every $\beta \in I$ with a $\to_\alpha$ b, a $\to_\beta$ c, c $\twoheadrightarrow_\alpha$ d, b $\twoheadrightarrow_\beta$ d, the reduction b $\twoheadrightarrow_\beta$ d consists of more than one step.

To prove: if every $\to_\alpha$ ($\alpha \in I$) which has splitting effect is relatively terminating, then $\to$ is confluent. (Note that this strengthens Newman's Lemma.)

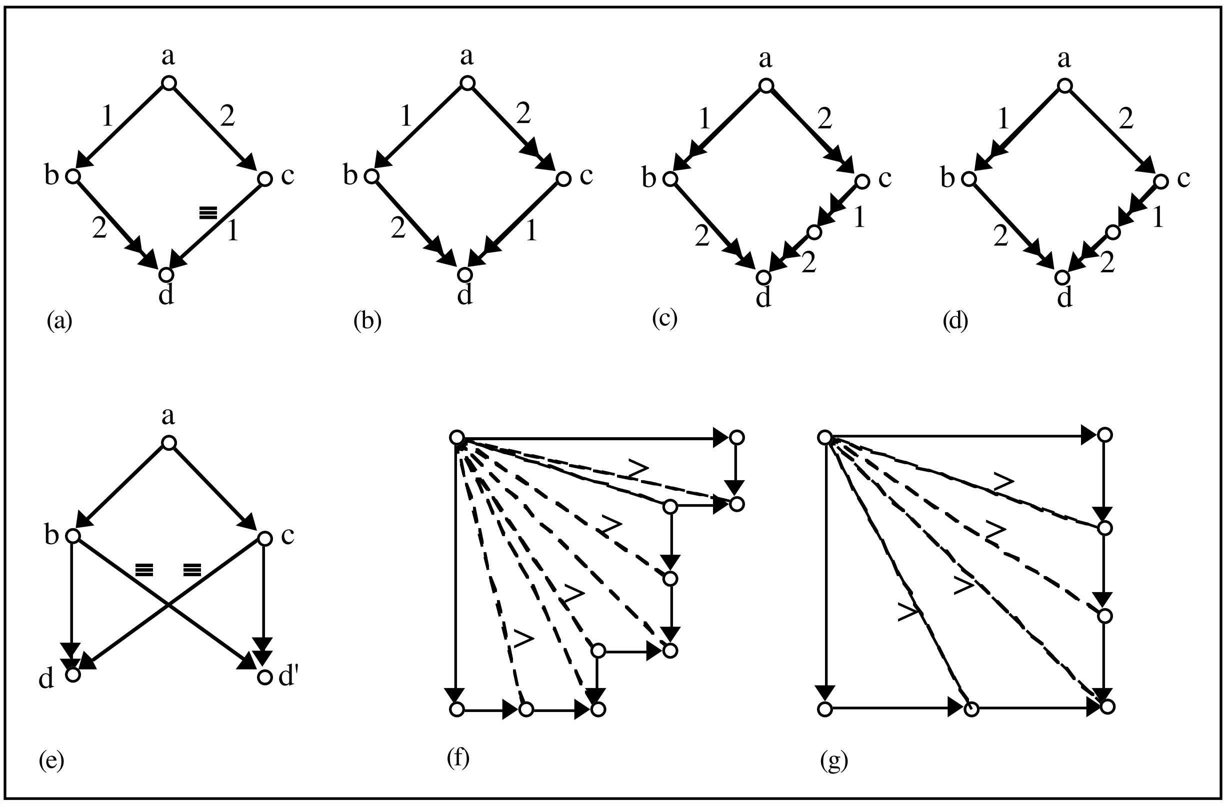


Figure 1.4

1.7.12. EXERCISE (Winkler & Buchberger [83]). Let $\langle A, \to \rangle$ be an ARS where the 'reduction' relation $>$ is a partial order and SN. (So $>$ is well-founded.) Suppose a $\to$ b implies a $>$ b. Then the following are equivalent:

(a) $\to$ is confluent,

(b) whenever a $\to$ b and a $\to$ c, there is a $\to$-conversion $b \equiv d_1 \leftrightarrow d_2 \leftrightarrow \ldots \leftrightarrow d_n \equiv c$ (for some $n \geq 1$) between b, c such that $a > d_i$ ($i = 1,\ldots,n$). Here each $\leftrightarrow$ is $\to$ or $\leftarrow$. (See Figure 1.4f.)

(Note that this strengthens Newman's Lemma.)

1.7.13. EXERCISE (Klop [80]). Let $\mathcal{A} = \langle A, \to \rangle$ be an ARS. Let $B \subseteq A$. Then $\mathcal{B}$ is *cofinal* in $\mathcal{A}$ if $\forall a \in A\ \exists b \in B\ \ a \twoheadrightarrow b$. Furthermore, $\mathcal{A}$ is said to have the *cofinality property* (CP) if in every reduction graph $\mathcal{G}(a)$, $a \in A$, there is a (possibly infinite) reduction sequence $a \equiv a_0 \to a_1 \to \ldots$ such that $\{a_n \mid n \geq 0\}$ is cofinal in $\mathcal{G}(a)$.

(i) Prove that for *countable* ARSs: $\mathcal{A}$ is CR $\Leftrightarrow$ $\mathcal{A}$ has CP.

(ii) Show that the condition of countability cannot be missed. (See Solutions.)

SOLUTION (from Klop [80]).

1.7.14. EXERCISE. Let $\langle A, \to \rangle$ be an ARS. For well-founded partial orderings $>$ on A we define the following properties.

(i) $\blacktriangle(>)$: whenever $b \leftarrow a \to c$ there is a $\to$-conversion $b \equiv d_0 \leftrightarrow d_1 \leftrightarrow \ldots \leftrightarrow d_n \equiv c$ (for some $n \geq 0$) such that $a > d_i$ for $i = 1,\ldots,n-1$. (So, in case $n = 0$ or 1 there is no partial order requirement.) Here each $\leftrightarrow$ is $\to$ or $\leftarrow$. (See Figure 1.4(f).)

(ii) $\blacklozenge(>)$: whenever $b \leftarrow a \to c$ there is a $\to$-conversion $b \equiv d_0 \to d_1 \to \ldots \to d_n \leftarrow d_{n+1} \leftarrow \ldots \leftarrow d_{n+m} \equiv c$ (for some $n, m \geq 0$) such that $a > d_i$ for $i = 1,\ldots,n+m-1$. (See Figure 1.4(g).)

(iii) The weakened properties $\blacktriangle^-(>)$ and $\blacklozenge^-(>)$ are obtained as follows (in addition to the clauses of the definition of $\blacktriangle(>)$ and $\blacklozenge(>)$, respectively): if for the diverging steps $b \leftarrow a \to c$ we can find an element d such that $b \to d \leftarrow c$, we do not require $a > d$ but only $a \geq d$.

(iv) We now define that the ARS $\langle A, \to \rangle$ has property $\blacktriangle$ if for some well-founded partial ordering $>$ on A, we have $\blacktriangle(>)$. Likewise for property $\blacklozenge$, and for $\blacktriangle^-$ and $\blacklozenge^-$.

(v) Let $\#\colon A \to \mathrm{ORD}$ be an ordinal assignment to elements of A. Analogous to (i-iii) above we define $\blacktriangle_o(\#)$, $\blacklozenge_o(\#)$, $\blacktriangle_o^-(\#)$, $\blacklozenge_o^-(\#)$, by replacing $a > b$ by $\#(a) > \#(b)$ and $a \geq b$ by $\#(a) \geq \#(b)$. (The latter $>$, $\geq$ denote the ordering between ordinals.) Analogous to (iv) we define $\blacktriangle_o$, $\blacklozenge_o$, $\blacktriangle_o^-$, $\blacklozenge_o^-$. All eight properties are criteria for confluence; somewhat surprisingly they are not all equivalent.

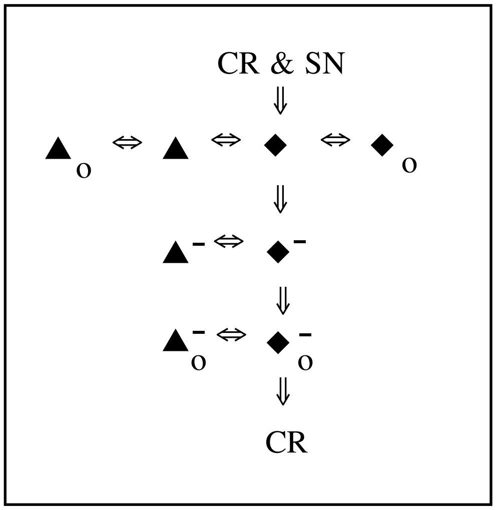


Figure 1.5

(vi) Prove that implications and equivalences hold as in Figure 1.5. Note that this entails SN $\Rightarrow$ ($\blacktriangle \Leftrightarrow$ CR), which is in fact Exercise 1.7.12. For an application of the confluence criterion $\blacklozenge_o^-$, see the next exercise.

Prove that the implications are strict, by considering the examples in Figure 1.6. Show that the ARS in (a) has properties $\neg\blacklozenge$, $\blacklozenge^-$, $\blacklozenge_o^-$. The ARSs in (e) and (f) satisfy $\neg\blacklozenge$, $\neg\blacklozenge^-$, $\blacklozenge_o^-$. The other three ARSs satisfy property $\blacklozenge$.

1.7.15. EXERCISE (Geser [89]). This exercise reformulates and slightly generalizes Exercise 1.7.11. Let $\langle A, \rightarrow_\alpha, \rightarrow_\beta \rangle$ be an ARS.

> **DEFINITION:** α/β ("α modulo β") is the reduction relation $\beta^*\alpha\beta^*$. So a $\rightarrow_{\alpha/\beta}$ b iff there are c, d such that a $\twoheadrightarrow_\beta$ c $\rightarrow_\alpha$ d $\twoheadrightarrow_\beta$ b.

Note that α is relatively terminating (in the sense of Exercise1.7.11) iff α/β is SN.

> **DEFINITION**. β is called *nonsplitting* (with respect to $\alpha \cup \beta$) if

$$\forall a,b,c \in A\ \exists d \in A\ (a \rightarrow_\beta b\ \&\ a \rightarrow_{\alpha\cup\beta} c \ \Rightarrow\ c \twoheadrightarrow_{\alpha\cup\beta} d\ \&\ b\ (\rightarrow_{\alpha\cup\beta})^{\equiv} d\ ).$$

Prove: If α/β is SN, α is WCR, and β is non-splitting, then $\alpha \cup \beta$ is confluent.

(Hint: Note that the transitive closure $(\alpha/\beta)^+$ is a well-founded partial ordering. This gives rise to an ordinal assignment to elements of A, such that a $\rightarrow_\alpha$ b $\Rightarrow$ #(a) > #(b) and a $\rightarrow_\beta$ b $\Rightarrow$ #(a) $\geq$ #(b). Now show that $\blacklozenge_o^-$ as in the previous exercise (vi) holds.)

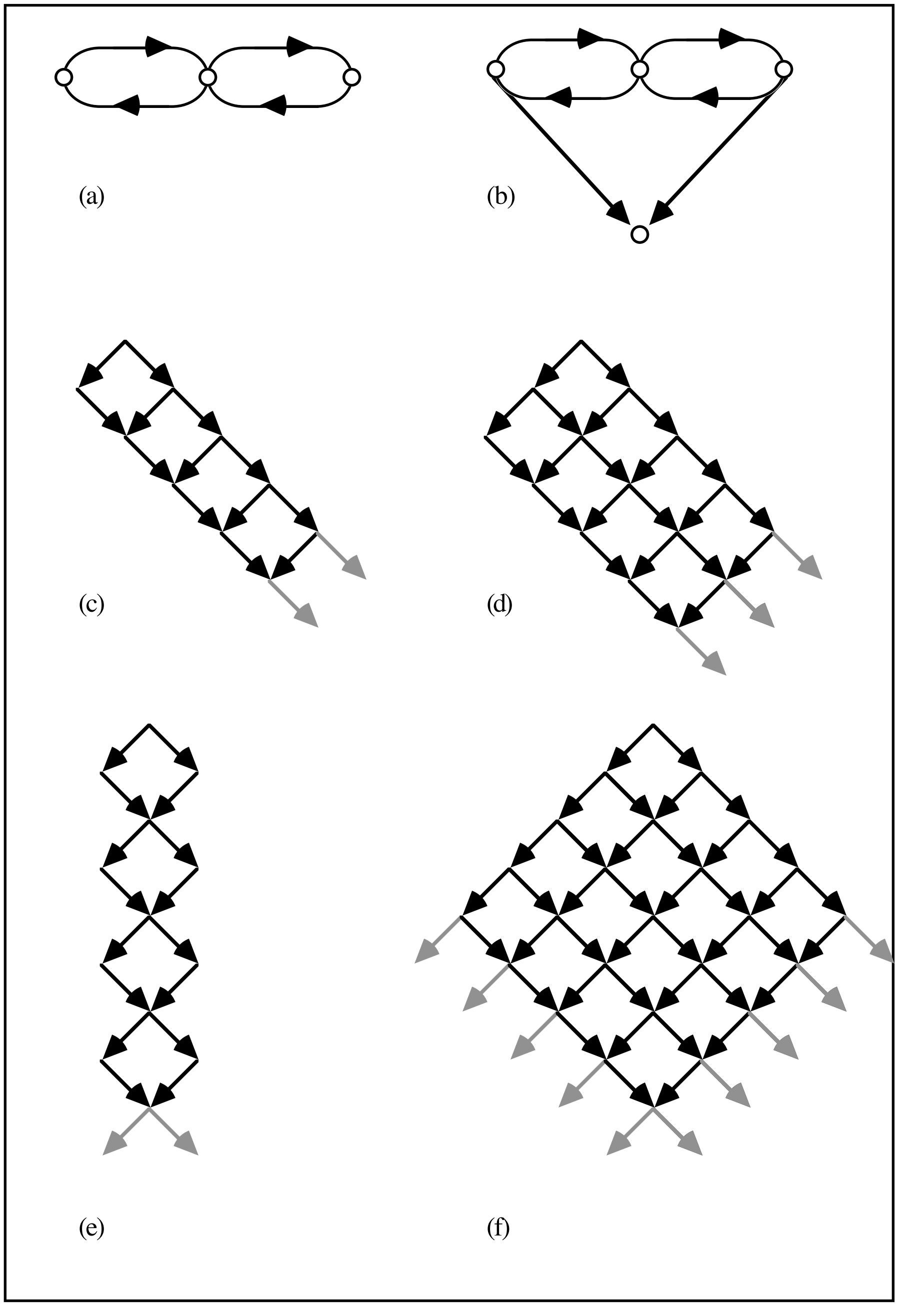


Figure 1.6

1.7.16. EXERCISE (Curien & Ghelli [90]). Let $\mathcal{A} = \langle A, \rightarrow_\alpha \rangle$ and $\mathcal{B} = \langle B, \rightarrow_\beta \rangle$ be ARSs. Suppose:

(i) $\mathcal{B}$ is confluent,
(ii) $\mathcal{A}$ is WN (weakly normalizing).
Moreover, let φ: A → B be a map such that:
(iii) $a \to_{\alpha} a' \Rightarrow \varphi(a) =_{\beta} \varphi(a')$, for all $a, a' \in A$,
(iv) φ translates α-normal forms into β-normal forms,
(v) φ is injective on α-normal forms.

Then $\mathcal{A}$ is confluent.

1.7.17. EXERCISE (De Bruijn [78]). Let $\mathcal{A} = \langle A, (\to_n)_{n \in I} \rangle$ be an ARS with I a partial order. Then, for a, b ∈ A, $a \twoheadrightarrow_{<n} b$ means that there is a sequence of reduction steps from a to b, each reduction step having index < n. Analogously, $a \twoheadrightarrow_{\leq n} b$ is defined. Furthermore, $\to_n^{\equiv}$ is the reflexive closure of $\to_n$. Prove:

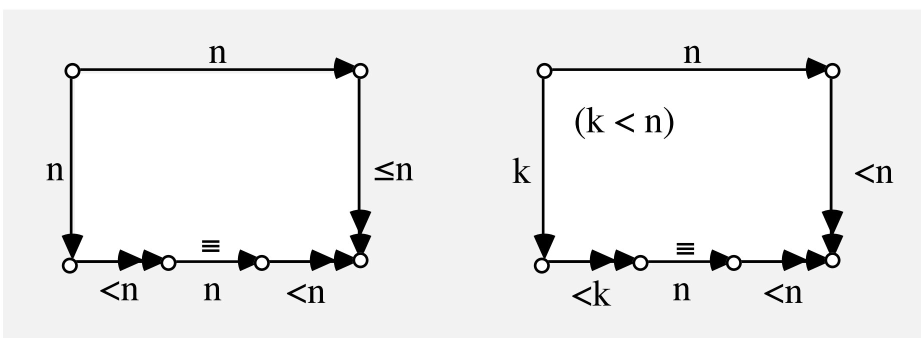


Figure 1.7

LEMMA. *Let* $\mathcal{A} = \langle A, (\to_n)_{n \in I} \rangle$ *be an ARS with* I *a well-founded linear order. Suppose that*

(i) $\forall a, b, c, n\, \exists d, e, f\, (a \to_n b \;\&\; a \to_n c \Rightarrow b \twoheadrightarrow_{\leq n} f \;\&\; c \twoheadrightarrow_{<n} d \to_n^{\equiv} e \twoheadrightarrow_{<n} f)$, *and*

(ii) $\forall a, b, c, n, k\, \exists d, e, f\, (k < n \;\&\; a \to_n b \;\&\; a \to_k c \Rightarrow b \twoheadrightarrow_{<n} f \;\&\; c \twoheadrightarrow_{<k} d \to_n^{\equiv} e \twoheadrightarrow_{<n} f)$.

(See Figure 1.7.) Then $\mathcal{A}$ *is confluent.*

1.6. Exercises: Criteria for Strong Normalization.

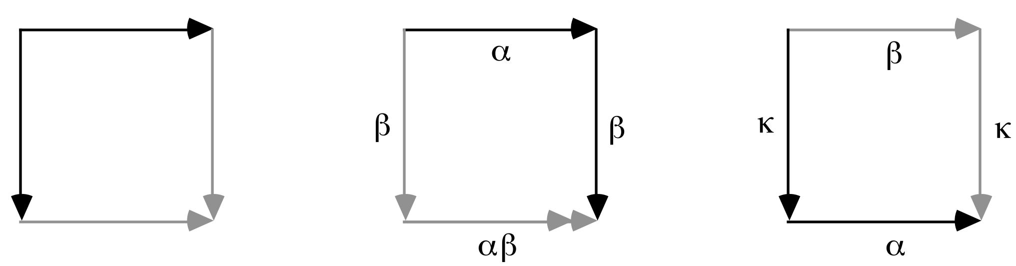
1.8.1. EXERCISE (Newman [42]). Let WCR^1 be the following property of ARSs $\langle A, \to \rangle$:

$\forall a,b,c \in A\ \ \exists d \in A\ \ (c \leftarrow a \rightarrow b\ \&\ b \neq c\ \Rightarrow\ c \rightarrow d \leftarrow b)$. (See Figure 1.8a.) Prove that WCR^1 & WN $\Rightarrow$ SN, and give a counterexample to the implication $WCR^{\leq 1}$ & WN $\Rightarrow$ SN.

1.8.2. EXERCISE (Bachmair & Dershowitz [86]). Let $\langle A, \rightarrow_\alpha, \rightarrow_\beta \rangle$ be an ARS such that $\forall a,b,c \in A\ \ \exists d \in A\ \ (a \rightarrow_\alpha b \rightarrow_\beta c\ \Rightarrow\ a \rightarrow_\beta d \twoheadrightarrow_{\alpha\beta} c)$. (In the terminology of Bachmair & Dershowitz [86]: β *quasi-commutes over* α.) (See Figure 1.8b.) Prove that β/α is SN iff β is SN. (For the definition of β/α, see Exercise 1.7.15.)

1.8.3. EXERCISE (Klop [80]). Let $\mathcal{A} = \langle A, \rightarrow_\alpha \rangle$ and $\mathcal{B} = \langle B, \rightarrow_\beta \rangle$ be ARSs. Let $\iota: A \rightarrow B$ and $\kappa: B \rightarrow A$ be maps such that
(i) $\kappa(\iota(a)) = a$ for all $a \in A$,
(ii) $\forall a,a' \in A\ \ \forall b \in B\ \ \exists b' \in B\ (b \rightarrow_\kappa a \rightarrow_\alpha a'\ \Rightarrow\ b \rightarrow_\beta b' \rightarrow_\kappa a')$ (Reductions in $\mathcal{A}$ can be 'lifted' to $\mathcal{B}$.) See Figure 1.8c.

Prove that $\mathcal{B}$ is SN implies that $\mathcal{A}$ is SN.



(a) (b) (c)

Figure 1.8

1.8.4.EXERCISE. (Geser [89]) Let $\langle A, \rightarrow_\alpha, \rightarrow_\beta \rangle$ be an ARS with two reduction relations α, β such that $\alpha \cup \beta$ is transitive. Then:

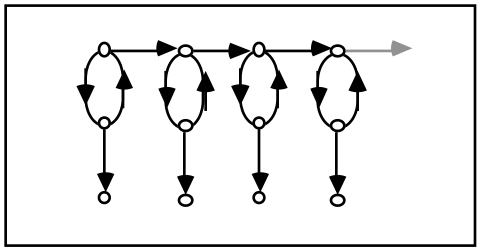
$$\alpha \cup \beta \text{ is SN} \Leftrightarrow \alpha \text{ is SN and } \beta \text{ is SN.}$$

(Hint: use Ramsey's Theorem, see Preliminaries.)

1.7. Exercises: Other properties of ARSs

1.9.1.EXERCISE. Define: $\mathcal{A}$ (or $\rightarrow$) has the *unique normal form property with respect to reduction* ($UN^{\rightarrow}$) if $\forall a,b,c \in A\ (a \twoheadrightarrow b\ \&\ a \twoheadrightarrow c\ \&\ b,c$ are normal forms $\Rightarrow\ b \equiv c)$. Show that UN $\Rightarrow$ $UN^{\rightarrow}$, but not conversely.

1.9.2.EXERCISE. Find a counterexample to the implication WCR & WN $\Rightarrow$ Ind.



1.9.3.EXERCISE. Let $\mathcal{A} = \langle A, \rightarrow \rangle$ be an ARS. Define: A is *consistent* if not every pair of elements in A is convertible. Note that if $\mathcal{A}$ is confluent and has two different normal forms, $\mathcal{A}$ is consistent. Further, let $\mathcal{A} = \langle A, \rightarrow_\alpha \rangle$, $\mathcal{B} = \langle B, \rightarrow_\beta \rangle$ be ARSs such that $A \subseteq B$. Then we define: $\mathcal{B}$ is a *conservative* extension of $\mathcal{A}$ if $\forall a,a' \in A$ ($a =_\beta a' \Leftrightarrow a =_\alpha a'$). Note that a conservative extension of a consistent ARS is again consistent. Further, note that a confluent extension $\mathcal{B}$ of $\mathcal{A}$ is conservative.

1.9.4. EXERCISE. (i) Let $\mathcal{A} = \langle A, \rightarrow \rangle$ be a countable ARS with decidable syntactic equality ($\equiv$). Moreover, let $\rightarrow$ be a recursively enumerable relation and let the set of normal forms of $\mathcal{A}$ be decidable. Show that if A is CR and WN, convertibility (=) in $\mathcal{A}$ is decidable.
(ii) Give an example of an ARS $\mathcal{A} = \langle \mathbb{N}, \rightarrow \rangle$ (where $\mathbb{N}$ is the set of natural numbers) such that $\rightarrow$ is decidable, but NF($\mathcal{A}$), the set of normal forms of $\mathcal{A}$, is undecidable.

1.9.5. EXERCISE (V. van Oostrom):
Give an example of an ARS which is WCR, $UN^{\rightarrow}$, but not $UN^{=}$
(Solution: Roel's counterexample, mixed with Hindley's counterexample.)

2

Decreasing Diagrams

As seen in Topic 1, we can infer CR from WCR when given the additional assumption of SN (Newman's Lemma). Also in this topic we will infer CR from WCR, this time given extra information on the nature of WCR, that is, on the form of the 'elementary diagrams' or e.d.'s that WCR provides. In the braid confluence problem we already encountered the procedure of tiling with e.d.'s to obtain a confluent diagram. We will now look in detail at this 'tiling game', starting with Huet's strong confluence lemma, in a sequence of introductory examples.

2.1 EXAMPLE. (i) DEFINITION. For an ARS $\mathcal{A} = \langle A, \rightarrow \rangle$ we define: $\rightarrow$ is *strongly confluent* if

$$\forall a,b,c \in A \;\; \exists d \in A \;\; (b \leftarrow a \rightarrow c \;\Rightarrow\; c \twoheadrightarrow d \leftarrow^{\equiv} b) \text{ (See Fig. 2.1(a))}$$

(Here $\rightarrow^{\equiv}$ is the relexive closure of $\rightarrow$, so b $\rightarrow^{\equiv}$ d is zero or one step.)

(ii) LEMMA (Huet [80]). *Let* $\mathcal{A}$ b*e strongly confluent. Then* $\mathcal{A}$ *is CR.*

The proof is simple. The assumption of strong confluence provides us with elementary diagrams (e.d.'s) as in Figure 2.1(a), which can be used to obtain CR as suggested in the diagram in Figure 2.1(b), where we profit from the fact that "splitting" occurs in the direction of our choice.
(This is so, because the quantification over a,b,c implicit in Figure 2.1(a) is universal, so we can mirror the e.d. in that figure around the main diagonal.)

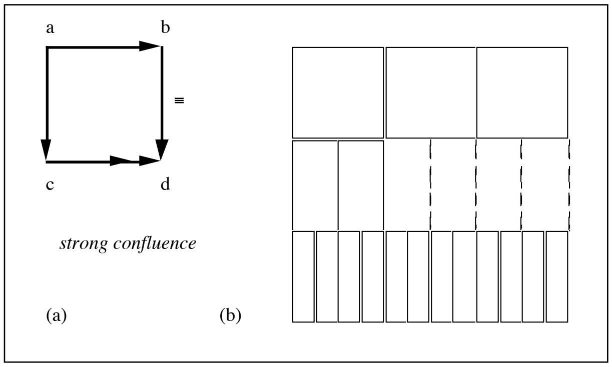


Figure 2.1

2.2 EXAMPLE. When splitting of e.d.'s would occur in both directions, our "diagram chase" to obtain confluence may very well fail: given e.d.'s of the form as in Figure 2.2(a), so corresponding to the WCR assumption

$$\forall a,b,c \in A\ \exists d,e,f \in A\ (c \leftarrow a \rightarrow b\ \Rightarrow\ c \rightarrow d \rightarrow e \leftarrow f \leftarrow b),$$

we may fail in our attempt to construct a confluent diagram by tiling; see Figure 2.2(b), where the diagram construction of diagram $\mathcal{D}$ falls in the trap of an infinite regress.

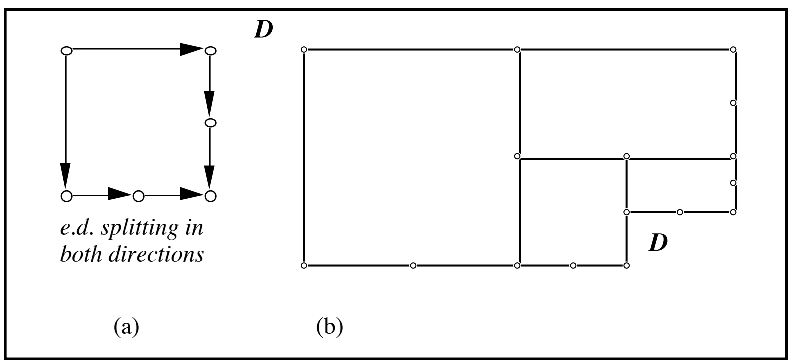


Figure 2.2

2.3. EXAMPLE. This tiling game is even more interesting when dealing with more than one reduction relation. Suppose we have two reduction relations, labeled (or indexed) with 1,2, and that we have for their union $\rightarrow_{12}$ WCR in the form of the e.d.'s as in Figure 2.3.

Question: does CR hold for $\rightarrow_{12}$?

Answer: No; for we may have a situation as in Figure 2.3, lower diagram.

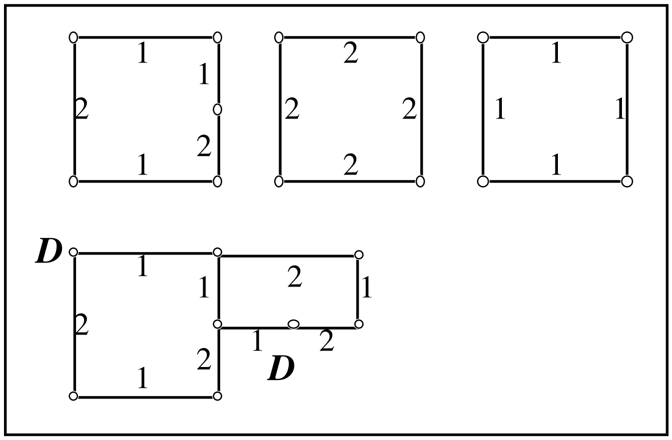


Figure 2.3

2.4. EXAMPLE. However, if we change the e.d.'s of Example 2.3 slightly as in Figure 2.4 we do have CR!

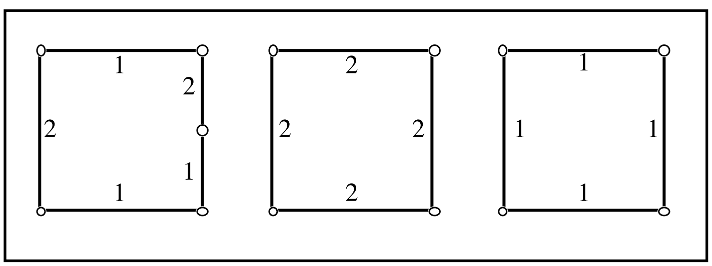


Figure 2.4

2.5. EXAMPLE. Question: do we have CR given the e.d.'s in Figure 2.5? This is not at all easy to see. The answer is yes, as will be clear at the end of this topic.

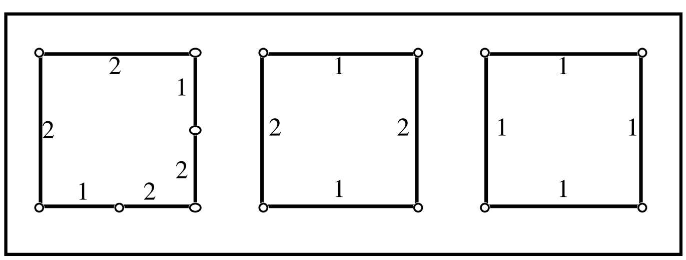


Figure 2.5

2.6. Reduction diagrams. We will now be more precise about elementary diagrams. They are of the following shapes; see Figure 2.6.

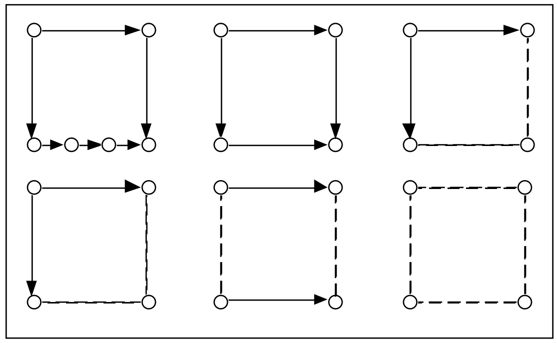


Figure 2.6

They are the 'atomic' or basic building blocks for constructing reduction diagrams. A non-trivial elementary diagram consists of two diverging steps (arrows), joined by two sequences of steps of arbitrary length. Note that in the e.d.'s we may use empty sides (the dashed sides, in some figures shaded), to keep matters orthogonal. This gives rise to some trivial e.d.'s as in the lower part of Figure 2.6. The e.d.'s are used as scalable 'tiles' with the intention to obtain a completed reduction diagram as in Figure 2.7. Usually we will forget the direction of the arrows (second picture in Figure 2.7): they always are from left to right, or downwards (except the empty 'steps' that have no direction).

2.7. Indexed Abstract Reduction Systems. In this topic we will consider an Abstract Reduction System (ARS) $\mathcal{A}$, equipped with a collection of rewrite or reduction relations $\to_\alpha$, indexed by some set I: $\mathcal{A} = \langle A, (\to_\alpha)_{\alpha \in I} \rangle$. The index set I is in this topic always a *well-founded partial order*. In examples, we will use the set of natural numbers with the usual ordering as index set. The union of the rewrite relations $\to_\alpha$ will be $\to$. We use the notation $\twoheadrightarrow$ for the transitive-reflexive closure of $\to$.

2.8. Multisets. We will use multisets over the index set I, together with the multiset ordering induced by that of I. It is well-known that this is again a well-founded partial order. (Furthermore, if I is a total order, then the p.o. of multisets over I is again total.)

We will use the notations: multiset ordering $\geq_\mu$, strict multiset ordering $>_\mu$, multiset union $\cup$.

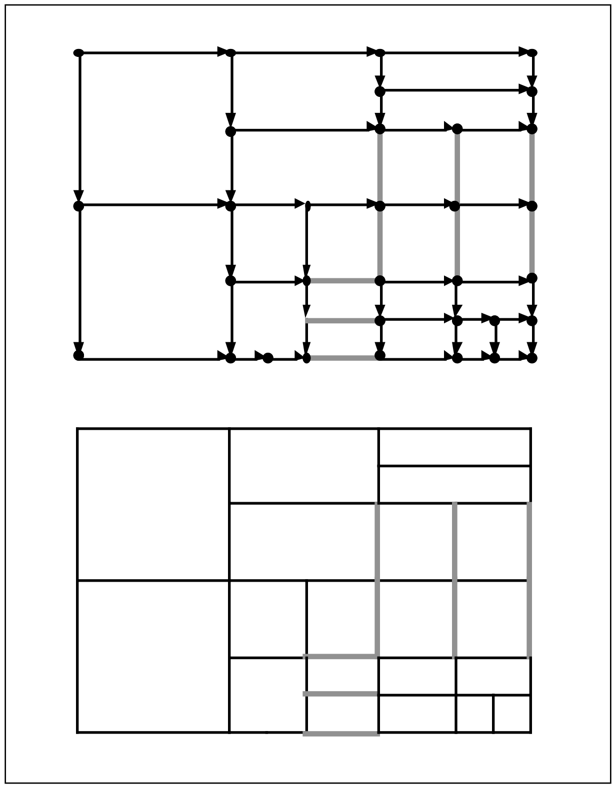


Figure 2.7

We will show somewhat more structure on the multiset p.o. If we have X $\geq_\mu$ Y, there is a 'descendant' relation between the elements of X and Y. Some elements of X are 'preserved' in Y: this is indicated by heavy arrows (see Figure 2.8). Some elements of X will be replaced by some elements in Y that are strictly smaller (in the p.o. I); this is indicated by light arrows. Heavy arrows cannot split, light arrows can. From an element of X also zero light arrows can exit: that element just disappears. (E.g. the '1' in Figure 2.8.) A descendant relation for X $\geq_\mu$ Y by means of 'multiset arrows' need not be unique, e.g. the pair of multisets in Figure 2.8 admits several other descendant relations.

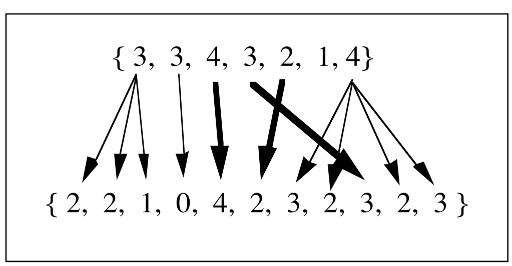
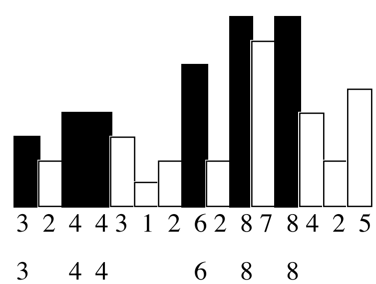


Figure 2.8

(Actually in the present treatment of this topic we will suppress all detailed proofs and therefore not really use this extra descendant structure for X $\geq_\mu$ Y, but mention of these heavy and light tracing arrows anticipates their use in the next topic.)

2.9. Monotonic filtering. We start with an important definition. Given a tuple σ of natural numbers, *filter*(σ) is the tuple obtained by 'reading' σ from left-to-right, removing the elements that are less than what was already encountered, and taking the tuple of the remaining elements. See example in Figure 2.9. Another operation on tuples is *multiset*; it yields the corresponding multiset. In the sequel we will be especially interested in *multiset* (*filter*(σ)).



monotonic filtering of string of natural numbers

Figure 2.9

2.10. Decreasing diagrams. Before defining what a decreasing diagram is, we need the notion of 'norm of a reduction sequence' in the ARS with indexed rewrite relations. This will be a tuple of natural numbers (in general, elements of I). Par abus de langage, we will also denote reduction sequences with σ, τ. If σ is a reduction sequence, *label*(σ) is the string of indexes of consecutive reduction steps in σ. Single steps will be denoted by α, β. So *label*(α) is the index of the step α.

2.10.1. DEFINITION.
(i) Let σ be a reduction sequence. Then |σ|, the *norm* of σ, is *multiset*(*filter*(*label*(σ))).
(ii) The norm of two diverging reductions σ, τ is $|\sigma| \cup |\tau|$. (Figure 2.10.)

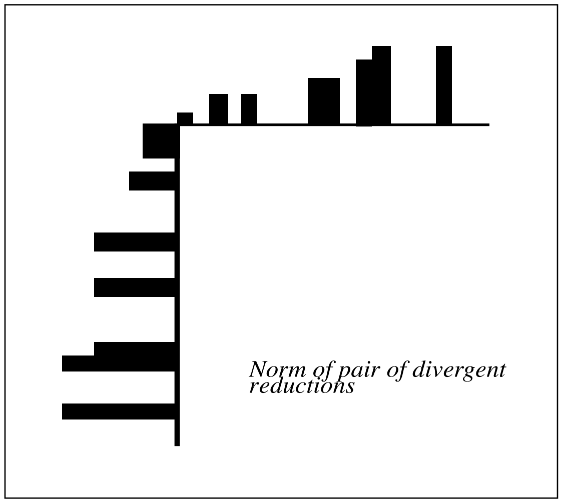


Figure 2.10

2.10.2. DEFINITION. Let σ: a $\twoheadrightarrow$ b, τ: a $\twoheadrightarrow$ c, σ': c $\twoheadrightarrow$ d, τ': b $\twoheadrightarrow$ d be reductions forming the reduction diagram $\mathcal{D}$ with corners a,b,c,d. (Figure 2.11.)

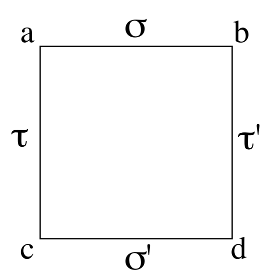


Figure 2.11

Then $\mathcal{D}$ is a *decreasing diagram*, if

$$|\sigma| \cup |\tau| \ \geq_\mu \ |\sigma{\cdot}\tau'| \text{ and}$$
$$|\sigma| \cup |\tau| \ \geq_\mu \ |\tau{\cdot}\sigma'|.$$

Note: we merely require $\geq_\mu$, not $>_\mu$!

2.11. Decreasing elementary diagrams. We will now see what the decreasingness condition means for elementary diagrams. Some consideration shows readily that decreasing e.d.'s have the following shape, as in Figure 2.12.
Explanation: Given two diverging steps a $\to_n$ b and a $\to_m$ c with indices n, m there is a common

reduct d such that

$$b \twoheadrightarrow_{<n} \cdot \rightarrow^{\equiv}_{m} \cdots \twoheadrightarrow_{<n \text{ or } <m} d \quad \text{and dually}$$

$$c \twoheadrightarrow_{<m} \cdot \rightarrow^{\equiv}_{n} \cdots \twoheadrightarrow_{<n \text{ or } <m} d$$

So from b we take some steps with indices < n, followed by 0 or 1 step with index m, followed by some steps with index < n or < m, with result d. Dually, from c we have a reduction to d as indicated.

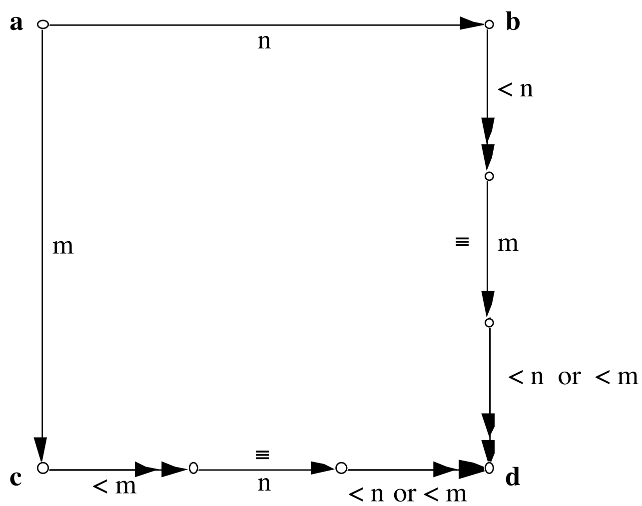


Figure 2.11

2.11.1. EXAMPLE. (i) So, we have examples as in Figure 2.13 of some decreasing and non-decreasing e.d.'s. (Note that the first e.d., upper-left, is an e.d. encountered in the braids confluence problem; so confluence of braids cannot be proved by an appeal on the theorem in this topic about confluence by decreasing diagrams!)

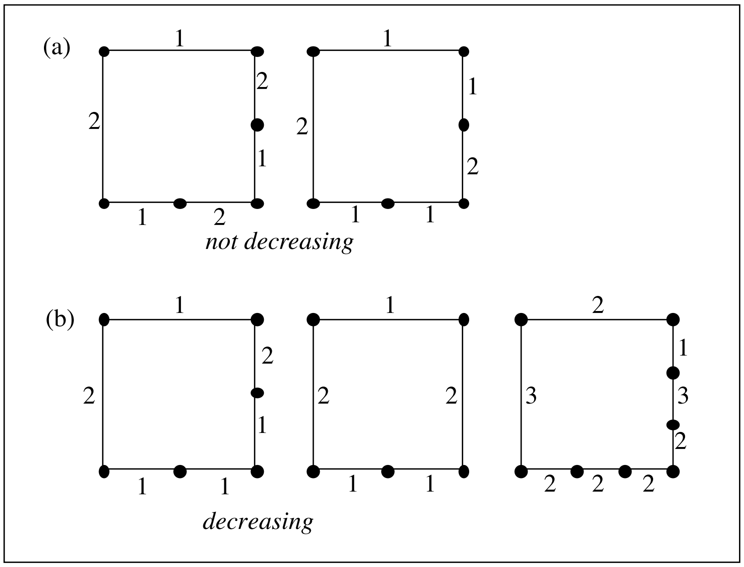


Figure 2.13

(ii) The e.d. in Example 2.2, Fig. 2.2 is not decreasing; of the e.d.'s in Fig.2.3 the first one is not decreasing, the other two are; the e.d.'s in Fig. 2.4 are decreasing; the e.d.'s in Fig. 2.5 are decreasing.

Now we will mention the two important properties of decreasing diagrams that give confluence. The first is indicated in Figure 2.14: pasting preserves decreasingness.

2.12. PROPOSITION. *Let two decreasing diagrams be joined as in Figure 2.14. Then the resulting diagram is again decreasing.*

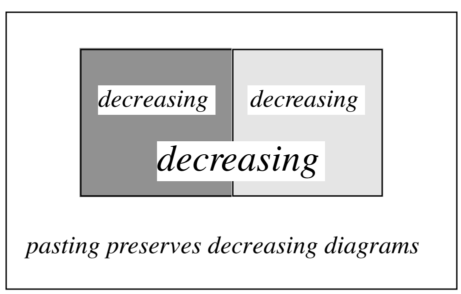


Figure 2.14

The second important property is indicated in Figure 2.15: inserting a decreasing diagram in a pair of co-initial reductions reduces the norm of the resulting pair of co-initial reductions.

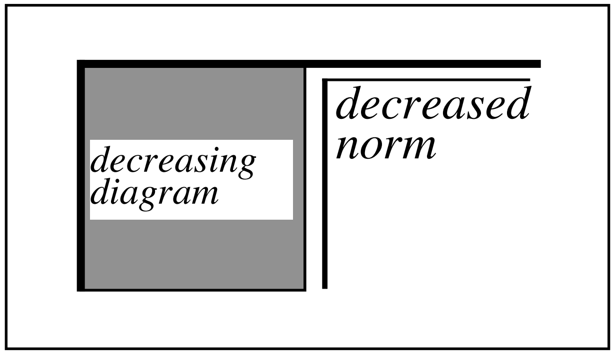


Figure 2.15

2.13. PROPOSITION. *Let a decreasing diagram be inserted as in Figure 2.15 into a pair of diverging reductions. Then the resulting pair of diverging reductions has a smaller norm.*

Finally, we can combine the two important properties to yield a proof of confluence, based on well-founded induction. See Figure 2.16.

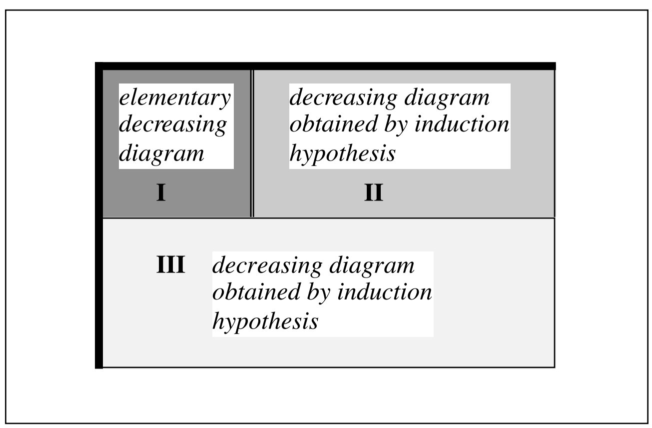


Figure 2.16

Let us give this final argument a bit more explicitly. See Fig. 2.17. The original norm is norm $(\beta.\tau, \alpha.\sigma) = |\beta.\tau| \cup |\alpha.\sigma|$. The induction hypothesis (IH) states that for all pairs of divergent reductions with smaller norm, tiling with decreasing e.d.'s succeeds. We have $|\beta| <_\mu |\beta.\tau|$, so norm $(\beta, \alpha.\sigma) <_\mu$ norm $(\beta.\ \tau, \alpha.\sigma)$. Now norm $(\beta', \sigma) <_\mu$ norm $(\beta, \alpha.\sigma) <_\mu$ norm $(\beta.\ \tau, \alpha.\sigma)$. Part I + II is again decreasing by Proposition 2.12. Hence (Proposition 2.13) norm $(\tau, \alpha'.\sigma') <_\mu$ norm $(\beta.\ \tau, \alpha.\sigma)$. Now IH yields part III.

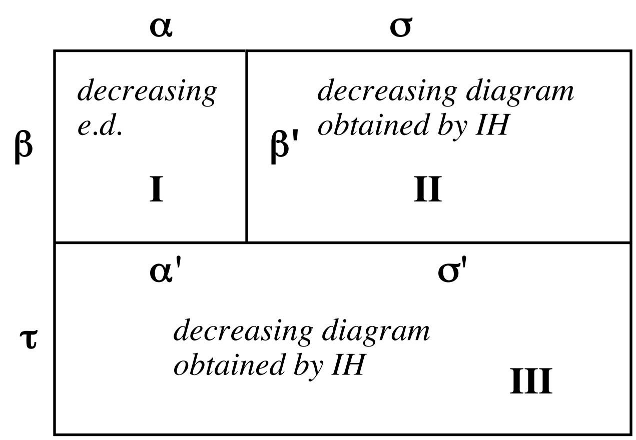


Figure 2.17

2.14. THEOREM (De Bruijn - Van Oostrom)
Every ARS with reduction relations indexed by a well-founded partial order I*, and satisfying the decreasing criterion for its e.d.'s, is confluent.*

2.15. REMARK. The unpublished note De Bruijn [78] is the first appearance of this theorem. There an asymmetrical version of the notion of decreasing elementary diagram is given. The notion of 'decreasing' as presented in this section was not present there and appears in Van Oostrom [94, 94a]. In Bezem et al. [96] the theorem is proved using the notion of 'trace-decreasing' which is slightly stronger than decreasing.

2.16. REMARK. The question arises how strong this method of decreasing diagrams is. In a way, it is best possible, at least for countable ARSs, since there is the following 'completeness' result:
Define an ARS $\mathcal{A} = \langle A, \rightarrow \rangle$ to have the property DCR (decreasing Church-Rosser), if there is an indexed ARS $\mathcal{A}^I = \langle A, (\rightarrow_\alpha)_{\alpha \in I} \rangle$ with rewrite relations $(\rightarrow_\alpha)_{\alpha \in I}$, such that $\mathcal{A}^I$ has decreasing e.d.'s with respect to some well-founded order on I, and such that the union of the rewrite relations $\rightarrow_\alpha$ is $\rightarrow$. So we have seen above that DCR $\Rightarrow$ CR. Now we have:

THEOREM (Van Oostrom [94]. *For countable ARSs*: DCR $\Leftrightarrow$ CR.

The proof, also present in Bezem et al. [96], employs the fact mentioned in Topic 1: CR $\Leftrightarrow$ CP for countable ARSs.

It seems to be a difficult exercise to establish the (conjectured) result that the condition

’countable’ is necessary.

2.17. Some applications. We have already seen some applications in the examples of decreasing diagrams.More interesting is that also Huet’s Strong Confluence Lemma in Example 2.1 and Newman’s Lemma are corollaries of confluence by decreasing diagrams. For both, a little trick is required, as follows.

(i) *Huet’s strong confluence lemma*. Note that e.d.’s corresponding to Figure 2.1(a), the assumption of strong confluence, are in general not decreasing. E.g. one with two steps in the lower side is not. Yet, this situation can be seen in the scope of the present method as follows. Take two copies of $\rightarrow$, one labeled with h (horizontal): $\rightarrow_h$, one with v (vertical): $\rightarrow_v$. Now the e.d.’s as in Figure 2.18 are decreasing, under the ordering h < v.

Therefore, $\mathcal{A}^{\{h,v\}} = \langle A, (\rightarrow_h, \rightarrow_v\rangle$ with the three types of e.d.’s as shown, is CR by the theorem. This means that the original $\mathcal{A} = \langle A, \rightarrow\rangle$ is also CR.

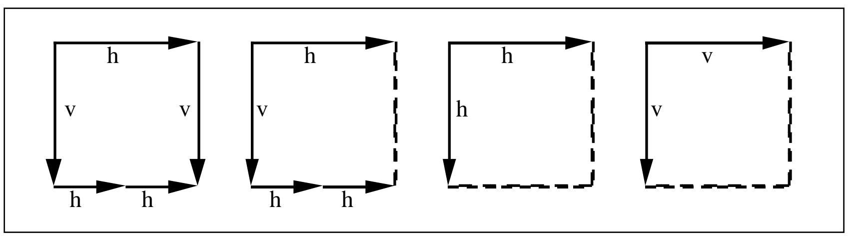


Figure 2.18

(ii) *Newman’s Lemma*. Let ARS $\mathcal{A} = \langle A, \rightarrow\rangle$ be SN and WCR. For the sake of exposition, let us assume that $\mathcal{A}$ is FB (finitely branching); then by SN each reduction graph G(a), $a \in A$, is finite, using König’s Lemma. Let $\underline{a}$ be the cardinality of G(a), so $\underline{a}$ is a natural number. Call it the size of a. Note that if $a \rightarrow b$, then $\underline{a} > \underline{b}$. Now take for each step $a \rightarrow b$, an indexed relation $\rightarrow_{\underline{a}}$ such that $a \rightarrow b \Leftrightarrow a \rightarrow_{\underline{a}} b$; in other words, label each step with the size of its left-hand side element. Now it is not hard to see that WCR gives us decreasing e.d.’s. Hence the labeled ARS is CR, and therefore the original one also.

The general argument, not assuming FB, is equally simple, using well-founded trees as size of an element instead of natural numbers.

References

BEZEM, M., KLOP, J.W. & VAN OOSTROM, V.,(1996) Diagram Techniques for Confluence. Information and Computation, Vol.141, No.2, p.172-204, 1998.

DE BRUIJN, N.G. (1978). *A note on weak diamond properties*. Memorandum 78-08, Eindhoven University of Technology, August 1978.

HUET, G. (1980). *Confluent reductions: Abstract properties and applications to term rewriting systems*. JACM, Vol.27, No.4 (1980), 797-821.

VAN OOSTROM, V. (1994). *Confluence for Abstract and Higher-Order Rewriting*. Ph.-D. thesis, Vrije Universiteit, Amsterdam, March 1994.

VAN OOSTROM, V. (1994a). *Confluence by decreasing diagrams*.
Theoretical Computer Science 126. p.259-280.

3

Infinite Diagrams

In this Topic we give a proof (sketch) of the 'confluence by decreasing diagrams' theorem in Topic 2 by an analysis of the geometry of, possibly infinite, reduction diagrams, resulting from two co-initial diverging finite reduction sequences, by 'tiling' with elementary reduction diagrams. Infinite diagrams arise this way, when we have a failure of confluence.

We will give several examples of infinite reduction diagrams, some of them exhibiting an interesting fractal-like boundary, some of them reminiscent to the pictures of M.C. Escher, with a repetition of the same pattern, receding in infinity.

Actually, we consider an enrichment of mere reduction diagrams, namely diagrams with a 'tree covering'. A tree covering of a diagram determines an ancestor-descendant relation between the edges appearing in a reduction diagram. By means of a tree covering an edge can be traced back to its ancestor edge on one of the original divergent reduction sequences. The theorem proved in this section states the impossibility of certain infinite diagrams with a tree-covering. Since the notion of decreasing diagram gives rise in a natural way to a tree covering—of the impossible kind—we have as an immediate corollary then the theorem of confluence by decreasing diagrams. The method of proof of our theorem is purely geometric. It employs topological notions such as condensation points of point sets in the real plane.

3.1. Infinite reduction diagrams and towers. We will consider infinite reduction diagrams as they arise from unsuccesfully tiling with elementary reduction diagrams (e.d.'s), defined in Topic 2. Some examples are given in Figure 3.1 and 3.2. The first example in Figure 3.1, the simplest infinite diagram, will be called the 'Escher-diagram'. Note the 'fractal-like' boundary that arises in

Figure 3.2.

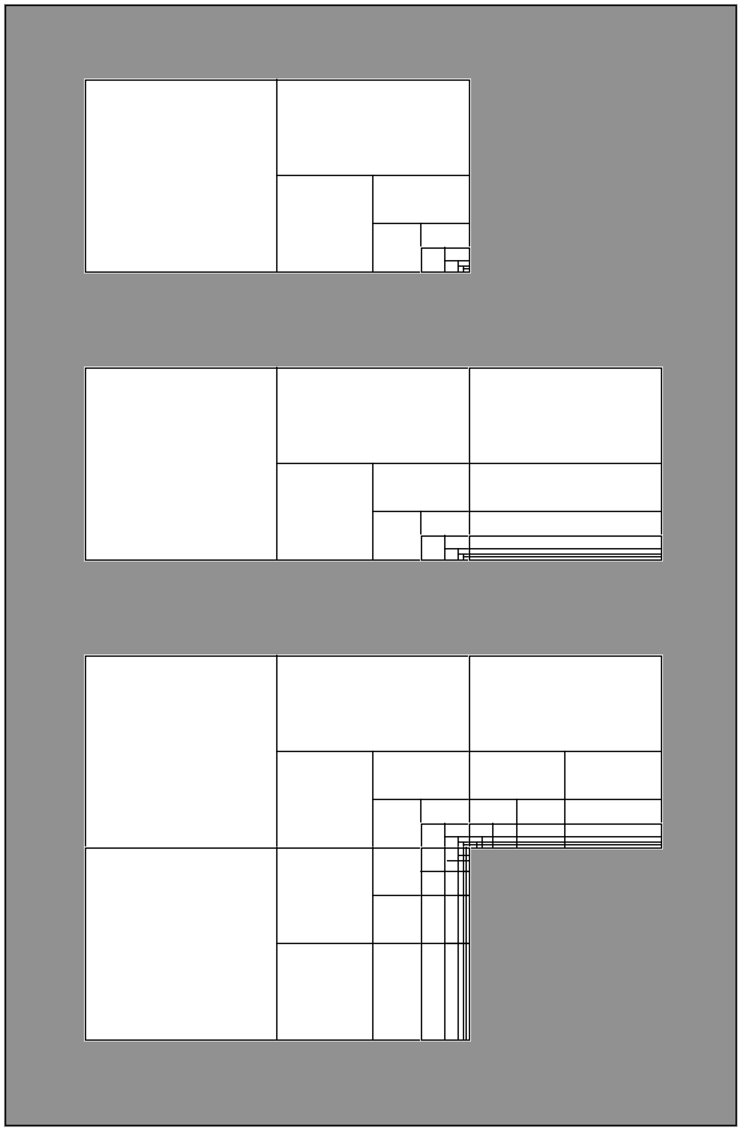


Figure 3.1

3.1.1. REMARK. Since we admit also empty steps, it is not immediately clear that an infinite diagram contains infinitely many non-empty edges. However, this is indeed the case; Bezem et al. (1996) proves the stronger fact that an infinite diagram possesses an infinite reduction containing infinitely many splitting steps. (An elementary diagram is 'splitting' if one of the converging sides contains two or more steps which then are called splitting steps. Clearly, splitting steps are non-empty.)

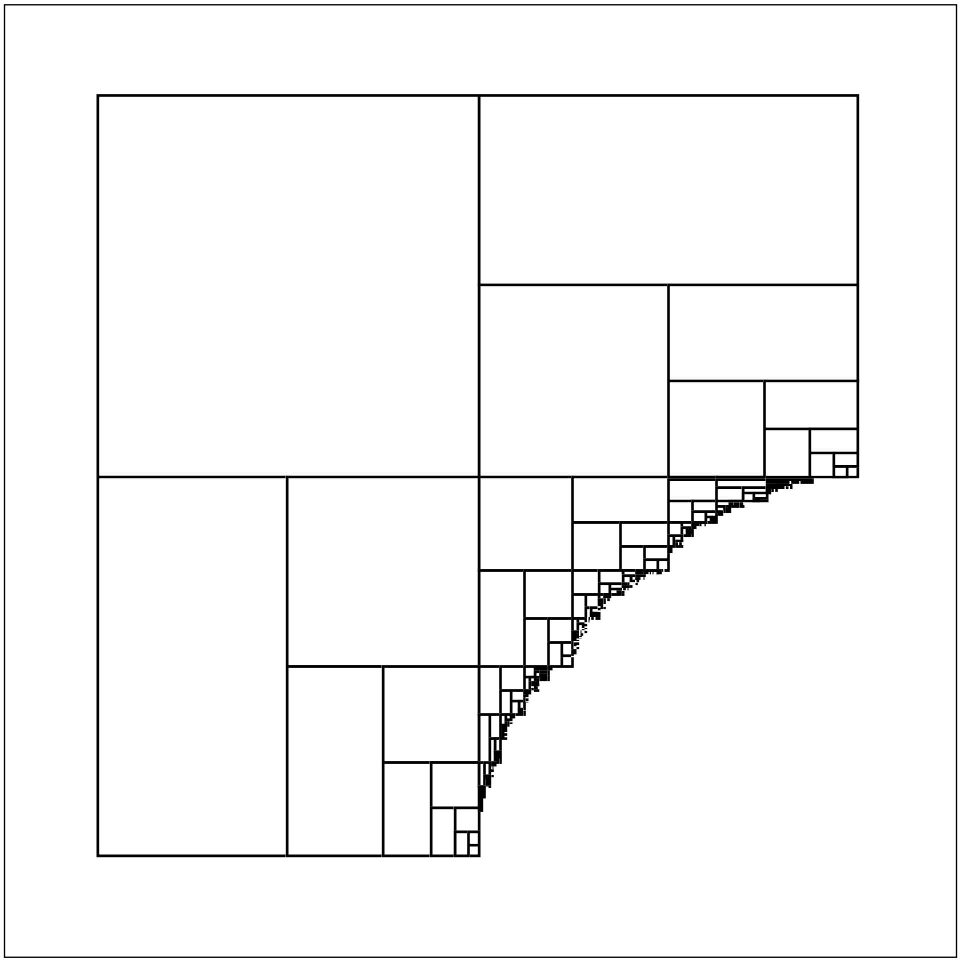


Figure 3.2

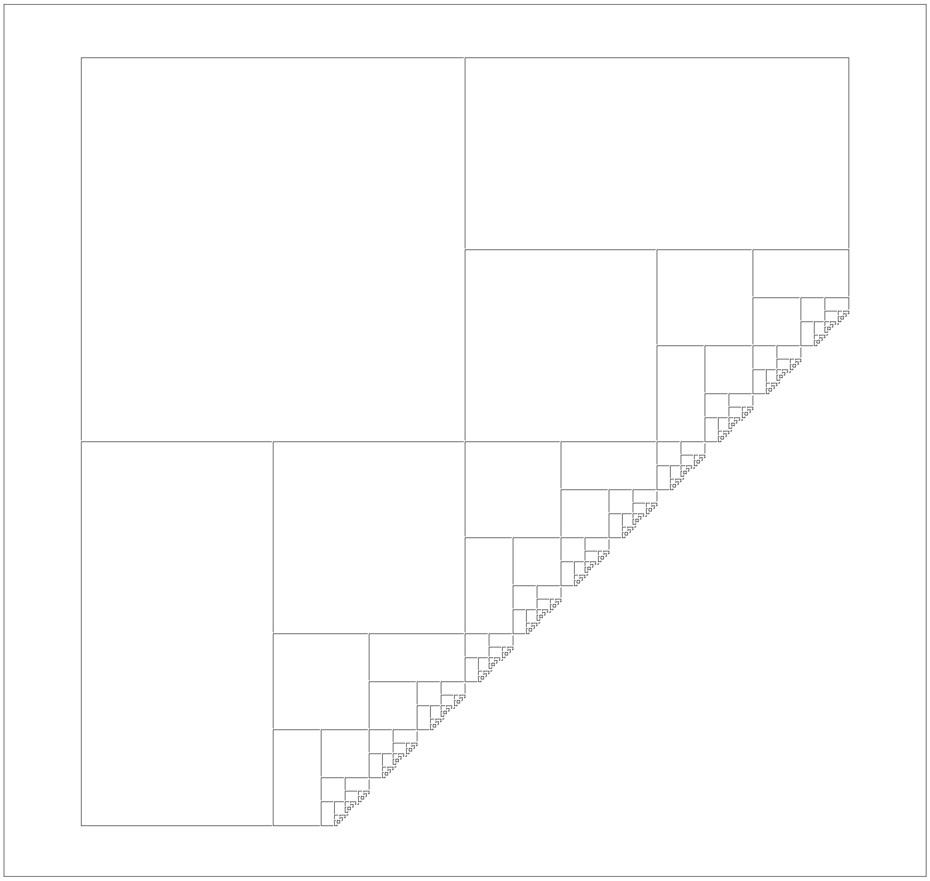


Figure 3.2a

A *tower* in an infinite reduction diagram is the result of adjoining elementary reduction diagrams in a ‘linear’ way, as suggested in Figure 3.3. We will always be interested in infinite towers. Towers can be either horizontal or vertical. Figure 3.4 displays two towers in the fractal-like diagram (Figure 3.2). Figure 3.4a displays (shaded) one of the two towers constituting the Escher diagram.

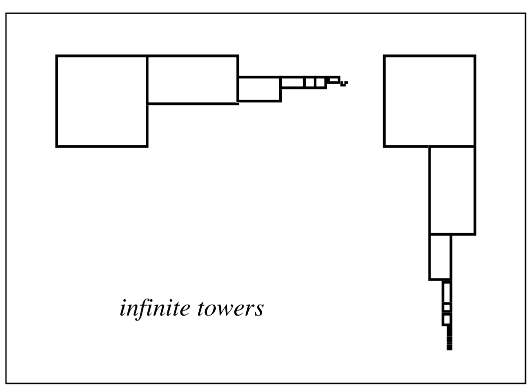


Figure 3.3

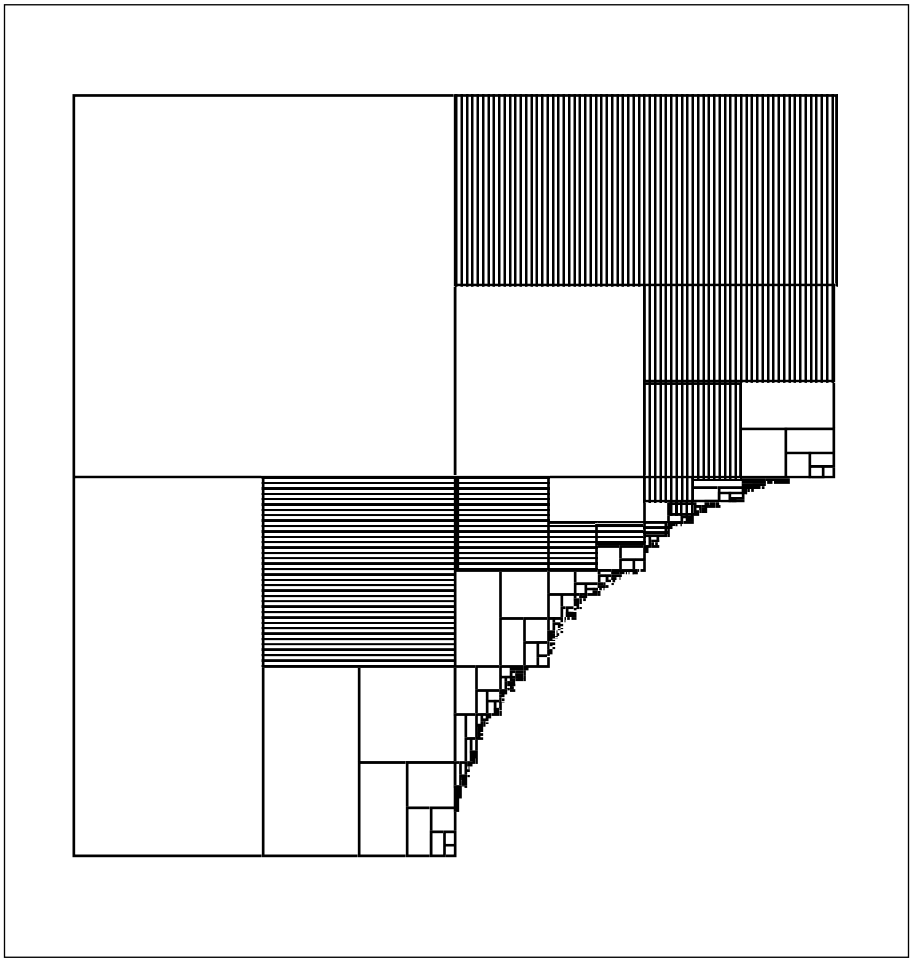


Figure 3.4

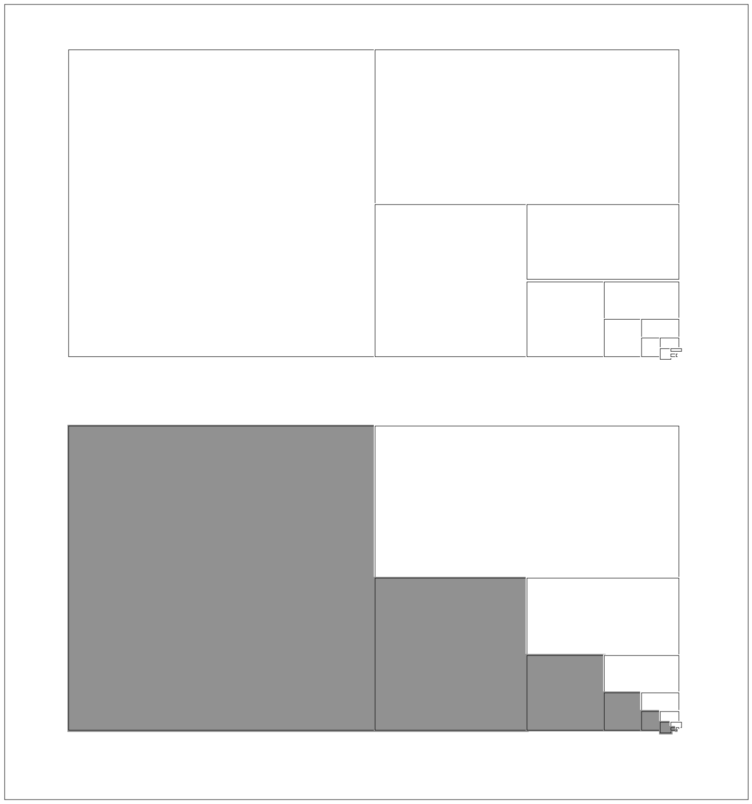


Figure 3.4a

3.2. PROPOSITION. *Every infinite diagram contains an infinite horizontal tower and an infinite vertical tower*.

PROOF. Consider the infinite diagram, and draw in each tile arrows from the left side to the steps in the right side (see Figure 3.4b). In this way finitely many trees arise. By the pigeon-hole principle and König's Lemma, one of these trees must have an infinite branch. This branch determines an infinite horizontal tower. Dually we find an infinite vertical tower. □

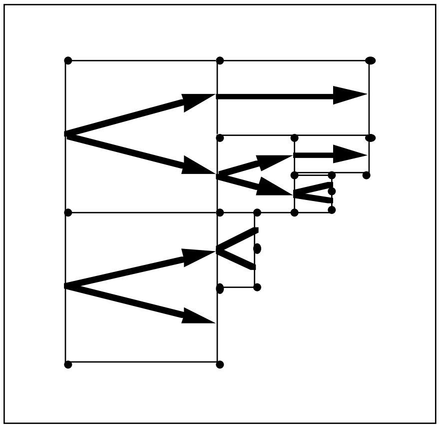


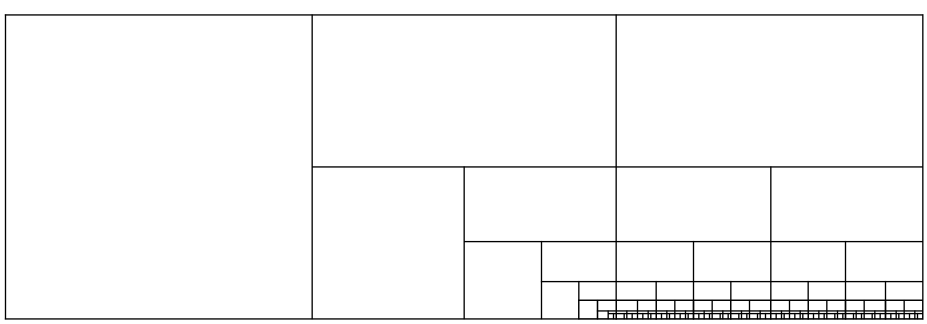
Figure 3.4b

Consider again the left-to-right trees in the preceding proof. Their branches are linearly ordered according to whether the one is 'above' the other. A branch σ is *above* branch τ, when after running together for some (possibly 0) steps, σ branches off to above compared to τ.

Furthermore it is clear that there is a highest infinite branch in the left-to-right trees of an infinite diagram. It is constructed in the obvious way: to start, choose the highest root of the left-right trees that has an infinite branch, then choose the highest successor with the same property, and so on.

Since branches in the left-right trees correspond with horizontal towers, there also exists a *highest horizontal infinite tower*. This will play an important role in the full proof (not given here.).

3.2.1. REMARK. In fact, the towers of a reduction diagram are linearly ordered by the relation 'above'. There may be continuum many towers (see Figure 3.4c).

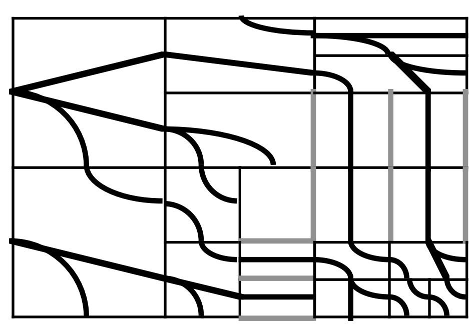


continuum many towers

Figure 3.4c

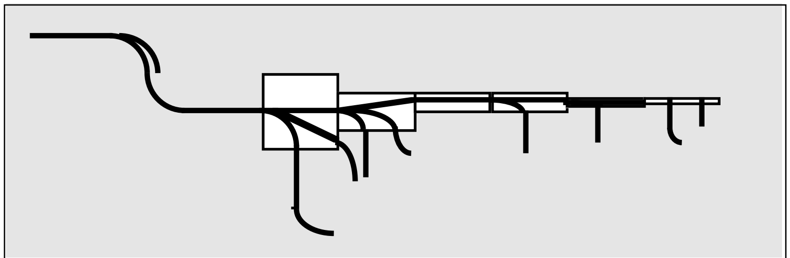
3.3. Tree coverings of reduction diagrams. Next, we will define the concept of a *tree covering* of a reduction diagram. Elementary reduction diagrams will be equipped with arrows leading from the initial (diverging) edges of the elementary reduction diagram to the opposite (converging) edges. Each converging edge is traced back via an arrow to one of the two initial edges (if the elementary diagram is not trivial; empty sides are not traced back). Figure 3.7 shows an example of a finite, completed reduction diagram with a tree covering. In this example the branches of the trees do not intersect, in general they may however.

3.4. DEFINITION. (i) A step in a branch is *straight* if it leads from an initial edge to an opposing edge.
(ii) A branch *changes orientation* if it goes from vertical to horizontal or dually.
(iii) An infinite branch is *meandering* if it changes orientation infinitely often.
(iv) Let τ be a horizontal branch. We say that τ branches off downward to branch σ, if τ, σ are concurrent for some steps, after which σ branches off to a lower opposing edge, or changes orientation. Likewise dually: a vertical branch may branch off to the right.
(v) There is exactly one tree covering all of whose steps are straight. We call it *the canonical tree covering*.



tree covering of reduction diagram

Figure 3.7



infinite branch of tree covering, contained in tower, branching off only in downward direction

Figure 3.8

We now formulate the main theorem of this topic. The full proof is omitted here and can be found in Klop et al. [99]. An intuition is sketched below. The full proof contains a notion not treated here, namely that of an 'upper boundary branch' of a tower.

3.5. THEOREM. *An infinite reduction diagram does not possess a tree covering such that*

(i) *all infinite branches are eventually contained in towers (i.e. straight),*

(ii) *infinite branches contained in horizontal towers split, eventually, only downwards,*

(iii) *infinite branches contained in vertical towers split, eventually, only to the right*.

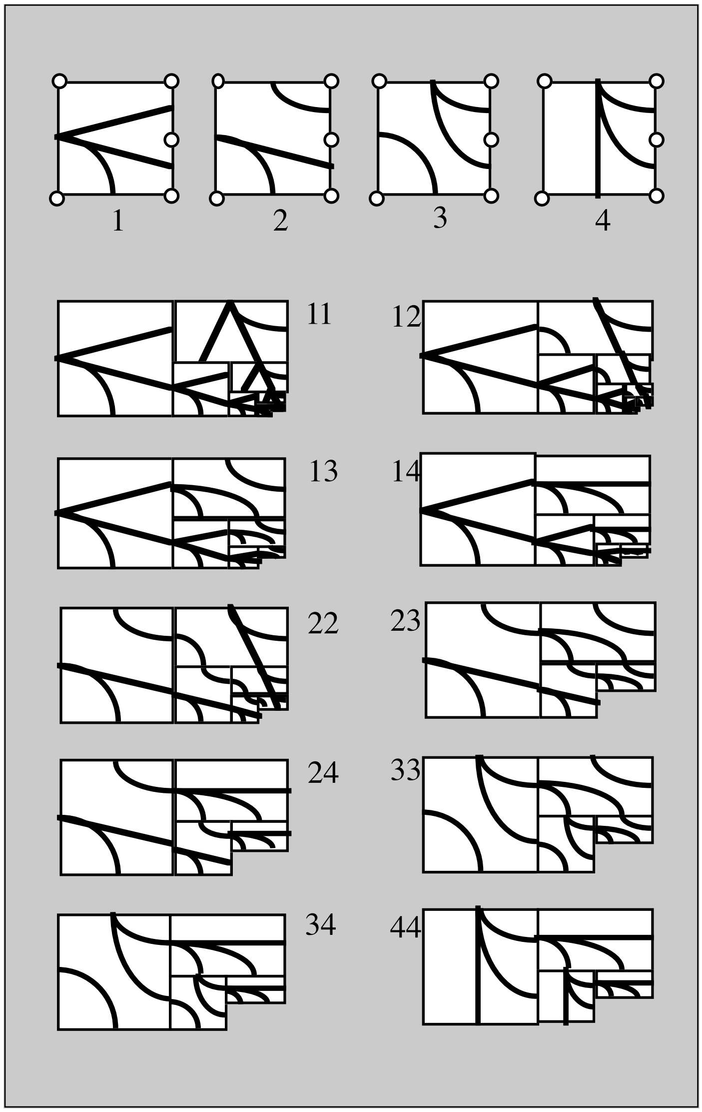


Figure 3.9

3.6. EXAMPLE. Figure 3.9 contains a number of 'periodic' tree coverings of the Escher diagram.

The upper part of Figure 6.1 gives some of the tree coverings (not exhaustive) of the elementary diagram of which the Escher diagram is built. (Note that the Escher diagram is indeed built from elementary diagrams of a single shape). These tree covered elementary diagrams are then used to build the Escher diagram in various combinations 11, 12, ... E.g. 23 means that the tree covered elementary diagram 2 is used, next the elementary diagram 3 (after mirroring); then the 23 configuration is recursively repeated.

Now the ten cases of Figure 3.9 have the following properties (see Table 1); indeed, no case has all three properties (i-iii) of the theorem.

	(i)	(ii)	(iii)
11	+	-	-
12	+	-	+
13	+	-	+
14	+	-	+
22	-	+	+
23	-	+	+
24	-	+	+
33	-	+	+
34	-	+	+
44	-	+	+

Table 1

Let us elaborate on the underlying intuition. Condition (i) says that there are no infinitely meandering branches. Let us simplify the situation by forbidding any meandering, so assume all branches are straight. This means that we are dealing with the canonical tree covering (Def. 3.4(v)). Now consider the lowest horizontal branch σ and the rightmost vertical branch τ. Now say an edge in diagram D is 'accounted for' if a branch of the tree covering under consideration passes through it. the branches σ and τ account for infinitely many edges, as they are infinite. But there remain infinitely many edges not touched by σ and τ. Some experiments make this clear; e.g. in the Escher diagram we find that the steps in bold are not accounted for (see Figure 3.9a). In Figure 3.9b this is the grey area, containing infinitely many edges. Now if σ and τ are not allowed to branch off towards this infinitely large area, the tree covering can never cover all these edges.

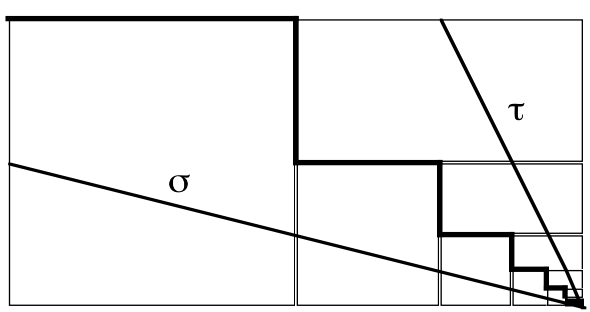


Figure 3.9a

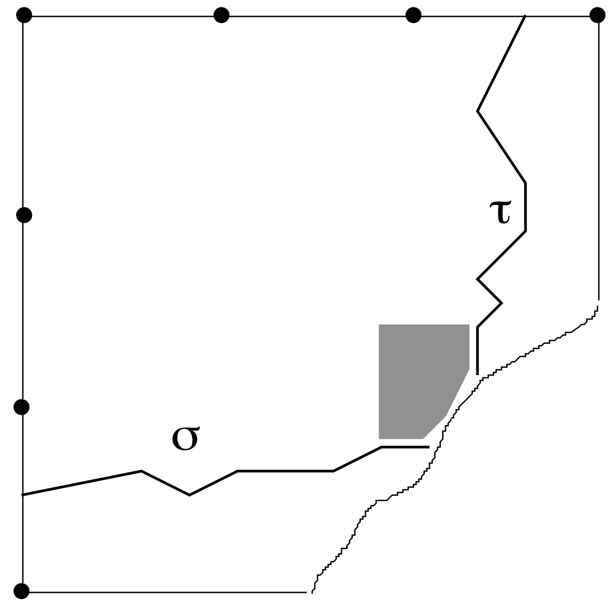


Figure 3.9b

3.7. Confluence by decreasing diagrams. Let us repeat the general form of an decreasing e.d. from the preceding topic (See Figure 3.10), together with an example of some decreasing e.d's as in Figure 3.10a).

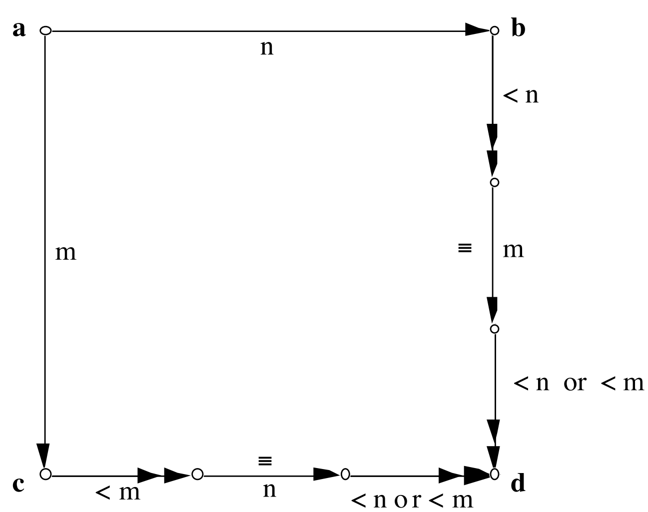


Figure 3.10

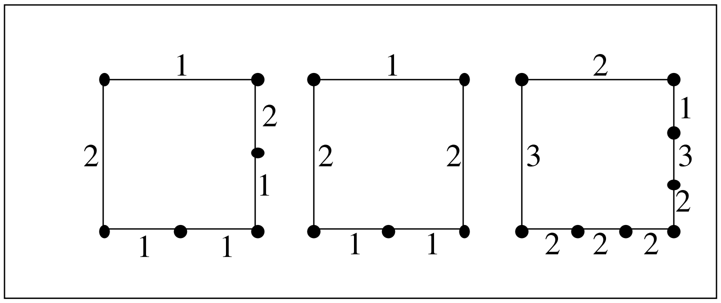


Figure 3.10a

We will now connect the present definition of decreasing e.d. with the tree coverings of this topic. In a decreasing elementary diagram we will trace back the converging steps to the two diverging steps. In doing so, it will be helpful to use a *heavy* arrow in case the index remains the same, and a *light* arrow in case the index decreases.

The heavy and light arrows are determined as follows. Consider the vertical reduction

$b \twoheadrightarrow_{<n} \cdot \rightarrow^{\equiv}_{m} \cdots \twoheadrightarrow_{<n \text{ or } <m}$ d. Now we let the first part of this reduction, consisting of steps with index less than the index n of the horizontal step $a \rightarrow_n b$, trace back lightly to that step. If the second part consists of 1 step with label m, it is traced back heavily to the vertical step $a \rightarrow c$. If it consists of 0 steps, we do nothing. The part consisting of steps with label less than n or m is treated as follows. If the step label is less than n we trace back lightly to $a \rightarrow b$, if less than m then lightly to $a \rightarrow c$, if both then we choose one. Likewise dually.

So a decreasing elementary diagram with the tracing arrows has one of the shapes of Figure 3.11: containing two heavy arrows, or one, or none. It is important that heavy arrows (along which the indices remain the same) are straight, while the light arrows (along which the indices decrease) *may* involve a change of orientation.

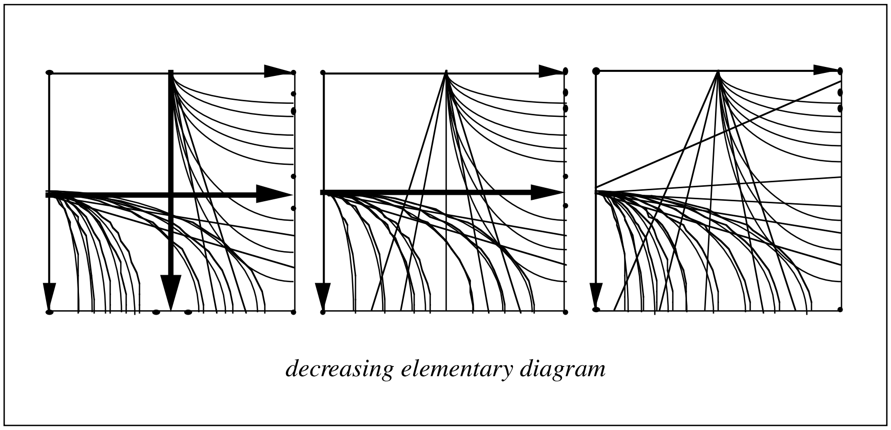


Figure 3.11
Elementary diagrams with tree covering

See Figure 3.12, consisting of the decreasing elementary diagrams of Figure 3.10a but now enriched with the tracing arrows (with the convention for heavy and light just mentioned).

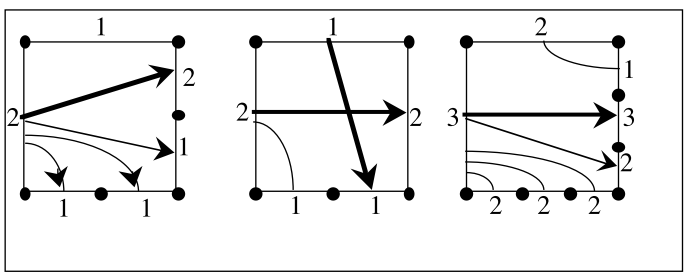


Figure 3.12

Note that the tracing pattern (the tree covering) is not uniquely determined by the decreasing elementary diagram; e.g. Figure 3.13 contains two tracings for the same elementary diagram.

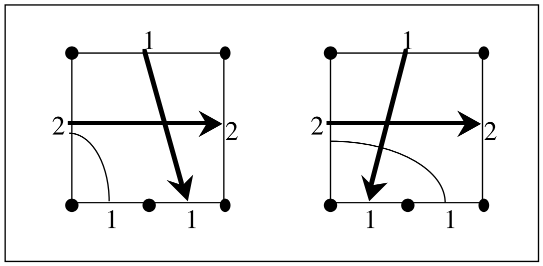


Figure 3.13

We now have

3.8. THEOREM. *Every diagram construction using decreasing elementary diagrams will terminate eventually in a finite confluent diagram.*

PROOF. Equip the decreasing elementary diagrams with heavy and light arrows as explained above. Note that heavy arrows preserve indices and are straight, while light ones decrease indices and may change orientation. Note furthermore that a horizontal heavy arrow cannot split off in upward direction (see Figure 3.14) and likewise dually.

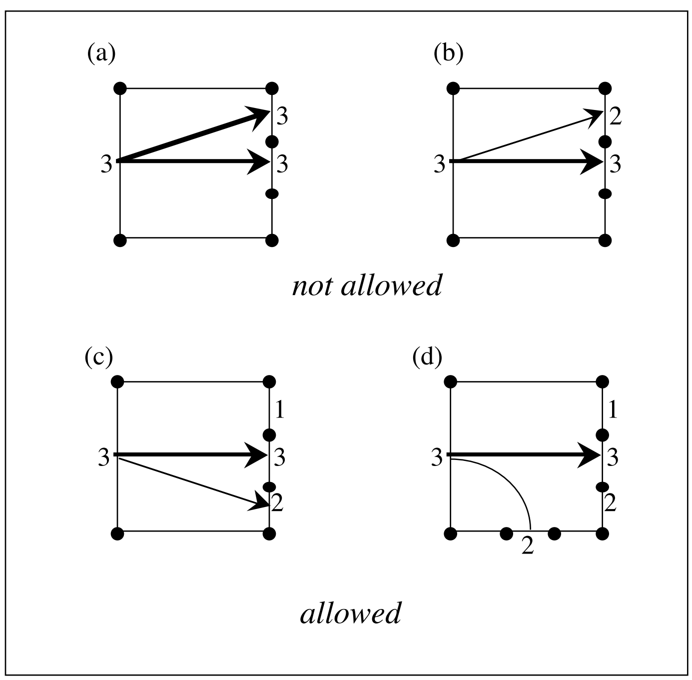


Figure 3.14

Now consider an infinite branch in the diagram enriched with heavy and light arrows. Because the partial order I is well-founded, eventually only heavy (index-preserving) arrows can occur in this branch. But these are straight. So, every infinite branch must be eventually straight (and thus contained in a tower).

Furthermore, from infinite horizontal branches we can only have split offs in downward direction (either by straight arrows as in Figure 3.14(c) or by a change in orientation as in 3.14(d). Likewise dually. That is, the three hypotheses of Theorem 3.5 are fulfilled. According to this theorem the diagram cannot be infinite. □

3.9. COROLLARY (De Bruijn–Van Oostrom; confluence by decreasing diagrams)
Every ARS with reduction relations indexed by a well-founded partial order I, *and satisfying the decreasing criterion for its elementary diagrams , is confluent*.

References

BEZEM, M., KLOP, J.W. & VAN OOSTROM, V., (1996) Diagram Techniques for Confluence. Information and Computation, Vol.141, No.2, p.172-204, 1998.

KLOP, J.W., De Vrijer,R.C. & VAN OOSTROM, V., (1999) A Geometric Proof of Confluence by Decreasing Diagrams Tech. Report Vrije Universiteit Amsterdam, 1999 (www.cs.vu.nl/~rdv/recent publications)

4

First-order Term Rewriting Systems

After the ‘abstract’ considerations of the preceding topics, in this topic we start with introducing term rewriting systems (TRSs). In contrast with the higher-order term rewriting systems in a later topic, we will call them also first-order TRSs; but often we will omit this qualification when it is clear.

4.1. Syntax of Term Rewriting Systems. A (first-order) Term Rewriting System (TRS) is a pair (Σ,R) of an *alphabet* or *signature* Σ and a set of reduction rules (rewrite rules) R. The alphabet Σ consists of:

(i) a countably infinite set of *variables* $x_1,x_2,x_3,...$ also denoted as $x,y,z,x',y',...$

(ii) a non-empty set of *function symbols* or *operator symbols* F,G,..., each equipped with an ‘arity’ (a natural number), i.e. the number of ‘arguments’ it is supposed to have. We not only (may) have unary, binary, ternary, etc., function symbols, but also 0-ary: these are also called *constant symbols*.

The set of terms (or expressions) ‘over’ Σ is $Ter(\Sigma)$ and is defined inductively:

(i) $x,y,z,... \in Ter(\Sigma)$,

(ii) if F is an n-ary function symbol and $t_1,...,t_n \in Ter(\Sigma)$ $(n \geq 0)$, then $F(t_1,...,t_n) \in Ter(\Sigma)$. The t_i $(i = 1,...,n)$ are the arguments of the last term.

Terms not containing a variable are called *ground* terms (also: *closed* terms), and $Ter_0(\Sigma)$ is the set of ground terms. Terms in which no variable occurs twice or more, are called *linear*.

Contexts are 'terms' containing one occurrence of a special symbol $\Box$, denoting an empty place. A context is generally denoted by C[ ]. If $t \in Ter(\Sigma)$ and t is substituted in $\Box$, the result is $C[t] \in Ter(\Sigma)$; t is said to be a subterm of C[t], notation $t \subseteq C[t]$. Since $\Box$ is itself a context, the trivial context, we also have $t \subseteq t$. Often this notion of subterm is not precise enough, and we have to distinguish *occurrences* of subterms (or symbols) in a term; it is easy to define the notion of occurrence formally, using sequence numbers denoting a 'position' in the term, but here we will be satisfied with a more informal treatment.

4.1.1. EXAMPLE. Let $\Sigma = \{A,M,S,0\}$ where the arities are 2,2,1,0 respectively. Then A(M(x,y),y) is a (non-linear) term, A(M(x,y),z) is a linear term, A(M(S(0),0),S(0)) is a ground term, A(M($\Box$,0),S(0)) is a context, S(0) is a subterm of A(M(S(0),0),S(0)) having two occurrences: A(M(**S(0)**,0),**S(0)**).

A *substitution* σ is a map from $Ter(\Sigma)$ to $Ter(\Sigma)$ satisfying $\sigma(F(t_1,...,t_n)) = F(\sigma(t_1),...,\sigma(t_n))$ for every n-ary function symbol F (here $n \geq 0$). So, σ is determined by its restriction to the set of variables. We also write t^σ instead of $\sigma(t)$.

A *reduction rule* (or rewrite rule) is a pair (t, s) of terms $\in Ter(\Sigma)$. It will be written as $t \to s$. Often a reduction rule will get a name, e.g. r, and we write r: $t \to s$. Two conditions will be imposed:

(i) the LHS (left-hand side) t is not a variable,

(ii) the variables in the right-hand side s are already contained in t.

A reduction rule r: $t \to s$ determines a set of *rewrites* $t^\sigma \to_r s^\sigma$ for all substitutions σ. The LHS t^σ is called a *redex* (from 'reducible expression'), more precisely an r-redex. A redex t^σ may be replaced by its '*contractum*' s^σ inside a context C[ ]; this gives rise to *reduction steps* (or one-step rewritings)

$$C[t^\sigma] \to_r C[s^\sigma].$$

A term without a redex is a *normal form*. We call $\to_r$ the *one-step reduction relation* generated by r. Concatenating reduction steps we have (possibly infinite) *reduction sequences* $t_0 \to t_1 \to t_2 \to$ ... or *reductions* for short. If $t_0 \to ... \to t_n$ we also write $t_0 \twoheadrightarrow t_n$, and t_n is a *reduct* of t_0, in accordance with the notations and concepts introduced in Topic 1 for ARSs.

It is understood that R does not contain two reduction rules that originate from each other by a 1-1 renaming of variables.

4.1.2. EXAMPLE. Consider Σ as in Example 4.1.1. Let (Σ, R) be the TRS (specifying the natural numbers with addition, multiplication, successor and zero) with the following set R of reduction rules:

r_1	A(x,0) → x
r_2	A(x,S(y)) → S(A(x,y))
r_3	M(x,0) → 0
r_4	M(x,S(y)) → A(M(x,y),x)

Table 4.1

Now M(S(S(0)), S(S(0))) ↠ S(S(S(S(0)))), since we have the following reduction:

M(S(S(0)),S(S(0))) →
A(M(S(S(0)),S(0)),S(S(0))) →
S(**A(M(S(S(0)),S(0)),S(0))**) →
S(S(**A(M(S(S(0)),S(0)),0)**)) →
S(S(**M(S(S(0)),S(0))**)) →
S(S(A(**M(S(S(0)),0)**,S(S(0))))) →
S(S(**A(0,S(S(0)))**)) →
S(S(S(**A(0,S(0))**))) →
S(S(S(S(**A(0,0)**)))) →
S(S(S(S(0)))).

Here in each step the bold-face redex is rewritten. Note that this is not the only reduction from M(S(S(0)), S(S(0))) to S(S(S(S(0)))).

Obviously, for each TRS (Σ,R) there is a corresponding ARS, namely (Ter(Σ), $(\rightarrow_r)_{r \in R}$). Here we have to be careful: it may make a big difference whether one discusses the TRS (Σ, R) consisting of all terms, or the TRS restricted to the ground terms (see the next example). We will adopt the convention that (Σ, R) has as corresponding ARS the one mentioned already, and we write $(\Sigma, R)_0$ if the ARS ($Ter_0(\Sigma)$, $(\rightarrow_r)_{r \in R}$) is meant. Via the associated ARS, all notions considered in Topic 1 (CR, UN, SN, ...) carry over to TRSs.

4.1.3. EXAMPLE. Let (Σ, R) be the TRS of Example 4.1.2 and consider (Σ, R') where R' = R ∪ {A(x, y) → A(y, x)}; so the extra rule expresses commutativity of addition. Now (Σ, R') is not WN: the term A(x, y) has no normal form. However, $(\Sigma, R')_0$ (the restriction to ground terms) is WN.

Whereas $(\Sigma, R)_0$ is SN, $(\Sigma, R')_0$ is no longer so, as witnessed by the infinite reductions possible in the reduction graph in Figure 4.1. The 'bottom' term in that reduction graph is a normal form.

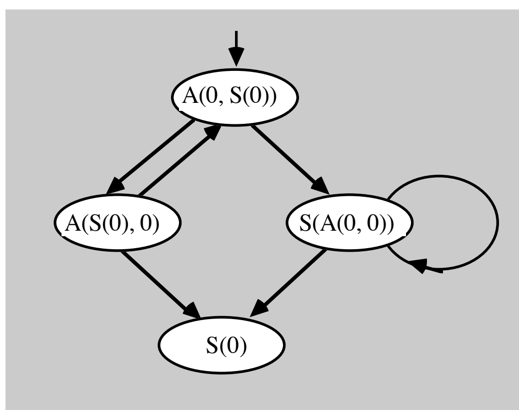


Figure 4.1

4.3. Semi-Thue systems. Semi-Thue Systems (STSs), as defined e.g. in Jantzen [88], can be 'viewed' as TRSs, as follows. We demonstrate this by the following example of a STS:

Let T = {(aba, bab)} be a one-rule STS. Then T corresponds to the TRS R with unary function symbols a, b and a constant o, and the reduction rule a(b(a(x))) → b(a(b(x))). Now a reduction step in T, e.g.: bbabaaa → bbbabaa, translates in R to the reduction step b(b(a(b(a(a(a(o))))))) → b(b(b(a(b(a(a(o))))))). It is easy to see that this translation gives an 'isomorphism' between T and R (or more precisely $(R)_0$, the restriction to ground terms).

4.4. Applicative Term Rewriting Systems. In some important TRSs there is a special binary operator, called *application* (Ap). E.g. Combinatory Logic (CL), based on S, K, I, has the rewrite rules

Ap(Ap(Ap(S,x),y),z)	→ Ap(Ap(x,z), Ap(y,z))
Ap(Ap(K,x),y)	→ x
Ap(I,x)	→ x

Table 4.2

Here S, K, I are constants. Often one uses the infix notation (t**.**s) instead of Ap(t, s), in which case the rewrite rules of CL read as follows:

((S**.**x)**.**y)**.**z	→ (x**.**z)**.**(y**.**z)
(K**.**x)**.**y	→ x
I**.**x	→ x

Table 4.3

As in ordinary algebra, the dot is mostly suppressed; and a further notational simplification is that many pairs of brackets are dropped in the convention of association to the left. That is, one restores the missing brackets choosing in each step of the restoration the leftmost possibility. Thus the three rules become:

Sxyz	→ xz(yz)
Kxy	→ x
Ix	→ x

Table 4.4

The TRS CL has 'universal computational power': every (partial) recursive function on the natural numbers can be expressed in CL. This feature is used in Turner [79], where CL is used to implement functional programming languages. Actually, an extension of CL is used there, called SKIM (for S,K,I-Machine); it is also an applicative TRS (see Table 4.5)

SKIM

Sxyz	→ xz(yz)
Kxy	→ x
Ix	→ x
Cxyz	→ xzy
Bxyz	→ x(yz)
Yx	→ x(Yx)
Uz(Pxy)	→ zxy
P_0(Pxy)	→ x
P_1(Pxy)	→ y
cond true xy	→ x
cond false xy	→ y
plus n m	→ n+m
times n m	→ n.m
eq n n	→ true
eq n m	→ false if n ≠ m

Table 4.5

Note that this TRS has infinitely many constants: apart from the constants S, K, ..., eq there is a constant n for each $n \in \mathbb{N}$. There are also infinitely many reduction rules, because the last four rules are actually *rule schemes*; e.g. plus n m → n+m stands for all reduction rules like plus 0 0 → 0, plus 0 1 → 1, ..., plus 37 63 → 100, ... (In fact, the extra constants in SKIM are there for reasons of efficient implementation; they can all be defined using only S and K.)

It is harmless to mix the applicative notation with the usual one, as in the following TRS, CL with test for syntactical equality:

Sxyz	$\to$ xz(yz)
Kxy	$\to$ x
Ix	$\to$ x
D(x,x)	$\to$ E

Table 4.6

However, some care should be taken: consider the following TRS

Sxyz	$\to$ xz(yz)
Kxy	$\to$ x
Ix	$\to$ x
Dxx	$\to$ E

Table 4.7

where D is now a *constant* (instead of a binary operator) subject to the rewrite rule, in full notation, Ap(Ap(D, x), x) $\to$ E. These two TRSs have very different properties: the first TRS is confluent, but the second is not.

4.5. Semantics of Term Rewriting Systems. Although we are almost always interested in syntactic methods to prove CR or UN, sometimes a semantical excursion may be convenient. Here is one:

4.5.1. THEOREM. *Let* $\mathcal{A}$ *be an algebra 'for' the TRS* R *such that for all normal forms* t, t' *of* R:

$$\mathcal{A} \models t = t' \quad \Rightarrow \quad t \equiv t'$$

Then R *has the property* UN *(uniqueness of normal forms).*

Here the phrase '$\mathcal{A}$ is an algebra for the TRS R' means that $\mathcal{A}$ has the same signature as R, and that reduction in R is sound with respect to $\mathcal{A}$, i.e. $t \twoheadrightarrow_R s$ implies $\mathcal{A} \models t = s$. The terms t, s need not be ground terms. The proof of the theorem is trivial.

4.6. Decidability of properties in Term Rewriting Systems.

We adopt the restriction in this subsection to TRSs R with finite alphabet and finitely many reduction rules. It is undecidable whether for such TRSs the property confluence (CR) holds. (This is so both for R, the TRS of all terms, and $(R)_0$, the TRS restricted to ground terms.)

For *ground TRSs*, i.e.TRSs where in every rule $t \to s$ the terms t, s are ground terms (not to be confused with $(R)_0$ above), confluence is decidable (Dauchet & Tison [84], Dauchet et al. [87], Oyamaguchi [87]).

For the termination property (SN) the situation is the same. It is undecidable for general TRSs, even for TRSs with only one rule (see for a proof Dauchet [89]). For ground TRSs termination is decidable (Huet & Lankford [78]).

For particular TRSs it may also be undecidable whether two terms are convertible, whether a term has a normal form, whether a term has an infinite reduction. A TRS where all these properties are undecidable is Combinatory Logic (CL), in Table 4.4.

For Recursive Program Schemes (RPSs), defined in Topic 5 as a subclass of the orthogonal TRSs, the properties SN and WN are decidable. Also for a particular RPS R it is decidable whether a term $t \in Ter(R)$ has the property SN or WN (Khasidashvili [90]).

We now discuss one of the main applications of term rewriting, namely: solving the word problem for structures defined by an equational specification. This involves also the introduction of 'equational logic'.

4.7. Equational specifications. An equational specification is a pair (Σ, E) where the signature (or alphabet) Σ is as in Section 4.1 for TRSs (Σ, R), and where E is a set of equations $s = t$ between terms $s, t \in Ter(\Sigma)$.

If an equation s = t is *derivable* from the equations in E, we write $(\Sigma, E) \vdash s = t$ or $s =_E t$. Formally, derivability is defined by means of the inference system "Equational Logic" of Table 4.8.

$(\Sigma, E) \vdash s = t$	if $s = t \in E$
$\dfrac{(\Sigma, E) \vdash s = t}{(\Sigma, E) \vdash s^{\sigma} = t^{\sigma}}$	for every substitution σ
$\dfrac{(\Sigma, E) \vdash s_1 = t_1 ,..., (\Sigma, E) \vdash s_n = t_n}{(\Sigma, E) \vdash F(s_1,\ldots,s_n) = F(t_1,\ldots,t_n)}$	for every n-ary $F \in \Sigma$
$(\Sigma, E) \vdash t = t$	
$\dfrac{(\Sigma, E) \vdash t_1 = t_2 ,\ (\Sigma, E) \vdash t_2 = t_3}{(\Sigma, E) \vdash t_1 = t_3}$	
$\dfrac{(\Sigma, E) \vdash s = t}{(\Sigma, E) \vdash t = s}$	

Table 4.8: Equational Logic

Let Σ be a signature. Then a Σ-*algebra* $\mathcal{A}$ is a set A together with functions $F^{\mathcal{A}}: A^n \to A$ for every n-ary function symbol $F \in \Sigma$. (If F is 0-ary, i.e. F is a constant, then $F^{\mathcal{A}} \in A$.) An equation s = t (s, t $\in Ter(\Sigma)$) is assigned a meaning in $\mathcal{A}$ by interpreting the function symbols in s,t via the corresponding functions in $\mathcal{A}$. Variables in s = t are (implicitly) universally quantified. If the universally quantified statement corresponding to s = t (s, t $\in Ter(\Sigma)$) is true in $\mathcal{A}$, we write $\mathcal{A} \models s = t$ and say that s = t is *valid* in $\mathcal{A}$. $\mathcal{A}$ is called a *model* of a set of equations E if every equation in E is valid in $\mathcal{A}$. Abbreviation: $\mathcal{A} \vDash E$. The *variety* of Σ-algebras defined by an equational specification (Σ, E), notation Alg(Σ, E), is the class of all Σ-algebras $\mathcal{A}$ such that $\mathcal{A} \vDash E$. Instead of $\forall \mathcal{A} \in \mathrm{Alg}(\Sigma, E)\ \ \mathcal{A} \vDash F$, where F is a set of equations between Σ-terms, we will write $(\Sigma, E) \vDash F$. There is the well-known completeness theorem for equational logic of Birkhoff [35]:

4.7.1. THEOREM. *Let* (Σ, E) *be an equational specification. Then for all* s, t ∈ Ter(Σ):

$$(\Sigma, E) \vdash s = t \iff (\Sigma, E) \vDash s = t. \qquad \Box$$

Now the *validity problem* or *uniform word problem* for (Σ, E) is:

Given an equation s = t *between Σ-terms, decide whether or not* $(\Sigma, E) \vDash s = t$.

According to Birkhoff's completeness theorem for equational logic this amounts to deciding $(\Sigma, E) \vdash s = t$. Now we can state why *complete* TRSs (i.e. TRSs which are SN and CR) are important. Suppose for the equational specification (Σ, E) we can find a complete TRS (Σ, R) such that for all terms s, t ∈ Ter(Σ):

$$t =_R s \iff E \vdash t = s \qquad (\circledast)$$

Then we have a positive solution of the validity problem. The decision algorithm is simple:

(1) *Reduce* s *and* t *to their respective normal forms* s', t'
(2) *Compare* s' *and* t': $s =_R t$ *iff* $s' \equiv t'$.

We are now faced with the question how to find a complete TRS R for a given set of equations E such that (❋) holds. In general this is not possible, since not every E (even if finite) has a solvable validity problem. The most famous example of such an E with unsolvable validity problem is the set of equations obtained from CL, Combinatory Logic, in Tables 4.3, 4.4 above after replacing '→' by

'=': see Table 4.9. (For a proof of the unsolvability see Barendregt [84].) So the validity problem of (Σ, E) can be solved by providing a complete TRS (Σ, R) for (Σ, E). Note however, that there are equational specifications (Σ, E) with decidable validity problem but without a complete TRS (Σ, R) satisfying (❇), see Remarks 4.7.2 and 4.7.3 below.

Sxyz	= xz(yz)
Kxy	= x
Ix	= x

Table 4.9

It is important to realize that we have considered up to now equations s = t between possibly *open* Σ-terms (i.e. possibly containing variables). If we restrict attention to equations s = t between *ground* terms s, t, we are considering the *word problem* for (Σ, E), which is the following decidability problem:

> *Given an equation* s = t *between ground terms* s, t ∈ Ter(Σ)*, decide whether or not* (Σ, E) ⊨ s = t *(or equivalently,* (Σ, E) s = t *).*

Also for the word problem, complete TRSs provide a positive solution. In fact, we require less than completeness (SN and CR) for all terms, but only for ground terms. (See Example 4.1.3 for an example where this makes a difference.) It may be that a complete TRS for E cannot be found with respect to all terms, while there does exist a TRS which is complete for the restriction to ground terms.

4.7.2. REMARK. Let (Σ, E) be the specification given by the equations

$$x + 0 = x$$
$$x + S(y) = S(x + y)$$
$$x + y = y + x$$

(i) Then there is no complete TRS R 'for' E, i.e. such that for all terms s, t $\in$ Ter(Σ): s $=_R$ t $\Leftrightarrow$ s $=_E$ t. (Consider in a supposed complete TRS R, the normal forms of the open terms x + y and y + x.)

(ii) For the restriction to ground terms though, there does exist such a TRS. (Just orient the first two equations.)

4.7.3. REMARK. In Klop [92] the following observation due to J.A.Bergstra is proved:

THEOREM. *Let* (Σ, E) *be the specification with* $\Sigma = \{0, +\}$ *and* E = {x + y = y + x}. *Then there is no finite TRS* R *such that the restriction to ground terms,* $(R)_0$, *is complete and such that* $=_R$ *and* $=_E$ *coincide on ground terms.*

4.8. Modularity. We will now consider what happens when TRSs are *combined*. For simplicity, we will suppose that such a combination is in fact a disjoint union.

4.8.1. DEFINITION. Let $\mathcal{R}_1 = (\Sigma_1, R_1)$ and $\mathcal{R}_2 = (\Sigma_2, R_2)$. Then :

(i) $\mathcal{R}_1 \cup \mathcal{R}_2 = (\Sigma_1 \cup \Sigma_2, R_1 \cup R_2)$.

(ii) If Σ_1, Σ_2 are disjoint (i.e. they have no function or constant symbols in common), then $\mathcal{R}_1 \cup \mathcal{R}_2$ is a disjoint union.

Further, we will say that a property $\mathcal{P}$ of TRSs is a *modular* property, if we have for disjoint unions $\mathcal{R}_1 \cup \mathcal{R}_2$:

$$\mathcal{R}_1 \cup \mathcal{R}_2 \models \mathcal{P} \Leftrightarrow \mathcal{R}_1 \models \mathcal{P} \ \& \ \mathcal{R}_2 \models \mathcal{P}.$$

The first significant result about modularity of properties is the following fact:

4.8.1. THEOREM (Toyama (1987). *CR is a modular property.*

The main problem in theory about modularity is the presence of *collapsing* reduction steps. A rewrite rule is collapsing if its RHS is a variable; application of such rules disturbs the layer structure that terms in a combined TRS $\mathcal{R}_1 \cup \mathcal{R}_2$ have. Figure 4.2 depicts such steps. In the picture the terms from the one signature are white, from the other signature black. A collapsing step removes one triangle of some color.

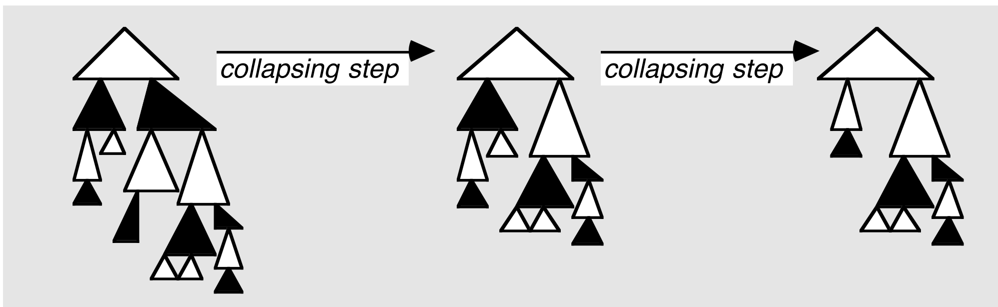


Figure 4.2

So, confluence is a 'modular' property. One might think that the same is true for termination (SN), but Toyama [87] gives a simple counterexample: take

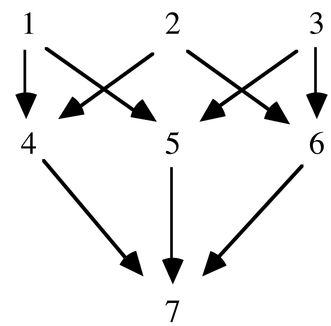
$R_1 = \{f(0,1,x) \to f(x,x,x)\}$
$R_2 = \{\underline{or}(x,y) \to x, \underline{or}(x,y) \to y\}$

then R_1, R_2 are both SN, but $R_1 \cup R_2$ is not, since there is the infinite reduction:

$f(\underline{or}(0,1), \underline{or}(0,1), \underline{or}(0,1)) \to f(0, \underline{or}(0,1), \underline{or}(0,1)) \to$
$f(0, 1, \underline{or}(0,1)) \to f(\underline{or}(0,1), \underline{or}(0,1), \underline{or}(0,1)) \to ....$

In this counterexample R_2 is not confluent and thus one may conjecture that 'confluent and terminating' (or CR & SN, or complete) is a modular property . Again this is not the case, as the following counterexample shows R_1 has the eleven rules

$F(4,5,6,x) \to F(x,x,x,x)$
$F(x,y,z,w) \to 7$



and R_2 has the three rules

$G(x,x,y) \to x$
$G(x,y,x) \to x$
$G(y,x,x) \to x.$

Now R_1 and R_2 are both complete, but $R_1 \cup R_2$ is not:

$F(G(1,2,3), G(1,2,3), G(1,2,3), G(1,2,3)) \twoheadrightarrow$
$F(G(4,4,3), G(5,2,5), G(1,6,6), G(1,2,3)) \twoheadrightarrow$
$F(\quad 4, \qquad 5, \qquad 6, \qquad G(1,2,3)) \to$
$F(G(1,2,3), G(1,2,3), G(1,2,3), G(1,2,3)).$

4.8.2. REMARK. A simpler counterexample to the modularity of completeness is given in Drosten [89]: R_1 consists of rules $F(0, 1, x) \to F(x, x, x)$, $F(x, y, z) \to 2$, $0 \to 2$, $1 \to 2$ while R_2 consists of rules $D(x, y, y) \to x$, $D(x, x, y) \to y$. Now R_1, R_2 are complete; however, their disjoint sum is not. To see this, consider the term $F(M, M, M)$ where $M \equiv D(0, 1, 1)$ and show that $F(M, M, M)$ has a cyclic reduction.

The last counterexample (and also that in Remark 4.8.2) involves a non-leftlinear TRS. This is essential, as the following theorem indicates. First we define this concept:

4.8.3. DEFINITION. (i) A term is *linear* if it contains no multiple occurrences of the same variable,

non-linear otherwise. (E.g. G(x, x, y) is non-linear.)
(ii) A reduction rule $t \to s$ is *left-linear* if t is a linear term. (iii) A TRS is *left-linear* if all its reduction rules are left-linear.

4.8.4. THEOREM (Toyama, Klop & Barendregt [89a,b]).
Let $\mathcal{R}_1$, $\mathcal{R}_2$ be left-linear disjoint TRSs. Then: $\mathcal{R}_1 \cup \mathcal{R}_2$ is complete iff $\mathcal{R}_1$ and $\mathcal{R}_2$ are complete.

Some useful information concerning the inference of SN for $\mathcal{R}_1 \cup \mathcal{R}_2$ from the SN property for $\mathcal{R}_1$ and $\mathcal{R}_2$ separately is given in Rusinowitch [87] and Middeldorp [89b], in terms of 'collapsing' and 'duplicating' rewrite rules:

4.8.5. DEFINITION. (i) A rewrite rule $t \to s$ is a *collapsing* rule (c-rule) if s is a variable.
(ii) A rewrite rule $t \to s$ is a *duplicating* rule (d-rule) if some variable has more occurrences in s than it has in t.
<u>Example</u>: $F(x,x) \to G(x,x)$ is not a d-rule, but $F(x,x) \to H(x,x,x)$ is. Also $P(x) \to G(x,x)$ is a d-rule.

4.8.6. THEOREM. *Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be disjoint TRSs both with the property* SN.
(i) *If neither $\mathcal{R}_1$ nor $\mathcal{R}_2$ contain c-rules, $\mathcal{R}_1 \cup \mathcal{R}_2$ is* SN.
(ii) *If neither $\mathcal{R}_1$ nor $\mathcal{R}_2$ contain d-rules, $\mathcal{R}_1 \cup \mathcal{R}_2$ is* SN.
(iii) *If one of the TRSs $\mathcal{R}_1$, $\mathcal{R}_2$ contains neither c- nor d-rules, $\mathcal{R}_1 \cup \mathcal{R}_2$ is* SN.

Statements (i) and (ii) are proved in Rusinowitch [87]; statement (iii) is proved in Middeldorp [89b].

4.8.7. REMARK. (i)An equivalent way (due to E. Ohlebusch) of stating the theorem of Middeldorp and Rusinowitch is as follows: *Let $\mathcal{R}_1$, $\mathcal{R}_2$ be two disjoint terminating TRSs such that their union $\mathcal{R}_1 \cup \mathcal{R}_2$ is non-terminating. Then $\mathcal{R}_1$ contains a duplicating rule and $\mathcal{R}_2$ contains a collapsing rule, or vice versa.*
(ii) Another useful fact, proved in Middeldorp [89a], is that UN is a modular property.
(iii) It is an instructive exercise to prove that WN is a modular property.

4.8.8. EXAMPLES.

(i) Consider CL ∪ {D(x,x) → E}, Combinatory Logic with binary test for syntactic equality as in Table 4.6. Note that this is indeed a disjoint union. As we shall see in Topic 6, CL is confluent. By a simple exercise: the one rule TRS {D(x,x) → E} is confluent. Hence, by Toyama's theorem (4.8.1) the disjoint sum is confluent.

(ii) By contrast, the union CL ∪ {Dxx → E}, Combinatory Logic with applicative test for syntactic equality as in Table 4.7, is *not* confluent. (See Klop [80].) Note that this combined TRS is merely a union and not a disjoint union, since CL and {Dxx → E} have the function symbol Ap in common, even though hidden by the applicative notation.

(iii) Another application of Toyama's theorem (4.8.1): let $\mathcal{R}$ consist of the rules

if true then x *else* y	→ x
if false then x *else* y	→ y
if z *then* x *else* x	→ x.

(Here *true*, *false* are constants and *if - then - else* is a ternary function symbol.) Then CL ∪ $\mathcal{R}$ is confluent. Analogous to the situation in (ii), it is essential here that the if—then—else— construct is a ternary operator. For the corresponding applicative operator, the resulting TRS would not be confluent.

4.8.9. THEOREM (Middeldorp & Toyama [90]). *Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be constructor TRSs, possibly sharing constructor symbols but not defined symbols. Then:*

$$\mathcal{R}_1 \cup \mathcal{R}_2 \textit{ is complete } \Leftrightarrow \mathcal{R}_1, \mathcal{R}_2 \textit{ are complete.}$$

5

Critical Pair Completion

In Topic 4 we have seen that complete TRSs (i.e. TRSs that have the properties of CR and SN) are important for solving word problems. We continue with the question how to find a complete TRS (for the case of open terms, henceforth) for an equational specification (Σ, E). This is in fact what the Knuth-Bendix completion algorithm is trying to do. We will briefly explain the essential features of the completion algorithm, by an informal, "intuition-guided"completion of the equational specification of groups, the paradigm example treated extensively in Knuth & Bendix [70]. This paper was the start of the development of the critical pair completion technology. The following axioms E for groups are given.

E

$e.x$	$= x$
$I(x).x$	$= e$
$(x \cdot y) \cdot z$	$= x \cdot (y \cdot z)$

Table 5.1

First we give these equations a 'sensible' orientation:

1.	$e \cdot x$	$\to$	x
2.	$I(x) \cdot x$	$\to$	e

3. $(x\cdot y)\cdot z \;\to\; x\cdot(y\cdot z)$

(Note that the orientation in rules 1, 2 is forced, by the restrictions on the format of rewrite rules as defined in Topic 4. As to the orientation of rule 3, the other direction is just as 'sensible'.) These rules are not confluent, as can be seen by superposition of e.g. 2 and 3. Redex $I(x)\cdot x$ can be unified (after variable renaming) with a *non-variable* subterm of redex $(x\cdot y)\cdot z$ (the underlined subterm), with result $(I(x)\cdot x)\cdot z$. This term is subject to two possible reductions: $(I(x)\cdot x)\cdot z \to_2 e\cdot z$ and $(I(x)\cdot x)\cdot z \to_3 I(x)\cdot(x\cdot z)$. The pair of reducts $\langle e\cdot z, I(x)\cdot(x\cdot z)\rangle$ is called a *critical pair*, since the confluence property depends on the reduction possibilities of the terms in this pair. Formally, we have the following definition which at a first reading is not easily digested. The concept of a 'most general unifier' is supposed to be known.

5.1. DEFINITION. Let $\alpha \to \beta$ and $\gamma \to \delta$ be two rewrite rules such that α is *unifiable* (after renaming of variables) with a subterm of γ which is not a variable (a non-variable subterm). This means that there is a context C[ ], a non-variable term t and a 'most general unifier' σ such that $\gamma \equiv C[t]$ and $t^\sigma \equiv \alpha^\sigma$. The term $\gamma^\sigma \equiv C[t]^\sigma$ can be reduced in two possible ways: $C[t]^\sigma \to C[\beta]^\sigma$ and $\gamma^\sigma \to \delta^\sigma$.

Now the pair of reducts $\langle C[\beta]^\sigma, \delta^\sigma\rangle$ is called a *critical pair* obtained by the superposition of $\alpha \to \beta$ on $\gamma \to \delta$. If $\alpha \to \beta$ and $\gamma \to \delta$ are the same rewrite rule, we furthermore require that α is unifiable with a proper (i.e. not $\equiv \alpha$) non-variable subterm of $\gamma \equiv \alpha$.

5.2. DEFINITION. A critical pair $\langle s, t\rangle$ is called *convergent* if s and t have a common reduct.

Our last critical pair $\langle e\cdot z, I(x)\cdot(x\cdot z)\rangle$ is not convergent: $I(x)\cdot(x\cdot z)$ is a normal form and $e\cdot z$ only reduces to the normal form z. So we have the problematic pair of terms z, $I(x)\cdot(x\cdot z)$; problematic because their equality is derivable from E, but they have no common reduct with respect to the reduction available so far. Therefore we adopt a new rule

4. $I(x)\cdot(x\cdot z) \to z$

Now we have a superposition of rule 2 and 4: $I(I(y))\cdot(I(y)\cdot y) \to_4 y$ and $I(I(y))\cdot(I(y)\cdot y) \to_2 I(I(y))\cdot e$. This yields the critical pair $\langle y, I(I(y))\cdot e\rangle$ which cannot further be reduced. Adopt new rule:

5. $I(I(y))\cdot e \to y$ ***canceled later***

As it will turn out, in a later stage this last rule will become superfluous. We go on searching for critical pairs:

Superposition of 4, 1: $I(e)\cdot(e\cdot z) \to_4 z$ and $I(e)\cdot(e\cdot z) \to_1 I(e)\cdot z$.

Adopt new rule:

6. $I(e)\cdot z \to z$ ***canceled later***

Superposition of 3, 5: $(I(Iy))\cdot e)\cdot x \to_3 I(I(y))\cdot(e\cdot x)$ and $(I(Iy))\cdot e)\cdot x \to_5 y\cdot x$.

Adopt new rule:

7. $I(Iy))\cdot x \to y\cdot x$ ***canceled later***

Superposition of 5, 7: $I(I(y))\cdot e \to_7 y.e$ and $I(I(y))\cdot e \to_5 y$.

Adopt new rule:

8. $y\cdot e \to y$

Superposition of 5, 8: $I(I(y))\cdot e \to_5 y$ and $I(I(y))\cdot e \to_8 I(I(y))$.

Adopt new rule

9. $I(I(y)) \to y$ ***cancel 5 and 7***

(Rule 5 is now no longer necessary to ensure that the critical pair $\langle y, I(I(y))\cdot e\rangle$ has a common reduct, because: $I(I(y))\cdot e \to_9 y\cdot e \to_8 y$. Likewise for rule 7.)

Superposition of 6, 8: $I(e)\cdot e \to_6 e$ and $I(e)\cdot e \to_8 I(e)$.

Adopt new rule

10. $I(e) \to e$ ***cancel 6***

Superposition of 2, 9: $I(I(y))\cdot I(y) \to_2 e$ and $I(I(y))\cdot I(y) \to_9 y\cdot I(y)$.

Adopt new rule

11. $y\cdot I(y) \to e$

Superposition of 3, 11: $(y\cdot I(y))\cdot x \to_3 y\cdot(I(y)\cdot x)$ and $(y\cdot I(y))\cdot x \to_{11} e\cdot x$.

Adopt new rule

12. $y\cdot(I(y)\cdot x) \to x$

Superposition (again) of 3, 11: $(x\cdot y)\cdot I(x\cdot y) \to_{11} e$ and $(x\cdot y)\cdot I(x\cdot y) \to_3 x\cdot(y\cdot I(x\cdot y))$.

Adopt new rule

13. $x\cdot(y\cdot(y\cdot I(x\cdot y)) \to e$ ***canceled later***

Superposition of 13, 4: $I(x)\cdot(x\cdot(y\cdot I(x\cdot y))) \to_4 y\cdot I(x\cdot y)$ and $I(x)\cdot(x\cdot(y\cdot I(x\cdot y))) \to_{13} I(x)\cdot e$.

Adopt new rule

14. $y\cdot I(x\cdot y) \to I(x)$ ***canceled later***
cancel 13

Superposition of 4, 14: $I(y)\cdot(y\cdot I(x\cdot y)) \to_4 I(x\cdot y)$ and $I(y)\cdot(y\cdot I(x\cdot y)) \to_{14} I(y)\cdot I(x)$.

Adopt new rule

15. $I(x\cdot y) \to I(y)\cdot I(x)$ ***cancel 14***

At this moment the TRS has only convergent critical pairs, e.g.:

$$\begin{array}{ccc} I(y\cdot I(y)) & \to_{15} & I(I(y))\cdot I(y) \\ & & \downarrow_{9} \\ \downarrow_{11} & & y\cdot I(y) \\ & & \downarrow_{11} \\ I(e) & \to_{10} & e \end{array}$$

The significance of this fact is stated in the following lemma. The proof is a matter of a straightforward case analysis, as suggested in Figure 5.1.

5..3. CRITICAL PAIR LEMMA (Knuth & Bendix [70], Huet [80]).

A TRS R *is WCR iff all critical pairs are convergent*.

1.	$e\cdot x$	$\to$	x
2.	$I(x)\cdot x$	$\to$	e
3.	$(x\cdot y)\cdot z$	$\to$	$x\cdot(y\cdot z)$
4.	$I(x)\cdot(x\cdot z)$	$\to$	z
8.	$y\cdot e$	$\to$	y
9.	$I(I(y))$	$\to$	y
10.	$I(e)$	$\to$	e
11.	$y\cdot I(y)$	$\to$	e
12.	$y\cdot(I(y)\cdot x)$	$\to$	x
15.	$I(x\cdot y)$	$\to$	$I(y)\cdot I(x)$

Table 5.2

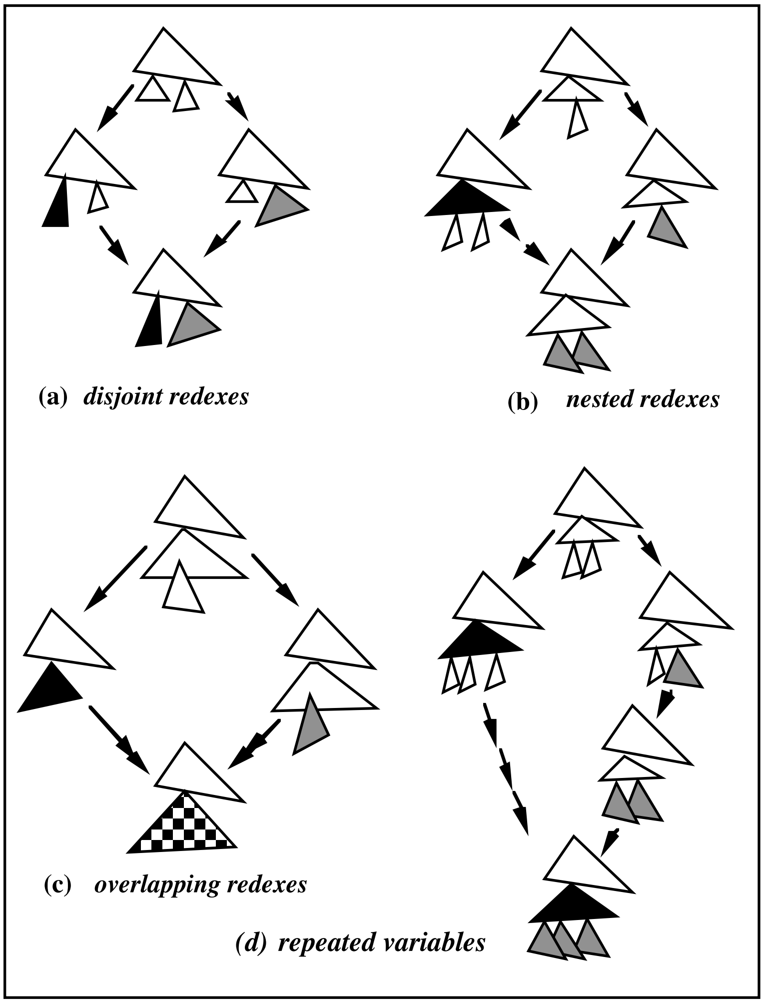


Figure 5.1

So the TRS R_c with rewrite rules as in Table 5.4 is WCR.

Furthermore, one can prove SN for R_c by the recursive path ordering technique explained in Topic 7. (In fact we need the extended lexicographic version, due to the presence of the associativity rule.) According to Newman's Lemma R_c is therefore CR and hence complete. We conclude that the validity problem for the equational specification of groups is solvable.

The following theorem of Knuth and Bendix is an immediate corollary of the Critical Pair Lemma 5.3 and Newman's Lemma:

5.4. COROLLARY (Knuth & Bendix [70]). *Let* R *be a TRS which is SN. Then* R *is CR iff all critical pairs of* R *are convergent.*

The completion procedure above by hand was naive, since we were not very systematic in searching for critical pairs, and especially since we were guided by an intuitive sense only of what direction to adopt when generating a new rule. In most cases there was no other possibility (e.g. at 4: $z \to I(x)\cdot(x\cdot z)$ is not a reduction rule due to the restriction that the lefthand-side is not a single variable), but in case 15 the other direction was at least as plausible, as it is even length-decreasing. However, the other direction $I(y)\cdot I(x) \to I(x\cdot y)$ would have led to disastrous complications (described in Knuth & Bendix [70]).

The problem of what direction to choose is solved in the actual Knuth-Bendix algorithm and its variants by preordaining a 'reduction ordering' on the terms.

5.5. DEFINITION. A *reduction ordering* > is a well-founded partial ordering among terms, which is closed under substitutions and contexts, i.e. if $s > t$ then $s^\sigma > t^\sigma$ for all substitutions σ, and if $s > t$ then $C[s] > C[t]$ for all contexts C[ ].

We now have immediately the following fact (noting that if R is SN, then $\to_R^+$ satisfies the requirements of Definition 5.5):

5.6. PROPOSITION. *A TRS* R *is SN iff there is a reduction ordering > such that* $\alpha > \beta$ *for every rewrite rule* $\alpha \to \beta$ *of* R.

In Figure 5.2 a simple version of the Knuth-Bendix completion algorithm is presented.
The program of Figure 5.2 has three possibilities: it may (1) terminate successfully, (2) loop infinitely, or (3) fail because a pair of terms s,t cannot be oriented (i.e. neither $s > t$ nor $t > s$). The third case gives the most important restriction of the Knuth-Bendix algorithm: equational specifications with commutative operators cannot be completed.

In case (1) the resulting TRS is complete. To show this requires a non-trivial proof, see e.g. Huet [81], or for a general method for such correctness proofs, Bachmair, Dershowitz & Hsiang [86].

Simple version of the Knuth-Bendix completion algorithm

Input: - an equational specification (Σ, E)
- a reduction ordering > on $Ter(\Sigma)$ (i.e. a program which computes >)

Output: - a complete TRS R such that for all $s,t \in Ter(\Sigma)$: $s =_R t \Leftrightarrow (\Sigma, E) \vdash s = t$

```
R := Ø;
while E ≠ Ø do
    choose an equation s = t ∈ E;
    reduce s and t to respective normal forms s' and t' with respect to R;
    if s' ≡ t' then
        E := E - {s = t}
    else
        if s' > t' then
            α := s'; β := t'
        else if t' > s' then
            α := t'; β := s'
        else
            failure
        fi;
        CP := {P = Q | ⟨P, Q⟩ is a critical pair between the rules in R and α → β};
        R := R ∪ {α → β};
        E := E ∪ CP - {s = t}
    fi
od;
success
```

Figure 5.2

5.7. EXAMPLE. Knuth & Bendix [70] contains completions of two specifications which closely resemble the specification of groups (see Table 5.3), called ‘L-R theory’ and ‘R-L theory’.

Using the completions, it is easy to see that $x \cdot e = x$ is not derivable in L-R theory and that in R-L theory the equations $e \cdot x = x$ and $x \cdot I(x) = e$ are not derivable. Furthermore, in L-R theory the equation $x \cdot e = x$ is not derivable. Hence the three theories are different, i.e. determine different varieties of algebras.

In fact, the variety of groups is the intersection of both the variety of L-R algebras and that of R-L algebras, and the latter two varieties are incomparable with respect to set inclusion. (See Figure 5.3.)

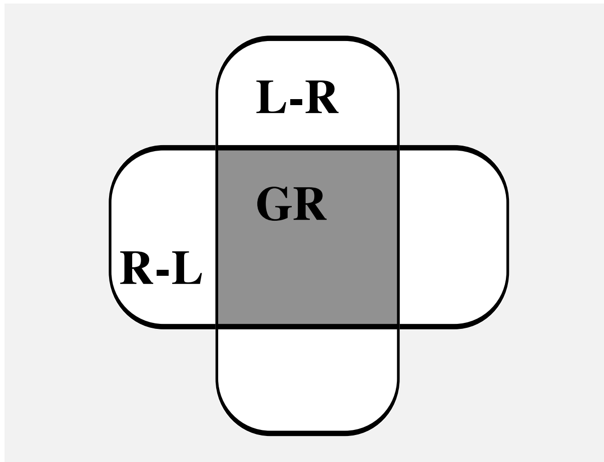


Figure 5.3

group theory	*L-R theory:*	*R-L theory:*
$e \cdot x = x$	$e \cdot x = x$	$x \cdot e = x$
$I(x) \cdot x = e$	$x \cdot I(x) = e$	$I(x) \cdot x = e$
$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$
completion:	*completion:*	*completion:*
$e \cdot x \rightarrow x$	$e \cdot x \rightarrow x$	
$x \cdot e \rightarrow x$		$x \cdot e \rightarrow x$
$I(x) \cdot x \rightarrow e$		$I(x) \cdot x \rightarrow e$
$x \cdot I(x) \rightarrow e$	$x \cdot I(x) \rightarrow e$	
$(x \cdot y) \cdot z \rightarrow x \cdot (y \cdot z)$	$(x \cdot y) \cdot z \rightarrow x \cdot (y \cdot z)$	$(x \cdot y) \cdot z \rightarrow x \cdot (y \cdot z)$
$I(e) \rightarrow e$	$I(e) \rightarrow e$	$I(e) \rightarrow e$
$I(x \cdot y) \rightarrow I(y) \cdot I(x)$	$I(x \cdot y) \rightarrow I(y) \cdot I(x)$	$I(x \cdot y) \rightarrow I(y) \cdot I(x)$
$x \cdot (I(x) \cdot y) \rightarrow y$	$x \cdot (I(x) \cdot y) \rightarrow y$	
		$e \cdot x \rightarrow I(I(x))$
$I(x) \cdot (x \cdot y) \rightarrow y$	$I(x) \cdot (x \cdot y) \rightarrow y$	
		$x \cdot I(I(y) \rightarrow x \cdot y$
$I(I(x)) \rightarrow x$		
	$x \cdot e \rightarrow I(I(x))$	
	$I(I(I(x))) \rightarrow I(x)$	$I(I(I(x))) \rightarrow I(x)$
		$x \cdot (y \cdot I(y)) \rightarrow x$
	$I(I(x)) \cdot y \rightarrow x \cdot y$	
		$x \cdot (I(I(y)) \cdot z) \rightarrow x \cdot (y \cdot z)$
		$x \cdot (y \cdot (I(y) \cdot z)) \rightarrow x \cdot z$
		$I(x) \cdot (x \cdot y) \rightarrow I(I(y))$

Table 5.3

6

Orthogonal term rewriting systems

In the preceding sections we have considered general properties of TRSs and how these properties are related; among them the most important property, confluence, with its consequence of uniqueness of normal forms. We will now consider a special class of TRSs, the orthogonal ones, which all have the confluence property as well as various other desirable properties concerned with reduction strategies.

6.1. DEFINITION. (i) A TRS R is *orthogonal* if R is left-linear and there are no critical pairs (Definition 5.1).
(ii) R is *weakly orthogonal* if R is left-linear and R contains only trivial critical pairs, i.e. if $\langle t, s \rangle$ is a critical pair then $t \equiv s$.

Left-linear means that the lefthand-sides of the rewrite rules contain no duplicated variables (Definition 4.8.3.) One problem with non-left-linear rules is that their application requires a test for syntactic equality of the arguments substituted for the variables occurring more than once. As terms may be very large, this may be very laborious. Another problem is that the presence of non-left-linear rules may destroy the CR property, as noted in Topic 4.

In Definition 5.1 we have already defined the notion of 'critical pair'. Since that definition is often found difficult, we will now explain the *absence* of critical pairs in a more ‘intuitive’ way. Let R be the TRS as in Table 6.1:

r_1	F(G(x, S(0)), y, H(z))	→ x
r_2	G(x, S(S(0)))	→ 0
r_3	P(G(x, S(0)))	→ S(0)

Table 6.1

Call the context F(G(❐, S(0)), ❐, H(❐)) the *pattern* of rule r_1. (Earlier, we defined a context as a term with exactly one hole ❐, but it is clear what a context with more holes is.) In tree form the pattern is the shaded area as in Figure 6.1.

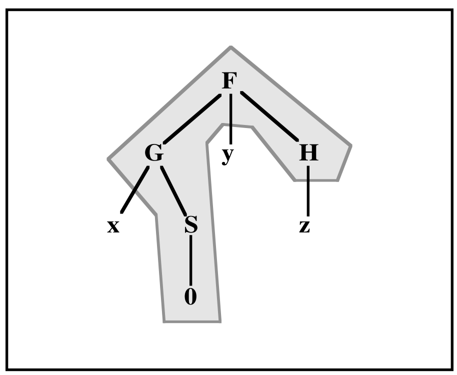


Figure 6.1

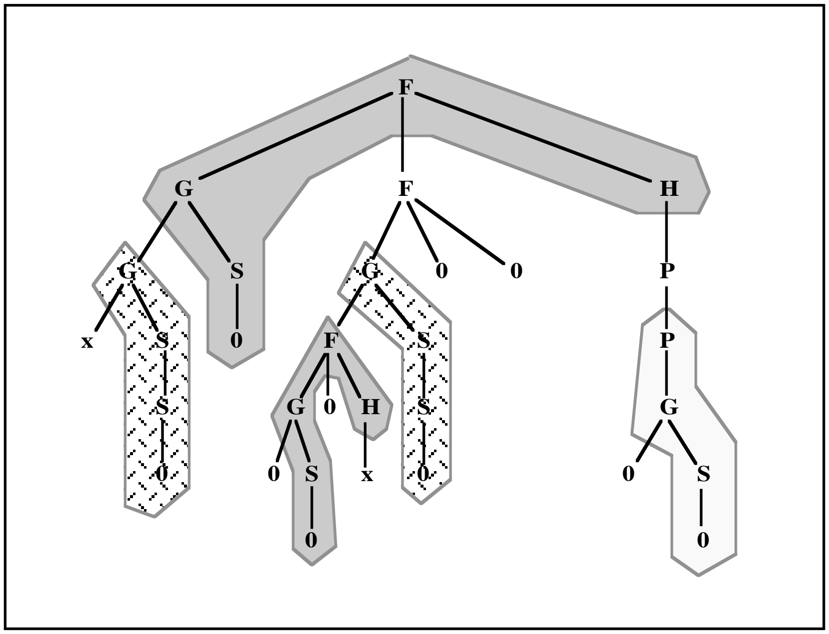


Figure 6.2

For a left-linear rule it is only its pattern that 'counts'.

The TRS R in Table 6.1 has the property that *in no term patterns can overlap,* i.e. R has the non-overlapping or non-ambiguity property. Figure 8.2 shows a term in R with all patterns indicated, and indeed they do not overlap.

Overlap can already occur in one rule, e.g. in the rule L(L(x)) → 0; see Figure 6.3(a). An overlap at the root (of the tree corresponding to a term), arising from the rules F(0,x,y) → 0, F(x,1,y) → 1, is shown in Figure 6.3(b). Another overlap at the root, arising from the rules for the non-deterministic or: or(x,y) → x, or(x,y) → y, is shown in Figure 6.3(c).

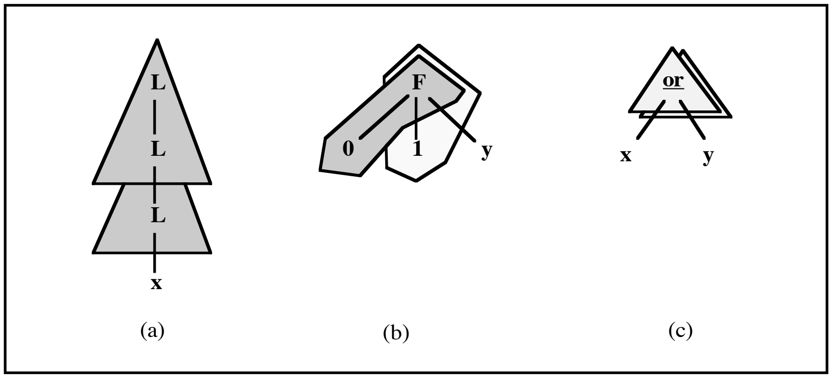


Figure 6.3

We will now formulate and sketch the proofs of the basic theorems for orthogonal TRSs. To that end, we need the notion of '*descendant*' in a reduction. Somewhat informally, this notion can be introduced as follows:

Let t be a term in a orthogonal TRS R, and let $s \subseteq t$ be a redex whose head symbol we will give a marking, say by underlining it, to be able to 'trace' it during a sequence of reduction (rewrite) steps. Thus if $s \equiv F(\mathbf{t})$, it is marked as $\underline{F}(\mathbf{t})$**.** First consider the rewrite step $t \rightarrow_{s'} t'$, obtained by contraction of redex s' in t. We wish to know what has happened in this step to the marked redex s. The following cases can be distinguished, depending on the relative positions of s and s' in t:

Case 1. The occurrences of s' *and* s *in* t *are disjoint*. Then we find back the marked redex s, unaltered, in t'.

Case 2. The occurrences of s *and* s' *coincide*. Then the marked redex has disappeared in t'; t' does not contain an underlined symbol.

Case 3. s' *is a proper subterm of the marked redex* s (so s' is a subterm of one of the arguments of s). Then we find back the marked redex, with some possible change in one of the arguments. (Here we need the orthogonality of R; otherwise the marked redex could have stopped being a redex in t' after the 'internal' contraction of s'.)

Case 4. s *is a proper subterm of* s'. Then the marked redex s is n times multiplied for some $n \geq 0$; if $n = 0$ s is erased in t'. The reduct t' now contains n copies of the marked redex, all of them still marked.

Now the marked redexes in t' are called the *descendants* of $s \subseteq t$ in the reduction step $t \rightarrow_{s'} t'$. It is obvious how to extend this definition by transitivity to sequences of rewrite steps

$$t \rightarrow_{s'} t' \rightarrow_{s''} t'' \rightarrow \ldots t^{(n-1)} \rightarrow_{s(n)} t^{(n)}.$$

6.2. PROPOSITION. *Let* R *be a orthogonal* TRS, $t \in \text{Ter}(R)$. *Let* t *contain, possibly among others, the mutually disjoint redexes* $s_1,...,s_n$. *Let these redexes be marked by underlining their head symbol.*

Furthermore, suppose that t t' *is the sequence of* n *rewrite steps obtained by contraction of all redexes* s_i *(in some order), and let* $t \rightarrow_s t''$ *be a rewrite step obtained from contracting redex* s. *(See Figure 64(a).)*

Then a common reduct t''' *of* t', t'' *can be found by contracting in* t'' *all marked redexes (which still are mutually disjoint). The reduction* $t' \twoheadrightarrow t'''$ *consists of the contraction of all descendants of* s *in* t' *after the reduction* $t \twoheadrightarrow t'$. ❒

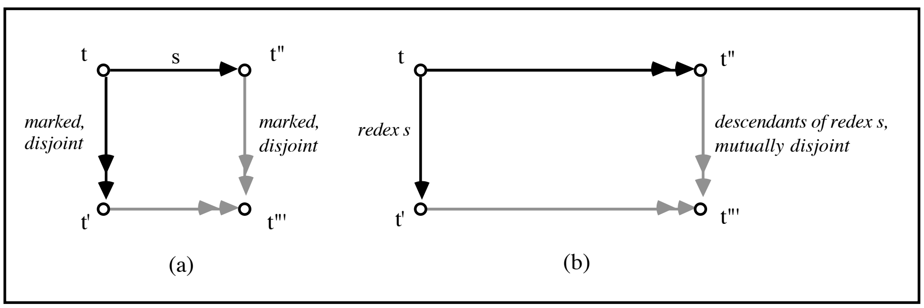


Figure 6.4

The proof is a matter of easy casuistics, left to the reader. An immediate corollary is the 'Parallel Moves Lemma':

6.3. PARALLEL MOVES LEMMA. *We consider reductions in an orthogonal TRS. Let* $t \twoheadrightarrow t''$, *and let* $t \rightarrow_s t'$ *be a one step reduction by contraction of redex* s. *Then a common reduct* t''' *of* t', t'' *can be found by contraction in* t'' *of all descendants of redex* s, *which are mutually disjoint. (See Figure 64(b).)* ❒

By repeated application of the Parallel Moves Lemma we now have

6.4. THEOREM. *Every orthogonal TRS is confluent.*

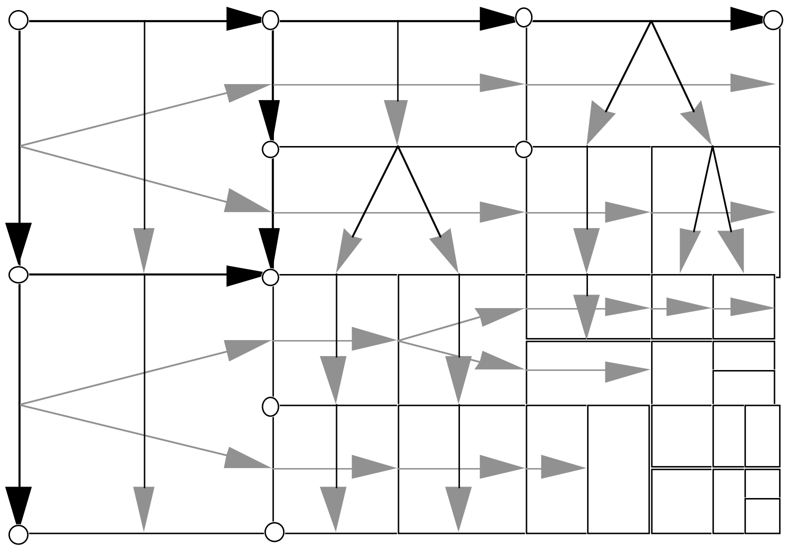


Figure 6.5

In fact, an analysis of the Parallel Moves Lemma yields more than mere confluence; it also asserts that common reducts can be found by means of a certain procedure, as suggested in Figure 6.5, namely by repeatedly adjoining 'elementary reduction diagrams'. We encountered this tiling procedure also in Topic 2 and 3; but this time the success of the tiling procedure is due to the orthogonality, not the decreasingness property. The shaded arrows in Figure 6.5 suggest how reduction steps propagate through the diagram (in an orthogonal fashion), corresponding to the notion of descendant.

By the same arguments we can also prove the confluence theorem for weakly orthogonal TRSs, including the strong version of confluence referring to diagram construction as just described.

The earliest proof of Theorem 8.4 is probably that of Rosen [73]; but earlier proofs of the confluence of CL (Combinatory Logic), work just as well for orthogonal TRSs in general. The confluence theorem for (weakly) orthogonal TRSs is also a special case of a theorem of Huet. We need a definition first:

8.5. DEFINITION. (Parallel reduction) $t \twoheadrightarrow_{||} s$ if $t \twoheadrightarrow s$ via a reduction of disjoint redexes.

8.6. THEOREM (Huet [80]). *Let* R *be a left-linear TRS. Suppose for every critical pair* $\langle t,s \rangle$ *we have* $t \twoheadrightarrow_{||} s$. *Then* $\twoheadrightarrow_{||}$ *is strongly confluent, hence* R *is confluent.* ❐

(For the definition of 'strongly confluent' see Topic 2, Example 2.1.) Note that confluence for orthogonal TRSs is a corollary by absence of critical pairs. Likewise for weakly ortogonal TRSs, by the triviality of their critical pairs.

8.7. EXAMPLES. (i) Combinatory Logic (Table 4.4) has rule patterns as in Figure 8.6; they cannot overlap. As CL is left-linear, it is therefore orthogonal and hence confluent.

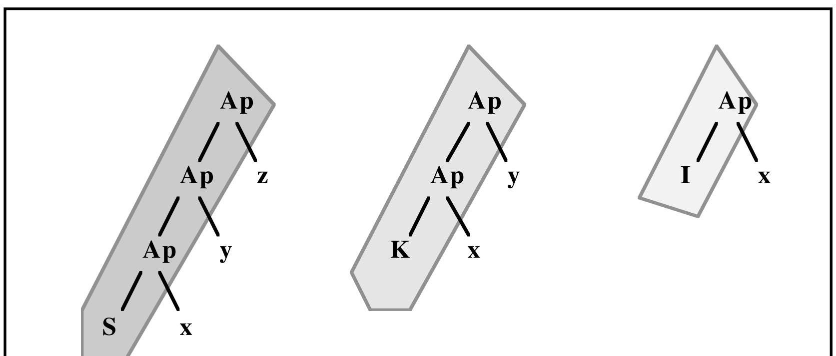


Figure 8.6

(ii) SKIM, in Table 4.5, is orthogonal.

(iii) Combinatory Logic with *parallel or*, CL ∪ <u>or</u>(x, <u>true</u>) → <u>true</u>, <u>or</u>(<u>true</u>, x) → <u>true</u>, is weakly orthogonal: the only critical pair is <<u>true</u>, <u>true</u>>, hence confluent.

(iv) Combinatory Logic with *nondeterministic choice,* CL ∪ <u>or</u>(x, y) → x, <u>or</u>(x, y) → y, is weakly orthogonal: the only critical pair is $\langle x, x \rangle$, hence confluent.

(iv) A *Recursive Program Scheme (RPS)* is a TRS with

- a finite set of function symbols F ={F_1,...,F_n} (the 'unknown' functions), where F_i has arity $m_i \geq 0$ (i = 1,...,n), and
- a finite set G = {G_1,...,G_k} (the 'known' or 'basic' functions), disjoint from F, where G_j has arity $p_j \geq 0$ (j = 1,...,k).
- The reduction rules of R have the form

$$F_i(x_1,...,x_{m_i}) \to t_i \qquad (i = 1,...,n)$$

where all the displayed variables are pairwise different and where t_i is an arbitrary term built from operators in F,G and the displayed variables. For each F_i (i = 1,...,n) there is exactly one rule.

Example:

$$F_1(x) \to G_1(x,F_1(x),F_2(x,x))$$
$$F_2(x,y) \to G_2(F_2(x,x),F_1(G_3))$$

Every RPS is orthogonal, hence confluent.

Apart from confluence, many interesting facts can be proved for orthogonal TRSs.

6.8. DEFINITION. (i) A TRS is *non-erasing* if in every rule t → s the same variables occur in t and in s. (E.g. CL is not non-erasing, due to the rule Kxy → x.)

(ii) A TRS is *weakly innermost normalizing* (WIN) if every term has a normal form which can be reached by an *innermost* reduction. (In an innermost reduction a redex may only be 'contracted' if it contains no proper subredexes.)

The next theorem was proved in Church [41] for the case of the non-erasing version of λ-calculus, the λI-calculus, where the restriction on term formation is adopted stating that in every abstraction term λx.M the variable x must have a free occurrence in M.

6.9. THEOREM. *Let* R *be orthogonal and non-erasing. Then:* R *is WN iff* R *is SN.* ❐

Another useful theorem, which also reduces the burden of a termination (SN) proof for orthogonal TRSs, is:

6.10. THEOREM (O'Donnell [77]). *Let* R *be an orthogonal TRS. Then:* R *is WIN iff* R *is SN.* ❐

The last two theorems can be refined to terms: call a term WN if it has a normal form, SN if it has no infinite reductions, WIN if it has a normal form reachable by an innermost reduction. The 'local' version of Theorem 6.9 then says that for a term in an orthogonal, non-erasing TRS the properties WN and SN coincide. Likewise there is a local version of Theorem 6.10. Thus, if in CL a term can be normalized via an innermost reduction, all its reductions are finite.

6.11. EXERCISE. In this exercise we sketch a proof of 'Church's Theorem' 6.9 and O'Donnell's Theorem 6.10.
(i) The following proposition has an easy proof:

6.11.1. PROPOSITION. Let t be a term in an orthogonal TRS, containing mutually disjoint redexes $s_1,...,s_n$, and a redex s. Let $t \twoheadrightarrow t'$ be the n-step reduction obtained by contraction, in some order, of the redexes $s_1,...,s_n$. Suppose s has after the reduction $t \twoheadrightarrow t'$ no descendants in t'.

Then for some $i \in \{1,...,n\}$: $s \subseteq s_i$. This means that either s coincides with some s_i, or is contained in an argument of some s_i.

(ii) We write "$\infty(t)$" if the term t has an infinite reduction $t \to \to ...$. So $\infty(t)$ iff t is not SN. Using Proposition 6.11.1 one can now prove (the proof is non-trivial):

6.11.2. ERASURE LEMMA. Let t be a term in an orthogonal TRS such that $\infty(t)$. Let $t \to_s t'$ be a reduction step such that $\neg \infty(t')$. Then the redex s must contain a proper subterm p with $\infty(p)$ that is erased in the step $t \to_s t'$ (i.e. has no descendants in t').

(iii) Using the Lemma it is now easy to prove Theorem 6.10: 'critical' steps $t \to t'$ in which $\infty(t)$ but $\neg \infty(t')$, cannot occur in a non-erasing TRS.

(iv) Also Theorem 6.11 follows immediately from the Lemma 6.11.2 by observing that an innermost contraction cannot erase a proper subterm which admits an infinite reduction, since otherwise the contracted redex would not have been innermost.

6.12. Reduction strategies. Terms in a TRS may have a normal form as well as admitting infinite reductions. So, if we are interested in finding normal forms, we should have some strategy at our disposal telling us what redex to contract in order to achieve that desired result. We will in this section present some strategies which are guaranteed to find the normal form of a term whenever such a normal form exists. We will adopt the restriction to orthogonal TRSs.

The strategies below will be of two kinds: one-step or sequential strategies (which point in each reduction step to just one redex as the one to contract) and many-step or parallel strategies (in which a set of redexes is contracted simultaneously). Of course all strategies must be computable.

Apart from the objective of finding a normal form, we will consider the objective of finding a 'best possible' reduction even if the term at hand does not have a normal form.

6.12.1. DEFINITION. Let R be a TRS.

(i) A *sequential reduction strategy* $\mathbb{F}$ *for* R is a map $\mathbb{F}$: Ter(R) → Ter(R) such that

(1) $t \equiv \mathbb{F}(t)$ if t is a normal form,

(2) $t \to \mathbb{F}(t)$ else.

(ii) A *parallel reduction strategy* $\mathbb{F}$ *for* R is a map $\mathbb{F}$: Ter(R) → Ter(R) such that

(1) $t \equiv \mathbb{F}(t)$ if t is a normal form,

(2) $t \to^+ \mathbb{F}(t)$ else.

Here $\to^+$ is the transitive (but not reflexive) closure of →. Instead of 'sequential' and 'parallel' we will also say 'one-step' and 'many-step', respectively.

6.12.2. DEFINITION. (i) A reduction strategy (one step or many step) $\mathbb{F}$ for R is *normalizing* if for each term t in R having a normal form, the sequence $\{\mathbb{F}^n(t) \mid n \geq 0\}$ contains a normal form.

(ii) $\mathbb{F}$ is *cofinal* if for each t the sequence $\{\mathbb{F}^n(t) \mid n \geq 0\}$ is cofinal in $\mathcal{G}(t)$, the reduction graph of t. (See Topic 1, Ex.13.8 for 'cofinal' and see Figure 6.7.)

A normalizing reduction strategy is good, but a cofinal one is even better: it finds, when applied on term t, the best possible reduction sequence starting from t (or rather, a best possible) in the following sense. Consider a reduction $t \to s$ as a gain in information; thus normal forms have maximum information. In case there is no normal form in $\mathcal{G}$(t), one can still consider infinite reductions as developing more and more information. Now the cofinal reductions $t \equiv t_0 \to t_1 \to t_2 \to \ldots$ are optimal since for every t' in $\mathcal{G}$(t) they contain a t_n with information content no less than that of t' (since $t' \twoheadrightarrow t_n$ for some t_n, by definition of 'cofinal').

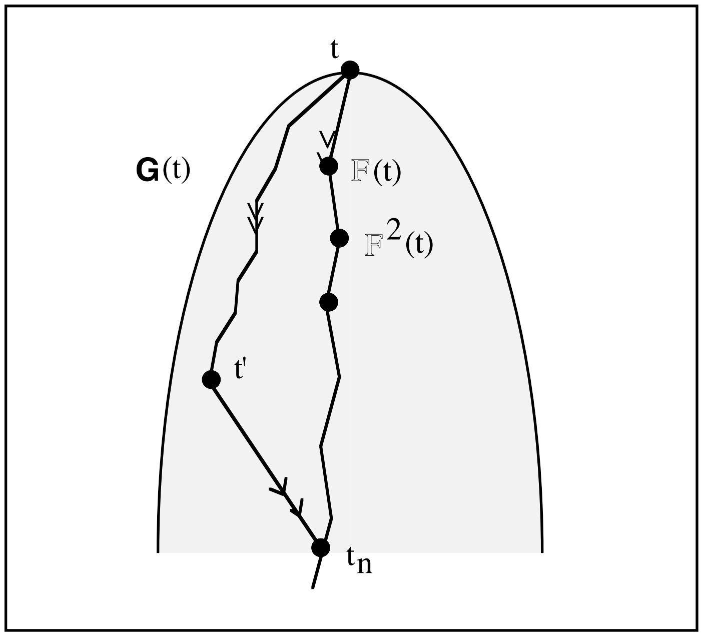


Figure 6.7

We now present some well-known reduction strategies.

6.12.3. DEFINITION. (i) The *leftmost-innermost* (one step) strategy is the strategy in which in each step the leftmost of the minimal or innermost redexes is contracted.

(ii) The *parallel-innermost* (many step) strategy contracts simultaneously all innermost redexes. Since these are pairwise disjoint, this is no problem.

(iii) The *leftmost-outermost* (one step) strategy: in each step the leftmost redex of the maximal (or outermost) redexes is contracted. Notation: $\mathbb{F}_{lm}$.

(iv) The *parallel-outermost* (many step) strategy contracts simultaneously all maximal redexes; since these are pairwise disjoint, this is no problem. Notation: $\mathbb{F}_{po}$.

(v) The *full substitution rule* (or *Kleene reduction*, or *Gross-Knuth reduction*): this is a many-step strategy in which all redexes are simultaneously contracted. Notation: $\mathbb{F}_{GK}$.

Strategies (i)-(iv) are well-defined for general TRSs. Strategy (v) is only defined for orthogonal TRSs, since for a general TRS it is not possible to define an unequivocal result of simultaneous reduction of a set of possibly nested redexes. The five strategies are illustrated by Figure 6.8, for the following TRS:

and(true, x) → x
and(false, x) → false
or(true, x) → true
or(false, x) → x

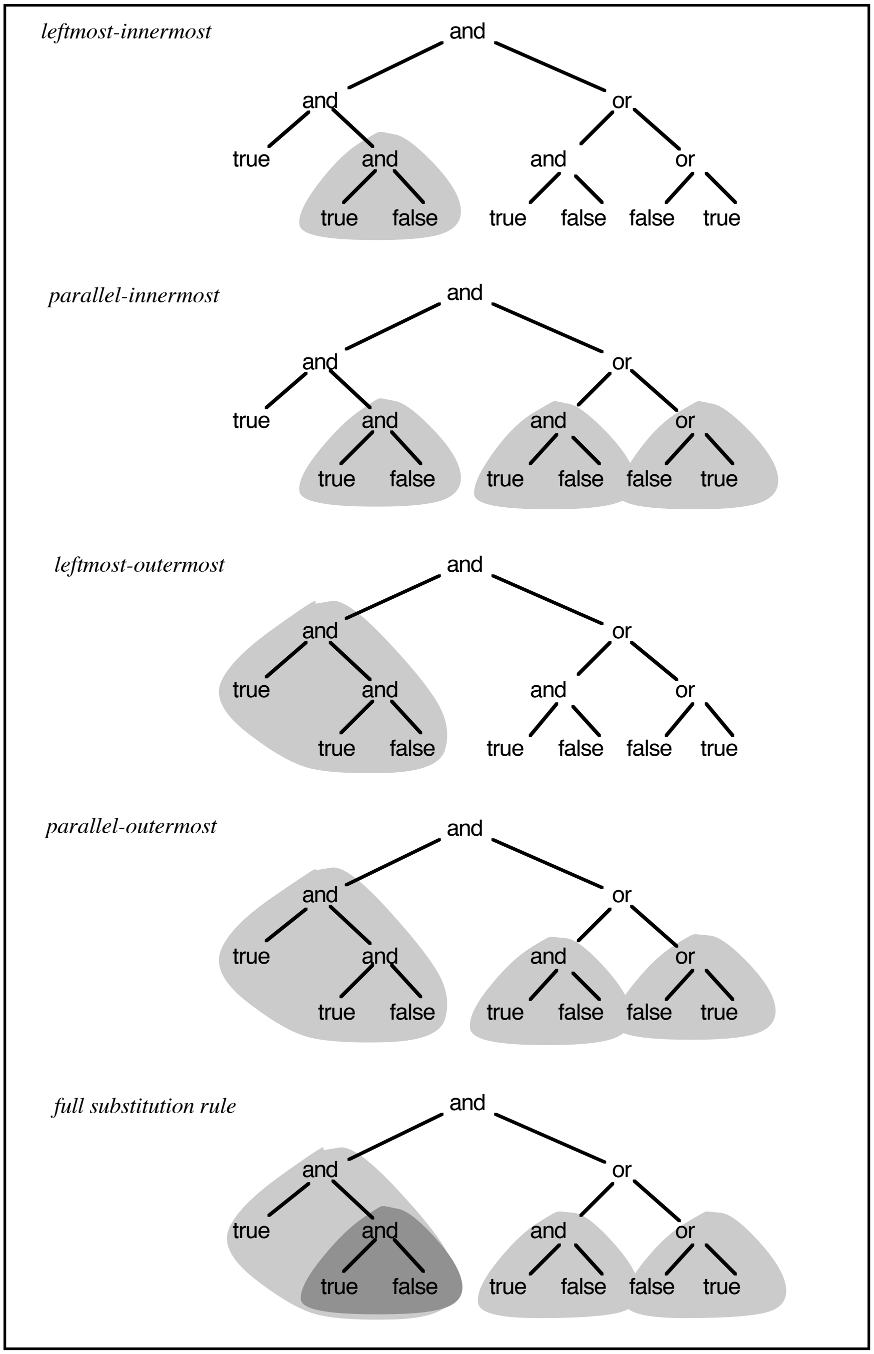


Figure 6.8

We will be mainly interested here in the strategies (iii)-(v), for a reason that will be clear by inspection of Table 3.2 below. We have the following facts (for proofs see O'Donnell [77] or Klop [80]):

6.12.4. THEOREM. *For orthogonal TRSs:*

(i) $\mathbb{F}_{GK}$ *is a cofinal reduction strategy.*

(ii) $\mathbb{F}_{po}$ *is a normalizing reduction strategy.*

6.12.5. REMARK. For λ-calculus this theorem also holds. Moreover, $\mathbb{F}_{lm}$ is there also a normalizing strategy, just as it is for the orthogonal TRS CL (Combinatory Logic). However, in general $\mathbb{F}_{lm}$ is not a normalizing strategy for orthogonal TRSs.

Even though $\mathbb{F}_{lm}$ is in general for orthogonal TRSs not normalizing, there is a large class of orthogonal TRSs for which it is:

6.12.6. DEFINITION. (O'Donnell [77]). An orthogonal TRS is *left-normal* if in every reduction rule $t \to s$ the constant and function symbols in the left-hand side t precede (in the linear term notation) the variables.

6.12.6.1. EXAMPLE. (i) CL (Combinatory Logic) is left-normal. (ii) RPSs (Recursive Program Schemes) as defined in Examples 8.7(iii) are all left-normal. (iii) $F(x, B) \to D$ is not left-normal; $F(B, x) \to D$ is left-normal.

6.12.6.2. THEOREM. (O'Donnell [77]).
Let R *be a left-normal orthogonal TRS. Then* $\mathbb{F}_{lm}$ *is a normalizing reduction strategy for* R.

6.12.7. Relaxing the constraints in $\mathbb{F}_{lm}$, $\mathbb{F}_{GK}$ and $\mathbb{F}_{po}$.

In the reduction strategy $\mathbb{F}_{GK}$ (full substitution) every redex is 'killed' *as soon as it arises*, and this repeatedly. Suppose we relax this requirement, and allow ourselves some time (i.e. some number of reduction steps) before getting rid of a particular redex—but with the obligation to deal with it *eventually*. The reductions arising in this way are all cofinal.

6.12.7.1. DEFINITION. (i) Let $R = t_0 \to t_1 \to \ldots$ be a finite or infinite reduction sequence. Consider some redex s in some term t_n of R. We say that s is *secured in* R if eventually there are no descendants of s left, i.e.

$$\exists m>n \ (t_m \text{ contains no descendants } s', s'', \ldots \text{ of } s \subseteq t_n).$$

(ii) R is *fair* if every redex in R is secured.

6.12.7.2. THEOREM. *For reductions* R *in orthogonal TRSs:*
R *is fair* $\Rightarrow$ R *is cofinal.*

Note that Theorem 6.12.4(i) is a corollary of the present theorem, since evidently reductions obtained by applying $\mathbb{F}_{GK}$ are fair.

A similar relaxation of constraints applies to the other two strategies $\mathbb{F}_{po}$ and $\mathbb{F}_{lm}$:

6.12.7.3. DEFINITION. (i) A reduction R is *leftmost-fair* if R ends in a normal form or infinitely many times the leftmost outermost redex is contracted in R.
(ii) R = $t_0 \to t_1 \to \ldots$ is *outermost-fair* if R does not contain a term t_n with an outermost redex which infinitely long stays an outermost redex but which is never contracted.

6.12.7.4. THEOREM. *Let* R *be an orthogonal TRS. Then:*
(i) *Outermost-fair reductions are normalizing.*
(ii) *If* R *is moreover left-normal, then leftmost-fair reductions are normalizing.*

We will now summarize some of the main properties of the various reduction strategies (and their ‘relaxed’ versions) in Table 9.1. Before doing so, we introduce one more property of strategies:

6.12.7.5. DEFINITION. A reduction strategy $\mathbb{F}$ for R is *perpetual*, if for all t: $\infty(t) \Rightarrow \infty(\mathbb{F}(t))$.

Here $\infty(t)$ means that t has an infinite reduction, i.e. $\neg SN(t)$. So a perpetual strategy is the opposite of a normalizing one; it tries to avoid normal forms whenever possible, and could therefore also be called ‘anti-normalizing’.

In Table 9.1 p, n, c stand for perpetual, normalizing, cofinal respectively. In case a property is not mentioned, it does not hold generally. Note that for the leftmost-outermost strategy, when applied to orthogonal TRSs in general, none of the three properties holds generally. Proofs that leftmost-outermost reduction is normalizing for left-normal orthogonal TRSs and that parallel-outermost reduction is normalizing for all orthogonal TRSs can be found in O'Donnell [77].

	orthogonal TRSs	*orthogonal left-normal TRSs*	*orthogonal non-erasing TRSs*
leftmost-innermost	p	p	p n
parallel-innermost	p	p	p n
leftmost-outermost (leftmost-fair)		n	p n
parallel-outermost (outermost-fair)	n	n	p n
full substitution (fair)	n c	n c	p n c

Table 9.1

For results regarding optimality (with respect to the number of steps) of orthogonal reduction strategies we refer to Khasidashvili [90b].

6.12.8. Computable reduction strategies. A strategy is *recursive* or *computable* if it is, after a coding of the terms into natural numbers, a recursive function. Obviously we are primarily interested in computable strategies; and indeed all five strategies in Definition 9.3 are computable. We may now ask whether there is always for an orthogonal TRS a *computable* *<u>one-step</u>* *normalizing* reduction strategy. A priori this is not at all clear, in view of TRSs such as the following one: CL extended with rules for the Berry-Kleene function F:

$$\text{FABx} \to \text{C}$$
$$\text{FBxA} \to \text{C}$$
$$\text{FxAB} \to \text{C}$$

which is an orthogonal TRS. This TRS seems to require a parallel reduction strategy (so, not a one-step or sequential strategy), because in a term of the form FMNL we have no way to see the 'right' argument for computation: a step in the third argument may be unnecessary, namely if the first and second argument evaluate to A and B respectively (which is undecidable due to the presence of CL); likewise a step in the other arguments may be unnecessary.

We can consider the same problem for the weakly orthogonal TRS obtained by extending CL with Parallel-or:

$$or(true, x) \rightarrow true$$
$$or(x, true) \rightarrow true.$$

It might be thought that such TRSs require a parallel evaluation. However, there is the following surprising fact.

6.12.8.1. THEOREM (Kennaway [89]).
For every weakly orthogonal TRS there exists a computable sequential normalizing reduction strategy.

In fact, Kennaway [89] proves this theorem for the larger class of weakly orthogonal Combinatory Reduction Systems; these are TRSs with bound variables, such as λ-calculus.

6.12.9. Standard reductions in orthogonal TRSs. For λ-calculus and CL there is a very convenient tool: the Standardization Theorem (see Barendregt [84], Klop [80]). For orthogonal TRSs there is unfortunately not a straightforward generalization of this theorem. The obstacle is the same as for the normalizing property of the leftmost reduction strategy, discussed in the previous section. When we restrict ourselves again to left-normal orthogonal TRSs, there is a straightforward generalization.

6.12.9.1. DEFINITION. **(Standard reductions)**
Let R be a TRS and $\mathsf{R} = t_0 \rightarrow t_1 \rightarrow \ldots$ be a reduction in R. Mark in every step of R all symbols to the left of the head symbol of the contracted redex, with '*'. Furthermore, markers are persistent in subsequent steps.

Then R is a *standard reduction* if in no step a redex is contracted with a marked head operator. (So the action in R moves literally from left to right, and an increasing left part of the term is ‘frozen’.)

6.12.9.2. STANDARDIZATION THEOREM for left-normal orthogonal TRSs.
Let R *be a left-normal orthogonal TRS. Then: if* t $\twoheadrightarrow$ s *there is a standard reduction in* R *from* t *to* s.

For a proof see Klop [80]. A corollary is our earlier theorem stating that $\mathbb{F}_{lm}$ is a normalizing strategy for left-normal orthogonal TRSs; this fact is in λ-calulus and CL literature also known as the *Normalization Theorem.*

9.10.3. EXAMPLE. *(Primitive recursive functions.)*
The primitive recursive functions from $\mathbb{N}$ to $\mathbb{N}$ are defined by the following inductive definition (Shoenfield [67]):

(i) The *constant* functions $C_{n,k}$, the *projection* functions $P_{n,i}$ and the *successor* function S are primitive recursive. (Here $C_{n,k}(x_1,...,x_n) = k$; $P_{n,i}(x_1,...,x_n) = x_i$; $S(x) = x+1$.)

(ii) If G, $H_1,...,H_k$ are primitive recursive, then F defined by

$$F(\mathbf{x}) = G(H_1(\mathbf{x}),...,H_k(\mathbf{x}))$$

(where $\mathbf{x} = x_1,...,x_n$) is primitive recursive.

(iii) If G and H are primitive recursive, then F defined by

$$F(0, \mathbf{x}) = G(\mathbf{x})$$
$$F(S(y), \mathbf{x}) = H(F(y, \mathbf{x}), y, \mathbf{x})$$

is primitive recursive. Here $\mathbf{x} = x_1,...,x_n$.

Observe that, by replacing every ‘=’ by ‘$\rightarrow$’ in the defining equations, every primitive recursive function is defined by a terminating, left-normal, orthogonal constructor TRS.

We conclude with an important theorem of Huet and Lévy about needed redexes and needed reduction, a normalizing strategy.

6.10.1. DEFINITION. Let $t \in Ter(R)$, R orthogonal. Let $s \subseteq t$ be a redex. Then s is a *needed* redex (needed for the computation of the normal form, if it exists) iff in all reductions $t \to \ldots \to t°$ such that $t°$ is a normal form, some descendant of s is contracted.

(So, trivially, any redex in a term without normal form is needed.)

6.10.2. THEOREM (Huet & Lévy [79]). *Let* t *be a term in an orthogonal TRS* R.

(i) *If* t *is not in normal form,* t *contains a needed redex.*

(ii) *Repeated contraction of a needed redex leads to the normal form, if it exists.*
(So, needed reduction is normalizing.)

(iii) *Let* t *have a normal form. Then there does not exist an infinite reduction in* R *of* t *containing infinitely many steps in which a needed redex is contracted.*

Part (iii) says that needed reduction is not only normalizing, but even 'hyper-normalizing', in the sense that in between performing needed reduction steps we can relax from this requirement and perform (finitely many) arbitrary reductions.

As it stands, these facts are not yet useful, since it is in general undecidable whether a redex is needed. But there are decidable subcases of great importance. We will not pursue this here.

7

Transfinite rewriting

In this topic we will consider infinite terms over a first order signature. We follow work reported in Kennaway, Klop, Sleep & de Vries [KKSV95a], Klop & de Vrijer [KV91]. A complete formal treatment, including full proofs, can be found in [KKSV95a]. This work was stimulated by earlier studies of infinite rewriting by Dershowitz, Kaplan & Plaisted [DKP89] and Farmer & Watro [FW89].

In many functional programming languages (e.g. Miranda ([Tur85]), Haskell (Hud88]), Clean ([PvE93]) one can express such infinite terms, e.g. 'streams' of natural numbers, or infinite trees. Recently such infinite objects also receive much attention from the point of view of coinductive techniques. In lambda calculus one is used to working with infinite objects in the form of Böhm Trees; we will discuss these in a later topic. For now the signature is first-order, so there are no bound variables as in lambda calculus.

So, our starting point is an ordinary TRS (Σ, R). In fact, we will suppose throughout that our TRSs are *orthogonal*. Now it is obvious that the rules of the TRS (Σ, R) just as well apply to infinite terms as to the usual finite ones. First, let us explain the notion of infinite term that we have in mind. As before, let Ter(Σ) be the set of finite Σ-terms. Then Ter(Σ) can be equipped with a distance function d such that for t, s ∈ Ter(Σ), we have $d(t, s) = 2^{-n}$ if the n-th level of the terms s, t (viewed as labeled trees) is the first level where a difference appears, in case s and t are not identical; furthermore, d(t, t) = 0. It is well-known that this construction yields (Ter(Σ), d) as a metric space.

Now infinite terms are obtained by taking the completion of this metric space, and they are represented by infinite trees. We will refer to the complete metric space arising in this way as $(Ter^{\infty}(\Sigma), d)$, where $Ter^{\infty}(\Sigma)$ is the set of finite and infinite terms over Σ.

A natural consequence of this construction is the emergence of the notion of *Cauchy convergence* as a possible basis for infinite reductions which have a limit: we say that $t_0 \to t_1 \to t_2 \to \ldots$ is an infinite reduction sequence with limit t, if t is the limit of the sequence $t_0, t_1, \ldots$ in the usual sense of Cauchy convergence. See Figure 7.1 for an example, based on a rewrite rule $F(x) \to P(x, F(S(x)))$ in the presence of a constant 0.

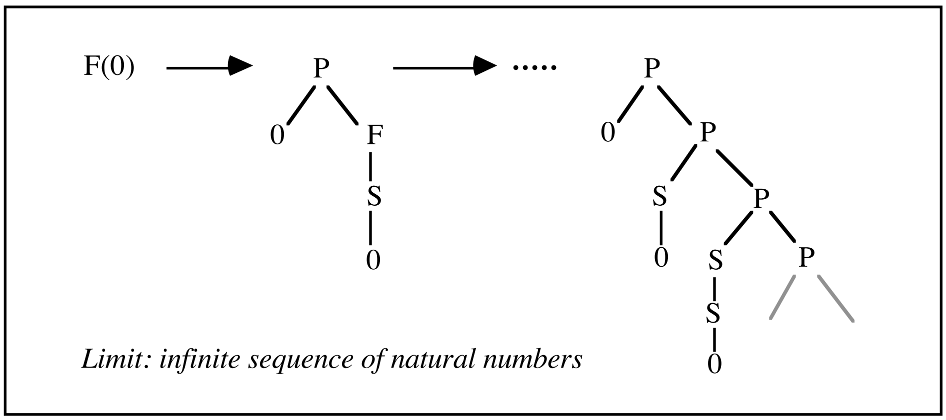
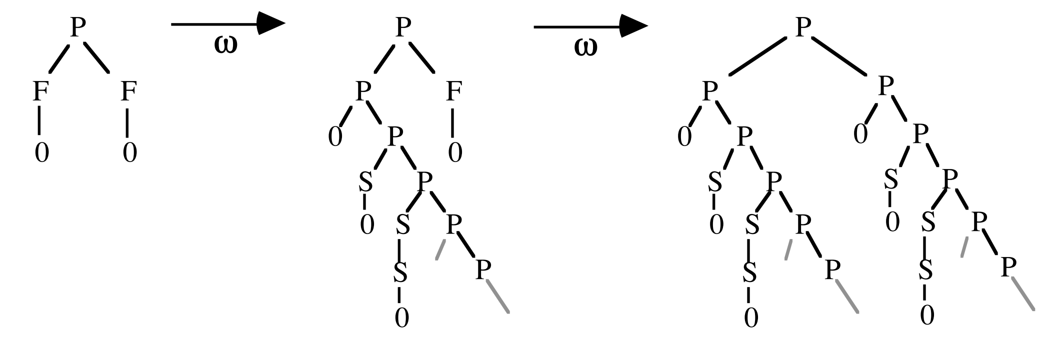


Figure 7.1

In the sequel we will however adopt a stronger notion of converging reduction sequence which turns out to have better properties. First, let us argue that it makes sense to consider not only reduction sequences of length ω, but even reduction sequences of length α for arbitrary ordinals α. Given a notion of convergence, and limits, we may iterate reduction sequences beyond length ω and consider e.g.

$$t_0 \to t_1 \to t_2 \to \ldots \to t_n \to \ldots$$
$$s_0 \to s_1 \to s_2 \to s_3 \to \ldots \ r$$

where $\lim_{n\to\infty} t_n = s_0$ and $\lim_{n\to\infty} s_n = r$. See Figure 7.2 for such a reduction sequence of length $\omega + \omega$, which may arise by evaluating first the left part of the term at hand, and next the right part. Of course, in this example a 'fair' evaluation is possible in only ω many reduction steps, but we do not want to impose fairness requirements at the start—even though we may (and will) consider it to be a desirable feature that reductions of length α could be 'compressed' to reductions of length not exceeding ω steps, yielding the same 'result'.



Transfinite reduction sequence of length $\omega + \omega$

Figure 7.2

We will give a formal definition now.

7.1. DEFINITION. Let (Σ, R) be a TRS. A *(Cauchy-) convergent* R-*reduction sequence of length* α (an ordinal) is a sequence $\langle t_\beta \mid \beta \leq \alpha \rangle$ of terms in $Ter^\infty(\Sigma)$, such that

(i) $t_\beta \to_R t_{\beta+1}$ for all $\beta < \alpha$,

(ii) $t_\lambda = \lim_{\beta<\lambda}$ for every limit ordinal $\lambda \leq \alpha$).

Here (ii) means: $\forall n\, \exists \mu < \lambda\, \forall \nu\, (\mu \leq \nu \leq \lambda \;\Rightarrow\; d(t_\nu, t_\lambda) \leq 2^{-n})$.

Notation: If $\langle t_\beta \mid \beta \leq \alpha \rangle$ is a Cauchy-convergent reduction sequence we write $t_0 \to_\alpha{}^c\, t_\alpha$ ('c' for 'Cauchy').

The notion of normal form as a final result has to be considered next. We simply generalize the old finitary notion of normal form to the present infinitary setting thus: a (possibly infinite) term is a normal form *when it contains no redexes*. The only difference with the finitary case is that here a redex may be itself an infinite term. But note that a redex is still so by virtue of a finite prefix, that was called in Topic 4 the redex pattern—this is so because our rewrite rules are orthogonal and hence contain no repeated variables.

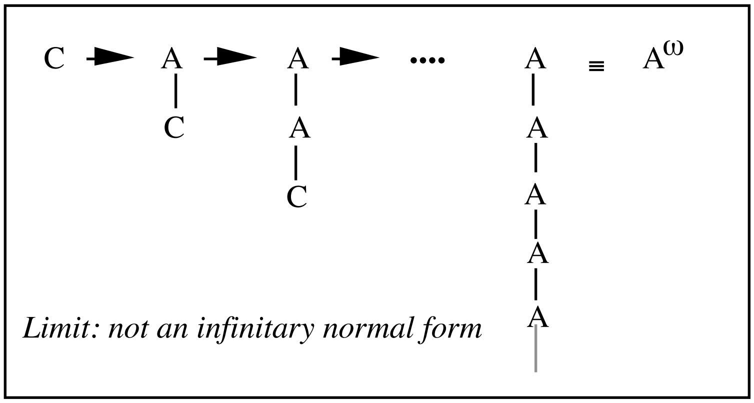


Figure 7.3

So, in Figure 7.3 we have, with as TRS {C → A(C), A(x) → x}, a (Cauchy-) converging reduction sequence with as limit the infinite term A(A(A(A..., abbreviated as A^{ω}; this limit is not a normal form: A^{ω} reduces to itself: $A^{\omega} \to A^{\omega}$, a nd only to itself. (Note that this step can be performed in infinitely many different ways, since every A in A^{ω} is the root of a redex.) Normal forms are shown in Figures 7.1, 7.2 as the rightmost terms (if no other reduction rules are present than the one mentioned above). Henceforth we will often drop the reference 'infinite' or 'infinitary'. Thus a term, or a normal form, may be finite or infinite. The notion of Cauchy converging reduction sequence that was considered so far, is not quite satisfactory. We would like to have the *compression property*:

$$t_0 \twoheadrightarrow_{\alpha}{}^{c}\, t_{\alpha} \;\Rightarrow\; t_0 \twoheadrightarrow_{\leq\omega}{}^{c}\, t_{\alpha}.$$

That is, given a reduction $t_0 \twoheadrightarrow_{\alpha}{}^{c}\, t_{\alpha}$, of length α, the result t_{α} can already be found in at most ω many steps. ('At most', since it may happen that a transfinite reduction sequence can be compressed to finite length, but not to length ω.) Unfortunately, $\twoheadrightarrow_{\alpha}{}^{c}$ lacks this property:

7.2. COUNTEREXAMPLE. Consider the orthogonal TRS with rules {A(x) → A(B(x)), B(x) → E(x)}. Then $A(x) \twoheadrightarrow_{\omega} A(B^{\omega}) \to A(E(B^{\omega}))$, so $A(x) \twoheadrightarrow_{\omega+1} A(E(B^{\omega}))$. However, we do not have $A(x) \twoheadrightarrow_{\leq\omega} A(E(B^{\omega}))$, as can easily be verified.

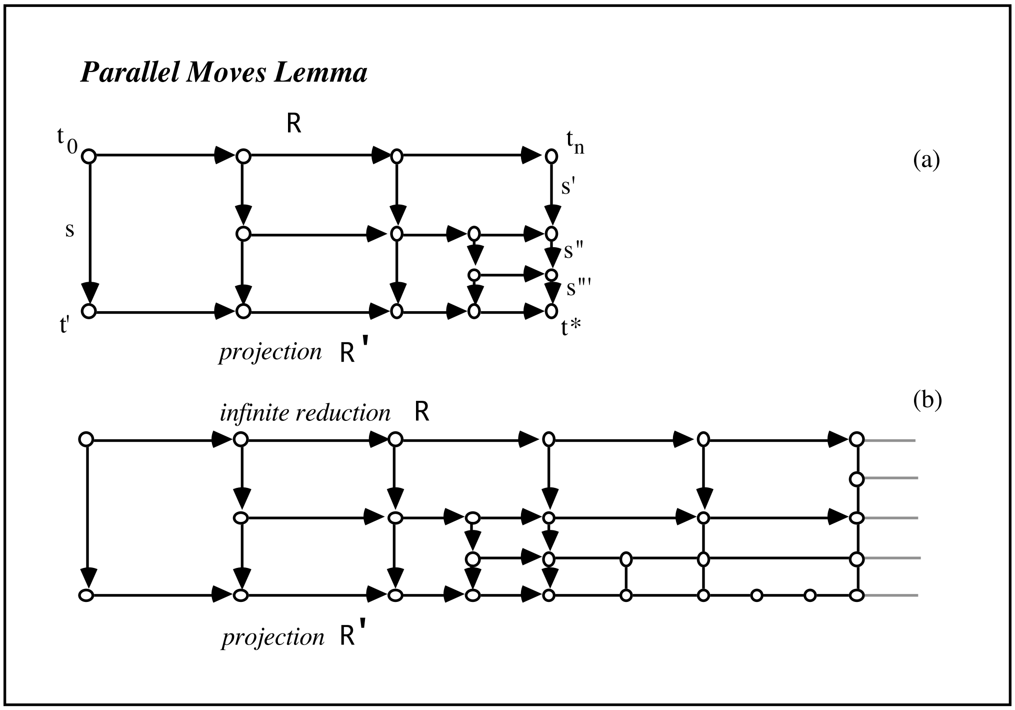


Figure 7.4

Another obstacle for $\rightarrow_\alpha{}^c$ is that the well-known Parallel Moves Lemma resists a generalization to the present transfinite case. We recall the PML in Figure 7.4(a): setting out a finite reduction R: $t_0 \twoheadrightarrow t_n$ against a one step reduction $t_0 \rightarrow_s t'$ (where s is the contracted redex), one can complete the reduction diagram in a canonical way, thereby obtaining as the righthand side of the diagram a reduction $t_n \twoheadrightarrow t^*$ which consists entirely out of contractions of all the *descendants of* s *along* R. Furthermore, the reduction R': $t' \twoheadrightarrow t^*$ arising as the lower side of this reduction diagram, is called the *projection of* R *over the reduction step* $t_0 \rightarrow_s t'$. Notation: R' = R/ ($t_0 \rightarrow_s t'$).

We would like to have a generalization of PML where R is allowed to be infinite, and converging to a limit. In this way we would have a good stepping stone towards establishing infinitary confluence properties. However, it is not clear at all how such a generalization can be established. The problem is shown in Figure 7.5. First note that we can without problem generalize the notion of 'projection' to infinite reductions, as in Figure 7.4(b): there R' is the projection of the infinite R over the displayed reduction step. This merely requires an iteration of the finitary PML, no infinitary version is needed. Now consider the two rule TRS {A(x, y) → A(y, x), C → D}. Let R be the infinite reduction A(C, C) → A(C, C) → A(C, C) → ... , in fact a reduction cycle of length 1. Note that R is converging, with limit A(C, C). The projection R' of R over the step A(C, C) → A(D, C), however, is no longer converging. For, this is A(D, C) → A(C, D) → A(D, C) → ..., a 'two cycle'. So, the class of infinite converging reduction sequences is not closed under projection. This means that in order to get some decent properties of infinitary reduction in this sense, one has to impose further restrictions.

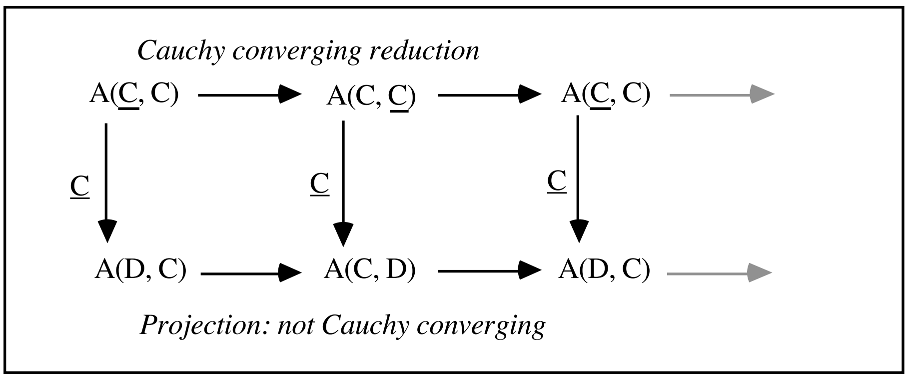


Figure 7.5

As the last example shows, there is a difficulty in that we loose the notion of descendants which is so clear and useful in finite reductions. Indeed, after the infinite reduction A(C, C) → A(C, C) → A(C, C) → ... , with Cauchy limit A(C, C), what is the descendant of the original underlined redex C in the limit A(C, C)? There is no likely candidate.

We will now describe the stronger notion of converging reduction sequence that does preserve the notion of descendants in limits. If we have a converging reduction sequence $t_0 \rightarrow_{s0} t_1 \rightarrow_{s1} \ldots t$, where s_i is the redex contracted in the step $t_i \rightarrow t_{i+1}$ and t is the limit, we now moreover require that

$$\lim_{i\to\infty} \text{depth}(s_i) = \infty. \qquad (*)$$

Here depth(s_i), the depth of redex s_i, is the distance of the root of t_i to the root of the subterm s_i. If the converging reduction sequence satisfies this additional requirement (*), it is called *strongly convergent*. The difference between the previous notion of (Cauchy) converging reduction sequence and the present one, is suggested by Figure 7.6. The circles in that figure indicate the root nodes of the contracted redexes; the shaded part is that prefix part of the term that does not change anymore in the sequel of the reduction. The point of the additional requirement (*) is that this growing non-changing prefix is required really to be non-changing, in the sense that no activity (redex contractions) in it may occur at all, even when this activity would by accident yield the same prefix.

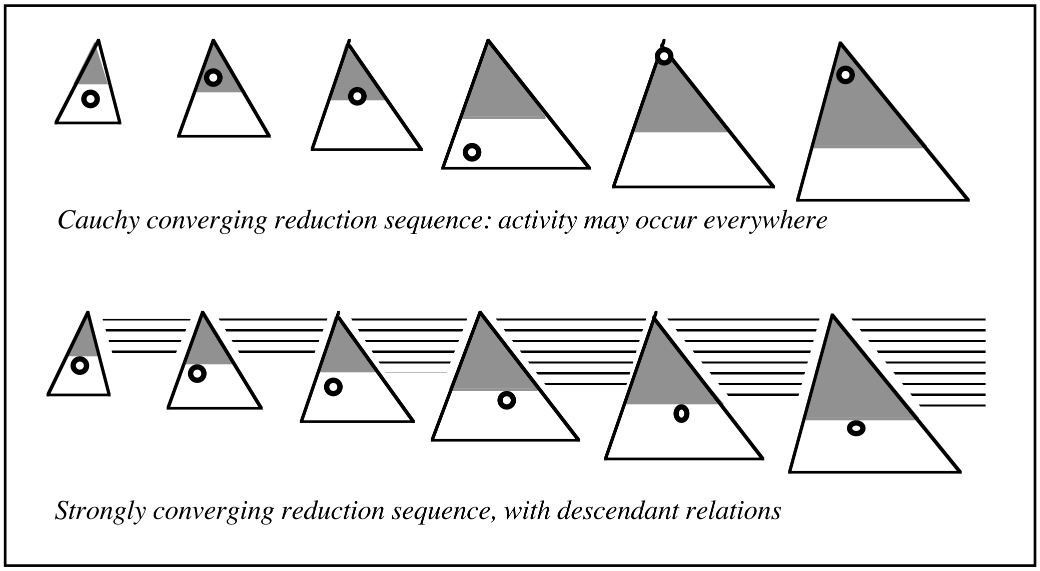


Figure 7.6

Note that there is now an obvious definition of descendants in the limit terms; the precise formulation is not hard to make explicit.

In fact, we define strongly converging reductions of length α for every ordinal α, by imposing the additional condition (*) whenever a limit ordinal $\lambda \leq \alpha$ is encountered. (It will turn out however that only countable ordinals may occur.) More formally:

7.3. DEFINITION. Let (Σ, R) be a TRS. A *strongly convergent* R-*reduction sequence of length* α is a sequence $\langle t_\beta \mid \beta \leq \alpha \rangle$ of terms in $Ter^\infty(\Sigma)$, such that

(i) $t_\beta \rightarrow_R t_{\beta+1}$ for all $\beta < \alpha$,

(ii) for every limit ordinal $\lambda \leq \alpha$: $\forall n\ \exists \mu < \lambda\ \forall \nu\ (\mu \leq \nu \leq \lambda \Rightarrow d(t_\nu, t_\lambda) \leq 2^{-n}\ \&\ depth(s_\nu) \geq n)$.

Here s_ν is the redex contracted in the step $t_\nu \rightarrow t_{\nu+1}$. (See Fig. 7.7.)

Notation: If $\langle t_\beta \mid \beta \leq \alpha \rangle$ is a strongly convergent reduction sequence we write $t_0 \twoheadrightarrow_\alpha t_\alpha$.

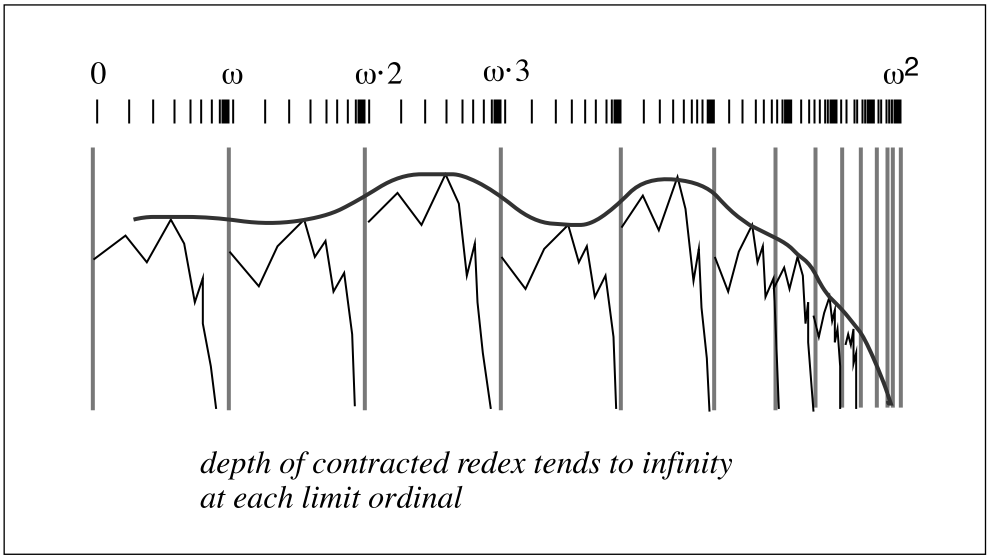


Figure 7.7

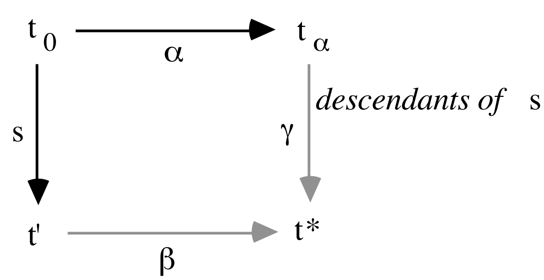
Henceforth all our infinitary reductions will be strongly convergent. Now we can state the benefits of this notion.

7.4. COMPRESSION LEMMA. *In every orthogonal TRS:*

$$t \to_\alpha t' \;\Rightarrow\; t \to_{\leq\omega} t'.$$

(Note that the counterexample 7.2 to compression for Cauchy converging reductions was not strongly converging.)

7.5. INFINITARY PARALLEL MOVES LEMMA. *In every orthogonal TRS:*



That is, whenever $t_0 \to_\alpha t_\alpha$ *and* $t_0 \to_s t'$, *where* s *is the contracted redex (occurrence), there are infinitary reductions* $t' \to_\beta t^*$ *and* $t_\alpha \to_\gamma t^*$. *The latter reduction consists of contractions of all* *descendants of* s *along the reduction* $t_0 \rightarrow_\alpha t_\alpha$.

Actually, by the Compression Lemma we can find $\beta, \gamma \leq \omega$.

As a side-remark, let us mention that in every TRS (even with uncountably many symbols and rules), all transfinite reductions have countable length. All countable ordinals can indeed occur as length of a strongly convergent reduction. (For ordinary Cauchy convergent reductions this is not so: the rewrite rule $C \rightarrow C$ yields arbitrarily long convergent reductions $C \rightarrow_\alpha{}^c C$. However, these are not strongly convergent.)

The infinitary PML is "half of the infinitary confluence property". The question arises whether full infinitary confluence (CR^∞)holds. That is, given $t_0 \rightarrow_\alpha t_1$, $t_0 \rightarrow_\beta t_2$, is there a t_3 such that $t_1 \rightarrow_\gamma t_3, t_2 \rightarrow_\delta t_3$ for some γ, δ? Using the Compression Lemma and the PML all that remains to prove is: given $t_0 \rightarrow_\omega t_1, t_0 \rightarrow_\omega t_2$, is there a t_3 such that $t_1 \rightarrow_{\leq\omega} t_3, t_2 \rightarrow_{\leq\omega} t_3$? Surprisingly, the answer is negative: *full infinitary confluence for orthogonal rewriting does not hold.* The counterexample is in Figure 7.8, consisting of an orthogonal TRS with three rules, two of which are 'collapsing rules'. (A rule $t \rightarrow s$ is collapsing if s is a variable.) Indeed, in Figure 7.8(a) we have $C \rightarrow_\omega A^\omega$, $C \rightarrow_\omega B^\omega$ but A^ω, B^ω have no common reduct as they only reduce to themselves. Note that these reductions are indeed strongly convergent. (Figure 7.8(b) contains a rearrangement of these reductions that we need later on.)

However, the good news is that in spite of the failure of CR^∞we do have unicity of (possibly infinite) normal forms (UN^∞).

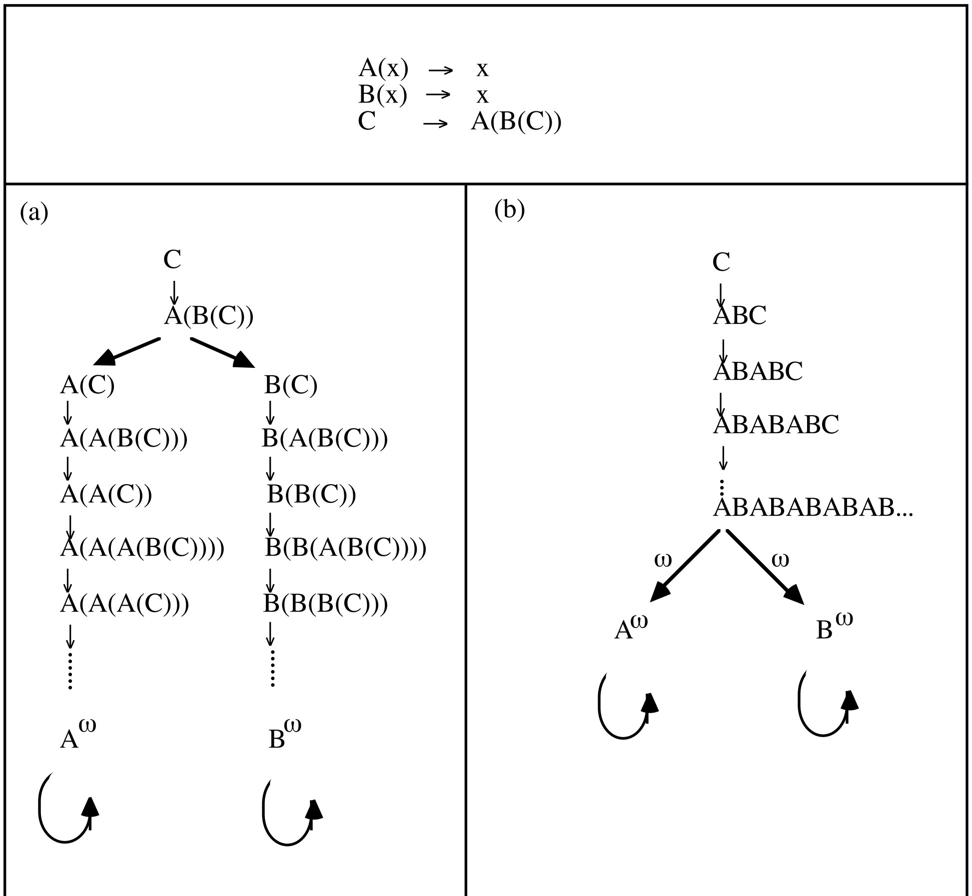
7.6. THEOREM. *For all orthogonal TRSs: Let* $t \rightarrow_\alpha t'$, $t \rightarrow_\beta t''$ *where* t', t'' *are (possibly infinite) normal forms. Then* $t' \equiv t''$.

Here $\equiv$ denotes syntactical equality. Note that in the ABC counterexample in Figure 7.8 the terms A^ω and B^ω are not normal forms.

This Unique Normal Form property, by the way, also holds for Cauchy converging reductions, that is, with $\rightarrow_\alpha$ replaced by $\rightarrow_\alpha{}^c$ and likewise for β. The reason is that we have:

$$t \rightarrow_\alpha{}^c t' \ \&\ t' \textit{ is a normal form} \ \Rightarrow\ t \rightarrow_{\leq\omega} t'.$$

(For $\alpha = \omega$ this is easy to prove; in fact a converging reduction of length ω to a normal form is already strongly convergent. For general α, the proof of the statement requires some work.)



Failure of infinitary confluence

Figure 7.8

We will now investigate the extent to which infinitary orthogonal rewriting lacks full confluence. It will turn out that non-confluence is only marginal, and that terms which display the bad behaviour are included in a very restricted class. The following definition is inspired by a classical notion in λ-calculus; see Barendregt [84].

7.7. DEFINITION. (i) The term t is in *head normal form* (hnf) if $t \equiv C[t_1,...,t_n]$ where $C[t_1,...,t_n]$ is a non-empty context (prefix) such that no reduction of t can affect the prefix C[,...,]. More precisely, if $t \twoheadrightarrow s$ then $s \equiv C[s_1,...,s_n]$ for some s_i (i =1,...,n), and every redex of s is included in one of the s_i (i =1,...,n).

(ii) t *has a hnf* if t $\twoheadrightarrow$ s and s is in hnf.

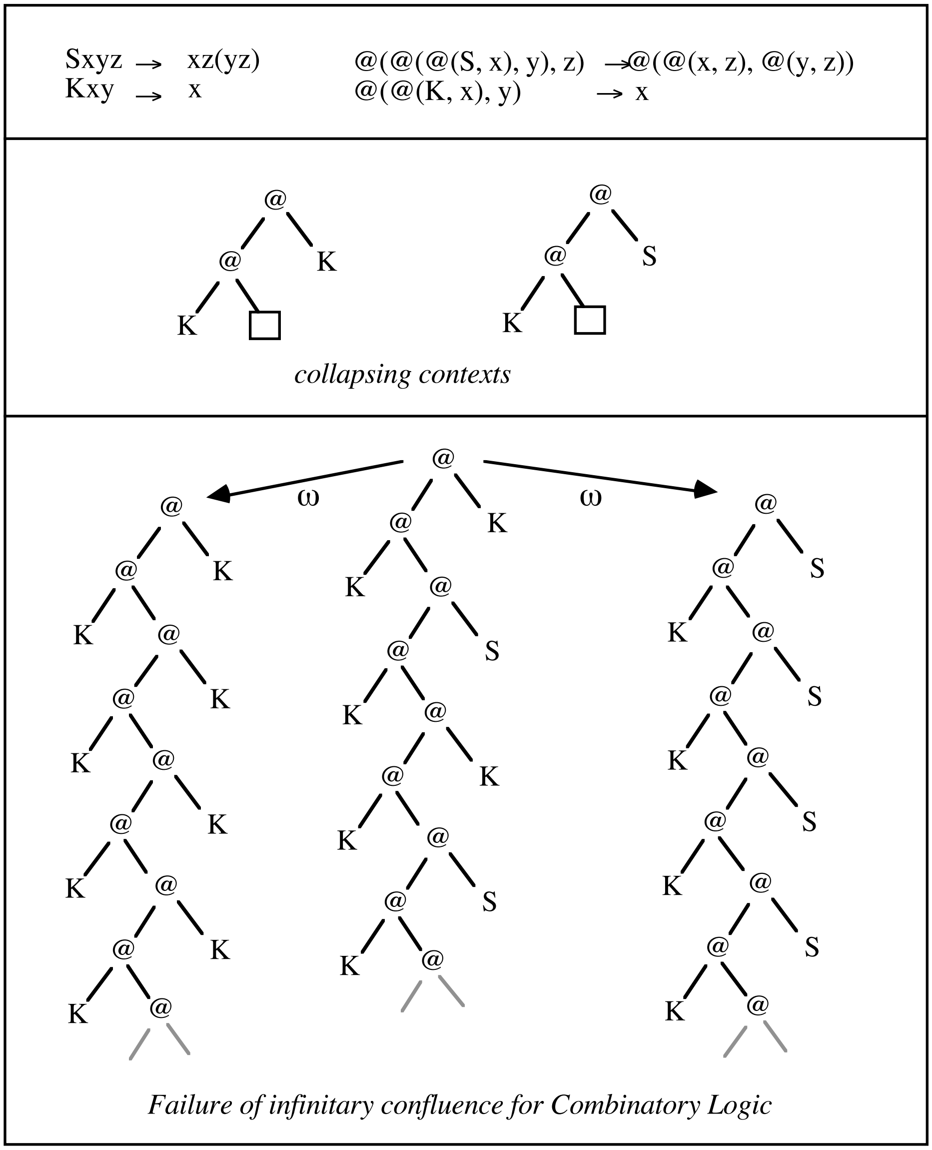
Actually, this definition is equivalent to one of DKP[89]; there a term t is called 'top-terminating' if there is no infinite reduction t → t' → t" → ... in which infinitely many times a redex contraction *at the root* takes place. So: t is top-terminating ⇔ t has a hnf. We need one more definition before formulating the next theorem.

7.8. DEFINITION. If t is a term of the TRS R, then the *family* of t is the set of subterms of reducts of t, i.e. $\{s \mid t \twoheadrightarrow_R C[s]$ for some context $C[\ ]\}$.

7.9. THEOREM. *For all orthogonal TRSs: Let* t *have no term without hnf in its family. Then* t *is infinitary confluent.*

Just as in λ-calculus, one can now formulate some facts about "Böhm trees", which are (possibly infinite) terms where the subterms without hnf are replaced by a symbol Ω for 'undefined'. As in λ-calculus, each term in an orthogonal TRS has a unique Böhm tree. It is also possible to generalize much of the usual theory for finitary orthogonal rewriting to the infinitary case. We mention the theory of Huet & Lévy about 'needed redexes', and results about reduction strategies (such as the parallel-outermost strategy). For more information we refer to KKSV[95a].

Here we want to reconsider the last theorem. Actually, it can be much improved. Consider again the ABC example in Figure 7.8. Rearranging the reductions $C \to_\omega A^\omega$, $C \to_\omega B^\omega$ as in Figure 7.8(b) into reductions $C \to_\omega (AB)^\omega \to_\omega A^\omega$ and $C \to_\omega (AB)^\omega \to_\omega B^\omega$ makes it more perspicuous what is going on: $(AB)^\omega$ is an infinite 'tower' built from two different collapsing contexts A(❒), B(❒), and this infinite tower can be collapsed in different ways.



collapsing contexts

Failure of infinitary confluence for Combinatory Logic

Figure 7.9

The ABC example (Figure 7.8) is not merely a pathological example; the same phenomenon (and therefore failure of infinitary confluence) occurs in Combinatory Logic, as in Figure 7.9, where an infinite tower built from the two different collapsing contexts K❒K and K❒S is able to collapse in two different ways. (Note that analogous to the situation in Figure 7.8, the middle term, built alternatingly from K❒K and K❒S, can be obtained after ω steps from a finite term which can easily be found by a fixed point construction.) Also for λ-calculus one can now easily construct a counterexample to infinitary confluence.

Remarkably, it turns out that the collapsing phenomenon is the *only* cause of failure of infinitary confluence. (The full proof is in KKVS[95a].) Thus we have:

7.10. THEOREM. (i) *Let the orthogonal TRS* R *have no collapsing rewrite rules* $t(x_1,...,x_n) \to x_i$. *Then* R *is infinitary confluent.*
(ii) *If* R *is an orthogonal TRS with as only collapsing rule:* $I(x) \to x$, *then* R *is infinitary confluent.*

Call an infinite term $C_1[C_2[...C_n[...]...]]$, built from infinitely many non-empty collapsing contexts $C_i[\ ]$, a *hereditarily collapsing* (hc) term. (A context C[ ] is collapsing if C[ ] contains one hole ❐ and C[ ] ↠ ❐.) Also a term reducing to a hc term is called a hc term. E.g. C from the ABC example in Figure 7.8 is a hc term. Clearly, hc terms do not have a hnf.

7.11. THEOREM. *Let* t *be a term in an orthogonal TRS, which has not a hc term in its family. Then* t *is infinitary confluent.*

This theorem can be sharpened somewhat, as follows. Let us introduce a new symbol ● (*black hole*) to denote hc terms, with the rewrite rule:

$t \to_{\bullet} \bullet$ *if* t *is a hc term.*

Of course this rule is not 'constructive', i.e. the reduction relation $\to_{\bullet}$ may be undecidable (as it is in CL, Combinatory Logic). However, we now have that orthogonal reduction extended with $\to_{\bullet}$ is infinitary confluent.

REFERENCES.

[Bar84] H.P. BARENDREGT. *The Lambda Calculus, its Syntax and Semantics*. 2nd ed., 1984. North-Holland 1984.

[DKP89] N. Dershowitz, S. Kaplan, and D.A. Plaisted. Infinite normal forms. In ICALP '89, volume 372 of Springer LNCS, p. 249-262, 1989.

[FW89] W.M. Farmer and R.J. Watro. Redex capturing in term graph rewriting. Technical report, M89-59, MITRE, 1989.

[Hud88] P. Hudak. Report on the functional programming language Haskell. Draft Proposed Standard, 1988.

[KKSV93] J.R. Kennaway, J.W. Klop, M.R. Sleep, and F.J. de Vries. An infinitary Church-Rosser property for non-collapsing orthogonal term rewriting systems. In M.R. Sleep, M.J. Plasmeijer, and M.C.J.D. van Eekelen, editors, Term Graph Rewriting - theory and practice, p. 47-59. Wiley, 1993.

[KKSV95a] J.R. Kennaway, J.W. Klop, M.R. Sleep, and F.J. de Vries.Infinitary lambda calculus and Böhm models. In Proceedings of RTA95, Kaiserslautern, Springer LNCS 914, p.257-270, 1995.

[KKSV95b] J.R. Kennaway, J.W. Klop, M.R. Sleep, and F.J. de Vries. Transfinite reductions in orthogonal term rewriting systems. Information and Computation, 119(1):18-38, 1995.

[KKSV97] J.R. Kennaway, J.W. Klop, M.R. Sleep, and F.J. de Vries. Infinitary lambda calculus. Theoretical Computer Science, 175(1):93-125, 1997.

[KV91] J.W. Klop and R.C. de Vrijer. Extended rewriting systems. In S. Kaplan and M. Okada, editors, Proceedings of the Workshop on Conditional and Typed Rewriting systems, Springer LNCS 516, p.26-50, Montreal, 1991.

[PvE93] M.J. Plasmeijer and M.C.J.D. van Eekelen. Functional Programming and Parallel Graph Rewriting. Addison Wesley, 1993.

[Tur85] D.A. Turner. Miranda: a non-strict functional language with polymorphic types. In J.-P. Jouannaud, editor, Proceedings of the ACM Conference on Functional Programming Languages and Computer Architecture, Springer LNCS 201, p.1-16, 1985.

8

Recursive Path Orders with Stars

In the chapters thus far we have seen several methods to prove CR, but the equally important property SN (strong Normalization) has not yet received much attention. In this chapter we will introduce a powerful method to prove SN, called Recursive Path Orders. It was developed by Dershowitz on the basis of a beautiful theorem of Kruskal. (See also the similar concept of 'path of subterm ordering' in Plaisted [78], discussed in Rusinowitch [87].)

We will in fact demonstrate not the usual formulation, but one "with stars". For general references on termination methods, see Huet & Oppen [80], Dershowitz [85].

8.1. DEFINITION. (i) Let $\mathbb{T}$ be the set of commutative finite trees with nodes labeled by natural numbers. Example: see Figure 7.1. This tree will also be denoted by: 3(5, 7(9), 8(0(1, 5))). Commutativity means that the 'arguments' may be permuted; thus 3(8(0(5, 1)), 5, 7(9)) denotes the same commutative tree.

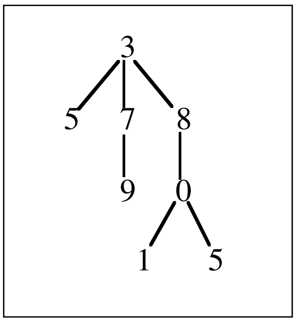


Figure 7.1

In order to introduce the next notion of 'embedding', we must make the definition of trees $t \in \mathbb{T}$ somewhat more precise. An element $t \in \mathbb{T}$ is a pair $(\langle D, \leq, \alpha_0 \rangle, \ell)$ where D is a finite set $\{\alpha_0, \beta, \gamma, ...\}$ with distinguished element α_0, called the root or the top of t, and partially ordered by $\leq$. We require that

(i) $\alpha_0 \geq \beta$ for all $\beta \in D$,

(ii) $\beta \leq \gamma$ and $\beta \leq \delta \Rightarrow \gamma \leq \delta$ or $\delta \leq \gamma$, for all $\beta, \gamma, \delta \in D$.

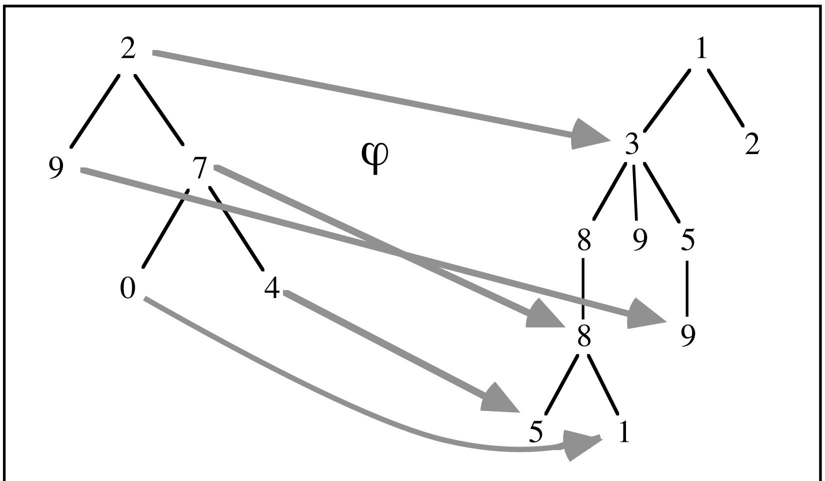
The set D is also called NODES(t), the set of nodes of t. Furthermore, $\ell : D \to \mathbb{N}$ is a map assigning labels (natural numbers) to the nodes of t. Finally, we use the notation $\alpha \wedge \beta$ for the supremum (least upper bound) of $\alpha, \beta \in D$. (The actual names α, β, ... of the nodes are not important, which is why they were suppressed in Figure 8.1.)

8.2. DEFINITION. Let $t, t' \in \mathbb{T}$. We say that t is *(homeomorphically) embedded in* t', notation $t \preceq t'$, if there is a map φ: NODES(t) $\to$ NODES(t') such that:

(i) φ is *injective*,
(ii) φ is *monotonic* ($\alpha \le \beta \Rightarrow \varphi(\alpha) \le \varphi(\beta)$),
(iii) φ is *sup preserving* ($\varphi(\alpha \wedge \beta) = \varphi(\alpha) \wedge \varphi(\beta)$),
(iv) φ is *label increasing* ($\ell(\alpha) \le \ell'(\varphi(\alpha))$, where ℓ, ℓ' are the labeling maps of t, t' respectively; $\le$ is the ordering of natural numbers).

8.3. REMARK. Actually, (ii) is superfluous as it follows from (iii).

8.4. EXAMPLE.
(i) $2(9, 7(0, 4)) \preceq 1(3(8(8(5, 1)), 9, 5(9)), 2)$ as the embedding in Figure 8.2 shows.
(ii) Note that we do *not* have $1(0, 0) \preceq 1(0(0, 0))$.



homeomorphic embedding

Figure 8.2

8.5. REMARK. A more elegant and equivalent definition of $s \preceq t$ can be given in terms of rewriting, as follows: on $\mathbb{T}$ we define the reduction $\twoheadrightarrow$:

(1) $n(\mathbf{t}) \twoheadrightarrow m(\mathbf{t})$ if $n > m$,
(2) $n(s, \mathbf{t}) \twoheadrightarrow s$.

(3) n(s, **t**) ⋙→ **t**

(Here **t** is $t_1,..., t_k$, $k \geq 0$.) Let ⋙→* be the transitive reflexive closure of ⋙→. Then:

$s \preceq t \Leftrightarrow t$ ⋙→* s.

8.6. EXAMPLE. (See also Figure 7.2)

1(3(8(8(5, 1)), 9, 5(9)), 2) ⋙→
3(8(8(5, 1)), 9, 5(9)) ⋙→
3(8(5, 1), 9, 5(9)) ⋙→
3(7(5, 1), 9, 5(9)) ⋙→ ⋙→
3(7(4, 0), 9, 5(9)) ⋙→
2(7(4, 0), 9, 5(9)) ⋙→
2(7(4, 0), 5(9)) ⋙→
2(7(4, 0), 9) = 2(9, 7(0, 4)).

Clearly, $\preceq$ is a partial order on $\mathbb{T}$. Moreover it satisfies the following remarkable property:

8.6. KRUSKAL'S TREE THEOREM.
Let $t_0, t_1, t_2,...$ *be a sequence of trees in* $\mathbb{T}$. *Then for some* $i < j$: $t_i \preceq t_j$.

The proof of this theorem as given in Kruskal [60] is very complicated. In Dershowitz [79] a short proof is given for a restricted case, namely for trees whose branching degree is uniformly bounded by some natural number N. In Dershowitz & Jouannaud [90] a proof sketch is given, using Ramsey's theorem and Higman's Lemma. We will now give a detailed proof of the theorem, first for the restricted case as in Dershowitz [79], then upgrading it to the general case (without restriction on the branching degree, other than this must be finite). Higman's Lemma will be found along the way, as a simple case of the first proof. All proofs are originally due to Nash-Williams [63].

8.7. DEFINITION. (i) The *branching degree* of a node s in $t \in \mathbb{T}$ is the number of immediate successor nodes of s.
(ii) $\mathbb{T}_N$ is the subset of $\mathbb{T}$ consisting of trees where all nodes have branching degree $\leq N$.

8.8. PROPOSITION. *Each infinite sequence of natural numbers* $n_0, n_1, n_2, ...$ *has a weakly ascending infinite subsequence.*

PROOF. We have to prove that there is a subsequence $n_{f(0)}, n_{f(1)}, n_{f(2)}, ...$ with $f(0) < f(1) < f(2) < ...$ such that $n_{f(0)} \leq n_{f(1)} \leq n_{f(2)} \leq ...$. For the sake of exposition we will give three proofs.

The first and simplest proof is as follows. Let $n_{f(0)}$ be a minimal element in the sequence, i.e. $n_{f(0)} \leq n_i$ for all $i = 0,1,2,...$. Now find in the sequence $n_{f(0)+1}, n_{f(0)+2}, n_{f(0)+3},...$ again a minimal element; this is $n_{f(1)}$. And so on.

The second proof is more sophisticated and presents an argument used later on. It is easy to see that every infinite sequence of natural numbers $n_0, n_1, n_2, \ldots$ must contain a pair n_i, n_j ($i < j$) such that $n_i \leq n_j$ (*). Now consider all weakly ascending subsequences (we will call these: chains) of the infinite sequence under consideration. If one of them is infinite, we are done. Otherwise, we can find infinitely many finite chains, each maximally prolonged to the right, and such that the next chain starts after the end of the previous chain. Consider the sequence of final elements of these chains. This infinite sequence contains again, by (*), a chain of length 2. But then some chain was not maximal, contradiction.

The third proof is even more sophisticated, and uses the infinite version of Ramsey's Theorem. Consider the partition of $[\mathbb{N}]^2$ into subsets X and Y defined as follows:

$$\{i,j\} \in X \text{ if } i < j \text{ and } n_i \leq n_j$$
$$\{i,j\} \in Y \text{ if } i < j \text{ and } n_i > n_j.$$

Then, according to Ramsey's theorem, there is an infinite $A \subseteq \mathbb{N}$ such that either $[A]^2 \subseteq X$ or $[A]^2 \subseteq Y$. In the first case we have found our infinite chain (the weakly ascending subsequence). The second case cannot occur, as it would entail the existence of an infinite descending subsequence. ❑

8.9. DEFINITION. (1) Let $t \in \mathbb{T}$. Then $|t|$ is the number of nodes of t.
(2) Notation: an infinite sequence of trees $t_0, t_1, \ldots$ will be written as **t**. The initial segment $t_0,\ldots,t_{n-1}$ is $(\mathbf{t})_n$. The set of infinite sequences of trees from $\mathbb{T}$ is $\mathbb{T}^\omega$.
(3) Let $\mathcal{D} \subseteq \mathbb{T}^\omega$. Then the sequence $\mathbf{t} \in \mathcal{D}$ is *minimal in* D if $\forall \mathbf{s} \in \mathcal{D}\ (\mathbf{s})_n = (\mathbf{t})_n \Rightarrow |s_n| \geq |t_n|$.
(4) Furthermore, we say that $\mathbf{s}, \mathbf{t} \in \mathcal{D}$ have distance 2^{-n} if $(\mathbf{s})_n = (\mathbf{t})_n$ but $(\mathbf{s})_{n+1} \neq (\mathbf{t})_{n+1}$. This induces a metric on $\mathbb{T}^\omega$. We say that is $\mathcal{D}$ closed if it is so with respect to this metric.

8.10. PROPOSITION. *Let* D $\subseteq \mathbb{T}^\omega$ *be non-empty and closed. Then* $\mathcal{D}$ *contains a minimal element (with respect to* $\mathcal{D}$*).*

PROOF. Choose a $\mathbf{s} \in \mathcal{D}$ such that $|s_0|$ is minimal. Choose from the set $\mathcal{D}_1 = \{\mathbf{t} \in \mathcal{D} \mid s_0 = t_0\}$ an $\mathbf{s}'$ such that $|s_1'|$ is minimal (in $\mathcal{D}_1$). And so on: from the set $\mathcal{D}_n = \{\mathbf{t} \in \mathcal{D}_{n-1} \mid (\mathbf{s})_n = (\mathbf{t})_n\}$we choose an $\mathbf{s}^{(n)}$ such that $|\mathbf{s}^{(n)}{}_n|$ is minimal (in $\mathcal{D}_n$). Clearly, the sequence of sequences $\mathbf{s}, \mathbf{s}', \ldots, \mathbf{s}^{(n)}, \ldots$ converges with respect to the metric of Definition 9; its limit can easily be shown to be minimal as required. ❑

8.11. EXERCISE. Give an example showing that the closure requirement cannot be missed in Proposition 8.10.

8.12. NOTATION. (1) Let **s**, **t** $\in \mathbb{T}^\omega$. Then **s** $\subseteq$ **t** means that **s** is a subsequence of **t**.
(2) Let **t** = $t_0, t_1, \ldots$ and let **s** = $s_{f(0)}, s_{f(1)}, \ldots$ be a subsequence of **t**, such that for all i, $s_{f(i)}$ is a *proper* subtree of $t_{f(i)}$. Then we write **s** $\subseteq\subseteq$ **t** and call s a subsubsequence of t. (See Figure 8.4.)

8.13. DEFINITION. **s** = $s_0, s_1, s_2, \ldots$ is a *chain* if $s_0 \preceq s_1 \preceq s_2 \preceq \ldots$, where $\preceq$ is the embedding relation as in Kruskal's Tree Theorem.

8.14. KRUSKAL'S TREE THEOREM (restricted case)
Let $t_0, t_1, t_2, \ldots$ *be a sequence of trees in* $\mathbb{T}_N$. *Then for some* $i < j$: $t_i \preceq t_j$.

PROOF.We will suppose, for a proof by contradiction, that there is a counterexample sequence to the restricted version of Kruskal's Tree Theorem that we want to prove. That is, the set $C \subseteq \mathbb{T}_N{}^\omega$ of sequences **s** such that for no $i < j$ we have $s_i \preceq s_j$, is supposed to be non-empty.

CLAIM. *Let* **t** *be a minimal element from* C. *Suppose* **s** $\subseteq\subseteq$ **t**.
(1) *Then for some* $i < j$: $s_i \preceq s_j$.
(2) *Even stronger,* **s** *contains a subsequence which is a chain.*

Proof of the claim. (Note that a minimal element **t** exists by the assumption $C \neq \emptyset$ and by Proposition 10; C can easily be shown to be closed.) Let **s**, **t** be as in the claim. Let s_0 be a proper subtree of $t_{f(0)} = t_k$. Consider the sequence $t_0, \ldots, t_{k-1}, s_0, s_1, s_2, \ldots$, that is, $(\mathbf{t})_k$ followed by **s**. By minimality of **t**, this sequence is not in C. Hence it contains an embedded pair of elements (the earlier one embedded in the later one). The embedded pair cannot occur in the prefix $(\mathbf{t})_k$ because **t** $\in$ C. It can also not be of the form $t_i \preceq s_j$, since then **t** would contain the embedded pair $t_i \preceq t_{f(j)}$. So, the embedded pair must occur in the postfix **s**.

As to part (2) of the claim, suppose **s** does not contain an infinite chain as subsequence. We now apply the same argument as in the second proof of Proposition 8: Then **s** contains an infinite number of finite chains, each starting to the right of the end of the previous finite chain and each maximal in the sense that it cannot be prolonged by an element occurring to the right of it in **s**. Now consider the last elements of these finite chains. These last elements constitute an infinite subsubsequence of **t**, containing by (1) of the claim an embedded pair. But that means that one of the maximal finite chains can be prolonged, a contradiction. This proves the claim.

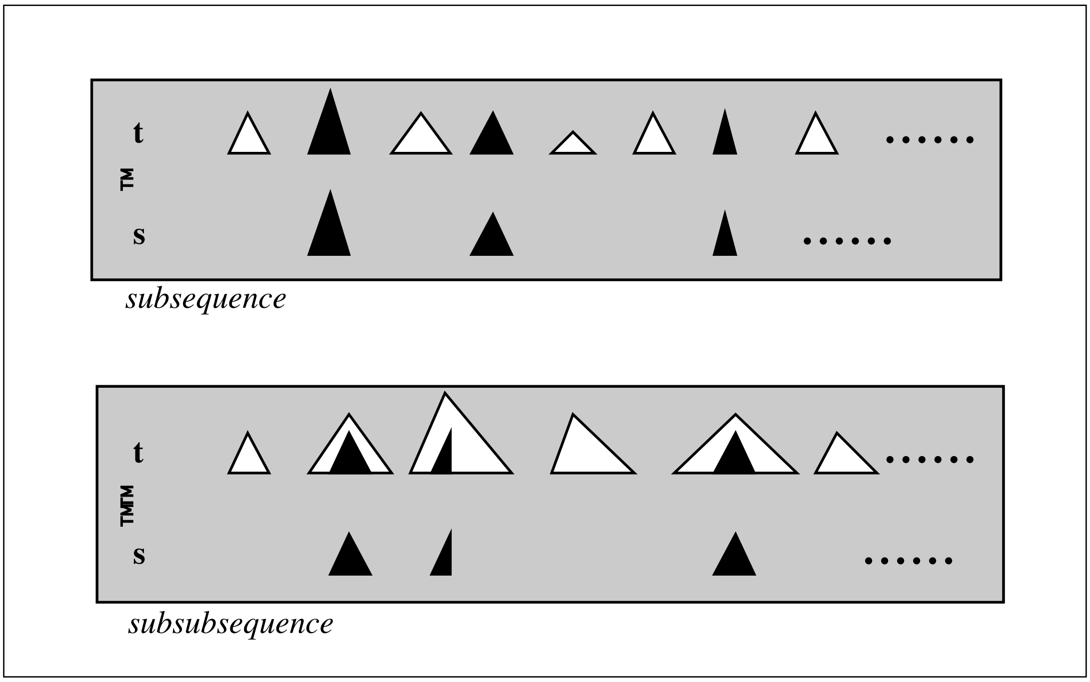


Figure 8.4

We will now apply a sieve procedure to the minimal counterexample sequence **t** $\in$ C. By Proposition 8 we can take a subsequence **t**' of **t** such that the root labels are weakly ascending. Of **t**' we take a subsequence **t*** with the property that the branching degrees of the roots are a weakly ascending sequence. By the claim every subsubsequence of **t*** still contains an infinite embedding chain.

Let us 'freeze' the elements in **t**', that is, we impose an ordering of the successors of each node in some arbitrary way. So the frozen trees in **t**' are no longer commutative trees, and we can speak of the first, second etc. 'arguments' of a node. (An argument of a node α is the subtree with as root a successor node β of α.)

The next step in the sieve procedure is done by considering the sequence of first arguments of (the roots of) the elements in t*. As this is a subsubsequence, it contains an infinite chain. Accordingly, we thin t* out, to the subsequence t**. This sequence has the property that its first arguments form a chain. Next, t** is thinned out by considering the sequence of the second arguments of t**. Again, this sequence contains a chain, and thinning t** accordingly yields the subsequence t***.

After at most N steps of the last kind, we are through. The result is then a chain, since the roots already satisfied the embedding condition (they form a weakly ascending chain), and the arguments are also related as chains. (See Figure 5.) However, this contradicts the assumption that **t** contains no embedded pair. Hence C is empty, and the restricted version of Kruskal's Tree Theorem is proved. ❑

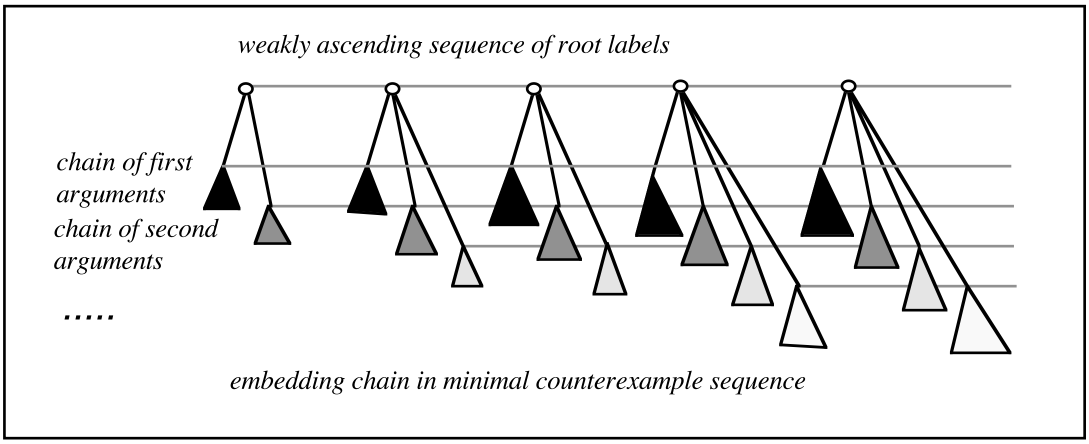


Figure 8.5

In the case that the branching degrees of the trees are not restricted, the above argument is not sufficient as the final part of the proof would not be applicable. However, the theorem and its proof above are ‘self-enhancing’, and we will now show how with little extra effort we can extend the theorem. To do this, we first introduce the terminology of well-quasi-orders and well-partial-orders.

8.15. DEFINITION. The binary relation $\leq$ is a *quasi-order* (qo) if it is reflexive and transitive. (So the relation $\twoheadrightarrow$ in a TRS is a qo.) If in addition $\leq$ is anti-symmetric (i.e. $x \leq y$ & $y \leq x \Rightarrow x = y$ for all x,y) then $\leq$ is a partial order (po). For a qo $\leq$ we define: $x < y$ iff $x \leq y$ & $\neg\, y \leq x$. (For a po it is equivalent to define $x < y$ iff $x \leq y$ & $x \neq y$, as usual.)

8.16. DEFINITION. Let $\langle X, \leq \rangle$ be a qo. A subset $Y \subseteq X$ is called a *cone* if $x \in Y$ & $x \leq y \Rightarrow y \in Y$ for all $x,y \in X$. The cone *generated by* $Y \subseteq X$, notation $Y\uparrow$, is the set $\{x \in X \mid \exists y {\in} Y\ y \leq x\}$. (It is the intersection of all cones containing Y.) A cone Z is *finitely generated* if $Z = Y\uparrow$ for some finite Y.

8.17. DEFINITION. Let $\langle X, \leq \rangle$ be a qo (po, respectively). Then $\langle X, \leq \rangle$ is a *well-quasi-order* (wqo) or *well-partial-order* (wpo) respectively, if every cone of X is finitely generated.

8.18. DEFINITION. Let $\langle X, \leq \rangle$ be a qo. A subset $Y \subseteq X$ is an *anti-chain* if the elements of Y are pairwise incomparable, i.e. for all $x,y \in Y$ such that $x \neq y$ we have neither $x \leq y$ nor $y \leq x$.

8.19. PROPOSITION. *Let* $\langle X, \leq \rangle$ *be a qo. Then the following conditions are equivalent:*

(i) $\langle X, \leq \rangle$ *is a wqo;*

(ii) X *contains no infinite descending chains* $x_0 > x_1 > x_2 > \ldots$ *and all anti-chains of* X *are finite;*

(iii) *for every infinite sequence of elements* $x_0, x_1, x_2, \ldots$ *in* X *there are* i, j *such that* $i < j$ *and* $x_i \leq x_j$.

PROOF. Exercise. ❑

8.19.1. EXAMPLE.
(i) The natural numbers with the usual ordering is a wqo (even a wpo).
(ii) The natural numbers with the discrete ordering is not a wqo.
(iii) Any finite set with the discrete ordering is a wqo.

8.19.2. REMARK. Another definition of wpo is: a wpo is a po such that every extension of it to a linear order is a well-ordering.

Now our first and easy upgrading of Theorem 8.14 is as follows. As labels at the nodes of trees we now admit the elements of some wqo. The definition of embedding (see Definition 8.2) generalizes by replacing the ordering of the natural numbers by that of the present wqo.

8.20. PROPOSITION. *Each infinite sequence of elements* $n_0, n_1, n_2, \ldots$ *in a wqo has a weakly ascending infinite subsequence.*

PROOF. Cf. the three proofs of Proposition 8. The first proof doesn't apply anymore. The second proof carries over verbatim. The third proof is slightly more complicated now. Define the partition of $[\mathbb{N}]^2$ into sets X, Y, Z as follows:

$$\{i,j\} \in X \text{ if } i < j \;\&\; n_i \preceq n_j$$
$$\{i,j\} \in Y \text{ if } i < j \;\&\; n_j \preceq n_i$$
$$\{i,j\} \in Z \text{ if } i < j \;\&\; n_i\,,\, n_j \text{ are unrelated.}$$

Now we find an infinite homogeneous set A. If its is homogeneous for X, we are done. For Y is impossible, as a wqo does not have infinite descending chains. For Z is also impossible, as a wqo does not have infinite anti-chains. ❑

8.20. KRUSKAL'S TREE THEOREM (restricted case; for wqo as label set)
Let $t_0, t_1, t_2, \ldots$ *be a sequence of trees in* $\mathbb{T}_N$. *Then for some* $i < j$: $t_i \preceq t_j$.
Rephrased: $\langle \mathbb{T}_N, \preceq \rangle$ *is again a wqo.*

PROOF. Verbatim as the proof of Theorem 14 above, now using Proposition 20 instead of 8. ❑

Next, we will upgrade the theorem by removing the restriction on branching degrees.

8.21. DEFINITION (embedding of words over a wqo)
Let $\langle S, \preceq \rangle$ be a wqo. Then $\langle S^*, \preceq^* \rangle$, the qo of words over S, is defined by: S^* is the set of words over S; $s_1 \ldots s_m \preceq^* t_1 \ldots t_n$ iff there is a 1-1 monotonic function f: $\{1,\ldots,m\} \to \{1,\ldots,n\}$ such that $s_i \preceq t_{f(i)}$ for each $i = 1,\ldots,m$.

8.22. COROLLARY (Higman's Lemma). *The set of words over a wqo is again a wqo.*

PROOF. Just turn a word 90 degrees, so that it is a unary-branching tree labeled from top to bottom with the elements from left to right making up the word. (See Figure 6.) Now apply Theorem 20. ❑

8.22.1. EXAMPLE. Consider words over the alphabet {a,b,c}. Then any infinite sequence of such words will contain an embedding pair of words, the first embeddable in the later one.

8.22.2. EXERCISE. Show that the set of words over a wpo is again a wpo.

8.22.2. REMARK. Define the ordinal associated to a wpo to be the sup of the ordinals of all linear extensions of that wpo. (A linear extension of a wpo is a well-founded po, hence it 'is' an ordinal.). A theorem of D. de Jongh asserts that if the ordinal of a wpo is α, then the ordinal of the wpo of words over the first wpo is $\leq \omega^{\beta}$, $\beta = \omega^{\alpha+1}$. (See Schmidt [78], De Jongh [77].)

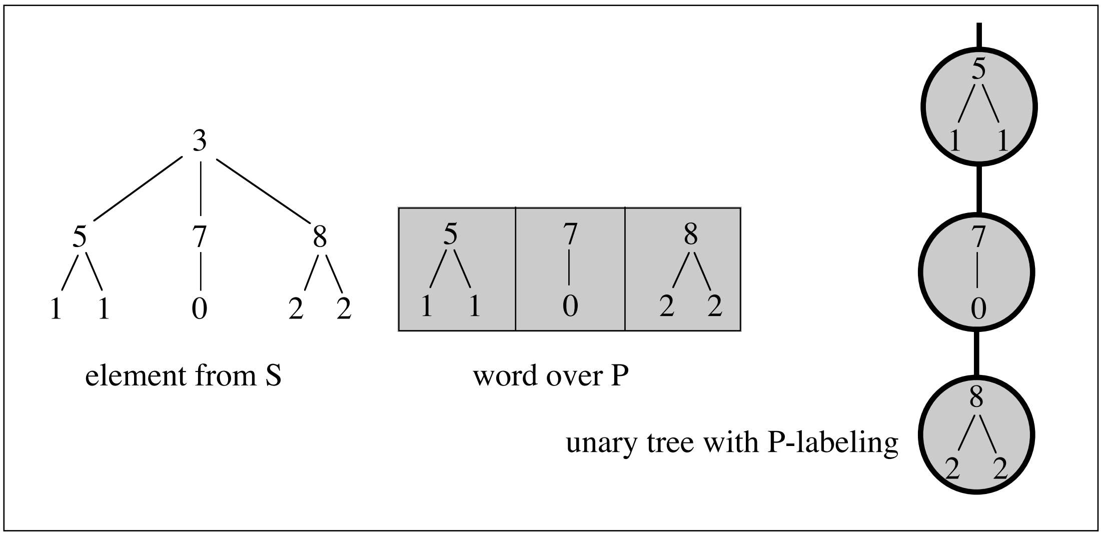


Figure 6

8.23. KRUSKAL'S TREE THEOREM (general case; for wqo as label set)
Let $t_0, t_1, t_2,...$ *be a sequence of trees in* $\mathbb{T}$. *Then for some* $i < j$: $t_i \preceq t_j$.
Rephrased: $\langle \mathbb{T}, \preceq \rangle$ *is again a wqo.*

PROOF. Consider again a minimal element **s** in the counterexample set C, as in the proof of Theorem 14. Now consider all proper subterms of elements s_n of **s**. Call this set P. We claim that P is a wqo. For suppose not, then there would be a sequence of elements in P, $\mathbf{t} = t_0, t_1,...$ without embedded pair. Now let t_0 be a subterm of s_k. Remove from **t** all elements (except possibly t_0) that are subterms of $s_0,...,s_{k-1}$. Call the result **t**'; this is still an infinite sequence without embedding pair. Now append **t**' behind $(\mathbf{s})_k$. As in the proof of Theorem we show that this sequence still cannot have an embedded pair; but this contradicts the minimality of **s**. This ends the proof of the claim.

By Corollary 8.22, the set of words over P is again a wqo. Now consider the minimal counterexample **s**, and (as in the proof of Theorem 14) filter out a subsequence **s*** such that the root labels form an embedding chain. Then consider of each element of **s*** the tuple (word) of its immediate subterms (see Figure 8.6). These must have, as words, an embedded pair, since P is a wqo. But together with the embedding of the root labels, this precisely constitutes an embedding of the corrsponding total elements in **s***. ❑

8.24. EXERCISE. Prove that $\langle \mathbb{T}, \preceq \rangle$ with labels a wpo is in fact a partial order; so Kruskal's theorem then states that $\langle \mathbb{T}, \preceq \rangle$ is a well-partial-order.

8.25. EXERCISE. Give a simple example showing that the well-partial-order $\langle \mathbb{T}, \preceq \rangle$, with natural numbers as labels, is not a linear order.

8.26. EXERCISE.
Show that ➪+ is a linear order. As it is well-founded, it corresponds to an ordinal. For connections with the ordinal Γ_0, the first impredicative ordinal, see Dershowitz [85]. For more about Kruskal's Tree Theorem and the connection with large ordinals, as well as a version of the Tree Theorem which is independent from Peano's Arithmetic, see Smorynski [82] and Gallier [87].

Note that the inverse of ➪+ is a linearization of $\preceq$, therefore it is well-founded.

8.27. EXERCISE.
Every element of $\langle \mathbb{T}, \preceq \rangle$ labeled with natural numbers corresponds with an ordinal, namely its place in the well-founded ordering given by ➪+. Place the elements of $\langle \mathbb{T}, \preceq \rangle$ along the sequence of ordinals as far as possible.

7.7. DEFINITION. Let $\mathbb{T}^*$ be the set of trees as above where now some of the nodes may be marked with (a single) *. So $\mathbb{T} \subseteq \mathbb{T}^*$. Example: see Figure 7.3(b); this tree will be denoted by 3*(5,7(9*),8*(0(1,5))).

7.8. NOTATION. As before, $n(t_1,...,t_k)$ will be written as n(**t**). The t_i (i = 1,...,k) are now elements of $\mathbb{T}^*$. Further, if $t \equiv n(t_1,...,t_k)$ then t* stands for $n^*(t_1,...,t_k)$.

7.9. DEFINITION. On $\mathbb{T}^*$ we define a reduction relation ➪ as follows.

(i) *place marker at the top:*

n(**t**) ➪ n*(**t**) (**t** = $t_1,...,t_k$; k ≥ 0)

(ii) *make copies below lesser top:*

if n > m, then n*(**t**) ➪ m(n*(**t**),...,n*(**t**)) (j ≥ 0 copies of n*(**t**))

(iii) *push marker down:*

n*(s,**t**) ➪ n(s*,...,s*,**t**) (j ≥ 0 copies of s*)

(iv) *select argument:*

$n^*(t_1,...,t_k) ⇨ t_i$ $(i \in \{1,...,k\}, k \geq 1)$

It is understood that these reductions may take place in a context, i.e.

if t ⇨ s, then n(---, t,---) ⇨ n(---, s,---).

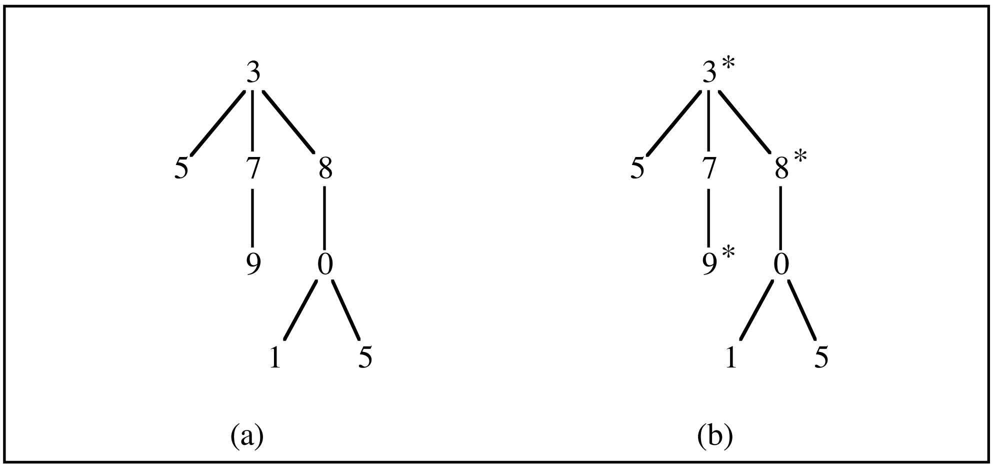


Figure 7.3

We write ⇨+ for the transitive (but not reflexive) closure of ⇨.

7.10. EXAMPLE. (i) Figure 7.4 displays a reduction in $\mathbb{T}^*$.

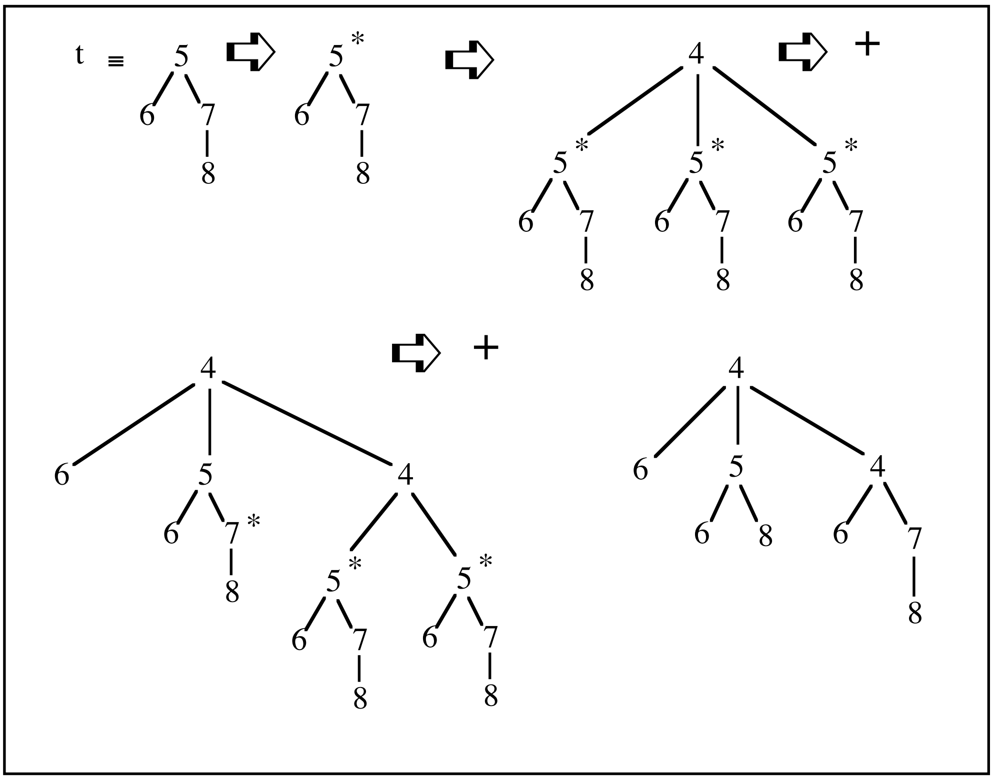


Figure 7.4

(ii) $n(\mathbf{t}) \Rightarrow^+ m(\mathbf{t})$ if $n > m$.

(iii) $n(s, \mathbf{t}) \Rightarrow^+ n(\mathbf{t})$.

(iv) 0^* is a normal form with respect to $\Rightarrow$.

Intuitively, attaching a marker as in rule (1) signifies a command to make the term smaller. The other rules express one step of the execution of this command, which is fully executed if all *-markers have disappeared.

Now we have the following proposition (of which (i) is nontrivial to prove):

7.11. PROPOSITION. (i) $\Rightarrow^+$ *is a strict partial order on* $\mathbb{T}$, (ii) *if* $s \preceq t$, *then* $t \Rightarrow^* s$.

(Here $\Rightarrow^*$ is the transitive-reflexive closure of $\Rightarrow$.) Note that the reverse of (ii) does not hold; for, if $t \Rightarrow^* s$, then s may have more nodes than t (see e.g. the example in Figure 7.4 above), hence: not $s \preceq t$. Note also that clause (iii) in Definition 7.2 (sup preserving) is necessary: we do not have, e.g., $1(0(0, 0)) \Rightarrow^* 1(0, 0)$ (without clause (iii) $1(0, 0)$ could be embedded in $1(0(0, 0))$). The proof of (ii) is trivial, using Remark 7.5.

Clearly, the reduction $\Rightarrow$ is not SN in $\mathbb{T}^*$; for, consider the second step in Figure 7.4: there the right hand side contains a copy of the left-hand side. However:

7.12. THEOREM. *The relation* $\Rightarrow^+$*, restricted to* $\mathbb{T}$*, is a well-founded partial ordering. Or, rephrased, the relation* $\Rightarrow^+$*, restricted to* $\mathbb{T}$*, is SN.*

PROOF. Suppose there is an infinite sequence of trees $t_0, t_1, t_2, \ldots$ in $\mathbb{T}$ such that

$$t_0 \Rightarrow^+ t_1 \Rightarrow^+ t_2 \Rightarrow^+ \ldots \Rightarrow^+ t_i \Rightarrow^+ \ldots \Rightarrow^+ t_j \Rightarrow^+ \ldots$$

then by Kruskal's Tree Theorem for some $i < j$ we have $t_i \preceq t_j$, hence $t_j \Rightarrow^* t_i$. But then we have

$$t_i \Rightarrow^+ \ldots \Rightarrow^+ t_j \Rightarrow^* t_i, \text{ so } t_i \Rightarrow^+ t_i,$$

which is impossible as $\Rightarrow^+$ is a *strict* partial order.

7.13. APPLICATION (Dershowitz [87]). Let a TRS R, computing disjunctive normal forms, as in Table 7.1 be given. To prove that R is SN.

$\neg\neg x$	$\to$	x
$\neg(x \vee y)$	$\to$	$(\neg x \wedge \neg y)$
$\neg(x \wedge y)$	$\to$	$(\neg x \vee \neg y)$
$x \wedge (y \vee z)$	$\to$	$(x \wedge y) \vee (x \wedge z)$
$(y \vee z) \wedge x$	$\to$	$(y \wedge x) \vee (z \wedge x)$

Table 7.1

Choose a 'weight' assignment $\vee \to 1$, $\wedge \to 2$, $\neg \to 3$. Now a reduction in R corresponds to a ➪+ reduction in $\mathbb{T}$ (and hence it is also SN) using the fact that we have LHS ➪+ RHS for each rule LHS $\to$ RHS of R:

3(3(t))	➪+	t
3(1(t,s))	➪+	2(3(t),3(s))
3(2(t,s))	➪+	1(3(t),3(s))
2(t,1(s,r))	➪+	1(2(t,s),2(t,r))
2(1(s,r),t)	➪+	1(2(s,t),2(r,t))

E.g. the second rule:

3(1(t,s)) ➪ 3*(1(t,s)) ➪ 2(3*(1(t,s)), 3*(1(t,s))) ➪+
2(3(1*(t,s)), 3(1*(t,s))) ➪+ 2(3(t),3(s)).

7.14. REMARK. It is also possible to formulate Kruskal's Tree Theorem in a form somewhat closer to the terminology of term rewriting. The difference is that we now work with function symbols having fixed arities, that the 'term trees' are not commutative, and that the embedding relation looses the aspect of label increasingness (clause (iv) of Definition 7.2). First define the following relation:

7.14.1. DEFINITION. Let t, s $\in$ Ter(Σ). We say that t is *embeddable in* s, notation t << s, if s $\twoheadrightarrow_S$ t with respect to the TRS (Σ, S) consisting of the rules $F(t_1, ..., t_n) \to t_i$ for all $1 \le i \le n$ and all n-ary F $\in \Sigma$. (S stands for simplification.)

7.14.2. KRUSKAL'S TREE THEOREM. *Let* $t_1, t_2, ...$ *be a a sequence of terms, such that in the sequence only finitely many symbols (function symbols, constants, variables) appear. Then for some* i, j *with* i < j *we have* $t_i << t_j$.

(Note that the finiteness condition is now necessary,. Otherwise the infinite sequence of different variables $x_0, x_1, x_2, ...$ would refute the theorem.)

7.14.3. The recursive path ordering is now defined as follows, using the auxiliary signature $\Sigma^* = \Sigma \cup \{F^* \mid F \in \Sigma\}$ (F a function or constant symbol from Σ; F^* has the same arity as F). So $Ter(\Sigma) \subseteq Ter(\Sigma^*)$. Now suppose Σ finite and suppose that function and constant symbols of Σ are partially ordered by >. We define a reduction relation ⇨ on $Ter(\Sigma^*)$, with the following reduction rules.

(1)	$F(\mathbf{t})$	⇨ $F^*(\mathbf{t})$
(2)	$F^*(\mathbf{t})$	⇨ $G(F^*(\mathbf{t}), ..., F^*(\mathbf{t}))$ *if* $F > G$
(3)	$F^*(\mathbf{t})$	⇨ t_i $(i = 1, ..., n)$
(4)	$F^*(\mathbf{p}, G(\mathbf{s}), \mathbf{q})$	⇨ $F(\mathbf{p}, G^*(\mathbf{s}), \mathbf{q})$

Table 7.2

Here $\mathbf{t} = t_1, ..., t_n$ and $\mathbf{s} = s_1, ..., s_m$ with $t_i, s_i \in Ter(\Sigma^*)$. Furthermore, $F, G \in \Sigma$ are function symbols with arities n, $m \geq 0$ respectively (so in rule (2) there are in the right-hand side m copies of $F^*(\mathbf{t})$). In rule (1), (2) the arity of F may be 0; in rule (3), (4) it is clear that the arity of F has to be at least 1. In (4), $\mathbf{p}, G(\mathbf{s}), \mathbf{q}$ is a sequence of n elements from $Ter(\Sigma^*)$, where $\mathbf{p}$, $\mathbf{q}$ may be empty sequences. With ⇨* we denote the transitive reflexive closure of ⇨, with ⇨+ the transitive closure. Note that the simplification reduction $\twoheadrightarrow_S$ is contained in ⇨*, i.e. if s $\twoheadrightarrow_S$ t then s ⇨* t. The rest is analogous to the case of commutative trees above.

So ⇨+ is a well-founded ordering on $Ter(\Sigma)$. This ordering is called the *recursive path ordering*. The recursive path ordering can be used for termination proofs of TRSs as follows.

7.14.3. THEOREM. Let (Σ, R) be a TRS with finite Σ. Suppose the function and constant symbols of Σ can be partially ordered in such a way that for the corresponding recursive path order ⇨+ we have, for every reduction rule $s \to t$ of R, that s ⇨+ t. Then R is SN.

The proof follows immediately since s ⇨+ t implies $C[s^\sigma]$ ⇨+ $C[t^\sigma]$ for every context C[ ] and instantiation σ.

7.15. EXAMPLE. Consider the string rewrite system (or Semi-Thue System), see section 4.3 in chapter 4, given as an example in Dershowitz & Jouannaud [90], over the alphabet {0, 1} and with the rules:

(1) $10 \to 0001$
(2) $01 \to 1$
(3) $11 \to 0000$

(4) $00 \to 0$

So in proper TRS format these rules stand for $1(0(x)) \to 0(0(0(1(x))))$, etc.With the ordering $1 > 0$ we have (dropping all brackets in the convention of association to the right):

(1) 10x ➪+ 000x, since 10x ➪ 1*0x ➪ 01*0x ➪ 001*0x ➪ 0001*0x ➪ 00010*x ➪ 0001x. Likewise for rules (2-4). Hence this STS is SN.

7.16. REMARK. (i) The termination proof method above does not work when a rule is present of which the left-hand side is embedded (in the sense of Definition 7.14.1) in the right-hand side, as in $f(s(x)) \to g(s(x), f(p(s(x))))$. (For, if it would, then we would have a contradiction with the acyclicity of ➪+). For an extension of Kruskal's Theorem, leading to a method which also can deal with this case, see Kamin & Lévy [80] and Puel [86].

(ii) Another example that resists a direct application of RPO as above, is:

$$g(x,y) \to h(x,y)$$
$$h(f(x),y) \to f(g(x,y)).$$

There are several ways to deal with such cases though, but not treated here.

(iii) A third example were the proof method above does not work, is when an associativity rule

$$(x.y).z \to x.(y.z)$$

is present.

(iv) The same problem occurs in the TRS for Ackermann's function:

$$A(0,x) \to S(x)$$
$$A(S(x),0) \to A(x,S(0))$$
$$A(S(x),S(y)) \to A(x,A(S(x),y))$$

What we need here is the *lexicographic path ordering* of Kamin and Lévy, see Dershowitz [85]. Essentially this says that a reduction in complexity in the first argument of A outweighs an increase (strictly bounded by the complexity of the original term) in the second argument. In fact, an ordering with the same effect can easily be described in the framework of reduction with markers * as follows. Add to the rules in definition 7.14.3:

(5) *simplify left argument*

n*(**t**) ➪ $n(t_1$*, n*(**t**),..., n*(**t**)) (**t** = $t_1,...,t_k$ ($k \geq 1$); k-1 copies of n*(**t**))

Example: A(S(x),S(y)) ⇨
A*(S(x),S(y)) ⇨
A(S*(x),A*(S(x),S(y))) ⇨
A(x,A*(S(x),S(y))) ⇨
A(x,A(S(x),S*(y))) ⇨
A(x,A(S(x),y)).

7.17. REMARK. Nested multisets. Nested multisets over $(\mathbb{N}, <)$ arise (roughly) as follows. First take the multisets over $(\mathbb{N}, <)$, result $(\mathbb{N}^{\mu}, <^{\mu})$. Iterating this we have multisets of multisets, $(\mathbb{N}^{\mu\mu}, <^{\mu\mu})$. Iterating this ω times and taking the limit we have the nested multisets over $\mathbb{N}$, notation $(\mathbb{N}^{\mu *}, <^{\mu *})$; they can be represented as finite commutative trees $\in \mathbb{T}$ with natural numbers at terminal nodes and 0 at all non-terminal nodes. The nested multiset ordering $<^{\mu *}$ is now just the recursive path order $⇨^{+}$. Replacing the natural number n by the multiset of n copies of 0, {0,...,0}, we see that the nested multiset order over $\mathbb{N}$ is the same as that over {0}. Hence we can take all labels of trees $\in \mathbb{T}$ equal to 0. Figure 7.5 contains a comparison of two such nested multisets by means of a ⇨-reduction. Note that ⇨ now uses all clauses in Definition except (ii).

7.18. REMARK. Higman's Lemma.

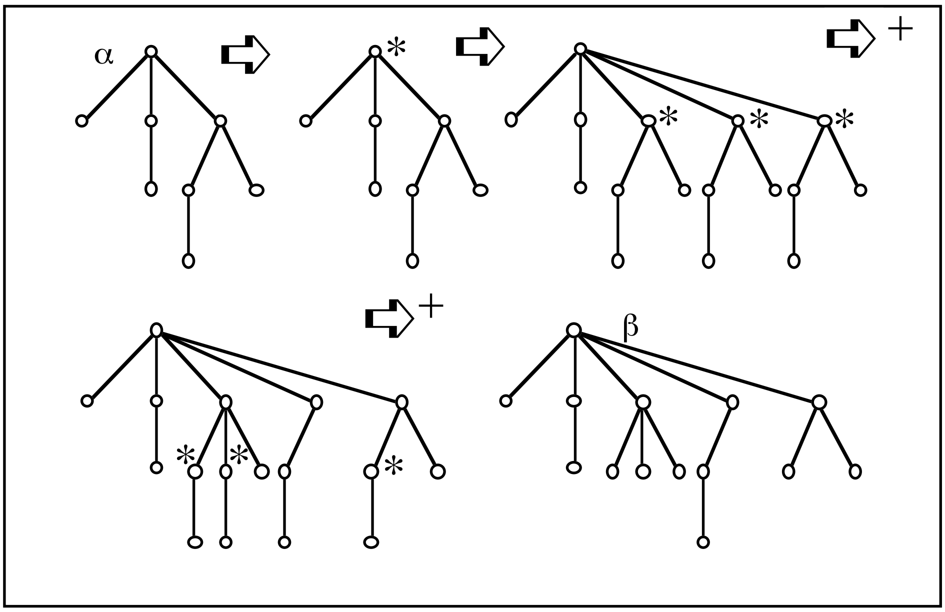


Figure 7.5

7.18. REMARK. $⇨^{+}$ is a linear order. As it is well-founded, it corresponds to an ordinal. For connections with the ordinal Γ_0, the first impredicative ordinal, see Dershowitz [85]. For more about Kruskal's Tree

Theorem and the connection with large ordinals, as well as a version of the Tree Theorem which is independent from Peano's Arithmetic, see Smorynski [82] and Gallier [87].

9

Higher-order rewriting

In this topic we introduce a new syntactic feature to term rewriting, that is, new as compared to the first-order TRSs that we have dealt with so far; namely *bound variables*. Of course bound variables are familiar already from lambda calculus and predicate logic. Roughly, we will take the union of the syntax of lambda calculus and of first-order TRSs, in order to arrive at an encompassing framework for term rewriting. The original idea stems from an unpublished note of P. Aczel (Aczel [78]), who proved the Church-Rosser theorem for ‘consistent reduction schemes’. In this topic we will explain this notion, under the name of Combinatory Reduction Systems (CRSs), by several examples. A closely related approach to ‘higher-order rewriting’ as the terminology is nowadays, was given by T. Nipkow (Nipkow [91]), on the basis of simply typed lambda calculus as the underlying substitution mechanism (Higher-order Rewrite Systems, HRSs). Still other higher-order rewrite frameworks have been introduced, e.g. Khasidashvili [90] (Expression Reduction Systems). The theory of higher-order rewriting was further developed by Van Raamsdonk [96, 99] and Van Oostrom[94, 99]. For a survey on higher-order rewriting, see Van Raamsdonk [99]. We start with a sequence of examples before presenting the formal definitions.

9.1. EXAMPLE. **λ-calculus**. The only rewrite rule is the β-reduction rule:

$$(\lambda x.Z_1(x))Z_2 \to Z_1(Z_2),$$

presented in an informal notation; the formal notation would use a substitution operator [:=] and we would write $(\lambda x.M)N \to M\ [x := N]$. Still, this informal notation has a direct appeal, and in the sequel we will make it formal; this is essential for the syntax definition of CRSs.

9.2. EXAMPLE.**Polyadic λ-calculus**. Here we have n-ary λ-abstraction and reduction rules (β_n) for every $n \geq 1$:

$$(\beta_n) \qquad (\lambda x_1 x_2 ... x_n.\ Z_0(x_1,x_2,...,x_n))Z_1 Z_2 ... Z_n \to Z_0(Z_1,Z_2,...,Z_n).$$

9.3. EXAMPLE.**μ-calculus**. This is the well-known notation system designed to deal with recursively defined objects (processes, program statements, ...) with as basic rewrite or reduction rule:

$$\mu x.Z(x) \to Z(\mu x.Z(x)),$$

which would read in the usual notation: $\mu x.Z(x) \to Z([x := \mu x.Z(x)]$.

9.4. EXAMPLE.**Proof Normalization**.

$$\mathsf{P}(\mathsf{L}Z_0)(\lambda x.Z_1(x))(\lambda y.Z_2(y)) \to Z_1(Z_0)$$
$$\mathsf{P}(\mathsf{R}Z_0)(\lambda x.Z_1(x))(\lambda y.Z_2(y)) \to Z_2(Z_0)$$

The operational meaning of this pair of rewrite rules should be self-explaining: according to whether Z_0 is prefixed by L or R it is substituted in the left or the right part of the 'body' of the redex headed by P, for all the free occurrences of x respectively y. The rules occur as *normalization procedures for proofs* in Natural Deduction (Prawitz [71], p.252), Girard [87]), albeit not in the present linear notation. The rules concer "∨-reduction". (For more explanation see Klop [80].)

9.5. EXAMPLE. **λ-calculus with δ-rules of Church**. This is an extension of λ-calculus with a constant δ and a possibly infinite set of rules of the form

$$\delta M_1 ... M_n \to N$$

where the M_i (i = 1,...,n) and N are closed terms and the M_i are moreover in "βδ-normal form", i.e. contain no β-redex and no subterm as in the left-hand side of a δ-rule. To ensure orthogonality (defined below) there should moreover not be two left-hand sides of different δ-rules of the form

$\delta M_1...M_n$ and $\delta M_1...M_m$, $m \geq n$. (So every left-hand side of a δ-rule is a normal form with respect to the other δ-rules.)

The account below follows Klop [80] (see also Klop, Van Oostrom & Van Raamsdonk [93]). The concept of a CRS was first suggested in Aczel [78], where a confluence proof for a subclass of the orthogonal CRSs was given, the 'consistent reduction schemes'. We will now give the formal definitions.

9.6. Alphabet of a Combinatory Reduction System.

The *alphabet* of a CRS consists of

(i) a set $\text{Var} = \{x_n \mid n \geq 0\}$ of *variables* (also written as x,y,z,...);

(ii) a set Mvar of *metavariables* $\{Z_n^k \mid k,n \geq 0\}$; here k is the *arity* of Z_n^k;

(iii) a set of function symbols F, G,..., each with a fixed arity; sometimes these function symbols have a familiar notation such as λ, μ, ∃, ... ;

(iv) a binary operator for abstraction, written as [-].-;

(v) improper symbols ( ) and [ ].

The arities k of the metavariables Z_n^k can always be read off from the term in which they occur—hence we will often suppress these superscripts. E.g. in $(\lambda x.Z_0(x))Z_1$ the Z_0 is unary and Z_1 is 0-ary.

9.7. Term formation in a Combinatory Reduction System.

9.7.1. DEFINITION. The set Mter of *meta-terms* of a CRS with alphabet as in 9.6 is defined inductively as follows:

(i) constants and variables are meta-terms;

(ii) if t is a meta-term, x a variable, then ([x]t) is a meta-term, obtained by *abstraction*;

(iii) if F is an n-ary function symbol ($n \geq 0$) and $t_1,...,t_n$ are metaterms, then $F(t_1,...,t_n)$ is a metaterm;

(iv) if $t_1,...,t_k$ ($k \geq 0$) are meta-terms, then $Z_n^k(t_1,...,t_k)$ is a meta-term (in particular the Z_n^0 are meta-terms).

Note that meta-variables Z_n^{k+1} are not meta-terms; they need arguments. Meta-terms in which no metavariable Z occurs, are *terms*. Ter is the set of terms.

9.7.2. NOTATION.

(i) An iterated abstraction meta-term $[x_1](...([x_{n-1}]([x_n]t))...)$ is written as $[x_1,...,x_n]t$ or $[\mathbf{x}]t$ for $\mathbf{x} = x_1,...,x_n$. For a unary function symbol F, the meta-term $F([\mathbf{x}]t)$ will be written as $F[\mathbf{x}].t$. For instance, $\lambda x.t$ abbreviates $\lambda\ ([x]t)$.

(iii) We will not be very precise about the usual problems with renaming of variables, α-conversion etc. That is, this is treated like in λ-calculus when one is not concerned with implementations. Thus we will adopt the following conventions:

- All occurrences of abstractors $[x_i]$ in a meta-term are different; e.g. $\lambda xx.t$ is not legitimate, nor is $\lambda x.(t\lambda x.t')$.
- Furthermore, terms differing only by a renaming of bound variables are considered syntactically equal. (The notion of 'bound' is as in λ-calculus: in [x]t the free occurrences of x in t (hence by (i) *all* occurrences) are bound by the abstractor [x].)

9.7.3. DEFINITION. A term is *closed* if every variable occurrence is bound.

9.8. Rewrite rules of a Combinatory Reduction System.

A *rewrite* (or *reduction*) *rule* in a CRS is a pair (t, s), written as $t \to s$, where t,s are meta-terms such that:

(i) t has the form $F(t_1,...,t_n)$;

(ii) t, s are closed meta-terms;

(iii) the metavariables Z_n^k that occur in s, also occur in t;

(iv) the metavariables Z_n^k in t occur only in the form $Z_n^k(x_1,...,x_k)$ where the x_i $(i = 1,...,k)$ are variables (no meta-terms). Moreover, the x_i are pairwise distinct.

If, moreover, no metavariable Z_n^k occurs twice or more in t, the rewrite rule $t \to s$ is called *left-linear*.

In order to generate actual rewrite steps from the rewrite rules, we have to define substitution:

9.9. Extracting the reduction relation. It requires some subtlety to extract from the rewrite rules the actual rewrite relation that they generate. First we define *substitutes* (we adopt this name from KAHRS [92]).

9.9.1. DEFINITION. Let t be a term in a CRS.

(i) Let $x_1,...,x_n$ be a string of pairwise distinct variables. Then $\underline{\lambda}(x_1,...,x_n).t$ is an n-ary substitute. We use $\underline{\lambda}$ as a 'meta-λ' to distinguish it from the one of λ-calculus.

(ii) The variables $x_1,...,x_n$ occurring in t are bound in the substitute $\underline{\lambda}$ $(x_1,...,x_n).t$. They may be renamed in the usual way, provided no name clashes occur. Renamed versions of a substitute are considered identical. The free variables in $\underline{\lambda}$ $(x_1,...,x_n).t$ are the free variables of t except $x_1...x_n$.

(iii) An n-ary substitute $\underline{\lambda}$ $(x_1,...,x_n).t$ may be applied to an n-tuple $(t_1,...,t_n)$ of terms from the CRS, resulting in the following *simultaneous substitution*:

$$(\underline{\lambda}\ (x_1,...,x_n).t)\ (t_1,...,t_n) = t[x_1:= t_1,..., x_n:= t_n]$$

9.9.2. DEFINITION. A *valuation* is a map σ assigning to an n-ary metavariable Z an n-ary substitute:

$$\sigma(Z) = \underline{\lambda}\ (x_1,...,x_n).t$$

Valuations are extended to a homomorphism on metaterms as follows:

(i) $\sigma(x) = x$ for $x \in Var$;

(ii) $\sigma([x]t) = [x]\ \sigma(t)$;

(iii) $\sigma(F(t_1,...,t_n)) = F(\sigma(t_1),..., \sigma(t_n))$

(iv) $\sigma(Z(t_1,...,t_n)) = \sigma(Z)\ (\sigma(t_1),..., \sigma(t_n))$.

(So if $\sigma(Z) = \underline{\lambda}\ (x_1,...,x_n).t$, then $\sigma(Z(t_1,...,t_n)) = t[x_1:=\sigma(t_1),...,x_n:=\sigma(t_n)]$.)

We will now formulate some 'safety conditions' for instantiating rewrite rules to actual rewrite steps. Intuitively, we could comprise their description as follows: *rename bound variables as much as possible*, in order to avoid name clashes, i.e. free variables x being captured unintentionally by abstractors [x].

9.9.3. DEFINITION. (i) Let $t \rightarrow s$ be a rewrite rule. A renaming of that rule (by renaming the bound variables in t, s) will be called a *variant* of the rule.

(ii) Let σ be a valuation. Then a variant of σ originates by renaming the bound variables in the substitutes σ(Z).

(iii) Let $t \rightarrow s$ be a rewrite rule and σ a valuation. Then $t \rightarrow s$ is called *safe* for σ, if for no Z in t, the substitute σ(Z) has a free variable x occurring in an abstraction [x] of t.

(iv) Furthermore, σ is called safe (with respect to itself) if there are no two substitutes $\sigma(Z)$ and $\sigma(Z')$ such that $\sigma(Z)$ contains a free variable x which appears also bound in $\sigma(Z')$.

Note that for every rewrite rule $t \to s$ and valuation σ there are variants σ' and $t' \to s'$ such that σ' is safe and $t' \to s'$ is safe for σ'.

Example. The η-reduction rule variant $\lambda x.\ Zx \to Z$, or in full notation $\lambda([x]@(Z, x)) \to Z$, is not safe for σ with $\sigma(Z) = x$. The variant $\lambda y.\ Zy \to Z$ is safe for σ.

9.9.4. DEFINITION. Let t be a term of a CRS and let s be a subterm occurrence in t. Then the result of replacing the occurrence of s by ø, indicating an open place, is called a *context*. We write C[ ] for a context. The result of replacing ø by a term s in C[ ] is C[s].

9.9.5. DEFINITION. (i) Let $t \to s$ be a rewrite rule version which is safe for the safe valuation σ. Then $\sigma(t) \to \sigma(s)$ is called a *rewrite*. The term $\sigma(t)$ is called a *redex*.

(ii) Let $\sigma(t) \to \sigma(s)$ be a rewrite, and C[ ] a context. Then $C[\sigma(t)] \to C[\sigma(s)]$ is called a *rewrite step* (reduction step).

(iii) As always, ÿ is the transitive reflexive closure of the one step rewrite relation $\to$ on terms.

9.9.6. REMARK. (i) We need $t \to s$ to be safe for σ, in (i) above, to prevent variable capture when evaluating the lefthand-side of the rule.

(ii) We need σ to be safe (with respect to itself) because otherwise undesired variable captures take place in evaluating the righthand-sides of rules. E.g. consider Z(Z') with σ such that $\sigma(Z) = \underline{\lambda}y.(\lambda x.xy)$ and $\sigma(Z') = x$ (so σ is not safe). Then $\sigma(Z(Z')) = \sigma(Z)(\sigma(Z')) = (\underline{\lambda}y.(\lambda x.xy))(x) = \lambda x.xx$, with variable capture.

(iii) Note that free variables in the rewrite $\sigma(t) \to \sigma(s)$ may be captured by the context C[ ] in which it is embedded to form a rewrite step $C[\sigma(t)] \to C[\sigma(s)]$; but that is intended!

9.9.7. EXAMPLE. In this example we write t^σ instead of $\sigma(t)$. We reconstruct a step according to the β-reduction rule of λ-calculus (written in the usual, applicative, notation):

$$(\lambda x.\ Z(x))Z' \to Z(Z').$$

Let the valuation $Z^\sigma = \underline{\lambda}x.\ yxx$, $Z'^\sigma = ab$ be given. Then we have the reduction step (in boldface):

$((\lambda x.\ Z(x))Z')^\sigma$

$= (\lambda x.\ Z(x)^\sigma)Z'^\sigma$

$= (\lambda x.\ Z^\sigma\ (x^\sigma))Z'^\sigma$

$= (\lambda x.(\underline{\lambda} x.yxx)(x))(ab)$

$= \mathbf{(\lambda x.\ yxx)(ab)} \rightarrow$

$(Z(Z'))^\sigma$

$= Z^\sigma\ (Z'^\sigma\ )$

$= (\underline{\lambda} x.yxx)(ab)$

$= \mathbf{y(ab)(ab)}$.

9.9.8. REMARK. (i) Note that in the CRS format there is no need for explicitly requiring that some variables are not allowed to occur in instances of metavariables. For instance, in F([x]Z), an instance of Z is not allowed to contain freee occurrences of x. In λ-calculus such a requirement cannot be made in the system itself; it has to be stated in the meta-language, as is done for the η-rule. In this sense the CRS formalism is more expressive than that of λ-calculus.
(ii) The requirement discussed in (i) is necessary: for, consider e.g. the rule $\tau x.\ xZ \rightarrow Z$. Suppose we would not require that Z cannot have free x's. Then $\tau x.\ xx \rightarrow x$; but that would mean that a closed term rewrites to an open term, i.e. free variables appear out of the blue, which of course is disallowed. One may ask why this is not the case for the rule $\tau x.\ xZ(x) \rightarrow Z(x)$; the answer is that this is not a legitimate rule because the righthand-side is not a closed metaterm.

We will now give a more precise definition of overlap and orthogonality.

9.9.9. DEFINITION. Let R be a CRS containing rewrite rules $\{r_i = t_i \rightarrow s_i \mid i \in I\}$.
(i) R is *non-overlapping* if the following holds:

(1) Let the left-hand side t_i of r_i be in fact $t_i(Z_1(\mathbf{x}_1),\ldots,Z_m(\mathbf{x}_m))$ where all metavariables in t_i are displayed. Now if the r_i-redex $\sigma(t_i(Z_1(\mathbf{x}_1),\ldots,Z_m(\mathbf{x}_m))$ contains an r_j-redex $(i \neq j)$, then this r_j-redex must be already contained in one of the $\sigma(Z_p(\mathbf{x}_p))$.

(2) Likewise if the r_i-redex *properly* contains an r_i-redex.

(ii) R is *left-linear* if all t_i are linear. A metaterm is linear if it does not contain multiple occurrences of the same metavariable. (Example: $\rho x.\ xZ(x)$ is linear; $\alpha xy.\ F(Z(x), Z(y))$ is not linear.)

(iii) R is *orthogonal* if it is non-overlapping and left-linear.

Above, all CRSs had an *unrestricted term formation* by some inductive clauses. However, often one will be interested in CRSs where some restrictions on term formation are present. A typical example is λI-calculus, where the restriction is that in a subterm λx. t there must be at least one occurrence of x in t. (This requirement makes the λI-calculus *non-erasing*, and Church proved that a λI-term is SN iff it is WN. (Cf. Topic 6, Theorem 6.9). As a consequence, if a λI-term t has a normal form then also every subterm of t has a normal form; this fact was Church's primary motivation for considering λI-calculus.)

Other typical examples of restricted term formation arise when *types* are introduced, as in typed λ-calculus (λ^τ-calculus) or typed Combinatory Logic (CL^τ) (see Hindley & Seldin [86]). In a simple way a type restriction occurs already when one considers *many-sorted* TRSs (not treated here). This leads us to the following definition:

9.9.10. DEFINITION. (i) Let (R, $\rightarrow_R$) be a CRS as defined above. Let T be a subset of Ter(R), which is closed under $\rightarrow_R$. Then (T, $\rightarrow_R$|T), where $\rightarrow_R$|T is the restriction of $\rightarrow_R$ to T, is a *substructure* of (R, $\rightarrow_R$) .

(ii) If (R, $\rightarrow_R$) is orthogonal, so are its substructures.

A large part of the theory for orthogonal TRSs carries over to orthogonal CRSs (see Klop [80]). The main fact is:

9.10. THEOREM. *All orthogonal CRSs are confluent.*

Just as for the case of first-order TRSs, one can define critical pairs. (See Nipkow [91] for doing so in the framework of HRSs.) When all critical pairs of a CRS are trivial and the CRS is left-linear, it is called *weakly orthogonal*. The paradigm example of a weakly orthogonal CRS is λβη-calculus. Now there is the following strengthening of Theorem 9.10 (see (Van Oostrom & van Raamsdonk [94b]):

9.11. THEOREM. *All weakly orthogonal CRSs are confluent.*

9.12. REMARK. (i) Also Church's Theorem (cf. Theorem 6.9) generalizes to orthogonal CRSs. Here the definition of 'non-erasing' reduction rule for CRSs generalizes from that for TRSs as follows: A rule $t \to s$ is non-erasing if all metavariables Z occurring in t, have an occurrence in s *which is not in the scope of a metavariable* (i.e. not occurring in an argument of a metavariable). Without this proviso, which for TRSs is vacuously fulfilled since there all metavariables in the rewrite rules are 0-ary, also rules like the β-reduction rule of λ-calculus $(\lambda x.Z(x))Z' \to Z(Z')$ would be non-erasing, which obviously is not the intention.

(ii) As to reduction strategies: here the situation resembles again that of TRSs. In fact, in Table 6.2 (Topic 6) one may replace "TRSs" everywhere by "CRSs". Similar for standardization and normalization: in general there is no standardization of reductions possible, but for left-normal CRSs (cf. Definition 6.12.6), among which λ-calculus, there is.

We conclude this topic with some examples of 'important' lambda calculi.

9.13. EXAMPLE. (Aczel [78]). λ-calculus extended with constants D_0, D_1, R_n, J, $\underline{0}$, S and rules as in Table 9.1 is a left-normal orthogonal CRS.

Pairing:	$D_0(DZ_1Z_2)$	$\to Z_1$
	$D_1(DZ_1Z_2)$	$\to Z_2$
Definition by cases:	$R_nQ_1Z_1...Z_n$	$\to Z_1$
	.....	
	$R_nQ_nZ_1...Z_n$	$\to Z_n$
Iterator:	$J\underline{0}Z_1Z_2$	$\to Z_2$
	$J(SZ_0)Z_1Z_2$	$\to Z_1(JZ_0Z_1Z_2)$

Table 9.1

9.14. EXAMPLE. PCF, a programming language for computable functions (Plotkin [77]), is the following CRS (see Table 9.2). PCF is a left-normal CRS (cf. Definition 6.12.6), hence the Standardization Theorem holds for PCF, and hence the Normalization Theorem (stating that leftmost reduction is normalizing.) PCF is an extension of simply typed lambda calculus.

Types:

(i) INT, BOOL are types (ground types)

(ii) if σ, τ are types, then $(\sigma \to \tau)$ is a type

Constants:

<u>true</u>	:	BOOL
<u>false</u>	:	BOOL
$\underline{\text{cond}}^{INT}$	:	BOOL → (INT → (INT → INT))
$\underline{\text{cond}}^{BOOL}$	:	BOOL → (BOOL → (BOOL → BOOL))
Y^σ	:	$(\sigma \to \sigma) \to \sigma$
<u>n</u> (for each $n \in \mathbb{N}$)	:	INT
<u>succ</u>	:	INT → INT
<u>pred</u>	:	INT → INT
<u>zero</u>	:	BOOL → INT

Variables: x_n^σ ($n \in \mathbb{N}$) : σ

Terms:

(i) x_n^σ is a term

(ii) constants are terms

(iii) if t, s are terms of type $\sigma \to \tau$ and σ respectively, then (ts) is a term of type τ

(iv) if t is a term of type τ, then $\lambda x_n^\sigma.t$ is a term of type $\sigma \to \tau$

`

Reduction rules:

$\underline{\text{cond}}^{INT}\ \underline{\text{true}}\ Z_1\ Z_2 \to Z_1$

$\underline{\text{cond}}^{INT}\ \underline{\text{false}}\ Z_1\ Z_2 \to Z_2$

$\underline{\text{cond}}^{BOOL}\ \underline{\text{true}}\ Z_1\ Z_2 \to Z_1$

$\underline{\text{cond}}^{BOOL}\ \underline{\text{false}}\ Z_1\ Z_2 \to Z_2$

$Y^\sigma Z \to Z(Y^\sigma Z)$

$(\lambda x^{\sigma}.Z_1(x^{\sigma}))Z_2$	$\rightarrow Z_1(Z_2)$	
$\underline{\text{succ}}\ \underline{n}$	$\rightarrow \underline{n+1}$	$(n \in \mathbb{N})$
$\underline{\text{pred}}\ \underline{n+1}$	$\rightarrow \underline{n}$	$(n \in \mathbb{N})$
$\underline{\text{zero}}\ \underline{0}$	$\rightarrow \underline{\text{true}}$	
$\underline{\text{zero}}\ \underline{n+1}$	$\rightarrow \underline{\text{false}}$	$(n \in \mathbb{N})$

Table 9.2

9.15. EXAMPLE. Gödel's $\mathcal{T}$ (or: Gödel's functions of finite type, or: primitive recursive functionals of finite type) is a CRS playing an important role in Proof Theory (see Hindley, Lercher & Seldin [72], p.127 and Barendregt [84], p.568.) It is the following extension of typed λ-calculus (see Table 9.3).

Types:	(i)	INT is a type
	(ii)	if σ, τ are types, then $(\sigma \rightarrow \tau)$ is a type
Constants:	$\underline{0}$	: INT
	$\underline{\text{succ}}$	: INT $\rightarrow$ INT
	R^{σ}	: $\sigma \rightarrow ((\sigma \rightarrow (\text{INT} \rightarrow \sigma)) \rightarrow (\text{INT} \rightarrow \sigma))$
Variables:	x_n^{σ}	: σ $\quad (n \in \mathbb{N})$
Terms:	(i)	x_n^{σ} is a term
	(ii)	constants are terms
	(iii)	if t, s are terms of type $\sigma \rightarrow \tau$ and σ respectively, then (ts) is a term of type τ
	(iv)	if t is a term of type τ, then λx_n^{σ}. t is a term of type $\sigma \rightarrow \tau$
Reduction rules:	$R^{\sigma}\ Z_1\ Z_2\ \underline{0}$	$\rightarrow Z_1$
	$R^{\sigma}\ Z_1\ Z_2\ (\underline{\text{succ}}\ Z_3)$	$\rightarrow Z_2\ (R^{\sigma}\ Z_1\ Z_2\ Z_3)\ Z_3$

$$(\lambda x^{\sigma}.\ Z_1(x^{\sigma}))\ Z_2 \quad \rightarrow Z_1(Z_2)$$

Table 9.3

(i) Gödels's $\mathcal{T}$ is an orthogonal CRS, hence confluent.

(ii) Note that it is not left-normal. But it is easy to formulate a variant of Gödel's $\mathcal{T}$ which is left-normal (by changing the order of the arguments of the R^{σ}), so for this variant the Standardization and Normalization Theorem hold.

References

ACZEL, P. (1978). *A general Church-Rosser theorem*. Preprint, Univ. of Manchester.

ARTIN, E.(1926). *Theorie der Zöpfe*. Abh. math. Semin. Hamburg Univ. 4 (1926), 47-72.

ARTIN, E.(1947). *Theory of braids*. Ann. of Math. (2) 48 (1947) 101-126

ARTIN, E.(1947a). *Braids and permutations*. Ann. of Math. vol. 48 (1947), 643-649

BACHMAIR, L. & DERSHOWITZ, N. (1986). *Commutation, transformation, and termination*. Proc. of 8th International Conference on Automated Deduction (ed. J.H. Siekmann), Oxford, Springer LNCS 230, 5-20.

BACHMAIR, L., DERSHOWITZ, N. & HSIANG, J. (1986). *Orderings for equational proofs*. In: Proc. of the IEEE Symp. on Logic in Computer Science, Cambridge, Massachusetts, 346-357.

BACHMAIR, L. & PLAISTED, D.A. (1985). *Associative path orderings*. In: Proc. of 1st Intern. Conf. on Rewriting Techniques and Applications (ed. J.-P. Jouannaud), Dijon, Springer LNCS 202, 241-254.

BARENDREGT, H.P. (1984). *The Lambda Calculus, its Syntax and Semantics*, 2nd ed. North-Holland 1984.

BEZEM, M., KLOP, J.W. & VAN OOSTROM, V., (1996) Diagram Techniques for Confluence. Information and Computation, Vol.141, No.2, p.172-204, 1998.

BIRKHOFF, G. (1935). *On the structure of abstract algebras*. In: Proc. of the Cambridge Philosophical Society 31, 433-454.

DE BRUIJN, N.G. (1978). *A note on weak diamond properties*. Memorandum 78-08, Eindhoven University of Technology, August 1978.

DAUCHET, M., TISON, S., HEUILLARD, T. & LESCANNE, P. (1987). *Decidability of the confluence of ground term rewriting systems*. In: Proc. of the 2nd Symp. on Logic in Computer Science, Ithaca, NY, 1987, 353-359.

DERSHOWITZ, N. (1979). *A note on simplification orderings*. Information Processing Letters, Vol.9, No.5, 1979, 212-215.

DERSHOWITZ, N. (1987). *Termination of rewriting*. J. of Symbolic Computation 3 (1&2), 69-115, 1987. Corrigendum: Vol.4, No.3, 409-410.

DERSHOWITZ, N. & JOUANNAUD, J.-P. (1990). *Rewrite systems*. In: Formal models and semantics, Handbook of Theoretical Computer Science, Vol.B (J. van Leeuwen, editor), Elsevier - The MIT Press, Chapter 6, p.243-320.

[DKP89] N. Dershowitz, S. Kaplan, and D.A. Plaisted. Infinite normal forms. In ICALP '89, volume 372 of Springer LNCS, p. 249-262, 1989.

DROSTEN, K. (1989). *Termersetzungssysteme*. Informatik-Fachberichte 210, Springer-Verlag.

[FW89] W.M. Farmer and R.J. Watro. Redex capturing in term graph rewriting. Technical report, M89-59, MITRE, 1989.

GALLIER, J.H. (1987). *What's so special about KruskalÆs Theorem and the ordinal Γ_0*. Techn. Rep. MS-CIS-87-27, Univ. of Pennsylvania, Philadelphia, PA, 1987.

GARSIDE, F.A. (1969). *The braid group and other groups*. Quart. J. Math. 20 (1969) 235-254.

GIRARD, J.-Y. (1987). *Proof theory and logical complexity*. Vol.I, Bibliopolis, Napoli, 1987.

HINDLEY, J.R. (1964). *The Church-Rosser property and a result in combinatory logic*. Ph.D. Thesis, Univ. Newcastle-upon-Tyne, 1964.

[Hud88] P. Hudak. Report on the functional programming language Haskell. Draft Proposed Standard, 1988.

HUET, G. (1980). *Confluent reductions: Abstract properties and applications to term rewriting systems*. JACM, Vol.27, No.4 (1980), 797-821.

HUET, G. & LÉVY, J.-J. (1979). *Call-by-need computations in non-ambiguous linear term rewriting systems*. Rapport INRIA nr.359.

HUET, G. & OPPEN, D.C. (1980). *Equations and rewrite rules: A survey*. In: Formal Language Theory: Perspectives and Open Problems (ed. R. Book), Academic Press, 1980, 349-405.

JANTZEN, M. (1988). *Confluent string rewriting*. EATCS (European Association for Theoretical ComputerScience) Monographs on Theor. Comp. Sci., No.14, Springer Verlag, 1988.

KAMIN, S. & LÉVY, J.-J. (1980). *Two generalizations of the recursive path ordering*. Unpublished manuscript, University of Illinois.

NEWMAN, M.H.A. (1942). *On theories with a combinatorial definition of* "*equivalence*". Annals of Mathematics, 43(2):223-243, April 1942.

KAHRS, S. (1991). *λ-rewriting*. Ph.D. Thesis, Universität Bremen.

KAHRS, S. (1992b). *Compilation of Combinatory Reduction Systems*. Preprint, University of Edinburgh.

KENNAWAY, J.R. (1989). *Sequential evaluation strategies for parallel-or and related systems*. Annals of Pure and Applied Logic, 43: 31-56.

KENNAWAY, J.R., KLOP, J.W., SLEEP, M.R. & DE VRIES, F.J. (1993). *An infinitary Church-Rosser property for non-collapsing orthogonal term rewriting systems*. In M.R. Sleep, M.J. Plasmeijer and M.C.J.D. van Eekelen, editors, Term -Graph Rewriting - theory and practice, p.47-59, Wiley, 1993.

KENNAWAY, J.R., KLOP, J.W., SLEEP, M.R. & DE VRIES, F.J. (1995). *Transfinite reductions in orthogonal term rewriting systems*. Information and Computation, 119(1):18-38, 1995.

KHASIDASHVILI, Z. (1990). *Expression reduction systems*. In: Proc. of I. Vekua Institute of Applied Mathematics, University of Tiblisi, Georgia, 1990.

KHASIDASHVILI, Z. (1992). *Church-Rosser Theorem in Orthogonal Combinatory Reduction Systems*. Preprint, INRIA Rocquencourt, 1992.

KNUTH, D.E. & BENDIX, P.B. (1970). *Simple word problems in universal algebras*. In: Computational Problems in Abstract Algebra (ed. J. Leech), Pergamon Press, 1970, 263-297.

KLOP, J.W. (1992). *Term rewriting systems*. In Vol.2 of Handbook of Logic in Computer Science (eds. S. Abramsky, D. Gabbay & T. Maibaum), Oxford University Press 1992, p.1-116

KLOP, J.W. (1980). *Combinatory Reduction Systems*. Mathematical Centre Tracts Nr.127, CWI, Amsterdam.

KLOP, J.W., VAN OOSTROM, V. & VAN RAAMSDONK, F. (1993). C*ombinatory reduction systems, introduction and survey*. Theoretical Computer Science, 121:279-308,1993.

KLOP, J.W., DE VRIJER, R.C. & VAN OOSTROM, V., (1999) A Geometric Proof of Confluence by Decreasing Diagrams Tech. Report Vrije Universiteit Amsterdam, 1999 (www.cs.vu.nl/~rdv/recent publications)

KRUSKAL, J.B. (1960). *Well-Quasi-Ordering, the Tree Theorem, and Vazsonyi's Conjecture*. Transactions of the AMS 95, 210-225.

KRUSKAL, J.B. *The theory of well-quasi-ordering: A frequently discovered concept*. J. Combinatorial Theory A13 (1972), 297-305.

MIDDELDORP, A. (1990). *Modular properties of term rewriting systems*. Ph.-D. Thesis, Vrije Universiteit Amsterdam, 1990.

MIDDELDORP, A. (1989). *Modular aspects of properties of term rewriting systems related to normal forms*. In: Proc. of 3rd Intern. Conf. on Rewriting Techniques and Applications, Chapel Hill, Springer LNCS 355, 263-277.

MIDDELDORP, A. (1989). *A sufficient condition for the termination of the direct sum of term rewriting systems*. In: Proc. of 4th IEEE Symposium on Logic in Computer Science, Pacific Grove, 396-401.

NASH-WILLIAMS, C.St.J.A. (1963). *On well-quasi-ordering finite trees*. In: Proc. of the Cambridge Philosophical Society 59 (4), 833-835.

NEDERPELT, R.P. (1973). *Strong normalization for a typed lambda calculus with lambda structured types*. Ph.D. Thesis, Eindhoven 1973.

NEWMAN, M.H.A. (1942). *On theories with a combinatorial definition of "equivalence"*. Annals of Math. 43, Nr.2 (1942), 223-243.

NIPKOW, T. (1991). *Higher-order critical pairs*. In: Proc. of LICS 91 (Sixth Annual Symposium on Logic in Computer Science), p.342-349, Amsterdam 1991.

NIPKOW, T. & BAADER, F. (1998). Term Rewriting and all that. Cambridge University Press, 1998.
NIPKOW, T. (1992). *Orthogonal Higher-Order Rewrite Systems are Confluent*. Preprint, TU München.

O'DONNELL, M.J. (1977). *Computing in systems described by equations*. Springer LNCS 58.

VAN OOSTROM, V. (1994a). *Confluence by decreasing diagrams*. Theoretical Computer Science 126. p.259-280.

VAN OOSTROM, V. (1994). *Confluence for Abstract and Higher-Order Rewriting*. Ph.-D. thesis, Vrije Universiteit, Amsterdam, March 1994.

VAN OOSTROM, V. & VAN RAAMSDONK, F. (1994). WEak orthogonality implies confluence: The higher-order case. In A. Nerode and Yu. V. Matiyasevich, editors, Proc. third Symp. on Logical Foundations of Computer Science, Springer LNCS 813, p. 379-392, 1994.

OYAMAGUCHI, M. (1987). *The Church-Rosser property for ground term rewriting systems is decidable*. Theoretical Computer Science 49 (1), 1987.

PLAISTED, D.A. (1978). *A recursively defined ordering for proving termination of term rewriting systems*. Report R-78-943, Univ. of Illinois, Urbana, IL.

PLOTKIN, G.D. (1977). *LCF as a programming language*. TCS 5, 223-257.

PvE93] PLASMEIJER, M.J. & VAN EEKELEN, M.C.J.D. (1993). Functional Programming and Parallel Graph Rewriting. Addison Wesley, 1993.

PRAWITZ, D. (1971). *Ideas and results in proof theory*. In: Proc. 2nd Scandinavian Logic Symposium (ed. J.E. Fenstad), North-Holland, 235-307.

PUEL, L. (1986). *Using unavoidable sets of trees to generalize Kruskal's theorem*. Rapport de Récherche du LIENS (Laboratoire d'Informatique de l'Ecole Normale Supérieure), Paris. (To appear: J. of Symbolic Computation, 1989.)

VAN RAAMSDONK, F. (1999). Higher-Order Rewriting. In: Proceedings RTA 99, Springer LNCS 1631, p.220-239.

ROSEN, B.K. (1973). *Tree-manipulating systems and Church-Rosser theorems*. JACM, Vol.20 (1973), 160-187.

RUSINOWITCH, M. (1987). *On termination of the direct sum of term rewriting systems*. Information Processing Letters 26, 65-70.

RUSINOWITCH, M. (1987a). *Path of Subterms Ordering and Recursive Decomposition Ordering Revisited*. J. Symbolic Computation (1987) 3, 117-131.

SHOENFIELD, J.R. (1967). *Mathematical Logic*. Addison-Wesley 1967.

SCHMIDT, G. & STRÖHLEIN, T. (1991). *Relations and Graphs - Discrete Mathematics for Computer Scientists*. Springer-Verlag, EATCS Monographs on Theoretical Computer Science, 1991

SMORYNSKI, C. (1982). *The variety of arboreal experience*. The Mathematical Intelligencer, Vol.4, No.4, 182-189, 1982.

STAPLES, J. (1975). *Church-Rosser theorems for replacement systems*. In: Algebra and Logic (ed. J. Crosley), Springer Lecture Notes in Mathematics 450, 291-307.

TOYAMA, Y. (1987). *Counterexamples to termination for the direct sum of Term Rewriting Systems*. Information Processing Letters 25, 141-143.

TOYAMA, Y. (1987). *On the Church-Rosser property for the direct sum of term rewriting systems*. JACM, Vol.34, No.1, 1987, 128-143.

TOYAMA, Y., KLOP, J.W. & BARENDREGT, H.P. (1989a). *Termination for the direct sum of left-linear term rewriting systems*. In: Proc. of 3rd Intern. Conf. on Rewriting Techniques and Applications, Chapel Hill, Springer LNCS 355, 477-491.

TURNER, D.A. (1979). *A new implementation technique for applicative languages*. Software Practice and Experience, Vol.9, 1979, 31-49.

TURNER, D.A. (1985). Miranda: a non-strict functional language with polymorphic types. In J.-P. Jouannaud, editor, Proceedings of the ACM Conference on Functional Programming Languages and Computer Architecture, volume 201 of Springer LNCS , p.1-16.

ZANTEMA, H. (1994). Termination of term rewriting: Interpretation and type elimination. J. of Symbolic Computation, 17:23-50, 1994.