Enno Ohlebusch

Advanced Topics in Term Rewriting

With 43 Figures

Abstract Reduction Systems

We will introduce term rewriting by first abstracting from the term structure. In other words, to start, we will concentrate on the so-called abstract reduction systems (ARSs).

ARSs were first studied by Newman [New42]. The approach to term rewriting via ARSs was used in Rosen's and Huet's seminal papers [Ros73, Hue80] and has been propagated by Klop [Klo92], for example.

The abstract approach has two main advantages. First, it is instructive to see which definitions and properties depend on the term structure and which are more fundamental. Second, abstract rewriting comprises several kinds of rewriting including term, string, graph, and conditional rewriting. Thus a repetition of similar definitions and concepts can be avoided by stating them once and only on an abstract level.

A reduction step in an ARS models a step in the transformation of some object (for example, a term, string, or graph). Abstract reduction may also be viewed as a stepwise execution of some computation. Given an object, we are interested in a final result of the computation. If an ARS *terminates*, then *every* computation ends and one gets a result in finite time, no matter how the computation proceeded. Moreover, this result will be unique if the system is *confluent*. Confluence and termination, and variants thereof, will be studied extensively in this book.

Most of the material in Sections 2.1–2.3 is folklore. We start with a rather general definition of ARSs.

Definition 2.0.1 An *abstract reduction system* $\mathcal{A} = (A, \{\to_\alpha\}_{\alpha\in I}, \vdash)$ is a triple consisting of a set of objects A, a set of binary relations $\to_\alpha$ on A

(where I is an index set), and a symmetric relation $\vdash\!\dashv$ on A. A relation $\to_\alpha$ is said to be a *reduction* relation *labeled* by α. The reduction relation of $\mathcal{A}$ is defined by $\to_{\mathcal{A}} = \bigcup_{\alpha\in I} \to_\alpha$. Whenever the ARS $\mathcal{A}$ can be inferred from the context, it will be suppressed in the notation $\to_{\mathcal{A}}$, i.e., we will write $\to$ instead of $\to_{\mathcal{A}}$.

First, we only consider ARSs for which the symmetric relation is the identity on A. These ARSs will simply be denoted by $(A, \{\to_\alpha\}_{\alpha\in I})$ in the following. Furthermore, if there is only one reduction relation on A, then we write $(A, \to)$ instead of $(A, \{\to\})$. Later, we will see that $(A, \{\to_\alpha\}_{\alpha\in I})$ and $(A, \{\to_\alpha\}_{\alpha\in I}, \vdash\!\dashv)$ can be viewed as useful presentations of $(A, \to)$ and $(A, \to, \vdash\!\dashv)$, respectively (useful in the sense that they allow us to establish confluence of the system). In Sections 2.5 and 2.6 we will study abstract reduction modulo an equivalence relation generated by the symmetric relation $\vdash\!\dashv$.

Definition 2.0.2 Let $\mathcal{A} = (A, \to)$ be an ARS.

1. We write $a \to b$ for $(a, b) \in \to$ and we say that a *reduces to* b *in one step* and that b is a *one-step reduct* of a.
2. The relation $\to^0$ denotes the identity on A, i.e., $\to^0 = \{(a, a) \mid a \in A\}$.
3. If $\ell \in \mathbb{N}$, then $\to^{\ell+1}$ denotes the $(\ell + 1)$-fold composition of $\to$, i.e., $\to^{\ell+1} = \to^\ell \cdot \to$.
4. Let $\ell \in \mathbb{N}$. If $a \to^\ell b$, then there is a *reduction sequence* (or *derivation*) from a to b of *length* ℓ, i.e., a sequence $a = a_0 \to a_1 \to \cdots \to a_\ell = b$ consisting of ℓ reduction steps.
5. The relation $\to^{\leq\ell}$ is defined by $\to^{\leq\ell} = \bigcup_{i=0}^{\ell} \to^i$. Thus, $a \to^{\leq\ell} b$ denotes a reduction sequence of length $\leq \ell$.
6. The reflexive closure of $\to$ is the relation $\to^{\leq 1}$. In the following, it will also be denoted by $\to^=$.
7. The transitive closure of $\to$ is $\to^+ = \bigcup_{\ell\in\mathbb{N}_+} \to^\ell$.
8. The reflexive transitive closure of $\to$ is $\to^* = \to^+ \cup \to^0$. If $a \to^* b$, we say that a *reduces* or *rewrites* to b and we call b a *reduct* of a.
9. The inverse of $\to$ is $\leftarrow = \{(a, b) \mid b \to a\}$. In other words, we write $a \leftarrow b$ if $b \to a$ and analogously $a \,{}^*\!\leftarrow b$ if $b \to^* a$.
10. The symmetric closure of $\to$ is $\leftrightarrow = \to \cup \leftarrow$. The reflexive transitive closure of $\leftrightarrow$ is called *conversion* or *convertibility* and is denoted by $\leftrightarrow^*$.
11. Two elements $a, b \in A$ are *joinable*, written $a \downarrow b$, if there is a $c \in A$ such that $a \to^* c \,{}^*\!\leftarrow b$. We call c a *common reduct* of a and b.

Definition 4.2.1 Let $\mathcal{R}$ be a TRS.

1. Let $l_1 \to r_1$ and $l_2 \to r_2$ be (variable) renamed versions of rewrite rules of $\mathcal{R}$ such that they have no variables in common. Suppose $l_1 = C[t]$ with $t \notin \mathcal{V}$ such that t and l_2 are unifiable. Let σ be a most general unifier of t and l_2. We call $\langle C[r_2]\sigma, r_1\sigma \rangle$ a *critical pair* of $\mathcal{R}$. If the two rules are renamed versions of the same rewrite rule of $\mathcal{R}$, we do not consider the case $C[\,] = \Box$.
2. $CP(\mathcal{R})$ denotes the set of all critical pairs between rules of $\mathcal{R}$.
3. A critical pair $\langle s, t \rangle \in CP(\mathcal{R})$ is called *joinable* if s and t are joinable.
4. A critical pair $\langle s, t \rangle \in CP(\mathcal{R})$ is called *trivial* if $s = t$.

Example 4.2.2 For

$$\mathcal{R} = \left\{ \begin{array}{rcl} b(w(x)) & \to & w(w(w(b(x)))) \\ w(b(x)) & \to & b(x) \\ b(b(x)) & \to & w(w(w(w(x)))) \\ w(w(x)) & \to & w(x) \end{array} \right.$$

the set $CP(\mathcal{R})$ consists of the critical pairs

$$\begin{array}{r} \langle b(b(x)),\ w(w(w(b(b(x))))) \rangle \\ \langle b(w(x)),\ w(w(w(b(w(x))))) \rangle \\ \langle w(w(w(w(b(x))))),\ b(w(x)) \rangle \\ \langle w(w(w(w(w(x))))),\ b(b(x)) \rangle \\ \langle b(w(w(w(b(x))))),\ w(w(w(w(w(x))))) \rangle \\ \langle b(w(w(w(w(x))))),\ w(w(w(w(b(x))))) \rangle \\ \langle w(b(x)),\ w(b(x)) \rangle \\ \langle w(w(x)),\ w(w(x)) \rangle \end{array}$$

4.3 Confluence of Nonterminating Systems

We have seen that critical pairs arise from "overlaps" between rewrite rules. Let us specify this notion.

Definition 4.3.3 A term s *overlaps* a term t if s is unifiable with a nonvariable subterm of t. s and t are called *nonoverlapping* if neither s overlaps t nor t overlaps s. A term s *overlays* a term t if s and t are unifiable. We say that a rewrite rule $l_2 \to r_2$ *overlaps* (*overlays*) a rewrite rule $l_1 \to r_1$ if there are renamed versions $l_2' \to r_2'$ and $l_1' \to r_1'$ of $l_2 \to r_2$ and $l_1 \to r_1$, respectively, which have no variables in common and satisfy l_2' overlaps (overlays) l_1'. Again, we do not consider the trivial case in which a rewrite rule overlaps itself at the root (overlays itself, that is).

Definition 4.3.4 Let $\mathcal{R}$ be a TRS.

1. $\mathcal{R}$ is *nonoverlapping* (or nonambiguous) if $CP(\mathcal{R}) = \emptyset$, i.e., there is no overlap between rewrite rules of $\mathcal{R}$.

2. $\mathcal{R}$ is an *overlay* system if every critical pair of $CP(\mathcal{R})$ originates from an overlay of rewrite rules of $\mathcal{R}$.

3. $\mathcal{R}$ is *orthogonal* if it is left-linear and nonoverlapping.

4. $\mathcal{R}$ is *almost orthogonal* if it is a left-linear overlay system in whicl every critical pair is trivial.

5. $\mathcal{R}$ is *weakly orthogonal* if it is left-linear and every critical pair i trivial.

Clearly, every orthogonal TRS is almost orthogonal and every almos orthogonal TRS is weakly orthogonal. Furthermore, note that constructo systems are special overlay systems.

Example 4.3.5 A standard example of an orthogonal TRS is combinator logic (CL), which is based on the constants S, K, and I with the followin rewrite rules:

$$\begin{array}{ll} Ap(Ap(Ap(S,x),y),z) & \to Ap(Ap(x,z),Ap(y,z)) \\ Ap(Ap(K,x),y) & \to x \\ Ap(I,x) & \to x \end{array}$$

The binary operator Ap is called the *application* operator. Because there i only this binary operator, the following notational conventions are commor

- infix notation $s \cdot t$ is used instead of $Ap(s,t)$,
- then the dot is suppressed, and
- brackets are deleted in the convention of association to the left.

If one adopts these conventions, then the three rules of CL become

$$\begin{array}{ll} Sxyz & \to xz(yz) \\ Kxy & \to x \\ Ix & \to x \end{array}$$

CL was developed by Schönfinkel [Sch24] and rediscovered by Curry an Feys [CF58]. CL has "universal computational power": every (partial) re cursive function on the natural numbers can be expressed in CL. This is on of the reasons variants of CL play an important role in the implementatio of functional programming languages such as Haskell, ML, and Miranda[1] see Turner [Tur79] and Peyton Jones [PJ87].

Our next goal is to show that every orthogonal TRS is confluent. As matter of fact, we will show a little more than that.

Definition 4.3.6 Let $A : s \to_{p,l\to r} t$ be a rewrite step in a TRS $\mathcal{R}$ and l $q \in \mathcal{P}os(s)$. The set $q\backslash A$ of *descendants* of q in t is defined by:

$$q\backslash A = \begin{cases} \{q\} & \text{if } q < p \text{ or } q \parallel p, \\ \{p.p_3.p_2 \mid r|_{p_3} = l|_{p_1}\} & \text{if } q = p.p_1.p_2 \text{ with } p_1 \in \mathcal{VP}os(l) \\ \emptyset & \text{otherwise} \end{cases}$$

[1] Miranda is a trademark of Research Software Ltd.

If $Q \subseteq \mathcal{P}os(s)$, then $Q\backslash A$ denotes the set $\bigcup_{q \in Q} q\backslash A$.

Let $q \in \mathcal{P}os(s)$ such that $s|_q$ is a redex. Then every $t|_{q'}$ with $q' \in q\backslash A$ is called a *descendant* of $s|_q$. The notion of descendant is extended to rewrite sequences in the obvious way.

We will use CL to illustrate the notion of descendant. The following reduction sequence shows that CL is not terminating:

$$\begin{aligned} & Ap(Ap(Ap(S,I),I), \underline{Ap}(Ap(S,I),I)) \\ \rightarrow\ & Ap(Ap(I, \underline{Ap}(Ap(S,I),I)), Ap(I, \underline{Ap}(Ap(S,I),I))) \\ \rightarrow\ & Ap(\underline{Ap}(Ap(S,I),I), Ap(I, \underline{Ap}(Ap(S,I),I))) \\ \rightarrow\ & Ap(\underline{Ap}(Ap(S,I),I), \underline{Ap}(Ap(S,I),I)) \end{aligned}$$

In the starting term, one redex is marked by underlining its root symbol. The descendants of this marked redex can easily be traced in the reduction sequence by simply searching for the marked symbols. In the preceding reduction sequence, every descendant of the marked redex is again a redex. This is no coincidence, as the following lemma shows.

Lemma 4.3.7 *In an orthogonal TRS $\mathcal{R}$, every descendant of a redex is again a redex.*

Proof We will show the lemma by tracing the descendants of a redex in a reduction step as in the preceding example. Let $s \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ and $q \in \mathcal{P}os(s)$ such that $v = s|_q$ is a redex, i.e., $v = l'\sigma'$ for some rule $l' \rightarrow r'$ in $\mathcal{R}$. Mark the root symbol of v by underlining it. So if $u = f(\ldots)$, then it is marked as $\underline{f}(\ldots)$. What happens to this redex in a rewrite step $s \rightarrow_{p,l \rightarrow r} t$ in which the redex $u = s|_p = l\sigma$ is contracted? We distinguish between the following cases:

Case $q < p$: If u is a proper subterm of v, then $root(t|_q)$ is underlined and $t|_q$ is the descendant of v. Because $\mathcal{R}$ is nonoverlapping, u must be a subterm of $x\sigma'$ for some $x \in \mathcal{V}ar(l')$. It follows that $t|_q$ is still a redex in t because the rule $l' \rightarrow r'$ is left-linear. It should be stressed that $t|_q$ need not necessarily be a redex if $\mathcal{R}$ is not orthogonal.

Case $q \parallel p$: If u and v occur at independent positions, then $root(t|_q)$ is underlined and $t|_q$, the descendant of v, is unaltered in t.

Case $q > p$: If v is a proper subterm of u, then it must be a subterm of $x\sigma$ for some $x \in \mathcal{V}ar(l)$ (again this is a consequence of nonoverlappingness). Now if the variable x appears m times in r, for some $m \geq 0$, then the marked redex appears m times in t, these copies of v are the descendants of v in t. If $m = 0$, then v has no descendant in t (v has been erased). Note that the descendants of v in t occur at pairwise disjoint positions.

Case $q = p$: If u and v coincide, then the marked redex has disappeared: t does not contain an underlined symbol. In other words, v has no descendant in t. □

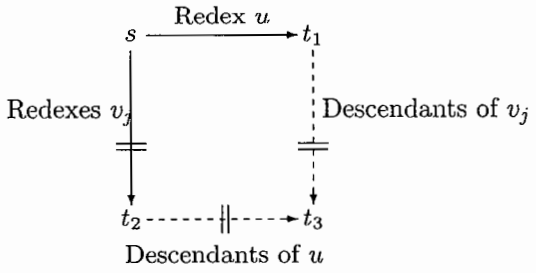

FIGURE 4.3. The parallel moves property.

The preceding lemma also holds for almost orthogonal systems but not for weakly orthogonal systems, as the following example shows. In the TRS $\{f(a) \to f(b), a \to b\}$, we have $\underline{f}(a) \to_{\mathcal{R}} \underline{f}(b)$ but $\underline{f}(b)$ is not a redex.

In order to show that every orthogonal TRS is confluent, we next introduce the parallel rewrite relation.

Definition 4.3.8 Let $\mathcal{R}$ be a TRS. We write $s \Vdash_{\mathcal{R}} t$ if t can be obtained from s by contracting a (possibly empty) set of pairwise disjoint redexes in s by $\mathcal{R}$. The relation $\Vdash$ is called the *parallel rewrite relation* w.r.t. $\mathcal{R}$. We sometimes also write $\Vdash$ instead of $\Vdash_{\mathcal{R}}$.

Lemma 4.3.9 *Let $\mathcal{R}$ be an orthogonal TRS. If $s \to_{\mathcal{R}} t_1$ by contracting redex u and $s \Vdash_{\mathcal{R}} t_2$ by contracting the pairwise disjoint redexes $v_1, \ldots, v_n$, then there is a term t_3 such that $t_1 \Vdash_{\mathcal{R}} t_3$ and $t_2 \Vdash_{\mathcal{R}} t_3$. Moreover, the redexes contracted in $t_1 \Vdash_{\mathcal{R}} t_3$ ($t_2 \Vdash_{\mathcal{R}} t_3$) are the descendants of $v_1, \ldots, v_n$ (u) in t_1 (t_2); see Figure 4.3.*

Proof Let $s|_p = u = l\sigma$ for some rewrite rule $l \to r \in \mathcal{R}$ and $s|_{q_j} = v_j$ for $1 \leq j \leq n$. It is relatively simple to prove the lemma by case analysis as in Lemma 4.3.7. The only interesting case is that in which u contains several terms out of $v_1, \ldots, v_n$ as proper subterms. For simplicity, let us assume that u contains $v_1, \ldots, v_k$, $1 \leq k \leq n$, as proper subterms. That is, $p < q_j$ for $1 \leq j \leq k$ and $p \parallel q_i$ for $k+1 \leq i \leq n$. Because $\mathcal{R}$ is nonoverlapping, every v_j must be a subterm of $x\sigma$ for some $x \in \mathcal{V}ar(l)$. It follows that $t_1|_p$ is still a redex in t_1 because the rule $l \to r$ is left-linear. The rest of the proof is left to the reader. □

Now we can prove the parallel moves lemma for orthogonal TRS.

Lemma 4.3.10 *Let $\mathcal{R}$ be an orthogonal TRS. If $s \Vdash_{\mathcal{R}} t_1$ and $s \Vdash_{\mathcal{R}} t_2$, then there is a term t_3 such that $t_1 \Vdash_{\mathcal{R}} t_3$ and $t_2 \Vdash_{\mathcal{R}} t_3$. Moreover, the redexes contracted in $t_1 \Vdash_{\mathcal{R}} t_3$ ($t_2 \Vdash_{\mathcal{R}} t_3$) are the descendants in t_1 (t_2) of the redexes contracted in $s \Vdash_{\mathcal{R}} t_2$ ($s \Vdash_{\mathcal{R}} t_1$); see Figure 4.4.*

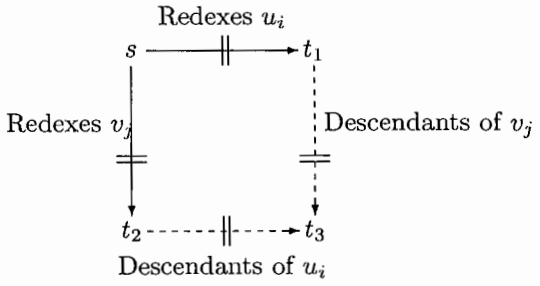

FIGURE 4.4. The parallel moves lemma.

Proof Let $u_1, \ldots, u_m$ be the redexes contracted in $s \Vdash_{\mathcal{R}} t_1$ and let $v_1, \ldots, v_n$ be the redexes contracted in $s \Vdash_{\mathcal{R}} t_2$. Let $s \to_{\mathcal{R}} \cdot \to_{\mathcal{R}} \ldots \to_{\mathcal{R}} \cdot \to_{\mathcal{R}} t_1$ be a reduction sequence in which the pairwise disjoint redexes are contracted in some order. According to Lemma 4.3.9, a common reduct t_3 of t_1 and t_2 can be obtained by contracting the descendants of $v_1, \ldots, v_n$ in t_1; see Figure 4.5, where $m = 3$. Moreover, the positions of the redexes contracted in the parallel reduction sequence $t_2 \Vdash_{\mathcal{R}} \cdot \Vdash_{\mathcal{R}} \ldots \Vdash_{\mathcal{R}} \cdot \Vdash_{\mathcal{R}} t_3$ are pairwise disjoint and coincide with the positions of the descendants of $u_1, \ldots, u_m$ in t_2. Therefore, $t_2 \Vdash_{\mathcal{R}} t_3$ by contracting the descendants of $u_1, \ldots, u_m$. □

Finally, we are in a position to prove that orthogonal systems are confluent; this was first shown by Rosen [Ros73] but "earlier proofs of the confluence of CL (combinatory logic) work just as well for orthogonal TRSs" as Klop [Klo92] remarks.

Theorem 4.3.11 *Every orthogonal term rewriting system is confluent.*

Proof For every TRS $\mathcal{R}$ we clearly have $\to_{\mathcal{R}} \subseteq \Vdash_{\mathcal{R}} \subseteq \to_{\mathcal{R}}^*$. Because $\Vdash_{\mathcal{R}}$ has the diamond property according to the parallel moves lemma 4.3.10, it follows from Lemma 2.4.2 that $\mathcal{R}$ is confluent. □

Theorem 4.3.11 is of utmost importance for inferring confluence of nonterminating TRSs. For example, it allows us to conclude that CL is confluent.

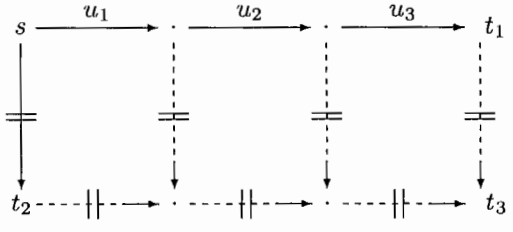

FIGURE 4.5. Proof of the parallel moves lemma.

Almost orthogonal TRSs also satisfy the parallel moves lemma. Moreover, weakly orthogonal TRSs satisfy the first part of the parallel moves lemma; cf. [Hue80] and [BN98, theorem 6.4.8]. Consequently, the preceding theorem remains valid if orthogonality is replaced with weak orthogonality.

In a confluent but nonterminating TRS some derivations starting from a term may lead to a normal form while others do not. Because we are usually interested in finding the normal form of a term, we need some *reduction strategy* that tells us what redex(es) we should contract to obtain that normal form. Such a reduction strategy is called *normalizing* if it finds the normal form whenever it exists. For orthogonal TRSs some "good" normalizing reduction strategies are known, and the notion of descendant plays a crucial role in this context. The reader is referred to Klop [Klo92, section 3.2] for details.