An Introduction to Knuth-Bendix Completion

Jan Willem Klop
Centre for Mathematics and Computer Science
Kruislaan 413, 1098 SJ Amsterdam
Department of Mathematics and Computer Science
Vrije Universiteit, de Boelelaan 1081, 1081 HV Amsterdam

Aart Middeldorp
Department of Mathematics and Computer Science
Vrije Universiteit, de Boelelaan 1081, 1081 HV Amsterdam

A self-contained introduction is given to the Knuth-Bendix completion method for equational specifications. After a short introduction to term rewriting systems and a presentation of some simple completion algorithms, we explain the recent abstract approach of Bachmair, Dershowitz and Hsiang to the correctness problem for such completion algorithms by means of proof orderings.

1. Introduction

Given the set of equational axioms $E = \{0 + x = x, (-x) + x = 0, (x + y) + z = x + (y + z)\}$ it is not an entirely trivial task to derive the equation $-0 = 0$. (See Example 3.5.) In a famous paper, D.E. Knuth and his student P. Bendix addressed the above question and similar ones by devising a technique attempting to find a set of oriented equations $\mathscr{R}$ or *term rewriting system* yielding the same equality as the one generated by E, and with the additional property that a sufficiently long computation of some expression t according to the oriented rules must lead to a unique 'normal form' of t. Such a term rewriting system $\mathscr{R}$ is called complete, and in fact it provides a positive solution for the *validity problem* or *uniform word problem* of the original specification E.

Nowadays, there are more powerful Knuth-Bendix completion techniques, such as the completion algorithm of Peterson & Stickel [14]). Also there are several applications other than deciding the validity problem; we mention 'inductionless induction' (see e.g. Huet & Hullot [9]), and 'computing with equations' (Dershowitz [5]). Furthermore, much recent activity concerns relating term rewriting completion techniques to the computational mechanism of resolution in logic programming (Dershowitz & Plaisted [6]), in an attempt to integrate functional programming and logic programming.

In this paper we will not present these other applications and developments, but instead focus on a very recent and elegant method of Bachmair, Dershowitz and Hsiang [2] to prove the *correctness* of large classes of 'Knuth-Bendix-like' completion algorithms. The method centers around the concept of

proof orderings, and has streamlined the previously tedious task of such correctness proofs considerably. The idea of proof orderings is as follows. A proof of an equation $t=s$ based on the equality axioms in E, is a sequence of elementary applications of the axioms: $t=t'=...=s$. Here the steps are without orientation. If a complete term rewriting system $\mathscr{R}$ for E is available (i.e. $\mathscr{R}$ generates the same equality as E does, but elementary steps in $\mathscr{R}$ now have a direction), then one can find a 'rewriting proof' or 'rewrite proof' $t \to t' \to ... \to r \leftarrow \leftarrow ... \leftarrow s$ having the form of two rewrite sequences of t,s leading to a common 'reduct' r. Such rewrite proofs are considered preferable to unoriented ones in E — because of the completeness of $\mathscr{R}$ it is decidable whether a rewrite proof between t,s exists. For, completeness says that every term has a unique normal form, to be reached eventually by repeated rewritings, and terms which can be proved equal must have the same normal form. Hence the decision algorithm is simple: compute the normal forms of t and s; if and only if they coincide does there exist a (rewrite) proof of $t=s$.

Actually, in the course of transforming E to a complete term rewrite system $\mathscr{R}$, one has to deal with intermediate situations where a part of E has been replaced by some rewrite rules; so we have a pair $(E',\mathscr{R}')$ where $E' \cup \mathscr{R}'$ generates the same equality as E does. In this situation an equality proof between t,s may be partially oriented, e.g.: $t \to t' = t'' \leftarrow ... = s' \to s$, a proof where each elementary step is either directed ($\to$ or $\leftarrow$) or undirected ($=$). Now proofs are ordered according to some complexity measure with the effect that equality proofs which have more orientation as in a rewrite proof, are less complex. The rewrite proofs have least complexity. This notion of 'proof ordering' plays a crucial role in proving the correctness of algorithms that are designed to transform a set of equations E to a complete term rewriting system via intermediate stages $(E',\mathscr{R}')$.

Our paper is self-contained. After a short introduction to equational specifications and term rewriting systems, we perform an informal completion of the axioms for group theory (the well-known example of a successful completion in Knuth & Bendix [11]), and present some completion algorithms. Finally, the 'abstract' approach via proof orderings is explained.

2. Equational specifications

Definition 2.1. An *equational specification* is a pair (Σ, E). The *signature* or *alphabet* Σ consists of a countably infinite set of *variables* $x_1, x_2, x_3, ...$, also denoted as $x, y, z, x', ...$, and a non-empty set of *function symbols* $F, G, ...$, each equipped with an 'arity', i.e. the number of 'arguments' it is supposed to have. Function symbols of arity 0 are called *constants*. E is a set of equations $s = t$ between terms s, t. The set of terms built from Σ, notation $T(\Sigma)$, is the smallest set such that $x \in T(\Sigma)$ for every variable $x \in \Sigma$, and if $t_1,...,t_n \in T(\Sigma)$ then $F(t_1,...,t_n) \in T(\Sigma)$, for $F \in \Sigma$ with arity n ($n \geqslant 0$). Terms not containing variables are called *ground* terms or *closed* terms.

DEFINITION 2.2. A *substitution* σ is a mapping from $T(\Sigma)$ to $T(\Sigma)$ satisfying $\sigma(F(t_1,...,t_n)) = F(\sigma(t_1),...,\sigma(t_n))$ for every n-ary function symbol F $(n \geqslant 0)$ and terms $t_1,...,t_n \in T(\Sigma)$. We write t^σ instead of $\sigma(t)$, henceforth.

EXAMPLE 2.3. An equational specification of groups is the pair (Σ_1, E_1) where Σ_1 contains the function symbols 0 (nullary), $-$ (unary), $+$ (binary) and E_1 consists of the following equations, using infix notation for $+$: $\{0 + x = x,\ (-x) + x = 0,\ (x + y) + z = x + (y + z)\}$.

DEFINITION 2.4. Let (Σ, E) be an equational specification. If an equation $s = t$ between terms $s, t \in T(\Sigma)$ is derivable from the equations E, we write $(\Sigma, E) \vdash s = t$ or $s =_E t$. *Derivability* is defined by means of the inference system of Table 1.

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

TABLE 1.

DEFINITION 2.5. Let Σ be a signature. A Σ-*algebra* $\mathscr{A}$ is a set A together with functions $F^{\mathscr{A}}: A^n \to A$ for every n-ary function symbol $F \in \Sigma$. (If F is a constant then $F^{\mathscr{A}} \in A$.)

EXAMPLE 2.6. Let (Σ_1, E_1) be the specification of Example 2.3. The set $A_1 = \{a, b, c\}$ with constant a and functions $-^{\mathscr{A}_1}$ and $+^{\mathscr{A}_1}$ defined by Table 2 is a Σ_1-algebra (denoted by $\mathscr{A}_1$).

$+^{\mathscr{A}_1}$	a	b	c	$-^{\mathscr{A}_1}$	
a	a	b	c	a	a
b	b	c	a	b	c
c	c	a	b	c	b

TABLE 2.

Let Σ be a signature and let $\mathscr{A}$ be a Σ-algebra. An equation $s = t$ between terms of $T(\Sigma)$ is assigned a meaning in $\mathscr{A}$ by interpreting the function symbols in s and t via the corresponding functions in $\mathscr{A}$. Variables in $s = t$ are (implicitly) universally quantified.

EXAMPLE 2.7. The meaning of the equation $(-x)+x=0$ in the Σ_1-algebra $\mathscr{A}_1$ of Example 2.6 is $\forall x \in A_1 \ +^{\mathscr{A}_1}(-^{\mathscr{A}_1}(x), x) = a$.

It is clear that $\forall x \in A_1 \ +^{\mathscr{A}_1}(-^{\mathscr{A}_1}(x), x) = a$ is a true statement. We say that $(-x)+x=0$ is *valid* in $\mathscr{A}_1$ or that $\mathscr{A}_1$ is a *model* of $(-x)+x=0$ and we write $\mathscr{A}_1 \models (-x)+x=0$.

DEFINITION 2.8. A Σ-algebra $\mathscr{A}$ is a *model* of a set of equations E between terms of $T(\Sigma)$, notation $\mathscr{A} \models E$, if every equation $s=t$ of E is valid in $\mathscr{A}$. The *variety* defined by an equational specification (Σ, E), notation $Alg(\Sigma, E)$, is the class of all Σ-algebras $\mathscr{A}$ such that $\mathscr{A} \models E$. Instead of $\forall \mathscr{A} \in Alg(\Sigma, E)\ \mathscr{A} \models F$, where F is a set of equations between terms of $T(\Sigma)$, we will write $(\Sigma, E) \models F$.

DEFINITION 2.9. Let (Σ, E) be an equational specification. The *validity problem* or *uniform word problem* for (Σ, E) is:

> *Given an equation $s=t$ between terms $s, t \in T(\Sigma)$, decide whether or not $(\Sigma, E) \models s=t$.*

The following theorem is the well-known completeness result of Birkhoff [4].

THEOREM 2.10. *Let (Σ, E) be an equational specification. For all terms $s, t \in T(\Sigma)$ we have $(\Sigma, E) \vdash s=t$ if and only if $(\Sigma, E) \models s=t$.*

A celebrated example of an equational specification with an unsolvable validity problem is Combinatory Logic, the specification in Table 3 with a binary operation 'application' $(\cdot)$ and constants S, K, I (see Barendregt [3]).

$((S\cdot x)\cdot y)\cdot z$	$=$	$(x\cdot z)\cdot(y\cdot z)$
$(K\cdot x)\cdot y$	$=$	x
$I\cdot x$	$=$	x

TABLE 3.

3. TERM REWRITING SYSTEMS

DEFINITION 3.1. A *term rewriting system* (TRS) is a pair $(\Sigma, \mathscr{R})$. Here Σ is a signature and $\mathscr{R}$ is a set of pairs (s,t) with $s, t \in T(\Sigma)$ subject to two constraints: (1) the left-hand side s is not a variable, (2) the variables which occur in the right-hand side t also occur in s. Pairs (s,t) will be called 'rewriting rules' and will henceforth be written as $s \to t$.

We usually write $\mathscr{R}$ instead of $(\Sigma, \mathscr{R})$ under the assumption that Σ does not contain function symbols which do not occur in the rewriting rules of $\mathscr{R}$. We will often give the rewriting rules a name, e.g. r and write $r : s \to t$ or $s \xrightarrow{r} t$.

DEFINITION 3.2. A *context* is a 'term' which contains a single occurrence of a special symbol $\square$. We denote contexts by $C[\ ], C_1[\ ], \ldots$. If $C[\ ]$ is a context and $t \in T(\Sigma)$ then $C[t] \in T(\Sigma)$ is the result of replacing the symbol $\square$ by t; t is said to be a *subterm* of $C[t]$, notation $t \subseteq C[t]$.

DEFINITION 3.3. The rewriting rules of a TRS $\mathscr{R}$ give rise to *reduction steps*, as follows: $s \to_{\mathscr{R}} t$ if and only if $s \equiv C[s_1^\sigma]$ and $t \equiv C[t_1^\sigma]$ for some context $C[\ ]$, substitution σ and rewriting rule $r : s_1 \to t_1$ of $\mathscr{R}$. (The symbol $\equiv$ denotes syntactic equality.) The subterm s_1^σ of s is called a *redex*, more precisely an *r-redex*.

If we want to specify the rewriting rules used in this reduction step, we write $s \xrightarrow{r}_{\mathscr{R}} t$. We usually omit the subscript $\mathscr{R}$.

DEFINITION 3.4. The transitive-reflexive closure of $\to_{\mathscr{R}}$ will be denoted by $\twoheadrightarrow_{\mathscr{R}}$. Further, $\to_{\mathscr{R}}^+$ denotes the transitive closure of $\to_{\mathscr{R}}$. If $s \twoheadrightarrow_{\mathscr{R}} t$ we say that s *reduces* to t. We write $s \leftarrow_{\mathscr{R}} t$ if $t \to_{\mathscr{R}} s$. The symmetric closure of $\to_{\mathscr{R}}$ is denoted by $\leftrightarrow_{\mathscr{R}}$ (so, $\leftrightarrow_{\mathscr{R}} = \to_{\mathscr{R}} \cup \leftarrow_{\mathscr{R}}$). The transitive-reflexive closure of $\leftrightarrow_{\mathscr{R}}$ is called *conversion*; it will be denoted by $=_{\mathscr{R}}$. When no confusion can arise, the subscript $\mathscr{R}$ will be dropped.

EXAMPLE 3.5. By orienting the equations in Example 2.3 from left to right we obtain the following set $\mathscr{R}$ of rewriting rules: $\{0 + x \to x,\ (-x) + x \to 0,\ (x + y) + z \to x + (y + z)\}$. So we have the reduction $(0 + (-0)) + 0 \twoheadrightarrow 0$, obtained from the sequence of reduction steps

$$\underline{(0 + (-0)) + 0} \to 0 + \underline{((-0) + 0)} \to \underline{0 + 0} \to 0.$$

In each step the underlined redex is rewritten. Note that although -0 does not reduce to 0, we do have $-0 = 0$, as shown by the following conversion, which may serve to illustrate that even in simple cases finding an equality proof can be quite complicated.

$$\begin{aligned}
-0 \leftarrow 0 + (-0) \ \leftarrow\ & ((-(-(-0))) + (-(-0))) + (-0) \\
\to\ & (-(-(-0))) + ((-(-0)) + (-0)) \\
\to\ & (-(-(-0))) + 0 \leftarrow (-(-(-0))) + (0 + 0) \\
\leftarrow\ & (-(-(-0))) + (((-(-0)) + (-0)) + 0) \\
\to\ & (-(-(-0))) + ((-(-0)) + ((-0) + 0)) \\
\to\ & (-(-(-0))) + ((-(-0)) + 0) \\
\leftarrow\ & ((-(-(-0))) + (-(-0))) + 0 \to 0 + 0 \to 0.
\end{aligned}$$

DEFINITION 3.6. Let $\mathscr{R}$ be a TRS. A term s is a *normal form* if there is no term t such that $s \to_{\mathscr{R}} t$. A term s *has a normal form* if there is a normal form t such that $s \twoheadrightarrow_{\mathscr{R}} t$. $\mathscr{R}$ is *weakly normalizing* (WN) if every term has a normal form. $\mathscr{R}$ is *strongly normalizing* (SN) if there are no infinite reduction sequences $t_0 \to t_1 \to t_2 \to \ldots$.

DEFINITION 3.7. Let $\mathscr{R}$ be a TRS. $\mathscr{R}$ is *weakly confluent* or *weakly Church-Rosser* (WCR) if for all terms s, t_1, t_2 with $s \to_{\mathscr{R}} t_1$ and $s \to_{\mathscr{R}} t_2$ we can find a term t_3 such that $t_1 \twoheadrightarrow_{\mathscr{R}} t_3$ and $t_2 \twoheadrightarrow_{\mathscr{R}} t_3$ (see Figure 4(a)). Such a term t_3 is called a *common reduct* of t_1 and t_2. $\mathscr{R}$ is *confluent* or has the *Church-Rosser*

property (CR) if for all terms s, t_1, t_2 with $s \twoheadrightarrow_{\mathscr{R}} t_1$ and $s \twoheadrightarrow_{\mathscr{R}} t_2$ we can find a term t_3 such that $t_1 \twoheadrightarrow_{\mathscr{R}} t_3$ and $t_2 \twoheadrightarrow_{\mathscr{R}} t_3$ (see Figure 4(b)).

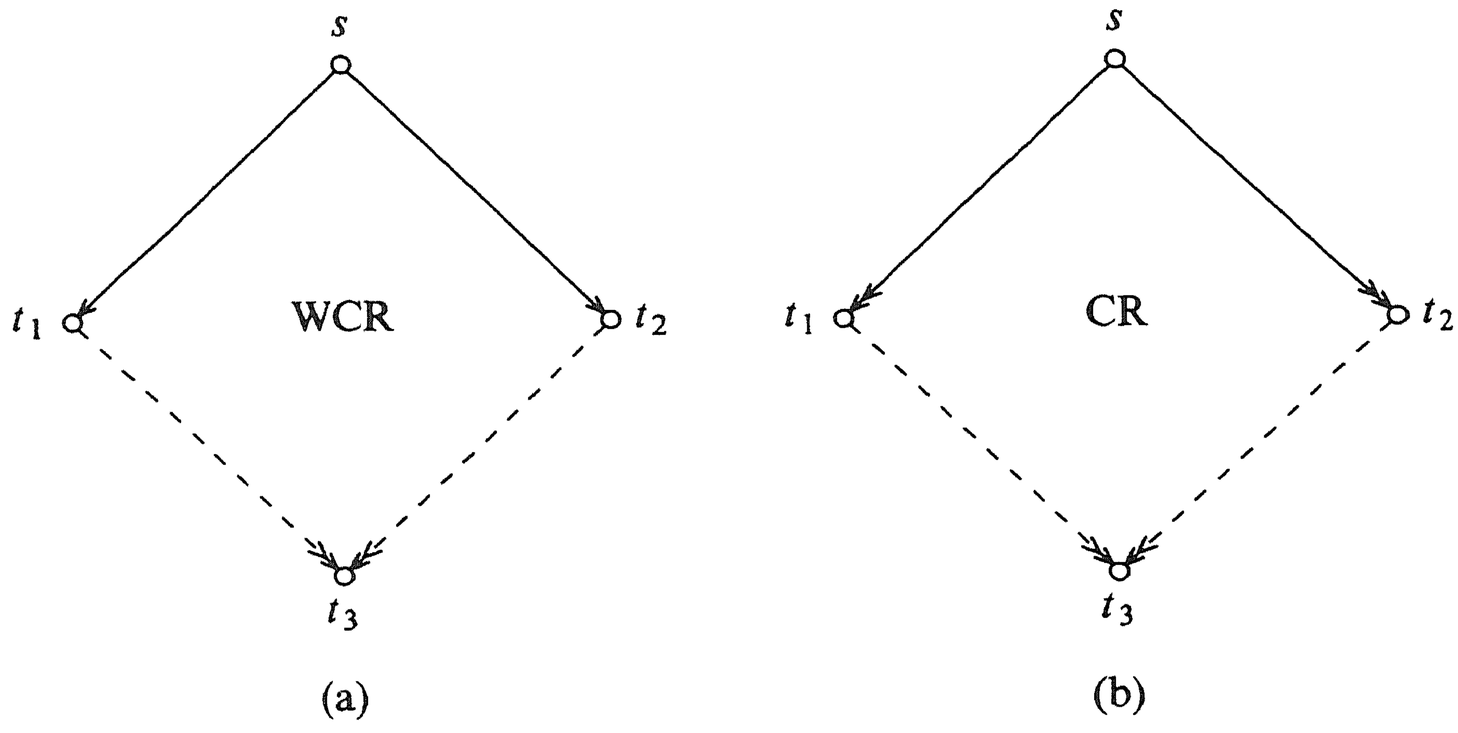


FIGURE 4.

It is a nice exercise, left to the reader, to find a TRS which is WCR but not CR. However, we do have the following extremely useful fact.

LEMMA 3.8 (NEWMAN [13]). *Let $\mathscr{R}$ be a* TRS. *If $\mathscr{R}$ is* WCR *and* SN *then $\mathscr{R}$ is* CR.

In Section 5 we give a proof of this lemma illustrating the developments there. Often one finds in the literature a different but equivalent definition of confluence, as suggested by the following proposition whose proof is easy.

PROPOSITION 3.9. *Let $\mathscr{R}$ be a* TRS. *$\mathscr{R}$ is* CR *if and only if for all terms t_1, t_2 with $t_1 =_{\mathscr{R}} t_2$ we can find a term t_3 such that $t_1 \twoheadrightarrow_{\mathscr{R}} t_3 \twoheadleftarrow_{\mathscr{R}} t_2$.*

EXAMPLE 3.10. The TRS of Example 3.5 is not CR: -0 and 0 are convertible ($-0 = 0$) but they have no common reduct because -0 and 0 are normal forms.

PROPOSITION 3.11. *Let $\mathscr{R}$ be a confluent* TRS. *Then $\mathscr{R}$ has unique normal forms, i.e. if $n_1 =_{\mathscr{R}} n_2$ and n_1, n_2 are normal forms then $n_1 \equiv n_2$.*

PROOF. Immediate from Proposition 3.9. □

DEFINITION 3.12. A TRS with the properties SN and CR is called *complete*.

4. KNUTH-BENDIX ALGORITHM: NAIVE APPROACH

We are interested in complete TRSs for the following reason. Let (Σ, E) be an equational specification. If we can find a complete TRS $\mathscr{R}$ with finitely many rewriting rules such that

$$s =_{\mathscr{R}} t \quad \Leftrightarrow \quad (\Sigma, E) \vdash s = t \qquad (\star)$$

for all terms $s, t \in T(\Sigma)$, then the validity problem for (Σ, E) is solvable. The algorithm is simple:

(1)	*Reduce s and t to their respective normal forms s′ and t′.*
(2)	*Compare s′ and t′: $s =_{\mathscr{R}} t$ if and only if $s' \equiv t'$.*

The Knuth-Bendix completion algorithm tries to construct a complete TRS $\mathscr{R}$ for a given equational specification (Σ, E) such that $(\star)$ holds. In this section we will explain the essential features of the completion algorithm first by an informal, 'intuition-guided' completion of the equational specification (Σ_1, E_1) of groups as in Example 2.3. First we give the equations a 'sensible' orientation:

$r_1: 0 + x \to x$, $r_2: (-x) + x \to 0$, $r_3: (x + y) + z \to x + (y + z)$.

(Note that the orientation in r_1, r_2 is forced, by the restrictions in Definition 3.1. As to the orientation of r_3, the other direction is just as 'sensible'.) However, these rules are not confluent: the r_2-redex $(-x) + x$ can be unified (after a renaming of variables) with a non-variable subterm of the r_3-redex $\underline{(x + y)} + z$ (the underlined subterm). The resulting term $((-x) + x) + z$ is subject to the two reductions:

$$\begin{array}{ccc} & ((-x) + x) + z & \\ {\scriptstyle r_2}\swarrow & & \searrow{\scriptstyle r_3} \\ 0 + z & & (-x) + (x + z) \end{array}$$

The pair of reducts $\langle 0 + z, (-x) + (x + z) \rangle$ 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.

DEFINITION 4.1. Let $\alpha \to \beta$ and $\gamma \to \delta$ be two rewriting rules such that α is *unifiable* (after variable renaming) with a non-variable subterm of γ. This means that there exists a context $C[\ ]$, a non-variable term t and a 'most general unifier' σ such that $\gamma \equiv C[t]$ and $t^\sigma \equiv \alpha^\sigma$. (For the concept of 'most general unifier, see the contribution of M. Bezem in this issue of the CWI Quarterly.) The term $\gamma^\sigma \equiv C[t]^\sigma$ can be reduced in two possible ways: $C[t]^\sigma \to C[\beta]^\sigma$ and $\gamma^\sigma \to \delta^\sigma$. 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 rewriting rule, we furthermore require that α is unifiable with a proper (i.e. $\not\equiv \alpha$) non-variable subterm of $\gamma \equiv \alpha$.

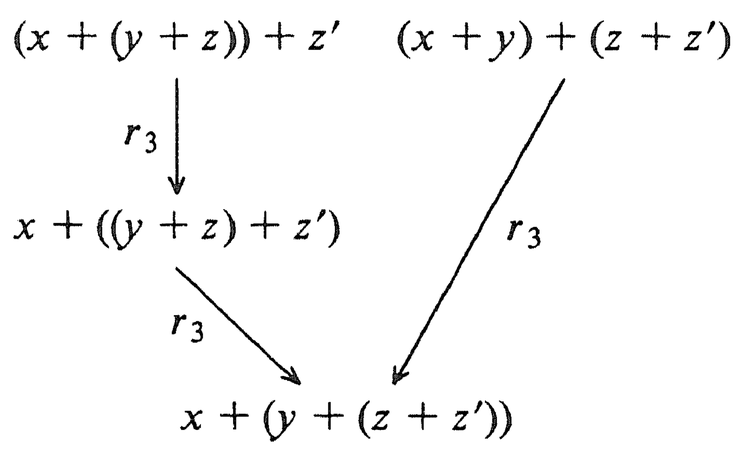
DEFINITION 4.2. A critical pair $\langle s, t \rangle$ is called *convergent* if s and t have a common reduct.

The critical pair $\langle 0 + z, (-x) + (x + z) \rangle$ is not convergent: $(-x) + (x + z)$

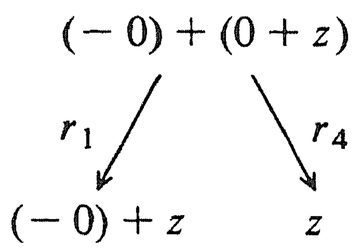
is a normal form and $0 + z$ reduces only to the normal form z. The rules r_1, r_2 and r_3 have the following critical pairs:

- $\langle x + z,\ 0 + (x + z)\rangle$ by superposition of r_1 on r_3,
- $\langle 0 + z,\ (-x) + (x + z)\rangle$ by superposition of r_2 on r_3,
- $\langle (x + (y + z)) + z',\ (x + y) + (z + z')\rangle$ by superposition of r_3 on r_3.

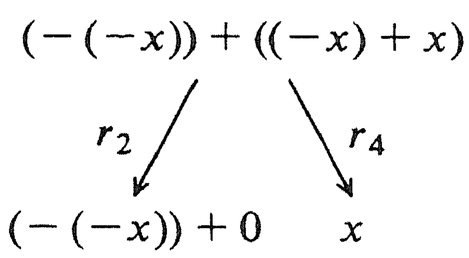
These are all the critical pairs caused by the rules r_1, r_2, r_3. They are found by searching for all possible 'overlaps' between the left-hand sides of the rules, where we are allowed to instantiate the variables. The pair $\langle x + z,\ 0 + (x + z)\rangle$ is convergent because $0 + (x + z) \to x + z$. The pair $\langle (x + (y + z)) + z',\ (x + y) + (z + z')\rangle$ is also convergent:



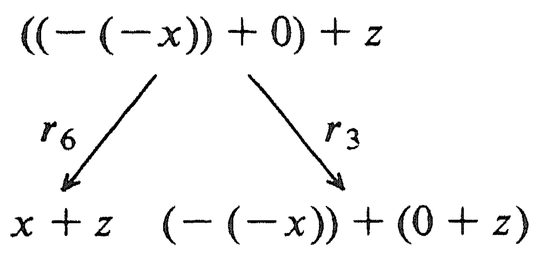
So only the pair $\langle 0 + z,\ (-x) + (x + z)\rangle$ is not yet convergent. By adopting the new rule r_4: $(-x) + (x + z) \to z$, the terms $0 + z$ and $(-x) + (x + z)$ have of course a common reduct. Note that the equality of z and $(-x) + (x + z)$ is derivable from E_1. The new TRS is still not confluent because the critical pair we get by superposing r_1 on r_4



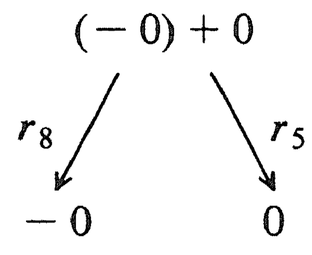
is not convergent. We add the rule r_5: $(-0) + z \to z$. We can superpose r_2 on r_4:



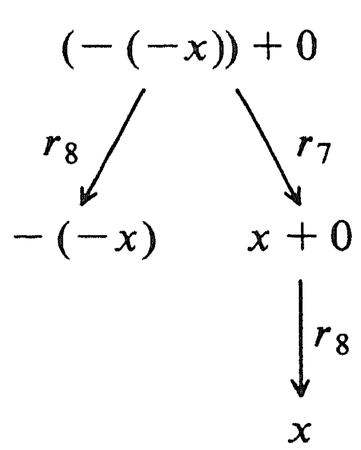
The resulting critical pair $\langle (-(-x)) + 0,\ x\rangle$ cannot be reduced. So we add the rule r_6: $(-(-x)) + 0 \to x$. Now r_6 can be superposed on r_3:



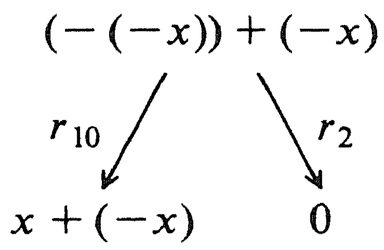
The term $(-(-x)) + (0 + z)$ reduces to $(-(-x)) + z$. To make the critical pair $\langle x + z, (-(-x)) + (0 + z)\rangle$ convergent, we add the rule $r_7: (-(-x)) + z \to x + z$. With this new rule r_6 can be simplified, because $(-(-x)) + 0 \xrightarrow{r_7} x + 0$. Therefore we replace r_6 by $r_8: x + 0 \to x$. Superposition of r_8 on r_5 yields



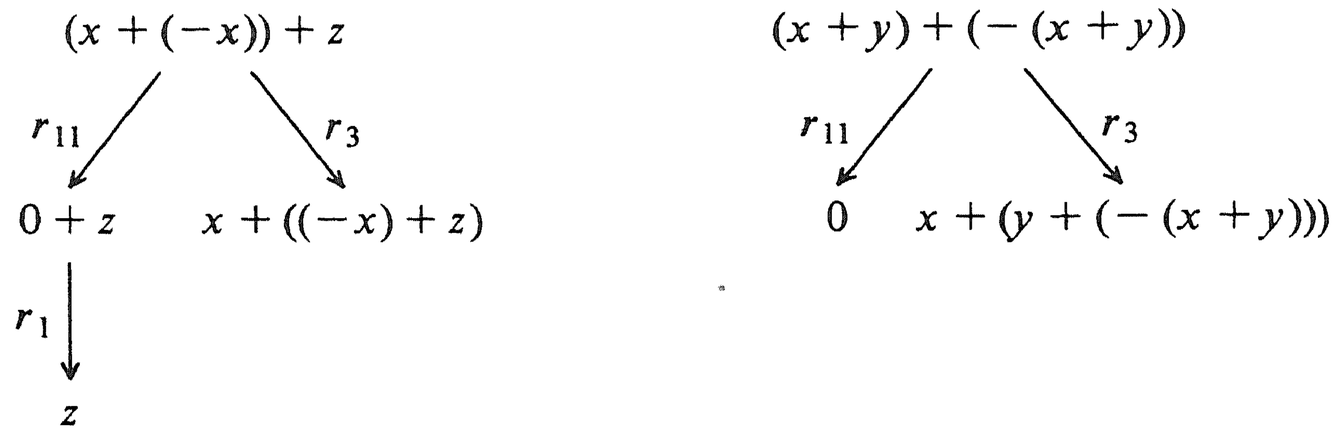
This results in the new rule $r_9: -0 \to 0$. Now r_5 is superfluous: $(-0) + z \xrightarrow{r_9} 0 + z \xrightarrow{r_1} z$. If we superpose r_8 on r_7 we get



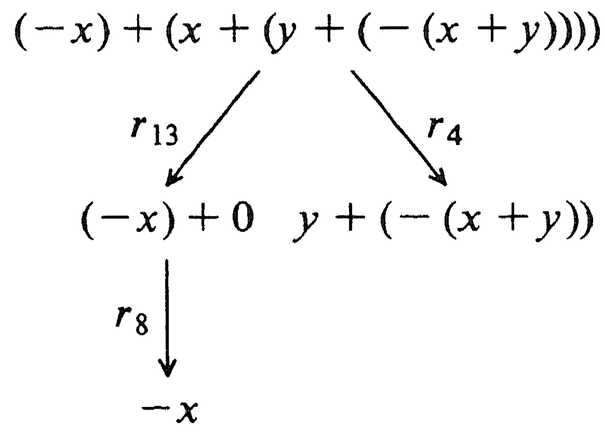
This results in the new rule $r_{10}: -(-x) \to x$. Now r_7 is superfluous: $(-(-x)) + z \xrightarrow{r_9} x + z$. We can superpose r_{10} on r_2:



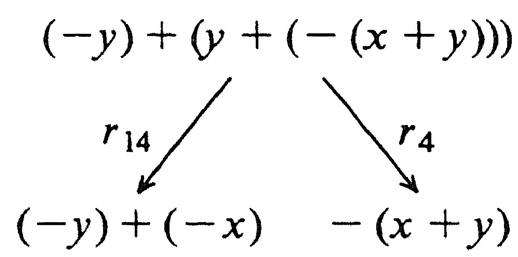
The critical pair $\langle x + (-x), 0\rangle$ is not convergent, so we add the rule $r_{11}: x + (-x) \to 0$. r_{11} can be superposed on r_3 in two ways:



This results in the rules $r_{12}: x + ((-x) + z) \to z$ and $r_{13}: x + (y + (-(x + y))) \to 0$. We can superpose r_{13} on r_4:



To make the critical pair $\langle (-x) + 0, y + (-(x + y)) \rangle$ convergent, we add the rule $r_{14}: y + (-(x + y)) \to -x$. Now r_{13} becomes superfluous: $x + (y + (-(x + y))) \xrightarrow{r_{14}} x + (-x) \xrightarrow{r_{11}} 0$. Superposition of r_{14} on r_4 yields

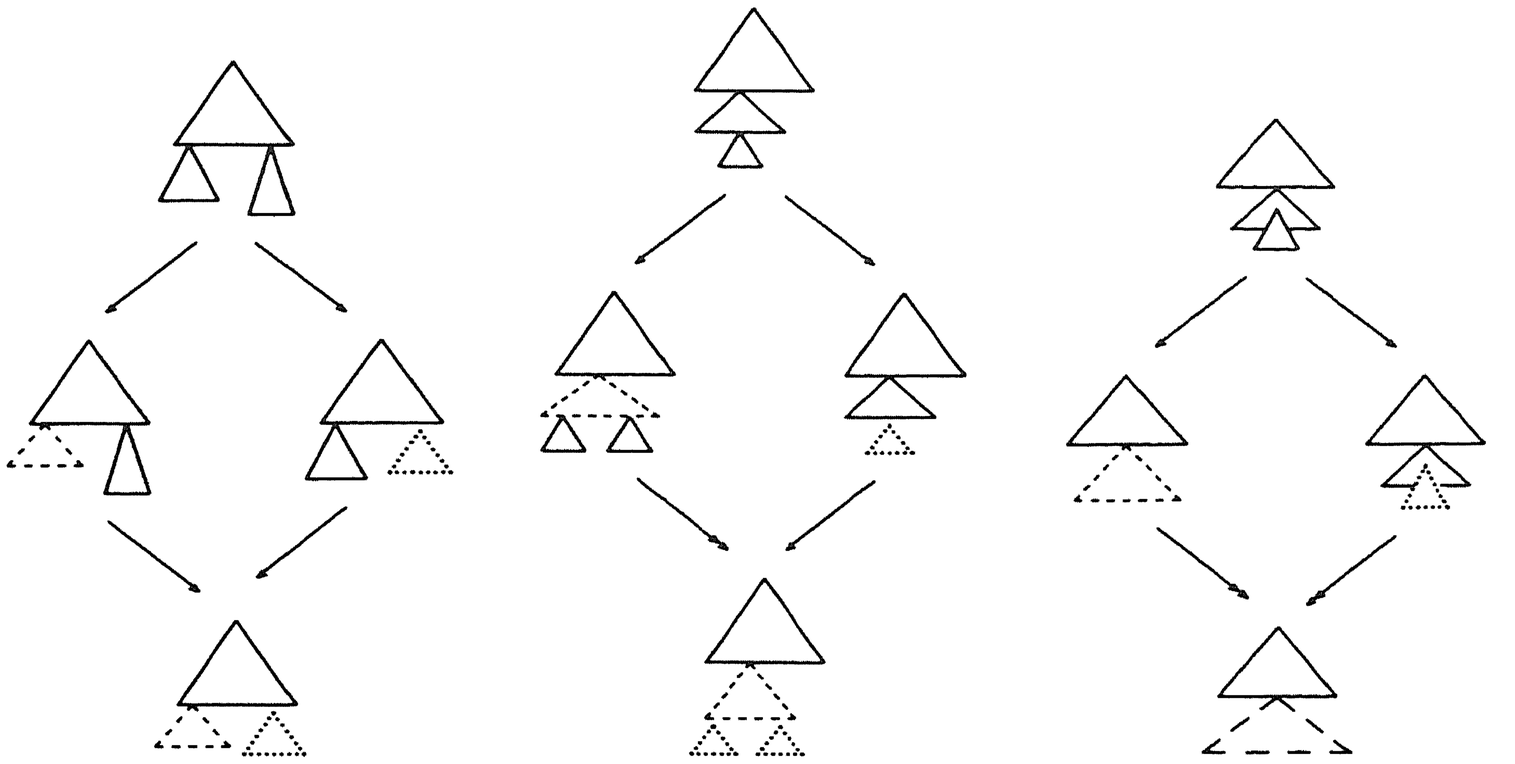


We add the rule $r_{15}: -(x + y) \to (-y) + (-x)$. Now r_{14} is no longer needed: $y + (-(x + y)) \to y + ((-y) + (-x)) \xrightarrow{r_{12}} -x$. At this moment the TRS has only convergent critical pairs. The significance of this fact is stated in the following lemma.

LEMMA 4.3 (HUET [7]). *A* TRS $\mathscr{R}$ *is* WCR *if and only if all critical pairs are convergent.*

PROOF SKETCH. If $\mathscr{R}$ is WCR then in particular all critical pairs have a common reduct. Conversely, consider two diverging reduction steps. We distinguish three cases, depending on the relative position of the two rewritten

redexes (see Figure 5). It is clear how to find converging reductions in the first two cases (disjoint and nested redexes). In the case of overlapping redexes we use the hypothesis of the lemma that every critical pair is convergent. □



(a) disjoint redexes (b) nested redexes (c) overlapping redexes

FIGURE 5.

So the TRS $\mathscr{R}_c$ with rewriting rules as in Table 6 is WCR.

$r_1 : 0 + x$	$\to x$	$r_9 : -0$	$\to 0$
$r_2 : (-x) + x$	$\to 0$	$r_{10} : -(-x)$	$\to x$
$r_3 : (x + y) + z$	$\to x + (y + z)$	$r_{11} : x + (-x)$	$\to 0$
$r_4 : (-x) + (x + z)$	$\to z$	$r_{12} : x + ((-x) + z)$	$\to z$
$r_8 : x + 0$	$\to x$	$r_{15} : -(x + y)$	$\to (-y) + (-x)$

TABLE 6.

$\mathscr{R}_c$ is also SN; a proof of this fact will not be given here. According to Newman's Lemma (3.8) $\mathscr{R}_c$ is 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 Lemma 3.8 and Lemma 4.3:

COROLLARY 4.4 (KNUTH & BENDIX [11]). *Let $\mathscr{R}$ be a* TRS *which is* SN. *Then $\mathscr{R}$ is* CR *if and only if all critical pairs of $\mathscr{R}$ are convergent.*

The complete TRS $\mathscr{R}_c$ of Table 6 was not obtained in a very systematic way. Critical pairs were generated without any strategy and especially the orientation of the new rewriting rules was only guided by our intuition. For most of the new rules there was no other possibility: e.g. the rule $r_4 : (-x) + (x + z) \to z$ cannot be reversed because $z \to (-x) + (x + z)$ is not a legitimate rewriting rule; and the orientation of $r_9 : -0 \to 0$ cannot be

changed because the rule $0 \to -0$ results in a TRS which is not SN. For rule $r_{15}: -(x+y) \to (-y)+(-x)$ the other direction was at least as plausible, as it is even length-decreasing. However, this would have led to disastrous complications (described in [11]).

The problem of the orientation of the new rules is solved in implementations of the Knuth-Bendix completion algorithm by preordaining a reduction ordering on the terms.

DEFINITION 4.5. 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 $\mathscr{R}$ is SN, then $\to_{\mathscr{R}}^+$ satisfies the requirements of Definition 4.5):

PROPOSITION 4.6. *A* TRS $\mathscr{R}$ *is* SN *if and only if there is a reduction ordering* $>$ *such that* $\alpha > \beta$ *for every rewriting rule* $\alpha \to \beta$ *of* $\mathscr{R}$.

In Figure 7 a simple version of the Knuth-Bendix completion algorithm is presented. Note that no attempts are made either to simplify the rules or to remove superfluous rules. As to the reduction ordering $>$ on $T(\Sigma)$ which is an input to the algorithm, this is a matter of ingenuity, or experimentation.

Simple version of the Knuth-Bendix completion algorithm

Input: - an equational specification (Σ, E),
— a reduction ordering $>$ on Ts (i.e. a program which computes $>$).
Output: — a complete TRS $\mathscr{R}$ such that
$\forall s, t \in Ts \quad s =_{\mathscr{R}} t \iff (\Sigma, E) \vdash s = t.$

$\mathscr{R} := \varnothing$;
while $E \neq \varnothing$ **do**
  choose an equation $s = t \in E$;
  reduce s and t to respective normal forms s' and t' with respect to $\mathscr{R}$;
  if $s' \equiv t'$ **then**
    $E := E - \{s = t\}$
  else
    if $s' > t'$ **then**
      $\alpha := s'$; $\beta := t'$
    else if $t' > s'$ **then**
      $\alpha := t'$; $\beta := s'$
    else
      failure
    fi;
    $CP := \{P = Q \mid \langle P, Q \rangle$ is a critical pair between the rules in $\mathscr{R}$ and $\alpha \to \beta\}$;
    $\mathscr{R} := \mathscr{R} \cup \{\alpha \to \beta\}$;
    $E := E \cup CP - \{s = t\}$
  fi
od;
success

FIGURE 7.

The program of Figure 7 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. For example, there exists no complete TRS for the specification of abelian groups of Table 8.

$0 + x$	$=$	x
$(-x) + x$	$=$	0
$(x + y) + z$	$=$	$x + (y + z)$
$x + y$	$=$	$y + x$

TABLE 8.

If we still want to compute with the above system, we have to work modulo the associativity and commutativity of the $+$-operator. This means that one does not consider terms individually, but equivalence classes of terms. We will not go into any details because completion modulo equations (like $x + y = y + x$ and $(x + y) + z = x + (y + z)$) is a very technical and complicated matter (see e.g. Peterson & Stickel [14], Jouannaud & Kirchner [10]).

In case (1) the resulting TRS $\mathscr{R}$ is complete. To show this requires a non-trivial proof (for an 'early' example of such a proof see Huet [8]). In Section 5 we will give an abstract formulation of Knuth-Bendix completion, which streamlines considerably this kind of correctness proofs.

The program of Figure 7 does not 'simplify' the rewriting rules themselves. We will now show how such an optimization can be performed after termination of the program.

DEFINITION 4.7. A TRS $\mathscr{R}$ is called *irreducible* if for every rewriting rule $\alpha \to \beta$ of $\mathscr{R}$ the following holds: (1) β is a normal form with respect to $\mathscr{R}$, (2) α is a normal form with respect to $\mathscr{R} - \{\alpha \to \beta\}$.

The next theorem states that every complete TRS can be transformed into an irreducible complete TRS.

THEOREM 4.8 (MÉTIVIER [12]). *Let $\mathscr{R}$ be a complete* TRS. *Then we can find an irreducible complete* TRS $\mathscr{R}'$ *such that the relations $=_{\mathscr{R}}$ and $=_{\mathscr{R}'}$ are the same.*

PROOF SKETCH. First we replace every rewriting rule $\alpha \to \beta$ of $\mathscr{R}$ by $\alpha \to \beta'$ where β' is the normal form of β with respect to $\mathscr{R}$. The resulting TRS is $\mathscr{R}_1$. Next, we leave away every rule $\alpha \to \beta$ in $\mathscr{R}_1$ such that α can be reduced by *another* rule $\alpha' \to \beta'$ of $\mathscr{R}_1$. Result: $\mathscr{R}'$. Now it is not hard to prove that $\mathscr{R}'$ is indeed complete and that $=_{\mathscr{R}}$ coincides with $=_{\mathscr{R}'}$, by first proving the analogous facts for $\mathscr{R}_1$ and $\mathscr{R}$. □

The Knuth-Bendix completion algorithm of Figure 9 simplifies the rewriting rules as much as possible. The only reason to perform simplification during the completion process - and not after its termination, which is possible according to Theorem 4.8 - is the efficiency of the completion process. Note that the program differentiates between a simplification of the left-hand side and a simplification of the right-hand side of a rewriting rule.

Efficient version of the Knuth-Bendix completion algorithm

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

Output: - a complete irreducible TRS $\mathscr{R}$ such that
$\forall s, t \in T(\Sigma) \quad s =_{\mathscr{R}} t \Leftrightarrow (\Sigma, E) \vdash s = t.$

$\mathscr{R} := \emptyset$;
while $E \neq \emptyset$ **do**
  choose an equation $s = t \in E$;
  reduce s and t to respective normal forms s' and t' with respect to $\mathscr{R}$;
  if $s' \equiv t'$ **then**
    $E := E - \{s = t\}$
  else
    if $s' > t'$ **then**
      $\alpha := s'$; $\beta := t'$
    else if $t' > s'$ **then**
      $\alpha := t'$; $\beta := s'$
    else
      failure
    fi;
    $\mathscr{R} := \{\gamma \to \delta' \mid \gamma \to \delta \in \mathscr{R}$ and δ' is a normal form of δ with respect to $\mathscr{R} \cup \{\alpha \to \beta\}\}$;
    $CP := \{P = Q \mid \langle P, Q \rangle$ is a critical pair between the rules in $\mathscr{R}$ and $\alpha \to \beta\}$;
    $E := E \cup CP \cup \{\gamma = \delta \mid \gamma \to \delta \in \mathscr{R}$ and γ is reducible by $\alpha \to \beta\} - \{s = t\}$;
    $\mathscr{R} := \mathscr{R} \cup \{\alpha \to \beta\} - \{\gamma \to \delta \mid \gamma$ is reducible by $\alpha \to \beta\}$
  fi
od;
success

FIGURE 9.

We conclude this section with a theorem stating that the Knuth-Bendix completion algorithm, given an equational specification and a reduction ordering, cannot generate two different complete irreducible TRSs. The proof is omitted.

DEFINITION 4.9. Let $>$ be a reduction ordering. We call a TRS $\mathscr{R}$ *compatible* with $>$ if $\alpha > \beta$ for every rewriting rule $\alpha \to \beta$ of $\mathscr{R}$.

THEOREM 4.10 (MÉTIVIER [12]). *Let $\mathscr{R}_1$ and $\mathscr{R}_2$ be two complete irreducible* TRS*s compatible with a given reduction ordering $>$. If $\mathscr{R}_1$ and $\mathscr{R}_2$ define the same convertibility, then $\mathscr{R}_1$ and $\mathscr{R}_2$ are equal (modulo renaming of variables).*

5. KNUTH-BENDIX ALGORITHM: ABSTRACT FORMULATION

There are many completion algorithms such as the two examples (in Figure 7 and 9) above, differing in order of execution of ways of optimization. The question is, how to prove that these algorithms are correct, i.e. deliver upon successful termination indeed a TRS $\mathscr{R}$ with the same equality as the one of the original set of equations E. As there is a whole family of completion algorithms, one needs to extract the 'abstract principles' of such algorithms; and this is done indeed by Bachmair, Dershowitz and Hsiang [2]. Their method for proving correctness of completion algorithms starts with the introduction of a derivation system where the objects are pairs $(E,\mathscr{R})$; each derivation step from $(E,\mathscr{R})$ to $(E',\mathscr{R}')$ preserves equality: $=_{E\cup\mathscr{R}}$ coincides with $=_{E'\cup\mathscr{R}'}$, and moreover, along a sequence of derivations the actual proofs of equations $t=s$ will be getting 'better and better', with a 'rewrite proof' as optimal proof format. See Figure 10, where it is shown how E (that is the pair $(E, \varnothing)$) is gradually transformed via pairs $(E',\mathscr{R}')$ to a TRS $\mathscr{R}$ (that is the pair $(\varnothing,\mathscr{R})$); along the way the two example proofs in the figure get more and more oriented until they are in rewrite form. (Here direction is downward; horizontal steps are without direction.)

There are two crucial ideas in this recent approach. One is the concept of a derivation system on pairs $(E,\mathscr{R})$ as discussed above. The other is the concept of ordering the proofs of equations $s=t$ according to their degree of orientation. We will now proceed to a more formal explanation.

DEFINITION 5.1. Let (Σ, E) be an equational specification. If $s =_E t$ by application of exactly one equation in E we write $s \leftrightarrow_E t$. So $s \leftrightarrow_E t$ if and only if there exist a context $C[\ ]$, a substitution σ and an equation $u = v$ (or $v = u$) in E such that $s \equiv C[u^\sigma]$ and $t \equiv C[v^\sigma]$.

DEFINITION 5.2. Let (Σ, E) be an equational specification and $\mathscr{R}$ a TRS with signature Σ. A *proof* in $E \cup \mathscr{R}$ of an equation $s = t$ between terms $s, t \in T(\Sigma)$ is a sequence of terms $(s_0,\ldots,s_n)$ such that $s_0 \equiv s$, $s_n \equiv t$, and for all $0 < i \leqslant n$ we have $s_{i-1} \leftrightarrow_E s_i$, $s_{i-1} \to_{\mathscr{R}} s_i$ or $s_{i-1} \leftarrow_{\mathscr{R}} s_i$. A *subproof* of $P \equiv (s_0,\ldots,s_n)$ is a proof $P' \equiv (s_i,\ldots,s_j)$ with $0 \leqslant i \leqslant j \leqslant n$. The notation $P[P']$ means that P' is a subproof of P. A proof of the form $s_0 \to\!\!\!\!\to_{\mathscr{R}} s_k \leftarrow\!\!\!\!\leftarrow_{\mathscr{R}} s_n$ is called a *rewrite proof*.

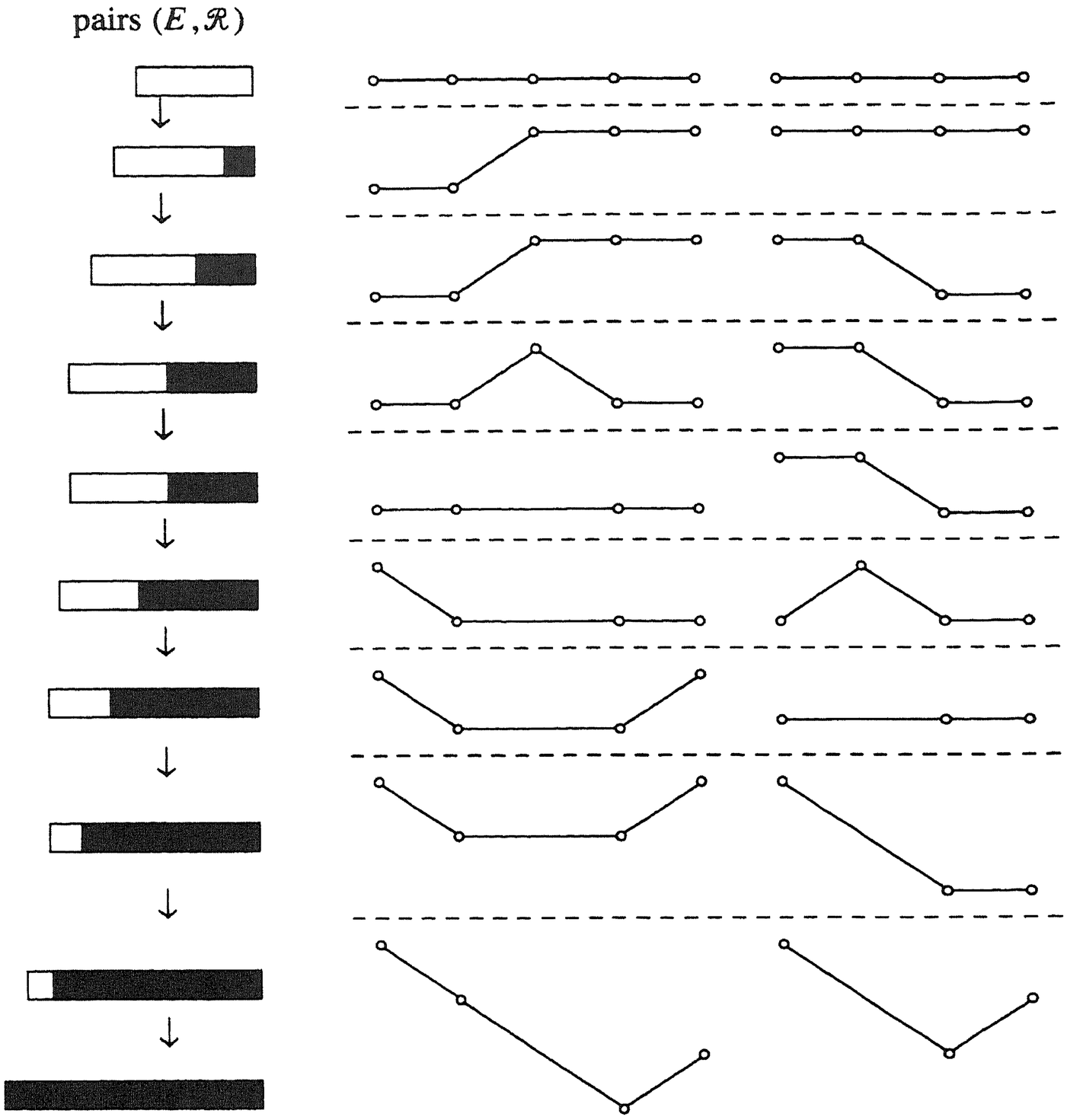


FIGURE 10.

By definition, $P \equiv (s)$ is a proof of $s = s$. In Figure 11 a proof is sketched.

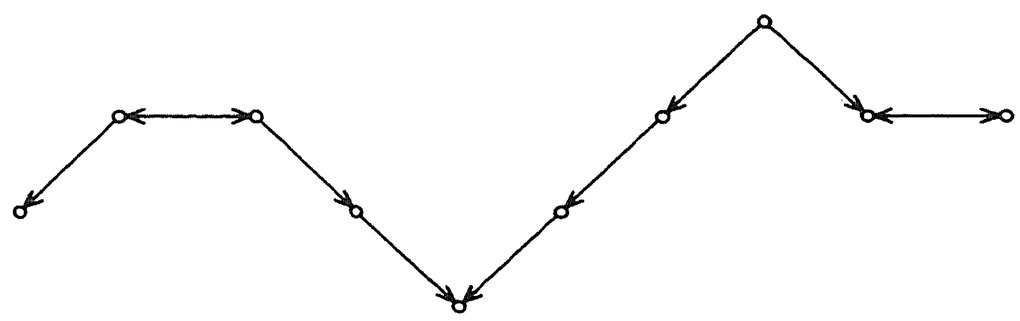


FIGURE 11.

Knuth-Bendix completion aims at transforming every proof $(s_0,\ldots,s_n)$ into a

rewrite proof $s_0 \twoheadrightarrow t \twoheadleftarrow s_n$. We now present an inference system for Knuth-Bendix completion. The objects of this system are pairs $(E, \mathscr{R})$.

Let $>$ be a reduction ordering. The inference system $\mathscr{BC}$ (*basic completion*) consists of the inference rules of Table 12.

(C_1)	*Orienting an equation*			
	$(E \dot\cup \{s \doteq t\}, \mathscr{R})$	$\pi_{\mathscr{BC}}$	$(E, \mathscr{R} \cup \{s \to t\})$	if $s > t$
(C_2)	*Adding an equation*			
	$(E, \mathscr{R})$	$\pi_{\mathscr{BC}}$	$(E \cup \{s = t\}, \mathscr{R})$	if $s \leftarrow_{\mathscr{R}} u \rightarrow_{\mathscr{R}} t$
(C_3)	*Simplifying an equation*			
	$(E \dot\cup \{s \doteq t\}, \mathscr{R})$	$\pi_{\mathscr{BC}}$	$(E \cup \{u = t\}, \mathscr{R})$	if $s \to_{\mathscr{R}} u$
(C_4)	*Deleting a trivial equation*			
	$(E \dot\cup \{s = s\}, \mathscr{R})$	$\pi_{\mathscr{BC}}$	$(E, \mathscr{R})$	

TABLE 12.

The notation $s \doteq t$ means $s = t$ or $t = s$; the symbol $\dot\cup$ denotes disjoint union. Note that the inference system does not contain inference rules for simplification of the rewriting rules.

DEFINITION 5.3. A (possibly infinite) sequence $(E_0, \mathscr{R}_0)$, $(E_1, \mathscr{R}_1)$, ... is called a *$\mathscr{BC}$-derivation* if $(E_{i-1}, \mathscr{R}_{i-1}) \pi^+_{\mathscr{BC}} (E_i, \mathscr{R}_i)$ for all $i > 0$. ($\pi^+_{\mathscr{BC}}$ is the transitive closure of $\pi_{\mathscr{BC}}$.)

The statements in the following proposition are easily proved.

PROPOSITION 5.4. *Let* $(E, \mathscr{R}) \pi^+_{\mathscr{BC}} (E', \mathscr{R}')$. (1) *If* $\mathscr{R}$ *is compatible with the reduction ordering* $>$ *then so is* $\mathscr{R}'$. (2) *The relations* $=_{E \cup \mathscr{R}}$ *and* $=_{E' \cup \mathscr{R}'}$ *are equal.*

Let $(E, \mathscr{R}) \pi^+_{\mathscr{BC}} (E', \mathscr{R}')$. Proposition 5.4(1) states that if $\mathscr{R}$ is SN then so is $\mathscr{R}'$ and Proposition 5.4(2) states that the inference rules of $\mathscr{BC}$ are *sound*, i.e. the set of equations provable in $(E, \mathscr{R})$ is the same as the set of equations provable in $(E', \mathscr{R}')$.

Although the same equations $s = t$ are provable in $E \cup \mathscr{R}$ and $E' \cup \mathscr{R}'$, proofs in $E' \cup \mathscr{R}'$ are in general 'simpler' than in $E \cup \mathscr{R}$. For example, by adding equations to E (inference rule C_2) some subproofs $s \leftarrow_{\mathscr{R}} u \rightarrow_{\mathscr{R}} t$ can be replaced by $s \leftrightarrow_{E'} t$. To formalize this aspect of the inference system $\mathscr{BC}$ we introduce orderings on proofs.

DEFINITION 5.5. A binary relation $\rightsquigarrow$ on proofs is *monotonic* if $Q \rightsquigarrow Q'$ implies that $P[Q] \rightsquigarrow P[Q']$ for all proofs P, Q and Q'. The relation $\rightsquigarrow$ is *stable* if

$$P \equiv (s, \ldots, u_i, \ldots, t) \rightsquigarrow (s, \ldots, v_j, \ldots, t) \equiv Q$$

implies that

$$(C[s^\sigma], \ldots, C[u_i^\sigma], \ldots, C[t^\sigma]) \rightsquigarrow (C[s^\sigma], \ldots, C[v_j^\sigma], \ldots, C[t^\sigma])$$

for all proofs P and Q, contexts $C[\ ]$ and substitutions σ. A *proof ordering* is a stable, monotonic, well-founded partial ordering on proofs.

To illustrate these concepts we will give an alternative proof of Newman's Lemma (3.8) using proof orderings. ('Alternative' with respect to the proofs that we have seen in the literature.) Let $\mathscr{R}$ be a TRS which is SN and WCR. Let $P \equiv (s_0,...,s_n)$ be a proof of the conversion $s_0 = s_n$. We define the *complexity* $\|P\|$ of the proof P as the multiset $[s_0,...,s_n]$. The ordering $\rightsquigarrow_{\mathscr{R}}$ which we use is the multiset extension of $\rightarrow^+_{\mathscr{R}}$. So $P \rightsquigarrow_{\mathscr{R}} P'$ if and only if $\|P\| \overset{m}{\rightarrow}{}^+_{\mathscr{R}} \|P'\|$, where $\overset{m}{\rightarrow}{}^+_{\mathscr{R}}$ denotes the multiset extension of $\rightarrow^+_{\mathscr{R}}$. (Example: $[s_0, s_1, s_2] \rightsquigarrow_{\mathscr{R}} [s_0, s_{11},...,s_{1n}, s_2]$ if $s_1 \rightarrow^+_{\mathscr{R}} s_{1i}$ $(i = 1,...,n, n \geqslant 0)$. Any element of the multiset may be replaced by arbitrarily many 'lesser' elements, less in the sense of the well-founded ordering $\rightarrow^+_{\mathscr{R}}$.)

PROPOSITION 5.6. *The relation $\rightsquigarrow_{\mathscr{R}}$ is a proof ordering.*

PROOF. Because $\mathscr{R}$ is SN, $\rightarrow^+_{\mathscr{R}}$ is a well-founded partial ordering. This implies that $\overset{m}{\rightarrow}{}^+_{\mathscr{R}}$ also is a well-founded partial ordering. The stability of $\rightsquigarrow_{\mathscr{R}}$ follows from the fact that $\rightarrow^+_{\mathscr{R}}$ is closed under substitutions and contexts. The monotonicity of $\rightsquigarrow_{\mathscr{R}}$ is a direct consequence of the definition of the multiset extension of an ordering. □

PROPOSITION 5.7. If $P \equiv (s_0,...,s_n)$ is not a rewrite proof then there is a proof P' of the equation $s_0 = s_n$ such that $P \rightsquigarrow_{\mathscr{R}} P'$.

PROOF. If P is not a rewrite proof then P contains a subproof $P_0 \equiv s_{k-1} \leftarrow s_k \rightarrow s_{k+1}$. Because $\mathscr{R}$ is WCR, s_{k-1} and s_{k+1} have a common reduct, say $s_{k-1} \twoheadrightarrow t \twoheadleftarrow s_{k+1} \equiv P_1$. The complexity of P_0 is $[s_{k-1}, s_k, s_{k+1}]$. It is clear that $s_k \rightarrow^+_{\mathscr{R}} u$ for every term $u \in \|P_1\|$. So $P_0 \rightsquigarrow_{\mathscr{R}} P_1$. Let P' be the proof $P[P_1 / P_0]$, i.e. the proof P in which the subproof P_0 is replaced by P_1. The desired result follows from the monotonicity of $\rightsquigarrow_{\mathscr{R}}$. □

PROOF OF LEMMA 3.8. Consider diverging reductions $t_1 \twoheadleftarrow s \twoheadrightarrow t_2$. Let P be the corresponding proof. If P is a rewrite proof (i.e. if $t_1 \equiv s$ or $s \equiv t_2$) then we are done. If P is not a rewrite proof then, according to Proposition 5.7, we can find a proof P_1 of $t_1 = t_2$ such that $P \rightsquigarrow_{\mathscr{R}} P_1$. If P_1 is not a rewrite proof then we can find a proof P_2 of $t_1 = t_2$ such that $P_1 \rightsquigarrow_{\mathscr{R}} P_2$. Because $\rightsquigarrow_{\mathscr{R}}$ is a well-founded ordering this process terminates in a rewrite proof of $t_1 = t_2$. We conclude that $\mathscr{R}$ is CR. □

The ordering which we use for completion is based on the given reduction ordering $>$ and on the elementary steps ($\rightarrow_{\mathscr{R}}$, $\leftarrow_{\mathscr{R}}$ or $\leftrightarrow_E$) in a proof.

DEFINITION 5.8. The *complexity* $\|P\|$ of a proof $P \equiv (s_0,...,s_n)$ is the multiset $[c(s_0, s_1),...,c(s_{n-1}, s_n)]$ where $c(s_{i-1}, s_i)$, the complexity of an elementary proof step, is defined by

$$c(s_{i-1}, s_i) = \begin{cases} [s_{i-1}] & \text{if } s_{i-1} \rightarrow_{\mathscr{R}} s_i, \\ [s_i] & \text{if } s_{i-1} \leftarrow_{\mathscr{R}} s_i, \\ [s_{i-1}, s_i] & \text{if } s_{i-1} \leftrightarrow_E s_i. \end{cases}$$

To compare the complexity of the elementary proof steps we use the multiset extension $\gg$ of the given reduction ordering $>$. The ordering $\rightsquigarrow_{\mathscr{BC}}$ is defined by $P \rightsquigarrow_{\mathscr{BC}} P'$ if and only if $\|P\| \gg^m \|P'\|$, where $\gg^m$ denotes the multiset extension of $\gg$.

PROPOSITION 5.9. *The ordering $\rightsquigarrow_{\mathscr{BC}}$ is a proof ordering.*

PROOF. Straightforward. □

DEFINITION 5.10. A proof ordering $\rightsquigarrow$ is *compatible* with $\mathscr{BC}$ if $(E, \mathscr{R}) \mathrel{\text{☞}}^{+}_{\mathscr{BC}} (E', \mathscr{R}')$ implies that for every proof P in $E \cup \mathscr{R}$ of an equation $s = t$ there exists a proof P' of $s = t$ in $E' \cup \mathscr{R}'$ such that $P \rightsquigarrow P'$ or $P \equiv P'$.

PROPOSITION 5.11. *The proof ordering $\rightsquigarrow_{\mathscr{BC}}$ is compatible with $\mathscr{BC}$.*

PROOF. We observe the following (see Figure 13):

- if $(E \dot{\cup} \{s \doteq t\}, \mathscr{R}) \mathrel{\text{☞}}_{\mathscr{BC}} (E, \mathscr{R} \cup \{s \to t\})$ by application of inference rule C_1 then $(s \leftrightarrow_{E \dot{\cup} \{s \doteq t\}} t) \rightsquigarrow_{\mathscr{BC}} (s \to_{\mathscr{R} \cup \{s \to t\}} t)$, because $[s, t] \gg [s]$;
- if $(E, \mathscr{R}) \mathrel{\text{☞}}_{\mathscr{BC}} (E \cup \{s = t\}, \mathscr{R})$ by application of inference rule C_2 with $u \to_{\mathscr{R}} s$ and $u \to_{\mathscr{R}} t$ then $(s \leftarrow_{\mathscr{R}} u \to_{\mathscr{R}} t) \rightsquigarrow_{\mathscr{BC}} (s \leftrightarrow_{E \cup \{s = t\}} t)$, because $[u] \gg [s, t]$;
- if $(E \dot{\cup} \{s \doteq t\}, \mathscr{R}) \mathrel{\text{☞}}_{\mathscr{BC}} (E \cup \{u = t\}, \mathscr{R})$ by application of inference rule C_3 with $s \to_{\mathscr{R}} t$ then $(s \leftrightarrow_{E \dot{\cup} \{s \doteq t\}} t) \rightsquigarrow_{\mathscr{BC}} (s \to_{\mathscr{R}} u \leftrightarrow_{E \cup \{u = t\}} t)$, because $[s, t] \gg [s]$ and $[s, t] \gg [\, , t]$;
- if $(E \dot{\cup} \{s = s\}, \mathscr{R}) \mathrel{\text{☞}}_{\mathscr{BC}} (E, \mathscr{R})$ by application of inference rule C_4 then $(s \leftrightarrow_{E \dot{\cup} \{s = s\}} s) \rightsquigarrow_{\mathscr{BC}} (s)$, because $[s, s] \gg [\,]$.

Now suppose $(E, \mathscr{R}) \mathrel{\text{☞}}_{\mathscr{BC}} (E', \mathscr{R}')$. Let P be a proof in $E \cup \mathscr{R}$ of an equation $s = t$. If P uses an equation of E or a rewriting rule of $\mathscr{R}$ which no longer exists in $E' \cup \mathscr{R}'$, then, according to the above observations and the stability of $\rightsquigarrow_{\mathscr{BC}}$, we can replace such a proof step by a simpler proof in $E' \cup \mathscr{R}'$. The result follows from the monotonicity of $\rightsquigarrow_{\mathscr{BC}}$. □

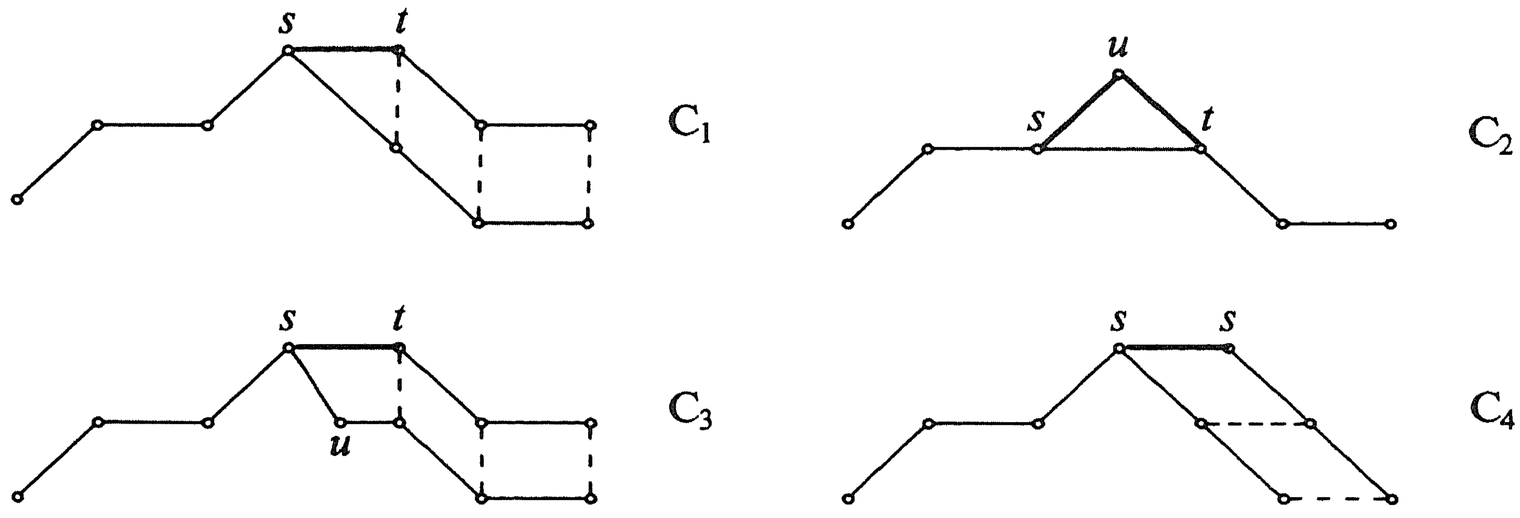


FIGURE 13.

Let $(E_0, \mathscr{R}_0)$, $(E_1, \mathscr{R}_1)$, ... be a $\mathscr{BC}$-derivation. Proposition 5.11 states that proofs in $E_j \cup \mathscr{R}_j$ are no more difficult than corresponding proofs in $E_i \cup \mathscr{R}_i$, for all $j > i$. The following condition implies that every proof in $E_i \cup \mathscr{R}_i$ of an

equation $s = t$ which is not a rewrite proof, can be simplified to a rewrite proof of $s = t$ in $E_j \cup \mathscr{R}_j$, for some $j \geqslant i$.

DEFINITION 5.12. A $\mathscr{BC}$-derivation $(E_0, \mathscr{R}_0)$, $(E_1, \mathscr{R}_1)$, ... is called *fair* if
(1) $\bigcap_{j>i} E_j = \varnothing$ for all $i \geqslant 0$, and
(2) if $\langle c, d \rangle \in \bigcap_{j \geqslant i} CP_j$ for some $i \geqslant 0$ then $c = d \in E_k$ for some $k \geqslant 0$. (CP_j is the set of all critical pairs between the rewriting rules of $\mathscr{R}_j$.)

PROPOSITION 5.13. *Let $(E_0, \mathscr{R}_0)$, $(E_1, \mathscr{R}_1)$, ... be a fair $\mathscr{BC}$-derivation and let P be a proof of $s = t$ in $E_i \cup \mathscr{R}_i$. If P is not a rewrite proof then there exists a proof P' in $E_j \cup \mathscr{R}_j$ of $s = t$ such that $P \rightsquigarrow_{\mathscr{BC}} P'$, for some $j \geqslant i$.*

PROOF. Let P be a proof $(s_0, \ldots, s_n)$ of $s = t$ in $E_i \cup \mathscr{R}_i$. Suppose that P is not a rewrite proof. We distinguish two cases:
(1) If $s_{k-1} \leftrightarrow_{E_i} s_k$ by application of $u = v \in E_i$, for some $1 \leqslant k \leqslant n$, then the equation $u = v$ will after some time be oriented into a rewriting rule, simplified, or deleted (because the derivation is fair). In all cases this results in a simpler proof of $s_{k-1} = s_k$ in $E_j \cup \mathscr{R}_j$, for some $j > i$. Because $\rightsquigarrow_{\mathscr{BC}}$ is a proof ordering compatible with $\mathscr{BC}$, this leads to a simpler proof P' in $E_j \cup \mathscr{R}_j$ of the equation $s = t$.
(2) If P does not contain any $\leftrightarrow$-steps then P contains a subproof $P_0 \equiv s_{k-1} \leftarrow_{\mathscr{R}_i} s_k \rightarrow_{\mathscr{R}_i} s_{k+1}$, for some $1 \leqslant k < n$. If the rewritten redexes in the reduction steps $s_{k-1} \leftarrow_{\mathscr{R}_i} s_k$ and $s_k \rightarrow_{\mathscr{R}_i} s_{k+1}$ do not overlap (see Figure 5(a, b)) then there is a proof $s_{k-1} \twoheadrightarrow_{\mathscr{R}_i} u \twoheadleftarrow_{\mathscr{R}_i} s_{k+1}$ which is simpler than P_0. If the rewritten redexes in the reduction steps $s_{k-1} \leftarrow_{\mathscr{R}_i} s_k$ and $s_k \rightarrow_{\mathscr{R}_i} s_{k+1}$ are overlapping (see Figure 5(c)) then $s_{k-1} = s_{k+1}$ contains an instance of a critical pair $c = d$ that after some time will be computed because the derivation is fair. In both cases P_0 can be replaced by a simpler proof and thus there is proof P' of $s = t$ which is simpler than P. □

DEFINITION 5.14. A *completion procedure* is a strategy for applying the inference rules of $\mathscr{BC}$ to given inputs (Σ, E) and $>$, in order to generate a $\mathscr{BC}$-derivation $(E_0, \mathscr{R}_0)$, $(E_1, \mathscr{R}_1)$, ... with $(E_0, \mathscr{R}_0) = (E, \varnothing)$.

Because a fair derivation may not be possible for certain inputs (Σ, E) and $>$, we allow for a completion procedure to fail. We say that a completion procedure is *fair* if it generates only fair derivations unless it fails.

DEFINITION 5.15. Let $(E_0, \mathscr{R}_0)$, $(E_1, \mathscr{R}_1)$, ... be a $\mathscr{BC}$-derivation. We denote the *limit* of the TRSs $\mathscr{R}_n$ by $\mathscr{R}^\infty$ ($\mathscr{R}^\infty = \bigcup_n \mathscr{R}_n$).

THEOREM 5.16 (BACHMAIR, DERSHOWITZ & HSIANG [2]). *Let C be a fair completion procedure that does not fail for certain inputs (Σ, E) and $>$.*
(1) *If $s =_E t$ then C will generate a pair $(E_i, \mathscr{R}_i)$ such that s and t have a common reduct in $\mathscr{R}_i$.*
(2) *$\mathscr{R}^\infty$ is a complete* TRS.

PROOF.

(1) Let $s =_E t$. The corresponding proof $P_0 \equiv s =_{E_0} t$ is not a rewrite proof. By Proposition 5.13 there is a proof P_1 of $s = t$ in $E_{i_1} \cup \mathscr{R}_{i_1}$ such that $P_0 \rightsquigarrow_{\mathscr{BC}} P_1$, for some $i_1 \geqslant 1$. If P_1 is not a rewrite proof then there is a proof P_2 of $s = t$ in $E_{i_2} \cup \mathscr{R}_{i_2}$ such that $P_1 \rightsquigarrow_{\mathscr{BC}} P_2$, for some $i_2 \geqslant i_1$. Because $\rightsquigarrow_{\mathscr{BC}}$ is a well-founded ordering this process terminates in a rewrite proof $P_n \equiv s \twoheadrightarrow u \twoheadleftarrow t$ of $s = t$ in $E_{i_n} \cup \mathscr{R}_{i_n}$, for some i_n.

(2) Straightforward.

□

It is now relatively easy to show that the program of Figure 7 is a completion procedure. If the program terminates successfully then the resulting TRS is complete according to the above theorem.

It is possible to extend the inference system $\mathscr{BC}$ with inference rules incorporating the simplification of the rewriting rules as performed in the more efficient version of the completion program in Figure 9. It is also possible to describe completion modulo certain equations, like commutativity for abelian groups, in the same formalism. We refer to [1] and [2] for further details.

REFERENCES

1. L. BACHMAIR and N. DERSHOWITZ (1987). Completion for rewriting modulo a congruence. *Proceedings of the 2nd International Conference on Rewriting Techniques and Applications,* Bordeaux, France, Lecture Notes in Computer Science 256, Springer-Verlag, 192-203.
2. L. BACHMAIR, N. DERSHOWITZ and J. HSIANG (1986). Orderings for equational proofs. *Proceedings of the IEEE Symposium on Logic in Computer Science,* Cambridge, Massachussetts, 346-357.
3. H.P. BARENDREGT (1984). *The Lambda Calculus - Its Syntax and Semantics*, 2nd edition, North-Holland.
4. G. BIRKHOFF (1935). On the structure of abstract algebras. *Proceedings of the Cambridge Philosophical Society 31,* 433-454
5. N. DERSHOWITZ, (1985). Computing with rewrite systems. *Information and Control 65,* 122-157.
6. N. DERSHOWITZ and D.A. PLAISTED (1985). Logic programming cum applicative programming. *Proceedings of the IEEE International Symposium on Logic Programming,* Boston, 54-66.
7. G. HUET (1980). Confluent reductions: abstract properties and applications to term rewriting systems. *Journal of the Association for Computing Machinery 27(4),* 797-821.
8. G. HUET (1981). A complete proof of correctness of the Knuth-Bendix completion algorithm. *Journal of Computer and Systems Sciences 23(1),* 11-21.
9. G. HUET and J.-M. HULLOT (1982). Inductive proofs for equational theories with constructors. *Journal of Computer and Systems Sciences 25(2),* 239-266.

10. J.-P. JOUANNAUD and H. KIRCHNER (1986). Completion of a set of rules modulo a set of equations. *SIAM Journal of Computing 15,* 1155-1194.
11. D.E. KNUTH and P. BENDIX (1970). Simple word problems in universal algebras. J. LEECH (ed.). *Computational Problems in Abstract Algebra,* Pergamon Press, 263-297.
12. Y. MÉTIVIER (1983). About the rewriting systems produced by the Knuth-Bendix completion algorithm. *Information Processing Letters 16,* 31-34.
13. M.H.A. NEWMAN (1942). On theories with a combinatorial definition of equivalence. *Annals of Mathematics 43(2),* 223-243.
14. G. PETERSON and M. STICKEL (1981). Complete sets of reductions for some equational theories. *Journal of the Association for Computing Machinery 28,* 233-264.