A complete proof of correctness of the Knuth-Bendix completion algorithm

Gérard Huet

INRIA

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105 78150 Le Chesnay
France
Tél. 954 90 20

Rapports de Recherche

N° 25

A COMPLETE PROOF OF CORRECTNESS OF THE KNUTH-BENDIX COMPLETION ALGORITHM

Gérard HUET

Juillet 1980

A COMPLETE PROOF OF CORRECTNESS OF THE KNUTH-BENDIX COMPLETION ALGORITHM

Gérard HUET

Résumé :

Nous donnons dans ce papier une description complète de l'algorithme de complétion de Knuth et Bendix. Nous prouvons complètement sa correction, en isolant avec soin les notions abstraites nécessaires, afin que la preuve puisse être étendue à d'autres versions et aux extensions de l'algorithme de base. Nous montrons que, dans les théories équationnelles remplissant les conditions d'application, l'algorithme définit une procédure de semi-décision pour la validité d'une équation. Lorsque l'algorithme termine, il génère un ensemble canonique de simplifications qui définit une procédure de décision pour la variété considérée.

Abstract :

We give in this paper a complete description of the Knuth-Bendix completion algorithm. We prove its correctness in full, isolating carefully the essential abstract notions, so that the proof may be extended to other versions and extensions of the basic algorithm. We show that it defines a semidecision algorithm for the validity problem in the equational theories for which it applies, yielding a decision procedure whenever the algorithm terminates.

A Complete Proof of Correctness of the Knuth-Bendix Completion Algorithm

Gérard Huet

Abstract

We give in this paper a complete description of the Knuth-Bendix completion algorithm. We prove its correctness in full, isolating carefully the essential abstract notions, so that the proof may be extended to other versions and extensions of the basic algorithm. We show that it defines a semidecision algorithm for the validity problem in the equational theories for which it applies, yielding a decision procedure whenever the algorithm terminates.

Introduction

We assume familiarity with the basic notions of the theory of term rewriting systems and in particular with the Knuth-Bendix completion method[7,2,5]. We recall that a term rewriting system $\mathcal{R}$ is a set of pairs of first-order terms $\lambda_i \rightarrow \rho_i$, such that all variables in ρ_i appear in λ_i. We say that term M *reduces by* $\mathcal{R}$ *to* N, and we write $M \rightarrow_{\mathcal{R}} N$, if and only if N is M, in which some occurrence of a substitution instance $\sigma(\lambda_i)$ of a left-hand side of $\mathcal{R}$ is replaced by the same instance $\sigma(\rho_i)$ of the corresponding right-hand side. An *$\mathcal{R}$-normal form* is a term irreducible by $\mathcal{R}$.

A *reduction ordering* $\succ$ is a well-founded partial ordering on terms closed by term replacement and substitution. That is, $M \succ N$ implies that $P[M] \succ P[N]$ for any term context $P[\,]$ and $\sigma(M) \succ \sigma(N)$ for any substitution σ. Note that if $\succ$ is a reduction ordering such that we have $\lambda \succ \rho$ for every $\lambda \rightarrow \rho$ in $\mathcal{R}$, then $\mathcal{R}$ is *nœtherian* (has the finite termination property), that is, there are no infinite reduction sequences.

If $\mathcal{E}$ is a set of equations $M_i = N_i$, we denote by $=_{\mathcal{E}}$ the congruence closure of $\mathcal{E}$. We extend this notation to term rewriting systems, and to unions of equations and rewrite rules. Note that $=_{\mathcal{R}}$ is $(\rightarrow_{\mathcal{R}} \cup \rightarrow_{\mathcal{R}}^{-1})^*$.

We say that $\mathcal{R}$ is *confluent* if and only if $M \rightarrow_{\mathcal{R}}^* N_1$ and $M \rightarrow_{\mathcal{R}}^* N_2$ imply the existence of P such that $N_1 \rightarrow_{\mathcal{R}}^* P$ and $N_2 \rightarrow_{\mathcal{R}}^* P$. We say that $\mathcal{R}$ is a *canonical set* if and only if it is both confluent and nœtherian.

The Knuth-Bendix theorem[7,2,5] gives a decision procedure for the confluence of nœtherian term rewriting systems. The basic idea is to consider the cases where two left-hand sides of $\mathcal{R}$ superpose in a nontrivial way to create an ambiguity

This work was partially supported by AFOSR Contract F49620-79-C-0099 and by ONR Contract N00014-75-C-0816, while the author was visiting SRI International.

of the form $M \rightarrow_{\mathcal{R}} N_1$ and $M \rightarrow_{\mathcal{R}} N_2$ (We say that (N_1, N_2) is a *critical pair.*) The system $\mathcal{R}$ is nonconfluent if and only if for some such pair, N_1 and N_2 reduce to two distinct $\mathcal{R}$-normal forms P_1 and P_2.

The Knuth-Bendix completion algorithm consists in attempting to complete a nonconfluent system into a confluent one by adding new rewrite rules, such as $P_1 \rightarrow P_2$. This must be done in such a way that the completed system is still finitely terminating. Of course, one round of completion is not sufficient in general, since new ambiguities may have been created. During this completion process, some newly introduced rule may simplify some old rule, either on its left or on its right-hand side. It is essential, both for efficiency and elegance, to keep all rules inter-reduced as much as possible. But then the question arises as to how the process can be carried out efficiently in an incremental fashion; that is, we do not want to recompute critical pairs between rules that have been previously considered. However, the rules that have been used to resolve these ambiguities may not exist any more, and so this step must be carefully justified. When a set of equations can be oriented so that the completion process terminates, the resulting canonical term rewriting system defines a decision procedure for the validity problem in the corresponding variety. This may be considered as compiling a set of axioms into a canonical-form algorithm.

We present here a complete proof of correctness of the Knuth-Bendix completion algorithm with incremental computation of critical pairs. We think it is important to present this proof in detail for the following reasons:

1) The importance of such theorem-proving methods is increasingly recognized, since many known decision procedures appear to be derivable through this approach[4,6].

2) The proof turned out to be more difficult than we had expected, and revealed critical conditions for the justification of rewrite rules simplifications, which may not be met by existing implementations. In particular, it is not enough to require that all the successive term rewriting systems $\mathcal{R}_1, \mathcal{R}_2, \ldots$ constructed by the algorithm be nœtherian. They must be terminating *for the same reason*; i.e., there must exist some uniform reduction ordering $\succ$ showing the termination of all these sets.

3) We give a meaning to the algorithm even in the case where it does not converge. We show that in this case we get a semidecision procedure for the validity problem in the original equational theory.

4) We believe our proof can be extended to the various extensions of the Knuth-Bendix procedure that have been proposed in the literature, for rewrite rules on equivalence classes of terms modulo permutations[8,9,10,11], for word problems in finitely presented algebras[1,4] and for inductive proofs[3].

In the following, $\mathcal{E}_i$ is a finite set of equations, and $\mathcal{R}_i$ is a finite rewriting system, for $i \in \mathcal{N}$. Every rewrite rule in $\mathcal{R}_i$ has a label, which is a unique integer. We denote by $k{:}\lambda \to \rho$ the rewrite rule $\lambda \to \rho$ with label k. Finally, every rewrite rule in $\mathcal{R}_i$ is marked or unmarked.

The Completion Algorithm.

Initial data: a (finite) set of equations $\mathcal{E}$, and a (recursive) reduction ordering $\succ$.

1. *Initialization:* Let $\mathcal{E}_0 = \mathcal{E}$, $\mathcal{R}_0 = \emptyset$, $i = 0$, $p = 0$.

2. If $\mathcal{E}_i \neq \emptyset$, then go to 4.

3. *Compute critical pairs:* If all rules in $\mathcal{R}_i$ are marked, stop with success. Otherwise, select an unmarked rule in $\mathcal{R}_i$, say with label k. Let $\mathcal{E}_{i+1}$ be the set of all critical pairs computed between rule k and any rule of $\mathcal{R}_i$ of label not greater than k. Let $\mathcal{R}_{i+1}$ be the same as $\mathcal{R}_i$, except that rule k is now marked. Increment i by 1 and go to 2.

4. *Reduce equation:* Select equation $M = N$ in $\mathcal{E}_i$. Let $M{\downarrow}$ (resp. $N{\downarrow}$) be an $\mathcal{R}_i$-normal form of M (resp. N) obtained by applying rules of $\mathcal{R}_i$ in any order, until none applies.

5. If $M{\downarrow} = N{\downarrow}$, then let $\mathcal{E}_{i+1} = \mathcal{E}_i - \{M = N\}$, $\mathcal{R}_{i+1} = \mathcal{R}_i$, increment i by 1 and go to 2.

6. If $M{\downarrow} \succ N{\downarrow}$, then let $\lambda = M{\downarrow}$, $\rho = N{\downarrow}$. Otherwise, if $N{\downarrow} \succ M{\downarrow}$, then let $\lambda = N{\downarrow}$, $\rho = M{\downarrow}$, else stop with failure.

7. *Add new rule:* Let K be the set of labels k of rules of $\mathcal{R}_i$ whose left-hand side λ_k is reducible by $\lambda \to \rho$, say to λ'_k. Let $\mathcal{E}_{i+1} = \mathcal{E}_i - \{M = N\} \cup \{\lambda'_k = \rho_k \mid k{:}\lambda_k \to \rho_k \in \mathcal{R}_i \text{ with } k \in K\}$. Increment p by 1. Let $\mathcal{R}_{i+1} = \{j{:}\lambda_j \to \rho'_j \mid j{:}\lambda_j \to \rho_j \in \mathcal{R}_i \text{ with } j \notin K\} \cup \{p{:}\lambda \to \rho\}$, where ρ'_j is a normal form of ρ_j, using rules from $\mathcal{R}_i \cup \{\lambda \to \rho\}$. The rules coming from $\mathcal{R}_i$ are marked or unmarked as they were in $\mathcal{R}_i$, the new rule $\lambda \to \rho$ is unmarked. Increment i by 1 and go to 2. ∎

When given a finite set of equations $\mathcal{E}$ and a reduction ordering $\succ$ on terms, the completion algorithm may stop with success, stop with failure, or loop forever. When it stops with failure, not much interesting may be said, except, as given by lemma 2 below, that every equation or rewrite rule generated so far is an equational consequence of $\mathcal{E}$. Either the algorithm should be tried again with a different ordering that will order the two terms $M{\downarrow}$ and $N{\downarrow}$ which were incomparable; or some new function symbol should be added with a definition in $\mathcal{E}$ that will reduce $M{\downarrow}$ or $N{\downarrow}$ (see [5,7]); or else the method is not applicable at all. This is what happens, for instance, with the original Knuth-Bendix completion algorithm

when a permutative equation (such as commutativity) is generated. In this case a different completion algorithm, for instance operating on congruence classes of terms under the permutative equation, should be used. See [8,9,10,11] for details.

We shall now interest ourselves in the remaining case; i.e., when all pairs of terms $M\downarrow$, $N\downarrow$ considered at step 6 are comparable in the ordering $\succ$. This will be the case, for instance, whenever $\succ$ is a total ordering, and will therefore apply to all the word completion algorithms considered in [4]. We shall show that the algorithm is correct, in that it gives us a semidecision algorithm for the $\mathcal{E}$-equality $=_{\mathcal{E}}$, which is a decision procedure when it terminates. The main difficulty in the proof is that the sets $\mathcal{R}_i$ do not increase monotonically, and so $\rightarrow_{\mathcal{R}_i}$ is not always contained in $\rightarrow_{\mathcal{R}_{i+1}}$. It must therefore be justified to mark the rules as we do above, since some critical pair may be simplified at some iteration i and not simplifiable at some further iteration j.

Let $\mathcal{R} = \bigcup_{i \geq 0} \mathcal{R}_i$, and let $\mathcal{R}_\infty$ be the set of all rules which belong to some $\mathcal{R}_k$ and to all $\mathcal{R}_j$'s with $j > k$; i.e., which are never reduced, neither on the left nor on the right, by other rules. $\mathcal{R}_\infty$ is the "limit" rewriting system constructed by the completion algorithm, and may be infinite. Note that $\mathcal{R}_\infty \subseteq \mathcal{R}$, and that both $\mathcal{R}$ and $\mathcal{R}_\infty$ are nœtherian. If the algorithm stops with success at iteration i, then $\mathcal{R}_\infty$ is $\mathcal{R}_i$.

Lemma 1. *At every iteration i, for all $\lambda \rightarrow \rho$ in $\mathcal{R}_i$, for every $j \geq i$, there exists $\lambda' \rightarrow \rho'$ in $\mathcal{R}_j$ such that λ is reducible by λ'.*

Proof. By construction, either there is in $\mathcal{R}_{i+1}$ a rule with left-hand side λ, or else there is some new rule $\lambda' \rightarrow \rho'$ in $\mathcal{R}_{i+1}$ such that λ' reduces λ. The result follows by a simple induction. ∎

Corollary. *If M is reducible by $\mathcal{R}_i$, it is reducible by every $\mathcal{R}_j$, with $j \geq i$.*

Lemma 2. *For every $i \geq 0$:*

a) $\quad =_{\mathcal{E}_{i+1}} \subseteq =_{(\mathcal{E}_i \cup \mathcal{R}_i \cup \mathcal{R}_{i+1})}$;

b) $\quad =_{\mathcal{R}_{i+1}} \subseteq =_{(\mathcal{E}_i \cup \mathcal{R}_i)}$.

Proof. We check a) and b) by inspection, using the set manipulations in the algorithm, and the fact that if $\langle P, Q \rangle$ is a critical pair of $\mathcal{R}_i$, then for some M we have $M \rightarrow_{\mathcal{R}_i} P$ and $M \rightarrow_{\mathcal{R}_i} Q$. ∎

Corollary. $\quad =_{\mathcal{R}} \subseteq =_{\mathcal{E}}$.

Lemma 3. $\forall i \geq 0 \quad \forall M = N \in \mathcal{E}_i \quad \exists P\,(M \rightarrow_{\mathcal{R}}^* P \;\&\; N \rightarrow_{\mathcal{R}}^* P)$.

Proof. We first remark that if $M = N$ in $\mathcal{E}_i$ is selected at step 4, in iteration j, then $M \rightarrow_{\mathcal{R}}^* M\downarrow$, $N \rightarrow_{\mathcal{R}}^* N\downarrow$, and the new rule $\lambda \rightarrow \rho$ introduced in $\mathcal{R}_{j+1}$ will be such that it will reduce $M\downarrow$ in $N\downarrow$ or $N\downarrow$ in $M\downarrow$. It is therefore enough to show that every element of $\mathcal{E}_i$ will be selected at some iteration. We prove this by showing that the inner loop will always terminate; i.e., for every i, $\mathcal{E}_i \neq \emptyset$ implies $\exists j > i \; \mathcal{E}_j = \emptyset$. First of all, note that there are two ways of going from step 4 to step 2, according to whether $M\downarrow = N\downarrow$ is true or not at step 5. If it is, then $|\mathcal{E}_i| + |\mathcal{R}_i|$ decreases. Otherwise, $|\mathcal{E}_i| + |\mathcal{R}_i|$ is constant. Therefore, we are left to show that we cannot loop indefinitely at step 7. For this, let $n = |\mathcal{E}_i| + |\mathcal{R}_i|$, and consider the $2 \times n$ tuple of terms formed from the left and right-hand sides of pairs of terms in both $\mathcal{E}_i$ and $\mathcal{R}_i$. At every passage at step 7, either the $2 \times n$ tuple decreases in $\succ^{2\times n}$ because some reduction has been effected, or else it is constant, but then $|\mathcal{E}_i|$ decreases (because necessarily $K = \emptyset$ in this case). This completes the proof of termination of the inner loop. ∎

Corollary. *We get* $=_{\mathcal{E}} \subseteq =_{\mathcal{R}}$ *for* $i = 0$, *and therefore* $=_{\mathcal{R}} = =_{\mathcal{E}}$.

Lemma 4. *For every* i, $\mathcal{R}_i$ *is in reduced form; that is, for every* $\lambda \rightarrow \rho$ *in* $\mathcal{R}_i$, ρ *is irreducible by* $\mathcal{R}_i$, *and* λ *is irreducible by* $\mathcal{R}_i - \{\lambda \rightarrow \rho\}$.

Proof. By construction. Actually, for this and the following lemmas, it does not matter whether the reductions of the right-hand sides at step 7 are effected with $\mathcal{R}_i \cup \{\lambda \rightarrow \rho\}$, or whether we use in these reductions rules whose right-hand sides have already been reduced. ∎

Corollary. $\mathcal{R}_\infty$ *is in reduced form.*

In order to prove the next lemma, we need a "fairness" assumption concerning the selection of the rule at step 3. This is to ensure that no rule will be ignored indefinitely by the selection process.

Fairness of Selection Hypothesis.

For every rule label k (i.e., integer such that at some iteration of step 7 we have $p = k$) there is an iteration i such that either the rule of label k is deleted from $\mathcal{R}_i$ (i.e. $k \in K$ at iteration i), or the rule of label k is selected at step 3.

In other words, we require that the critical pair scheduler takes into account the "age" of rules in its selection. This assumption will be met in practice. For instance, if the selection rule is to *select the rule with least complex left-hand side*, and there are only a finite number of terms whose complexity is less or equal

to any given one (e.g., if the complexity is measured by the number of function symbols), the hypothesis above is easily satisfied. Note that no such assumption is needed for the selection of the equation at step 4, since we were able to show the termination of the inner loop.

Lemma 5. *For all terms M, N_1 and N_2 such that $M \to_{\mathcal{R}_\infty} N_1$ and $M \to_{\mathcal{R}_\infty} N_2$, there exist P_1, P_2 and Q such that $M \to_{\mathcal{R}} P_1 \to^*_{\mathcal{R}} N_1$, $M \to_{\mathcal{R}} P_2 \to^*_{\mathcal{R}} N_2$, $P_1 \to^*_{\mathcal{R}} Q$ and $P_2 \to^*_{\mathcal{R}} Q$.*

The statement of lemma 5 is explained diagrammatically in Fig. 1.

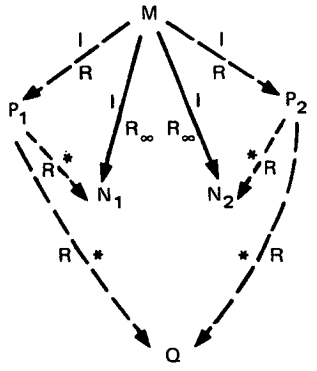

Fig. 1

Proof. This lemma expresses the "lifting" of the critical pairs closure. The proof is by cases on the relative positions of the left-hand sides of the rules $i\colon \lambda_1 \to \rho_1$ and $j\colon \lambda_2 \to \rho_2$ of $\mathcal{R}_\infty$ used to reduce M to respectively N_1 and N_2. When these left-hand sides do not share any function symbol of M, we get easily (see for instance the proof of lemma 2.1 in [2]) that $N_1 \to^*_{\mathcal{R}_\infty} Q$ and $N_2 \to^*_{\mathcal{R}_\infty} Q$ for some Q, yielding the desired result with $P_1 = N_1$ and $P_2 = N_2$. Let us now consider the case where λ_1 and λ_2 do superpose.

We may assume $j \geq i$ without loss of generality. By the fairness hypothesis, and since rule j cannot be deleted by definition of $\mathcal{R}_\infty$, we must select at step 3 the rule of label j in some iteration n, say $j\colon \lambda_2 \to \rho'_2$. By definition of $\mathcal{R}_\infty$ we must have in $\mathcal{R}_n$ some rule labeled i, say $i\colon \lambda_1 \to \rho'_1$. Let u_1 and u_2 be the occurrences in M where we reduce λ_1 and λ_2; i.e., (in the notations of [2]) $N_1 = M[u_1 \leftarrow \sigma_1(\rho_1)]$ and $N_2 = M[u_2 \leftarrow \sigma_2(\rho_2)]$. Now let $P_1 = M[u_1 \leftarrow \sigma_1(\rho'_1)]$ and $P_2 = M[u_2 \leftarrow \sigma_2(\rho'_2)]$. There exists a critical pair $\langle T_1, T_2 \rangle$ between the rules i and j of $\mathcal{R}_n$ such that $P_1/u = \sigma(T_1)$ and $P_2/u = \sigma(T_2)$ for some substitution σ, with u the minimum occurrence in $\{u_1, u_2\}$. We have either $T_1 = T_2$ or $T_2 = T_1$ in $\mathcal{E}_{n+1}$, and by lemma 3 we get $T_1 \to^*_{\mathcal{R}} T$ and $T_2 \to^*_{\mathcal{R}} T$ for some T. Therefore

$P_1 \to^*_{\mathcal{R}} Q$ and $P_2 \to^*_{\mathcal{R}} Q$, with $Q = P_1[u \leftarrow \sigma(T)]$. Finally, by construction we have $\rho'_1 \to^*_{\mathcal{R}} \rho_1$ and $\rho'_2 \to^*_{\mathcal{R}} \rho_2$, from which we get $P_1 \to^*_{\mathcal{R}} N_1$ and $P_2 \to^*_{\mathcal{R}} N_2$. ∎

Definition. We say that term M *contains* term N if and only if some subterm of M is a substitution instance of N. Note that M is reducible by a rule $\lambda \to \rho$ if and only if M contains λ. We now write $M \gg N$ if and only if M contains N and N does not contain M. Since $\gg$ is the composition of two well-founded orderings (strict superterm and strict instance), it is itself well-founded.

We are now ready for the main lemma, which shows the confluence of $\mathcal{R}_\infty$ and $\mathcal{R}$.

Lemma 6. *For every M:*

a) $\forall N \quad M \to^+_{\mathcal{R}} N \quad \Rightarrow \quad \exists P\,(M \to^+_{\mathcal{R}_\infty} P \;\&\; N \to^*_{\mathcal{R}_\infty} P)$;

b) $\forall N_1, N_2 \quad (M \to^*_{\mathcal{R}_\infty} N_1 \;\&\; M \to^*_{\mathcal{R}_\infty} N_2) \Rightarrow \exists P\,(N_1 \to^*_{\mathcal{R}_\infty} P \;\&\; N_2 \to^*_{\mathcal{R}_\infty} P)$;

c) $\forall N_1, N_2 \quad (M \to^*_{\mathcal{R}} N_1 \;\&\; M \to^*_{\mathcal{R}} N_2) \quad \Rightarrow \quad \exists P\,(N_1 \to^*_{\mathcal{R}_\infty} P \;\&\; N_2 \to^*_{\mathcal{R}_\infty} P)$.

Proof. We show simultaneously a), b) and c) by nœtherian induction on M, using the well-founded ordering $\succ$. We assume a), b) and c) for every term M' such that $M \succ M'$.
For a), let $M \to_{\mathcal{R}} M_1 \to^*_{\mathcal{R}} N$, with $k\colon \lambda \to \rho$ the rule of $\mathcal{R}$ used to reduce M in M_1. We use an induction on λ, with well-founded ordering $\gg$. There are two cases.

Case 1. There exists a rule with label k in $\mathcal{R}_\infty$, say $k\colon \lambda \to \rho'$, with $\rho \to^*_{\mathcal{R}} \rho'$. This implies that for some M_2 we have $M \to_{\mathcal{R}_\infty} M_2$ and $M_1 \to^*_{\mathcal{R}} M_2$. By induction hypothesis c) applied to M_1 we get P such that $M_1 \to^*_{\mathcal{R}_\infty} P$ and $N \to^*_{\mathcal{R}_\infty} P$. This case is illustrated by the diagram of Fig. 2.

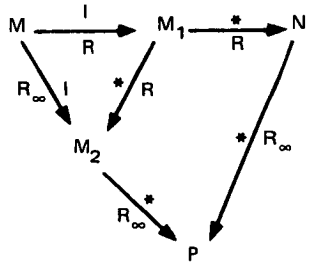

Fig. 2

Case 2. The rule with label k gets reduced on its left-hand side at some iteration i. That is, there is in $\mathcal{R}_i$ some $k\colon \lambda \to \rho'$, with $\rho \to^*_{\mathcal{R}} \rho'$, such that λ is reducible, say into λ', by the newly introduced rule $\lambda'' \to \rho''$. By transitivity of reducibility, M is reducible by $\lambda'' \to \rho''$ into say M'_1, and the reduction $\rho \to^*_{\mathcal{R}} \rho'$ corresponds to a reduction $M_1 \to^*_{\mathcal{R}} M_2$. Now we have $\lambda' = \rho' \in \mathcal{E}_{i+1}$, and by lemma 3 we get $M_2 \to^*_{\mathcal{R}} M_3$ and $M'_1 \to^*_{\mathcal{R}} M_3$ for some M_3. Using induction hypothesis c) at M_1, we get N' such that $M_3 \to^*_{\mathcal{R}_\infty} N'$ and $N \to^*_{\mathcal{R}_\infty} N'$. But since $\lambda \gg \lambda''$ we may apply induction hypothesis a) to the reduction $M \to^1_{\mathcal{R}} M'_1 \to^*_{\mathcal{R}} N'$, which gives us the desired P, according to the diagram in Fig. 3.

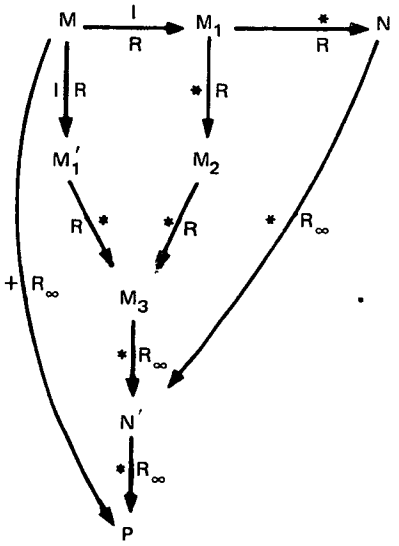

Fig. 3

Let us now show b). If either $M = N$ or $M = P$, it is obvious. Otherwise we have $M \to^1_{\mathcal{R}_\infty} M_1 \to^*_{\mathcal{R}_\infty} N_1$ and $M \to^1_{\mathcal{R}_\infty} M_2 \to^*_{\mathcal{R}_\infty} N_2$. By lemma 5 we get P_1, P_2 and P_3 such that $M \to^1_{\mathcal{R}} P_1$, $M \to^1_{\mathcal{R}} P_2$, $P_1 \to^*_{\mathcal{R}} M_1$, $P_2 \to^*_{\mathcal{R}} M_2$, $P_1 \to^*_{\mathcal{R}} P_3$ and $P_2 \to^*_{\mathcal{R}} P_3$. Using induction hypothesis c) in P_1 (and the fact that $\mathcal{R}_\infty \subseteq \mathcal{R}$) we get P_4 such that $N_1 \to^*_{\mathcal{R}_\infty} P_4$ and $P_3 \to^*_{\mathcal{R}_\infty} P_4$. Similarly, using induction hypothesis c) at P_2 concludes the proof of b), according to the diagram in Fig. 4.

We finally show c) from a) and b). The proof is straightforward, according to the diagram in Fig. 5. ∎

Corollary. *Part b) states the confluence of $\mathcal{R}_\infty$. Part c) implies the confluence of $\mathcal{R}$. From a) and b) we get easily $=_{\mathcal{R}} = =_{\mathcal{R}_\infty}$.*

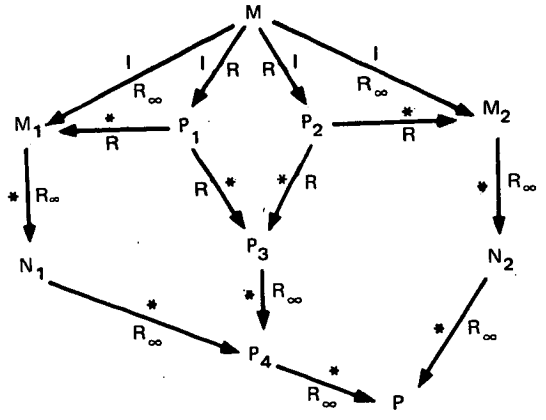

Fig. 4

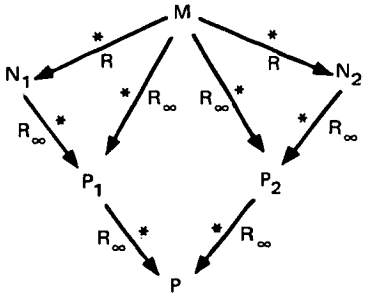

Fig. 5

Lemma 6 shows that the completion algorithm may be used as a semidecision algorithm for $=_{\mathcal{E}}$ as follows. Let M and N be terms for which we want to determine whether $M =_{\mathcal{E}} N$. We construct progressively $\mathcal{R}$-normal forms of M and N as $M \to^*_{\mathcal{R}_1} M_1 \to^*_{\mathcal{R}_2} M_2 \cdots \to^*_{\mathcal{R}_i} M_i$, $N \to^*_{\mathcal{R}_1} N_1 \to^*_{\mathcal{R}_2} N_2 \cdots \to^*_{\mathcal{R}_i} N_i$. If for some i we have $M_i = N_i$, then $M =_{\mathcal{E}} N$. Conversely, if $M =_{\mathcal{E}} N$ there must exist P in $\mathcal{R}_\infty$-normal form such that $M \to^*_{\mathcal{R}_\infty} P$ and $N \to^*_{\mathcal{R}_\infty} P$. If $M_i \neq P$, then M_i is not in $\mathcal{R}_\infty$-normal form, that is, is further reducible in some later iteration. Since all reductions must terminate, we must reach an iteration j at which $M_j = P = N_j$.

Finally, in the cases where the algorithm terminates, we get a decision procedure for $=_{\mathcal{E}}$, by reducing terms to their unique $\mathcal{R}_\infty$-normal form (and now $\mathcal{R}_\infty$ is $\mathcal{R}_k$ for the final iteration k). Let us now show that, conversely, when the algorithm does not terminate, we must necessarily have $\mathcal{R}_\infty$ infinite.

Lemma 7. *If $\mathcal{R}_\infty$ is finite, the completion algorithm terminates.*

Proof. If $\mathcal{R}_\infty$ is finite, then $\mathcal{R}_\infty = \mathcal{R}_k$ for some k. Since no further rule can be added after step k, all equations in $\mathcal{E}_k$ must be normalized by $\mathcal{R}_k$, and the algorithm terminates. ∎

Corollary. *If the completion algorithm does not terminate, the left-hand sides of $\mathcal{R}_\infty$ form an infinite set of terms that are pairwise incomparable in the ordering $\gg$.*

This corollary is useful to prove the termination of the completion algorithm, whenever the term language is such that there are no infinite sets of mutually incomparable elements relatively to the ordering $\gg$. This is the case for the word completion algorithm considered in [1]. Since the algorithm considered is the extension to a commutative-associative operator and since all left-hand sides consist only of multisets of generators, the ordering $\gg$ in this case corresponds to the standard vector ordering on n-tuples of natural numbers. The termination of the completion algorithm, yielding a decision procedure for the uniform word problem in finitely presented abelian monoids, follows therefore directly from the corollary above and Dickson's theorem. Similar methods are used in [4].

Conclusion

We have shown the correctness of a specific version of the Knuth-Bendix completion algorithm, in which critical pairs are computed incrementally. Our proof adapts easily to other versions. For instance, we may not compute all critical pairs of rule k and rules not greater than k at step 3; we could compute at one time just the critical pairs between two rules. This would have the advantage that some pairs may not be computed, since one of these critical pairs may give rise to a new rule that reduces rule k. This would have the drawback, however, that the data structure keeping track of which critical pair have been computed must now be more complicated than just marking rules. The fairness hypothesis must be revised accordingly.

Actually, we might even compute critical pairs one at a time, keeping track now of occurrences in left-hand sides. This might be sensible in implementations where left-hand sides of rules are arranged in a dictionary structure. We shall not consider further these implementation details, but we remark two advantages of this approach. First, a critical pair is computed only when *needed*. This might be a good solution in the cases where the completion algorithm does not terminate, and we want to use it only as a semidecision algorithm. It would then be interesting to drive the choice of the next critical pair we compute from the terms we are trying to prove equal. Second, this would generalize readily

to more complicated completion processes, where there might be an infinite (but recursively enumerable) set of critical pairs between two rules. We conjecture, for instance, that our completion algorithm and its proof could be generalized to show the correctness (as a semidecision algorithm) of the permutative completion algorithm of [9].

Finally, we remark that it should be easy to extend our completion algorithm to the confluence tests that consider congruence classes of terms modulo permutative axioms such as commutativity and associativity [10,11]. In particular, we conjecture that our lemmas could be extended to justify the incremental computation of critical pairs in the Peterson-Stickel completion algorithm described in [11].

References

1. Ballantyne A.M. and Lankford D.S., *New Decision Algorithms for Finitely Presented Commutative Semigroups.* Report MTP-4, Department of Mathematics, Louisiana Tech. U., May 1979.
2. Huet G., *Confluent Reductions: Abstract Properties and Applications to Term Rewriting Systems.* 18th IEEE Symposium on Foundations of Computer Science (1977), 30–45. To appear, JACM, October 1980.
3. Huet G. and Hullot J.M., *Proofs by Induction in Equational Theories with Constructors.* 21st IEEE Symposium on Foundations of Computer Science (1980), to appear.
4. Huet G. and Hullot J.M., *Canonical Form Algorithms for Finitely Presented Algebras.* In preparation.
5. Huet G. and Oppen D., *Equations and Rewrite Rules: a Survey.* "Formal Languages: Perspectives and Open Problems". Ed. Book R., Academic Press, 1980. Also Technical Report CSL-111, SRI International, January 1980.
6. Hullot J.M., *A Catalogue of Canonical Term Rewriting Systems.* Technical Report CSL-113, SRI International, April 1980.
7. Knuth D. and Bendix P., *Simple Word Problems in Universal Algebras.* "Computational Problems in Abstract Algebra". Ed. Leech J., Pergamon Press, 1970, 263–297.
8. Lankford D.S. and Ballantyne A.M., *Decision Procedures for Simple Equational Theories With Commutative Axioms: Complete Sets of Commutative Reductions.* Report ATP-35, Departments of Mathematics and Computer Sciences, U. of Texas at Austin, March 1977.

9. Lankford D.S. and Ballantyne A.M., *Decision Procedures for Simple Equational Theories With Permutative Axioms: Complete Sets of Permutative Reductions.* Report ATP-37, Departments of Mathematics and Computer Sciences, U. of Texas at Austin, April 1977.

10. Lankford D.S. and Ballantyne A.M., *Decision Procedures for Simple Equational Theories With Commutative-Associative Axioms: Complete Sets of Commutative-Associative Reductions.* Report ATP-39, Departments of Mathematics and Computer Sciences, U. of Texas at Austin, Aug. 1977.

11. Peterson G.E. and Stickel M.E., *Complete Sets of Reductions for Equational Theories With Complete Unification Algorithms.* Tech. Report, Dept. of Computer Science, U. of Arizona, Tucson, Sept. 1977.