Completion procedures as Semidecision procedures *

Maria Paola Bonacina **Jieh Hsiang**

Department of Computer Science
SUNY at Stony Brook
Stony Brook, NY 11794-4400, USA
{bonacina,hsiang}@sbcs.sunysb.edu

Abstract

In this paper we give a new abstract framework for the study of Knuth-Bendix type completion procedures, which are regarded as *semidecision procedures* for theorem proving.

First, we extend the classical proof ordering approach started in [6] in such a way that proofs of different theorems can also be compared. This is necessary for the application of proof orderings to theorem proving derivations. We use proof orderings to uniformly define all the fundamental concepts in terms of *proof reduction*.

A completion procedure is given by a set of *inference rules* and a *search plan*. The inference rules determine what can be derived from given data. The search plan chooses at each step of the derivation which inference rule to apply to which data. Each inference step either reduces the proof of a theorem or deletes a *redundant* sentence. Our definition of *redundancy* is based on the assumed proof ordering. We have shown in [16] that our definition subsumes those given in [50, 13].

We prove that if the inference rules are *refutationally complete* and the search plan is *fair*, a completion procedure is a semidecision procedure for theorem proving. The key part of this result is the notion of *fairness*. Our definition of fairness is the first definition of fairness for completion procedures which addresses the theorem proving problem. It is new in three ways: it is *target oriented*, that is it keeps the theorem to be proved into consideration, it is explicitly stated as a property of the search plan and it is defined in terms of proof reduction, so that expansion inferences and contraction inferences are treated uniformly. According to this definition of fairness, it is not necessary to consider all critical pairs in a derivation for the derivation to be fair. This is because not all critical pairs are necessary to prove a given theorem. Considering all critical pairs is an unnecessary source of inefficiency in a theorem proving derivation.

We also show that the process of disproving inductive theorems by the so called *inductionless induction* method is a semidecision process. Finally, we present according to our framework, some equational completion procedures based on Unfailing Knuth-Bendix completion.

*Research supported in part by grants CCR-8805734, INT-8715231 and CCR-8901322, funded by the National Science Foundation. The first author is also supported by Dottorato di ricerca in Informatica, Universitá degli Studi di Milano, Italy.

1 Introduction

The Knuth-Bendix completion procedure [49] computes a possibly infinite confluent rewrite system equivalent to a given set of equations [38]. If a set of equations E and an equation $s \simeq t$ are given, it semidecides whether $s \simeq t$ is a theorem of E, as first remarked in [51, 39]. These results hold if the procedure does not fail on an unoriented equation. Unoriented equations can be handled by adopting the Unfailing Knuth-Bendix method [35, 11], which gives a ground confluent set of equations.

Many completion procedures, related to Knuth-Bendix to different extents, have been designed. They include procedures for equational theories with special sets of axioms [54, 41, 7], Horn logic with equality [50, 27], first order logic [31, 32, 46, 8], first order logic with equality [33, 34, 36, 37, 57, 60, 12, 13], inductive theorem proving in equational and Horn theories [40, 29, 50] and logic programming [20, 21, 22, 15]. Surveys have been given in [24, 25].

A *completion procedure* is composed of *inference rules* and a *search plan*. The inference rules determine what can be derived from given data. The search plan chooses at each step of the derivation which inference rule to apply to which data and therefore it determines the unique derivation that the procedure computes from a given input.

The interpretation of Knuth-Bendix completion as generator of confluent systems is by far the most well known one, whereas theorem proving is basically regarded as a side effect of the generation of a confluent system. This view of completion is not acceptable from the theorem proving perspective, because a procedure which is guaranteed to eventually generate a confluent system cannot be efficient as theorem prover. In this work we reverse the traditional way of presenting completion procedures: we present them as semidecision procedures with the generation of confluent systems as a special side effect.

The interpretation of completion as semidecision procedure appeared first in [39]. Huet proved that if the search plan is fair, the limit of an unfailing Knuth-Bendix derivation is a confluent rewrite system and, as a consequence, if a theorem $s \simeq t$ is given to the procedure, it semidecides the validity of $s \simeq t$. The same result was obtained in a more general framework in [6].

We decouple the interpretation of completion as semidecision procedure from the interpretation of completion as generator of confluent systems. We prove that if the inference rules are *refutationally complete* and the search plan is *fair*, a completion procedure is a *semidecision procedure*. *Refutational completeness* means that for all unsatisfiable inputs, there exist successful derivations by the inference rules of the strategy. *Fairness* means that whenever successful derivations exist, the search plan guarantees that the computed derivation is successful, that is all the inference steps which are necessary to prove the goal are eventually done. In particular, all the critical pairs which are necessary to prove the goal are eventually considered. We give a new definition of fairness to capture this concept.

This notion of fairness is the key difference between completion for theorem proving and completion for the generation of confluent systems. In Huet's landmark paper [39] and in all the following work on completion [6, 9, 57, 13], fairness of a derivation consists in eventually considering all critical pairs. We call this property *uniform fairness* in order to distinguish it from fairness for theorem proving. Uniform fairness is necessary for the limit of a derivation to be confluent, but it is not necessary for theorem proving, because not all the critical pairs are necessary to prove a given theorem. All the definitions of fairness of completion procedures appeared so far in the literature [39, 9, 57, 13] require uniform fairness, because they do not separate theorem proving from the generation of a confluent system.

Therefore, we have proved Huet's classical result from weaker, strictly theorem proving oriented hypotheses, which do not imply any confluence property of the limit of the derivation, since such properties are not necessary for theorem proving. In our view, a completion procedure is first a theorem proving procedure, which also has the property of eventually generating confluent sets if it is uniformly fair.

The so called *inductionless induction* method is covered by the semidecision concept as well: completion for inductionless induction [40] is a *semidecision procedure for disproving inductive theorems.*

We conclude our work by presenting some completion procedures for equational logic: we show that the basic *Unfailing Knuth-Bendix procedure* [35, 11] and some of its extensions, such as the *AC-UKB procedure* [54, 41, 7, 1] with *Cancellation laws* [36], the *S-strategy* [35] and the *Inequality Ordered Saturation strategy* [3] fit nicely in our framework. To our knowledge, this is the first presentation of these extensions of the UKB procedure as sets of inference rules.

Our entire approach to completion procedures is coherently based on a notion of *proof reduction.* In theorem proving, one wants to reduce one single proof, the proof of the target theorem: the derivation halts successfully if the target has been reduced to some trivially true theorem, such as $s \simeq s$, whose proof is *empty.* In traditional completion, one wants to reduce all proofs: for instance, in Knuth-Bendix completion all equational proofs have to be reduced to rewrite proofs. In this proof reduction framework, fairness guarantees that the proof of the target is eventually reduced, whereas uniform fairness guarantees that all proofs are eventually reduced.

In order to formalize all concepts in terms of proof reduction, we need a notion of well founded *proof ordering.* Our starting point is the proof orderings approach originally given in [6, 9]. However, proof orderings as in [6] do not apply to a theorem proving derivation, because they allow to compare only two proofs of the same theorem. In theorem proving, the target is modified by inference steps applying to the theorem itself. Therefore, we give a new notion of proof ordering, where proofs of different theorems can be compared.

In this paper we concentrate on theorem proving. In [16], we compare our study of fairness with previous definitions of fairness in [9, 57, 13]. In [17], we complete our framework with a full generalization of Huet's theorem in [39] and of its extensions in [50, 13] for uniformly fair derivations.

In the following we assume that the reader is familiar with basic concepts and notations about theorem proving, term rewriting systems, completion procedures and orderings. We refer to [25, 26] for basic definitions and notations.

2 Completion procedures for theorem proving

2.1 Proof orderings for theorem proving

Given a finite set of sentences S, we denote by $Th(S)$ the *theory* of S, $Th(S) = \{\varphi | S \models \varphi\}$, and we say that S is a *presentation* of the theory $Th(S)$. The input for a theorem proving procedure is a pair $(S;\varphi)$, where S is a presentation of a theory and φ the *target.* A *theorem proving problem* is to decide whether $\varphi \in Th(S)$ and a *theorem proving derivation* is a sequence of deductions

$$(S_0;\varphi_0) \vdash (S_1;\varphi_1) \vdash \ldots \vdash (S_i;\varphi_i) \vdash \ldots,$$

where at each step the problem of deciding $\varphi_i \in Th(S_i)$ reduces to the problem of deciding $\varphi_{i+1} \in Th(S_{i+1})$. A step $(S_i;\varphi_i) \vdash (S_{i+1};\varphi_i)$, where the presentation is modified, is a *forward reasoning* step. A step $(S_i;\varphi_i) \vdash (S_i;\varphi_{i+1})$, where the target is modified, is a *backward reasoning* step, which derives a new goal from the current one. Informally, the derivation halts successfully at stage k if $\varphi_k \in Th(S_k)$ is trivially true and therefore it can be asserted that $\varphi_0 \in Th(S_0)$. In this section we introduce a notion of *proof ordering*, which allows us to describe a theorem proving derivation as a *proof reduction* process.

We denote proofs by capital Greek letters: $\Upsilon(S,\varphi)$ denotes a proof of φ from axioms in S. Proofs are often represented as trees whose nodes are labeled by sentences: the tree associated to $\Upsilon(S,\varphi)$ has φ as label of the root, elements in S as labels of the leaves and a node ψ has children $\psi_1 \ldots \psi_n$ if ψ is derived from $\psi_1 \ldots \psi_n$ by a step in $\Upsilon(S,\varphi)$. In equational logic, such a proof can also be represented as a chain [6]

$$s_1 \leftrightarrow_{l_1 \simeq r_1} s_2 \leftrightarrow_{l_2 \simeq r_2} \ldots \leftrightarrow_{l_{n-1} \simeq r_{n-1}} s_n,$$

where $s_1 \leftrightarrow_{l_1 \simeq r_1} s_2$ means that the equality of s_1 and s_2 is established by the equation $l_1 \simeq r_1$ because s_1 and s_2 are $c[l_1\sigma]$ and $c[r_1\sigma]$ for some context c and substitution σ. We write $s \rightarrow_{l \simeq r} t$ if $s \succ t$ is known a priori.

An ordering on proofs is defined in general starting from some ordering on the data involved in the proofs. We recall that a *simplification ordering* on terms is a monotonic and stable ordering, i.e. $s \succ t$ implies $c[s\sigma] \succ c[t\sigma]$ for all contexts c and substitutions σ, with the property that a term is greater than any of its subterms. A simplification ordering is well founded. A *complete simplification ordering* is also total on the set of ground terms. Some well known simplification orderings are the recursive path ordering [19], the lexicographic path ordering [45] and the Knuth-Bendix ordering [49]. Such orderings are surveyed in [23]. Our first basic assumption is to have a complete simplification ordering $\succ$ on terms. We prefer to have a simplification ordering, even if a well founded, monotonic and stable ordering total on ground terms is sufficient. Given a complete simplification ordering $\succ$ on terms, it is possible to define complete simplification orderings on equations, clauses and sets of clauses based on $\succ$, as shown for instance in [37].

A *proof ordering* is a monotonic, stable and well founded ordering on proofs [6]. As an example we give the following proof ordering from [27]:

Example 2.1 *A proof ordering to compare two ground equational proofs* $\Upsilon(E, s \simeq t) = s \leftrightarrow^*_E t$

and $\Upsilon'(E', s \simeq t) = s \leftrightarrow^*_{E'} t$, *can be defined as follows. We associate to a ground equational step* $s \leftrightarrow_{l\simeq r} t$ *the triple* (s, l, t), *if* $s \succ t$. *We compare these triples by the lexicographic combination* $>^e$ *of the complete simplification ordering* $\succ$, *the strict encompassment ordering* $\blacktriangleright$ *and again the ordering* $\succ$. *The encompassment ordering* $\trianglerighteq$ *is the composition of the subterm ordering and the subsumption ordering:* $t \trianglerighteq s$ *if* $t|u = s\sigma$ *for some position* u *and substitution* σ; $t \blacktriangleright s$ *if* $t \trianglerighteq s$ *and* $s \neq t$ *[25]. Then we compare two proofs* $\Upsilon(E, s \simeq t)$ *and* $\Upsilon'(E', s \simeq t)$ *by the multiset extension* ${>^e}_{mul}$ *of* $>^e$.

The proof orderings defined in [6] allow us to compare only two proofs $\Upsilon(S, \varphi)$ and $\Upsilon'(S', \varphi)$ of the same theorem φ in different presentations S and S' of a theory. This notion of proof ordering is not suitable for theorem proving, because in a theorem proving derivation

$$(S_0; \varphi_0) \vdash (S_1; \varphi_1) \vdash \ldots \vdash (S_i; \varphi_i) \vdash \ldots$$

both the presentation and the target are transformed. In order to compare the proof of φ_i in S_i and the proof of φ_{i+1} in S_{i+1}, we need a proof ordering such that two proofs $\Upsilon(S, \varphi)$ and $\Upsilon'(S', \varphi')$ may be comparable. Proof orderings with this property do exist and can actually be obtained quite easily. For instance the proof ordering of the previous example can be transformed into a proof ordering for proofs of different theorems as follows:

Example 2.2 *We can compare any two ground equational proofs* $\Upsilon(E, s \simeq t) = s \leftrightarrow^*_E t$ *and* $\Upsilon'(E', s' \simeq t') = s' \leftrightarrow^*_{E'} t'$ *by comparing the pairs* $(\{s, t\}, s \leftrightarrow^*_E t)$ *and* $(\{s', t'\}, s' \leftrightarrow^*_{E'} t')$ *by the lexicographic combination* $>_u$ *of the multiset extension* $\succ_{mul}$ *of the ordering* $\succ$ *on terms and the multiset extension* ${>^e}_{mul}$ *of* $>^e$.

Henceforth a *proof ordering* is a monotonic, stable, well founded ordering on proofs. The minimum proof is the *empty proof.* We denote by *true* the theorem whose proof is empty. For instance in equational logic, *true* is a trivial equality $s \simeq s$. Given a pair $(S; \varphi)$, we can select a minimal proof among all proofs of φ from S:

Definition 2.1 *Given a proof ordering* $>_p$, *we denote by* $\Pi(S, \varphi)$ a minimal proof *of* φ *from* S *with respect to* $>_p$, *i.e. a proof such that for all proofs* $\Upsilon(S, \varphi)$ *of* φ *from* S, $\Upsilon(S, \varphi) \not<_p \Pi(S, \varphi)$.

Having introduced this notion of proof ordering, we can regard a theorem proving derivation

$$(S_0; \varphi_0) \vdash (S_1; \varphi_1) \vdash \ldots \vdash (S_i; \varphi_i) \vdash \ldots,$$

as a process of reducing $\Pi(S_0, \varphi_0)$ to the empty proof and φ_0 to *true.* At each step $\Pi(S_i, \varphi_i)$ is replaced by $\Pi(S_{i+1}, \varphi_{i+1})$ and the derivation halts successfully at stage k if $\Pi(S_k, \varphi_k)$ is empty and φ_k is *true.*

Our generalization of the classical notion of proof orderings is more significant than it may seem at a first glance. Proof orderings were introduced in [6] to prove correctness of the Knuth-Bendix completion procedure as a procedure which generates possibly infinite, confluent term rewriting systems. A derivation by Knuth-Bendix completion in that context is a process of transforming a presentation

$$S_0 \vdash S_1 \vdash \ldots \vdash S_i \vdash \ldots.$$

In other words, it is a purely forward derivation. Since the purpose of such a derivation is to transform a presentation, it is sufficient to be able to compare $\Pi(S_i, \varphi)$ and $\Pi(S_{i+1}, \varphi)$ for any theorem φ in the theory.

This is not the case in theorem proving, since the purpose of a derivation is to prove a specific theorem. Theorem proving requires *backward reasoning*, since a theorem proving problem includes a target. Furthermore, backward reasoning is necessary to obtain a *target-oriented* and therefore presumably efficient procedure. The classical proof orderings approach does not apply to theorem proving because it does not provide for backward reasoning. On the other hand, our proof orderings approach applies to both theorem proving and traditional completion.

2.2 Inference rules and search plans

Since completion procedures are theorem proving strategies with special properties, we start by introducing some basic concepts about theorem proving strategies.

A *theorem proving strategy* is a pair $\mathcal{P} =< I; \Sigma >$, where I is a set of *inference rules* and Σ is a *search plan.* Inference rules in I decide what consequences can be deduced from the available data and Σ decides which inference rule and which data to choose next. We discuss first the inference rules and next the search plan. The general form of an inference rule f is:

$$f\colon \frac{S}{S'}$$

where S and S' are sets of sentences. The rule says that given S, the set S' can be inferred. We distinguish between *expansion* inference rules and *contraction* inference rules, as they are called in [27]. An expansion inference rule expands a given set S into a new set S' by deriving new sentences from sentences in S:

$$f\colon \frac{S}{S'} \text{ where } S \subset S'.$$

A contraction inference rule contracts a given set S into a new set S' by either deleting some sentences in S or replacing them by others:

$$f\colon \frac{S}{S'} \text{ where } S \not\subseteq S'.$$

Different schemes for inference rules, called *deduction* and *deletion*, are given in [13]. We further distinguish between inference rules which transform the presentation and inference rules which transform the target. We assume that targets are clauses and therefore can be regarded as sets of literals:

- *Presentation inference rules*:
  - *Expansion inference rules*: $f\colon \dfrac{(S;\varphi)}{(S';\varphi)}$ where $S \subset S'$.
  - *Contraction inference rules*: $f\colon \dfrac{(S;\varphi)}{(S';\varphi)}$ where $S \not\subseteq S'$.

- *Target inference rules*:
  - *Expansion inference rules*: f: $\frac{(S;\varphi)}{(S;\varphi')}$ where $\varphi \subset \varphi'$.
  - *Contraction inference rules*: f: $\frac{(S;\varphi)}{(S;\varphi')}$ where $\varphi \not\subseteq \varphi'$.

Example 2.3 Deduction *of a critical pair in Unfailing Knuth-Bendix completion is an expansion inference rule on the presentation, since it adds to the given set a new equation:*

$$\frac{(E \cup \{p \simeq q, l \simeq r\}; \hat{s} \simeq \hat{t})}{(E \cup \{p \simeq q, l \simeq r, p[r]_u\sigma \simeq q\sigma\}; \hat{s} \simeq \hat{t})} \quad \begin{array}{ll} p|u \notin X & (p|u)\sigma = l\sigma \\ p\sigma \not\preceq q\sigma, p[r]_u\sigma & \end{array}$$

where X is the set of variables, σ is a most general unifier and $\succ$ is the assumed complete simplification ordering on terms. Simplification *of the target is a contraction inference rule:*

$$\frac{(E \cup \{l \simeq r\}; \hat{s} \simeq \hat{t})}{(E \cup \{l \simeq r\}; \hat{s}[r\sigma]_u \simeq \hat{t})} \quad \begin{array}{l} \hat{s}|u = l\sigma \\ \hat{s} \succ \hat{s}[r\sigma]_u \end{array}$$

The inference rules are required to be sound. A presentation inference rule is *sound* if $Th(S') \subseteq Th(S)$, since an application of a presentation rule involves the presentation only. A target inference rule is *sound* if $Th(S \cup \{\varphi'\}) \subseteq Th(S \cup \{\varphi\})$.

Deduction by presentation rules is deduction of consequences from the axioms or *forward reasoning*. A target rule applies to both the presentation and the target to infer a new target: deduction by target rules is *backward reasoning*. The *Knuth-Bendix procedure* computing critical pairs and simplifying rewrite rules to obtain a canonical system performs exclusively forward reasoning. If a theorem to be proved is given as target, the procedure also performs some backward reasoning, when it simplifies the target.

Finally, a search plan Σ decides which inference rule should be applied to what data at any given step during a derivation. It may set a *precedence* on the inference rules and a *well founded ordering* on data and proceed accordingly:

Example 2.4 *A* Simplification-first *search plan [35] for Unfailing Knuth-Bendix completion is a search plan where Simplification has priority over Deduction. Therefore Deduction is considered only if Simplification does not apply to any equation. Equations can be sorted by the multiset extension $\succ_{mul}$ of the ordering on terms, or by size, or by age such as in a first-in first-out plan.*

2.3 Completion procedures

A completion procedure has then three components $< I_p, I_t; \Sigma >$, where I_p is the set of presentation inference rules, I_t is the set of target inference rules and Σ is the search plan.

A derivation by a completion procedure is a process of *proof reduction*. A target inference step modifies the target and therefore it affects the proof of the target. We require that the proof of the target is reduced:

Definition 2.2 *A target inference step $(S;\varphi) \vdash (S;\varphi')$ is* proof-reducing *if $\Pi(S,\varphi) \geq_p \Pi(S,\varphi')$. It is* strictly proof-reducing *if $\Pi(S,\varphi) >_p \Pi(S,\varphi')$.*

Example 2.5 *Simplification of the target as given in Example 2.3 is strictly proof-reducing. We assume the proof ordering* $>_u$ *introduced in Example 2.2. We have* $\{\hat{s}, \hat{t}\} \succ_{mul} \{\hat{s}', \hat{t}'\}$, *since* $\hat{s} \succ \hat{s}'$ *and* $\hat{t} = \hat{t}'$, *assuming* $\hat{s}$ *is simplified to* $\hat{s}'$. *Therefore* $\Pi(E, \hat{s} \simeq \hat{t}) >_u \Pi(E, \hat{s}' \simeq \hat{t}')$.

For a presentation inference step we allow more flexibility:

Definition 2.3 *Given two pairs* $(S; \varphi)$ *and* $(S'; \varphi')$, *the relation* $(S; \varphi) \rhd_{p,\mathcal{T}} (S'; \varphi')$ *holds if*

1. *either* $\Pi(S, \varphi) >_p \Pi(S', \varphi')$
2. *or*
   (a) $\Pi(S, \varphi) = \Pi(S', \varphi')$,
   (b) $\forall \psi \in \mathcal{T}$, $\Pi(S, \psi) \geq_p \Pi(S', \psi)$ *and*
   (c) $\exists \psi \in \mathcal{T}$ *such that* $\Pi(S, \psi) >_p \Pi(S', \psi)$.

Definition 2.4 *A presentation inference step* $(S; \varphi) \vdash (S'; \varphi)$ *is* proof-reducing *on* $\mathcal{T}$ *if* $(S; \varphi) \rhd_{p,\mathcal{T}} (S'; \varphi)$ *holds. It is* strictly proof-reducing *if* $\Pi(S, \varphi) >_p \Pi(S', \varphi)$.

The condition $(S_i; \varphi_i) \rhd_{p,\mathcal{T}} (S_{i+1}; \varphi_{i+1})$ says that either the step reduces the proof of the target, or it reduces the proof of at least one theorem in $\mathcal{T}$, while it does not increase the proof of any theorem in $\mathcal{T}$. A step which reduces the proof of the target is proof-reducing, regardless of its effects on other theorems. On the other hand, an inference step on the presentation may not immediately decrease the proof of the target and still be necessary to decrease it eventually. Such a step is proof-reducing too, if it does not increase any proof and strictly decreases at least one.

Example 2.6 *Deduction of a critical pair as given in Example 2.3 is proof-reducing. We assume the proof ordering* $>_u$ *introduced in Example 2.2. Given two equations* $l \simeq r$ *and* $p \simeq q$, *a* critical overlap *of* $l \simeq r$ *and* $p \simeq q$ *is any proof* $s \leftarrow_{l \simeq r} v \rightarrow_{p \simeq q} t$, *where* v *is* $c[p\tau]$ *for some context* c *and substitution* τ *and* $(p|u)\tau = l\tau$ *for some non variable subterm* $p|u$ *of* p. *The Deduction rule applied to* $l \simeq r$ *and* $p \simeq q$ *generates the critical pair* $p[r]_u\sigma \simeq q\sigma$, *where* σ *is the mgu of* $p|u$ *and* l *and therefore* $\tau = \sigma\rho$ *for some substitution* ρ. *Such a Deduction step affects a minimal proof by replacing any occurrence of the critical overlap* $s \leftarrow_{l \simeq r} v \rightarrow_{p \simeq q} t$ *by the equational step* $s \leftrightarrow_{p[r]_u\sigma \simeq q\sigma} t$, *justified by the critical pair. We have* $\{(v, l, s), (v, p, t)\} >^e{}_{mul} \{(s, p[r]_u\sigma, t)\}$ *or* $\{(v, l, s), (v, p, t)\} >^e{}_{mul} \{(t, q\sigma, s)\}$, *depending on whether* $s \succ t$ *or* $t \succ s$, *since* $v \succ s, t$. *Therefore* $\Pi(E, \psi) >_u \Pi(E \cup \{p[r]_u\sigma \simeq q\sigma\}, \psi)$ *if the minimal proof of* ψ *in* E *contains a critical overlap between* $l \simeq r$ *and* $p \simeq q$, $\Pi(E, \psi) = \Pi(E \cup \{p[r]_u\sigma \simeq q\sigma\}, \psi)$ *otherwise.*

This notion of proof reduction applies to presentation inference steps which are either expansion steps or contraction steps which replace some sentences by others. A contraction step which deletes sentences without adding any cannot reduce any minimal proof. In order to characterize these steps, we introduce a notion of *redundancy*:

Definition 2.5 *A sentence* φ *is* redundant *in* S *on domain* $\mathcal{T}$ *if* $\forall \psi \in \mathcal{T}$, $\Pi(S, \psi) = \Pi(S \cup \{\varphi\}, \psi)$.

A sentence is redundant in a presentation if adding it to the presentation does not affect any minimal proof.

Example 2.7 *An inference rule of Unfailing Knuth-Bendix completion, which deletes an equation without adding any is* Functional subsumption*:*

$$\frac{(E \cup \{p \simeq q, l \simeq r\}; \hat{s} \simeq \hat{t})}{(E \cup \{l \simeq r\}; \hat{s} \simeq \hat{t})} \quad (p \simeq q) \blacktriangleright (l \simeq r)$$

where $(p \simeq q) \blacktriangleright (l \simeq r)$ *means that* $p = c[l\sigma]$ *and* $q = c[r\sigma]$ *for some context* c *and substitution* σ*, where either* c *is not empty or* σ *is not a renaming of variables. An equation* $p \simeq q$ *subsumed by* $l \simeq r$ *is redundant according to the proof ordering* $>^e{}_{mul}$ *and therefore to the proof ordering* $>_u$ *as defined in Example 2.2. No minimal proof contains a step* $s \leftrightarrow_{p \simeq q} t$ *since the step* $s \leftrightarrow_{l \simeq r} t$ *is smaller: either* $\{(s,p,t)\} >^e{}_{mul} \{(s,l,t)\}$ *or* $\{(t,q,s)\} >^e{}_{mul} \{(t,r,s)\}$*, depending on whether* $s \succ t$ *or* $t \succ s$*, since* $p \blacktriangleright l$ *and* $q \blacktriangleright r$*.*

A notion of redundant clauses was introduced in [57] and in [13], where the term "redundant" was first used. We show in [16] that redundant clauses according to [57] and [13] are also redundant in our sense.

Definition 2.6 *An inference step* $(S; \varphi) \vdash (S'; \varphi')$ *is* reducing *on* $\mathcal{T}$ *if either it is proof-reducing on* $\mathcal{T}$ *or it deletes a sentence which is redundant in* S *on domain* $\mathcal{T}$*.*

Definition 2.7 *An inference rule* f *is* reducing *if all the inference steps* $(S; \varphi) \vdash_f (S'; \varphi')$ *where* f *is applied are reducing.*

We have finally all the elements to define a completion procedure:

Definition 2.8 *A theorem proving strategy* $\mathcal{C} =< I_p, I_t; \Sigma >$ *is a* completion procedure *on domain* $\mathcal{T}$ *if for all pairs* $(S_0; \varphi_0)$*, where* S_0 *is a presentation of a theory and* $\varphi_0 \in \mathcal{T}$*, the derivation*

$$(S_0; \varphi_0) \vdash_{\mathcal{C}} (S_1; \varphi_1) \vdash_{\mathcal{C}} \ldots \vdash_{\mathcal{C}} (S_i; \varphi_i) \vdash_{\mathcal{C}} \ldots$$

has the following properties:

- *monotonicity:* $\forall i \geq 0$, $Th(S_{i+1}) \subseteq Th(S_i)$,
- *relevance:* $\forall i \geq 0$, $\varphi_{i+1} \in Th(S_{i+1})$ *if and only if* $\varphi_i \in Th(S_i)$ *and*
- *reduction:* $\forall i \geq 0$*, the step* $(S_i; \varphi_i) \vdash_{\mathcal{C}} (S_{i+1}; \varphi_{i+1})$ *is reducing on* $\mathcal{T}$*.*

The *domain* $\mathcal{T}$ is the set of sentences where the inference rules of the completion procedure are reducing. For instance, for the *Knuth-Bendix completion procedure* $\mathcal{T}$ is the set of all equations. For the *Unfailing Knuth-Bendix procedure*, $\mathcal{T}$ is the set of all ground equations.

The *monotonicity* and *relevance* properties establish the soundness of the presentation and the target inference rules respectively. Monotonicity ensures that a presentation inference step does not create new elements which are not true in the theory, while relevance ensures that a target inference step replaces the target by a new target in such a way that proving the latter is equivalent to proving the former. For instance, a simplification step which reduces a target φ to φ' satisfies the relevance requirement because if φ' is true, φ is true as well. An interesting expansion inference rule for the target, called *Ordered saturation* will be described in detail in Section 3.3.

Reduction is the property which characterizes completion procedures. Clearly, if all the inference rules of a procedure are reducing, the procedure has the reduction property. We shall see in the second part that the inference rules of the known equational completion procedures are reducing. Most inference rules are reducing because they are suitably restricted by the complete simplification ordering $\succ$ on terms. A complete simplification ordering on data turns out to be a key element in characterizing a theorem proving strategy as a completion procedure.

2.4 Completion procedures as semidecision procedures

Given an input pair $(S_0;\varphi_0)$, a completion procedure works by reducing the proof $\Pi(S_0,\varphi_0)$. If the proof of the target is minimal, the process halts. Since the empty proof is smaller than any other proof, the computation halts at stage k if $\Pi(S_k,\varphi_k)$ is empty and φ_k is *true*.

A procedure is *complete* if, whenever φ_0 is a theorem of S_0, the derivation from $(S_0;\varphi_0)$ reduces φ_0 to *true* and halts. Completeness involves both the inference rules and the search plan. First, it requires that whenever $\varphi_0 \in Th(S_0)$, there exist successful derivations by the inference rules of the procedure. Second, it requires that whenever successful derivations exist, the search plan guarantees that the computed derivation is successful. We call these two properties *refutational completeness* of the inference rules and *fairness* of the search plan respectively.

In order to describe them, we introduce a structure called *I-tree*. Given a theorem proving problem $(S_0;\varphi_0)$ and a set of inference rules I, the application of I to $(S_0;\varphi_0)$ defines a tree, the *I-tree rooted at* $(S_0;\varphi_0)$. The nodes of the tree are labeled by pairs $(S;\varphi)$. The root is labeled by the input pair $(S_0;\varphi_0)$. A node $(S;\varphi)$ has a child $(S';\varphi')$ if $(S';\varphi')$ can be derived from $(S;\varphi)$ in one step by an inference rule in I. The I-tree rooted at $(S_0;\varphi_0)$ represents all the possible derivations by the inference rules in I starting from $(S_0;\varphi_0)$.

A set I of inference rules is *refutationally complete* if whenever $\varphi_0 \in Th(S_0)$, the I-tree rooted at $(S_0;\varphi_0)$ contains successful nodes, nodes of the form $(S;true)$. More precisely, we define completeness as follows:

Definition 2.9 *A set $I = I_p \cup I_t$ of inference rules is* refutationally complete *if whenever $\varphi \in Th(S)$ and $\Pi(S,\varphi)$ is not minimal, there exist derivations*

$$(S;\varphi)\vdash_I(S_1;\varphi_1)\vdash_I\ldots\vdash_I(S';\varphi')$$

such that $\Pi(S,\varphi) >_p \Pi(S',\varphi')$.

A set of inference rules is refutationally complete if it can reduce the proof of the target whenever it is not minimal. Since a proof ordering is well founded, it follows that if $\varphi \in Th(S)$, the I-tree rooted at $(S;\varphi)$ contains successful nodes. The advantage of giving the definition of completeness in terms of proof reduction is that the problem of proving completeness of I is reduced to the problem of exhibiting a suitable proof ordering.

Given a completion procedure $\mathcal{C} =< I_p, I_t; \Sigma >$, $I = I_p \cup I_t$, the I-tree rooted at $(S_0; \varphi_0)$ represents the entire search space that the procedure can potentially derive from the input $(S_0; \varphi_0)$. The search plan Σ selects a path in the I-tree: the derivation from input $(S_0; \varphi_0)$ controlled by Σ is the path selected by Σ in the I-tree rooted at $(S_0; \varphi_0)$. Once both a set of inference rules and a search plan are given, the derivation from $(S_0; \varphi_0)$ is unique. A pair $(S_i; \varphi_i)$ reached at stage i of the derivation is a *visited node* in the I-tree. Each visited node $(S_i; \varphi_i)$ has generally many children, but the search plan selects only one of them to be $(S_{i+1}; \varphi_{i+1})$. A search plan Σ is *fair* if whenever the I-tree rooted at $(S_0; \varphi_0)$ contains successful nodes, the derivation controlled by Σ starting at $(S_0; \varphi_0)$ is guaranteed to reach a successful node. Similar to completeness, we define fairness in terms of proof reduction:

Definition 2.10 *A derivation*

$$(S_0; \varphi_0) \vdash_{\mathcal{C}} (S_1; \varphi_1) \vdash_{\mathcal{C}} \ldots \vdash_{\mathcal{C}} (S_i; \varphi_i) \vdash_{\mathcal{C}} \ldots$$

controlled by a search plan Σ is fair *if and only if for all $i \geq 0$, if there exists a path*

$$(S_i; \varphi_i) \vdash_I \ldots \vdash_I (S'; \varphi')$$

in the I-tree rooted at $(S_0; \varphi_0)$ such that $\Pi(S_i, \varphi_i) >_p \Pi(S', \varphi')$, then there exists an $(S_j; \varphi_j)$ for some $j > i$, such that $\Pi(S', \varphi') \geq_p \Pi(S_j, \varphi_j)$. A search plan Σ is fair *if all the derivations controlled by Σ are fair.*

If the inference rules allow to reduce the proof of the target at $(S_i; \varphi_i)$, a fair search plan guarantees that the proof of the target will be indeed reduced at a later stage $(S_j; \varphi_j)$.

If the inference rules are complete and the search plan is fair, a completion procedure on domain $\mathcal{T}$ is a *semidecision procedure* for $Th(S) \cap \mathcal{T}$ for all presentations S:

Theorem 2.1 *Let $\mathcal{C} =< I_p, I_t; \Sigma >$ be a completion procedure. If the set $I = I_p \cup I_t$ of inference rules is refutationally complete and the search plan Σ is fair, then for all derivations*

$$(S_0; \varphi_0) \vdash_{\mathcal{C}} (S_1; \varphi_1) \vdash_{\mathcal{C}} \ldots \vdash_{\mathcal{C}} (S_i; \varphi_i) \vdash_{\mathcal{C}} \ldots,$$

where $\varphi_0 \in Th(S_0)$, $\forall i \geq 0$, if $\Pi(S_i, \varphi_i)$ is not minimal, then there exists an (S_j, φ_j), for some $j > i$, such that $\Pi(S_i, \varphi_i) >_p \Pi(S_j, \varphi_j)$.

Proof: if $\Pi(S_i, \varphi_i)$ is not minimal, then by completeness of the inference rules, there exists a path $(S_i; \varphi_i) \vdash_I \ldots \vdash_I (S'; \varphi')$ such that $\Pi(S_i, \varphi_i) >_p \Pi(S', \varphi')$. By fairness of the search plan, there exists an $(S_j; \varphi_j)$, for some $j > i$, such that $\Pi(S_i; \varphi_i) >_p \Pi(S', \varphi') \geq_p \Pi(S_j, \varphi_j)$. □

Corollary 2.1 *If a completion procedure $\mathcal{C}$ on domain $\mathcal{T}$ has refutationally complete inference rules and fair search plan, then for all inputs $(S_0; \varphi_0)$, if $\varphi_0 \in Th(S_0)$ then*

- *the derivation $(S_0; \varphi_0) \vdash_{\mathcal{C}} (S_1; \varphi_1) \vdash_{\mathcal{C}} \ldots \vdash_{\mathcal{C}} (S_i; \varphi_i) \vdash_{\mathcal{C}} \ldots$ halts at stage k for some $k \geq 0$ and*
- *$\varphi_k = true$.*

Proof: if $\varphi_0 \in Th(S_0)$, the derivation halts at some stage k by Theorem 2.1 and the well foundedness of $>_p$. Therefore, the proof $\Pi(S_k, \varphi_k)$ is minimal. Since we assume a proof ordering such that any two proofs can be compared, the only minimal proof is the empty proof and φ_k is *true*. □

In the following, we often write that a completion procedure is *complete* as a short hand for a completion procedure with complete inference rules and fair search plan.

2.5 Completion procedures as generators of decision procedures

In this paper we regard a completion procedure as a theorem proving procedure. In [17] we extend our framework to include completion procedures as *generators of decision procedures*.

If the search plan of a completion procedure satisfies a stronger fairness property, which we call *uniform fairness*, the procedure generates a possibly infinite *saturated set*. Uniform fairness is the fairness property which has been required so far for completion procedures [39, 9, 57, 13]. It basically consists in eventually considering all the inference steps. Saturated set is a generalization of confluent system: a set is saturated if no non-trivial consequences can be added [50, 13]. In [17], we define both uniform fairness and saturated set in terms of our notion of redundancy and we show that our definitions are equivalent to those given in [50, 13].

If a presentation is saturated, the derivations from that presentation are *linear input* derivations [18], that is derivations where each inference step applies to the goal to be proved. If linear input derivations from a saturated set are guaranteed to be well founded, a saturated set is a *decision procedure* and the completion procedure is a *generator of decision procedures*.

The well foundedness of the derivations is implied by additional requirements, which depend on the logic. In equational logic, a derivation $s \rightarrow^* \circ \leftarrow^* t$ made only of well founded simplification steps by a confluent rewrite system is a well founded linear input derivation and a confluent system is a decision procedure. Sufficient conditions for well-foundedness of derivations for ground targets in Horn logic with equality are also known [50, 13].

In [17], we give a full generalization of the classical results in [39], which covers also the extensions to Horn logic with equality in [50, 13].

Very few theories have a finite saturated presentation and even fewer satisfy the additional requirement for a saturated presentation to be a decision procedure. Therefore, the interpretation of completion as semidecision procedure which we have developed here is more useful in practice.

3 Completion procedures in equational logic

In the second part of this work we give a new presentation of some Knuth-Bendix type completion procedures for equational logic, in the framework developed so far.

3.1 Unfailing Knuth-Bendix completion

The *Unfailing Knuth-Bendix procedure* [35, 11] is a semidecision procedure for equational theories. A presentation is a set of equations E_0 and a theorem is an equational theorem $\forall \bar{x} s_0 \simeq t_0$. A derivation by UKB has the form

$$(E_0; \hat{s}_0 \simeq \hat{t}_0) \vdash_{UKB} (E_1; \hat{s}_1 \simeq \hat{t}_1) \vdash_{UKB} \ldots (E_i; \hat{s}_i \simeq \hat{t}_i) \vdash_{UKB} \ldots$$

where we denote by $\hat{s}_0 \simeq \hat{t}_0$ an equality which contains only universally quantified variables and therefore can be regarded as a ground equality. A derivation halts at stage k if $\hat{s}_k$ and $\hat{t}_k$ are identical. We assume that $\succ$ is a complete simplification ordering such that $\forall s, true \prec s$ and we extend the encompassment ordering to equations: $(p \simeq q) \trianglerighteq (l \simeq r)$ if $p|u = l\sigma$ and $q|u = r\sigma$, $(p \simeq q) \triangleright (l \simeq r)$ if $(p \simeq q) \trianglerighteq (l \simeq r)$ but $(p \simeq q) \neq (l \simeq r)$. At each step of the completion process the pair $(E_{i+1}; \hat{s}_{i+1} \simeq \hat{t}_{i+1})$ is derived from the pair $(E_i; \hat{s}_i \simeq \hat{t}_i)$ by applying one of the following inference rules:

- *Presentation inference rules*:
  - *Simplification*:
    $$\frac{(E \cup \{p \simeq q, l \simeq r\}; \hat{s} \simeq \hat{t})}{(E \cup \{p[r\sigma]_u \simeq q, l \simeq r\}; \hat{s} \simeq \hat{t})} \quad \begin{array}{ll} p|u = l\sigma & p \succ p[r\sigma]_u \\ p \triangleright l \vee q \succ p[r\sigma]_u & \end{array}$$
  - *Deduction*:
    $$\frac{(E \cup \{p \simeq q, l \simeq r\}; \hat{s} \simeq \hat{t})}{(E \cup \{p \simeq q, l \simeq r, p[r]_u\sigma \simeq q\sigma\}; \hat{s} \simeq \hat{t})} \quad \begin{array}{ll} p|u \notin X & (p|u)\sigma = l\sigma \\ p\sigma \not\preceq q\sigma, p[r]_u\sigma & \end{array}$$
  - *Deletion*:
    $$\frac{(E \cup \{l \simeq l\}; \hat{s} \simeq \hat{t})}{(E; \hat{s} \simeq \hat{t})}$$
  - *Functional subsumption*:
    $$\frac{(E \cup \{p \simeq q, l \simeq r\}; \hat{s} \simeq \hat{t})}{(E \cup \{l \simeq r\}; \hat{s} \simeq \hat{t})} \quad (p \simeq q) \triangleright (l \simeq r)$$
- *Target inference rules*:
  - *Simplification*:
    $$\frac{(E \cup \{l \simeq r\}; \hat{s} \simeq \hat{t})}{(E \cup \{l \simeq r\}; \hat{s}[r\sigma]_u \simeq \hat{t})} \quad \begin{array}{l} \hat{s}|u = l\sigma \\ \hat{s} \succ \hat{s}[r\sigma]_u \end{array}$$
  - *Deletion*:
    $$\frac{(E; \hat{s} \simeq \hat{s}\})}{(E; true)}$$

The main inference rule of UKB is *Simplification*. A simplification step consists in applying an equation in E_i to simplify either another equation in E_i or the goal $\hat{s}_i \neq \hat{t}_i$. The step is performed only if $p \succ p[r\sigma]_u$, that is a term is replaced by a smaller term. The condition $p \triangleright l \vee q \succ p[r\sigma]_u$ is explained as follows. If $q \succ p$, that is simplification applies to the smaller side of $p \simeq q$, the condition $q \succ p[r\sigma]_u$ is trivially satisfied and no other restriction is needed. If $q \not\succ p$, simplification applies to the greater side of $p \simeq q$ or $p \simeq q$ is not ordered. If $p \triangleright l$, either p is a proper instance of l or l matches a proper subterm of p. Otherwise, $p \doteq l$, that is p and l are equal up to variables renaming, but $q \succ p[r\sigma]_u$, that is the newly generated term $p[r\sigma]_u$ is smaller than both sides of the simplified equation $p \simeq q$. We are going to see in a few paragraphs (Lemma 3.1) how these conditions ensures that Simplification is proof-reducing.

The *Deduction* inference rule is the only expansion rule of UKB. It works by checking for a *superposition* between two equations $p \simeq q$ and $l \simeq r$. Two given equations superpose if there exists a non variable subterm, say $p|u$, which unifies with mgu σ with a side l of the other equation $l \simeq r$. This means that the term $p\sigma$ is equal to both $q\sigma$ and $p[r]_u\sigma$. The new equation $p[r]_u\sigma \simeq q\sigma$ is called a *critical pair.* A critical pair is generated only if $p\sigma \not\preceq q\sigma, p[r]_u\sigma$, that is the two equations are applied according to the simplification ordering. The original definitions of Simplification and Deduction given in [35] have slightly different conditions. Simplification requires that $l\sigma \succ r\sigma$ and superposition requires that $p\sigma \not\preceq q\sigma$ and $l\sigma \not\preceq r\sigma$. We adopt here the conditions given in [27], because they put weaker requirements on simplification and stronger requirements on superposition than the original ones. However, these conditions may be more expensive to compute, since they require to perform both substitution application and term replacement.

The *Functional subsumption* inference rule deletes an equation $p \simeq q$ because it is subsumed by another equation $l \simeq r$, that is $p = c[l\sigma]$ and $q = c[r\sigma]$ for some context c and substitution σ, where either c is not empty or σ is not a renaming of variables.

Simplification is the most important among the above inference rules, because it reduces dramatically the number and the size of the generated equations. A search plan for UKB should give to Simplification the highest priority among all the inference rules, so that the target and the presentation are always kept fully simplified. A search plan with this property is called *Simplification-first* [35]. If Simplification is not applied, the Deduction inference rule rapidly saturates the memory space with equations, making impossible to reach a proof in reasonable time.

In order to characterize the UKB procedure as a completion procedure, we define a proof ordering $>_{UKB}$ to compare the proofs $\Pi(E_i, \hat{s}_i \simeq \hat{t}_i)$. We use the ordering $>_u$ introduced in Example 2.2. We recall that we write and equational proof step $s \leftrightarrow_{l\simeq r} t$ meaning that s and t are $c[l\sigma]$ and $c[r\sigma]$ for some context c and substitution σ. We write $s \rightarrow_{l\simeq r} t$ if $s \succ t$ is known a priori. Then $\Pi(E_i, \hat{s}_i \simeq \hat{t}_i) >_{UKB} \Pi(E_j, \hat{s}_j \simeq \hat{t}_j)$ holds if and only if $(\{\hat{s}_i, \hat{t}_i\}, \hat{s}_i \leftrightarrow^*_{E_i} \hat{t}_i) >_u (\{\hat{s}_j, \hat{t}_j\}, \hat{s}_j \leftrightarrow^*_{E_j} \hat{t}_j)$ holds.

Lemma 3.1 *The presentation inference rules of the UKB procedure are reducing.*

Proof: we show that Deduction and Simplification are proof-reducing, Deletion and Functional subsumption delete redundant equations:

- the proof for Deduction was given in Example 2.6.
- A Simplification step where an equation $p \simeq q$ is simplified to $p[r\sigma]_u \simeq q$ by an equation $l \simeq r$, affects a minimal proof by replacing a step $s \leftrightarrow_{p\simeq q} t$ by two steps $s \rightarrow_{l\simeq r} v \leftrightarrow_{p[r\sigma]_u\simeq q} t$.
  - If $t \succ s$, we have $\{(t, q, s)\} >^e{}_{mul} \{(s, l, v), (t, q, v)\}$ since $t \succ s$ and $s \succ v$.
  - If $s \succ t$,
    - if $p \blacktriangleright l$, we have
      - if $t \succ v$, $\{(s, p, t)\} >^e{}_{mul} \{(s, l, v), (t, q, v)\}$ since $p \blacktriangleright l$ and $s \succ t$,

· if $v \succ t$, $\{(s,p,t)\} >^e{}_{mul} \{(s,l,v),(v,q,t)\}$ since $p \blacktriangleright l$ and $s \succ v$;

∗ if $p \doteq l$ and $q \succ p[r\sigma]_u$, $t \succ v$ follows from $q \succ p[r\sigma]_u$ by stability and monotonicity of $\succ$ and we have $\{(s,p,t)\} >^e{}_{mul} \{(s,l,v),(t,q,v)\}$ since $t \succ v$ and $s \succ t$.

- A trivial equation $l \simeq l$ is redundant: no minimal proof contains a step $s \leftrightarrow_{l \simeq l} s$ since the subproof given by the single term s is smaller: $\{(s,l,s)\} >^e{}_{mul} \{\epsilon\}$, where the empty triple ϵ is the proof complexity of s.

- the proof for Functional subsumption was given in Example 2.7. □

Lemma 3.2 *The target inference rules of the UKB procedure are strictly proof-reducing.*

Proof: the proof for Simplification was given already in Example 2.5. For a Deletion step we have $\{\hat{s}_i, \hat{t}_i\} \succ_{mul} \{true\}$, since $true$ is smaller than any term. Therefore $\Pi(E_i, \hat{s}_i \simeq \hat{t}_i) >_{UKB} \Pi(E_i, \hat{s}_{i+1} \simeq \hat{t}_{i+1})$. □

We can then show that UKB is a completion procedure:

Theorem 3.1 *The Unfailing Knuth-Bendix procedure is a completion procedure on the domain $\mathcal{T}$ of all ground equalities.*

Proof: for all equational presentations E_0 and for all ground targets $\hat{s}_0 \simeq \hat{t}_0$ the derivation

$$(E_0; \hat{s}_0 \simeq \hat{t}_0) \vdash_{UKB} (E_1; \hat{s}_1 \simeq \hat{t}_1) \vdash_{UKB} \ldots (E_i; \hat{s}_i \simeq \hat{t}_i) \vdash_{UKB} \ldots$$

has the monotonicity, relevance and reduction properties. Monotonicity and relevance follow by soundness of the inference rules, which is proved among others in [39, 6, 9]. Reduction follows from Lemma 3.1 and Lemma 3.2. □

If a *fair* search plan is provided, the UKB procedure is a semidecision procedure for equational theories:

Theorem 3.2 (Hsiang and Rusinowitch 1987) [35], (Bachmair, Dershowitz and Plaisted 1989) [11] *An equation $\forall \bar{x} s \simeq t$ is a theorem of an equational theory E if and only if the Unfailing Knuth-Bendix procedure derives* true *from* $(E; \hat{s} \simeq \hat{t})$.

3.2 Extensions: AC-UKB and cancellation laws

Many equational problems involve associative and commutative (AC) operators. An AC function f satisfies the equations

$f(f(x,y),z) \simeq f(x,f(y,z))$ *(associativity)* and

$f(x,y) \simeq f(y,x)$ *(commutativity).*

Handling associativity and commutativity as any other equation turns out to be very inefficient, since commutativity may generate a very high number of equations through the Deduction inference rule. Also, many instances of commutativity may not be ordered by the chosen simplification ordering, so that simplification does not apply as often as it is desirable to reduce the size and the number of the equations.

The efficiency of the UKB strategy can be greatly improved if associativity and commutativity are not given in the input, but built in the inference rules. The UKB procedure with associativity and commutativity built in the inference rules is called *AC-UKB* [1]. The basic idea is to replace syntactic identity by equality *modulo AC*. Let AC be a set of associativity and commutativity axioms. Two terms s and t are equal modulo AC, if $s \simeq t$ is a theorem of AC, which we write $s =_{AC} t$. The inference rules of the UKB procedure are modified in such a way that any two terms which are equal modulo AC are regarded as identical.

The first modification is to require that the complete simplification ordering on terms $\succ$ is in some sense "compatible" with replacing identity by equality modulo AC. More precisely, this "compatibility" requirement is a *commutation* property. Given two relations R and S, we say that R *commutes* over S if $S \circ R \subseteq R \circ S$, where $\circ$ is composition of relations. The complete simplification ordering $\succ$ is required to commute over $=_{AC}$: this means that for any two terms s and t, if there is a third term r such that $s =_{AC} r$ and $r \succ t$, there is also a term r' such that $s \succ r'$ and $r' =_{AC} t$. Secondly, matching and unification are replaced by AC-matching and AC-unification. A term s matches a term t *modulo AC* if there is a substitution σ such that $s\sigma =_{AC} t$. Similarly, two terms s and t unify *modulo AC* if there is a substitution σ such that $s\sigma =_{AC} t\sigma$. Finally, the strict encompassment ordering $\blacktriangleright$ is replaced by the ordering $\blacktriangleright_{AC}$, that is $s \blacktriangleright_{AC} t$ if and only if $s \blacktriangleright r$ and $r =_{AC} t$ for some term r.

The set of inference rules of the UKB procedure is therefore modified as follows:

- *Presentation inference rules*:
  - *Simplification*:
    $$\frac{(E \cup \{p \simeq q, l \simeq r\}; \hat{s} \simeq \hat{t})}{(E \cup \{p[r\sigma]_u \simeq q, l \simeq r\}; \hat{s} \simeq \hat{t})} \quad \begin{array}{ll} p|u =_{AC} l\sigma & p \succ p[r\sigma]_u \\ p \blacktriangleright_{AC} l \vee q \succ p[r\sigma]_u & \end{array}$$
  - *Deduction*:
    $$\frac{(E \cup \{p \simeq q, l \simeq r\}; \hat{s} \simeq \hat{t})}{(E \cup \{p \simeq q, l \simeq r, p[r]_u\sigma \simeq q\sigma\}; \hat{s} \simeq \hat{t})} \quad \begin{array}{ll} p|u \notin X & (p|u)\sigma =_{AC} l\sigma \\ p\sigma \not\preceq q\sigma, p[r]_u\sigma & \end{array}$$
  - *Extension*:
    $$\frac{(E \cup \{f(p,q) \simeq r\}; \hat{s} \simeq \hat{t})}{(E \cup \{f(p,q) \simeq r, f(p,q,z) \simeq f(r,z)\}; \hat{s} \simeq \hat{t})} \quad \begin{array}{l} f \ is \ AC \\ f(p,q) \not\preceq r \end{array}$$
  - *Deletion*:
    $$\frac{(E \cup \{l \simeq l\}; \hat{s} \simeq \hat{t})}{(E; \hat{s} \simeq \hat{t})}$$
  - *Functional subsumption*:
    $$\frac{(E \cup \{p \simeq q, l \simeq r\}; \hat{s} \simeq \hat{t})}{(E \cup \{l \simeq r\}; \hat{s} \simeq \hat{t})} \quad (p \simeq q) \blacktriangleright_{AC} (l \simeq r)$$
- *Target inference rules*:
  - *Simplification*:
    $$\frac{(E \cup \{l \simeq r\}; \hat{s} \simeq \hat{t})}{(E \cup \{l \simeq r\}; \hat{s}[r\sigma]_u \simeq \hat{t})} \quad \begin{array}{l} \hat{s}|u =_{AC} l\sigma \\ \hat{s} \succ \hat{s}[r\sigma]_u \end{array}$$

– *Deletion*:
$$\frac{(E; \hat{s} \simeq \hat{s})}{(E; true)}$$

This set of inference rules is obtained from the set of inference rules of the UKB procedure by replacing identity by equality modulo AC as explained above and by adding a new inference rule, called *Extension*. The *Extension* inference rule is a specialized version of the *Deduction* inference rule, designed to compute superpositions of equations in E onto associativity axioms. Namely, if $f(p,q) \simeq r$ is an equation in E, f is AC and $f(p,q) \not\preceq r$, the equation $f(p,q) \simeq r$ trivially superposes onto the associativity axiom $f(f(x,y),z) \simeq f(x,f(y,z))$, yielding the critical pair $f(p,f(q,z)) \simeq f(r,z)$, which we write in *flattened* form as $f(p,q,z) \simeq f(r,z)$. These critical pairs are called *extended rules*. Computing the extended rules is sufficien to ensure completeness of the AC-UKB procedure: no other critical pairs between E and AC need to be computed [54].

The extension of UKB to AC-UKB is feasible because algorithms for AC-matching and AC-unification are available. An algorithm for AC-unification, its application in a completion procedure and the extended rules first appeared in [58, 59, 54]. The correctness of the AC-unification algorithm was proved in [28]. General theoretical frameworks for working with equations modulo a set of axioms A are given in [41] and in [7]. These results are surveyed in [25] and more specifically for unification problems in [44].

The UKB or AC-UKB procedure can be further improved by building in the inference rules for the *cancellation laws*. A function F is *right cancellable* if it satisfies the *right cancellation law*

$$\forall x,y,z \;\; f(x,y) = f(z,y) \;\supset\; x = z$$

The *left cancellation law* is defined symmetrically. Cancellation laws may reduce considerably the size of the equations. They are implemented as inference rules as follows [36]:

Cancellation 1:

$$\frac{(E \cup \{f(p,u) \simeq f(q,v)\}; \hat{s} \simeq \hat{t})}{(E \cup \{f(p,u) \simeq f(q,v), p\sigma \simeq q\sigma\}; \hat{s} \simeq \hat{t})} \;\; u\sigma = v\sigma$$

Cancellation 2:

$$\frac{(E \cup \{f(d_1,d_2) \simeq y\}; \hat{s} \simeq \hat{t})}{(E \cup \{f(d_1,d_2) \simeq y, d_1\sigma \simeq x\}; \hat{s} \simeq \hat{t})} \;\; \begin{array}{ll} y \in V(d_1) & \sigma = \{y \mapsto f(x,d_2)\} \\ y \notin V(d_2) & x \textit{ is a new variable} \end{array}$$

Cancellation 3:

$$\frac{(E \cup \{f(p_1,q_1) \simeq r_1, f(p_2,q_2) \simeq r_2\}; \hat{s} \simeq \hat{t})}{(E \cup \{f(p_1,q_1) \simeq r_1, f(p_2,q_2) \simeq r_2, p_1\sigma \simeq p_2\sigma\}; \hat{s} \simeq \hat{t})} \;\; \begin{array}{l} q_1\sigma = q_2\sigma \\ r_1\sigma = r_2\sigma \end{array}$$

where the function f is right cancellable. In *Cancellation 2*, if the substitution $\sigma = \{y \mapsto f(x,d_2)\}$ is applied to the given equation, it becomes $f(d_1\sigma,d_2) \simeq f(x,d_2)$, since y does not occur in d_2. The cancellation law reduces this equation to $d_1\sigma \simeq x$.

In order to prove that the UKB procedure with the cancellation inference rules is a completion procedure, we need to prove that the Cancellation inference rules are proof-reducing. We adopt the proof ordering $>_{UKBC}$ defined as follows: a ground equational step $s \simeq t$ justified by an equation $l \simeq r$ has complexity measure $(s, l\sigma, l, t)$, if s is $c[l\sigma]$, t is $c[r\sigma]$ and $s \succ t$. Complexity measures are compared by the lexicographic combination $>^{ec}$ of the orderings $\succ$, $\blacktriangleright$, $\blacktriangleright$ and $\succ$. Proofs are compared by the lexicographic combination $>_{uc}$ of the multiset extensions $\succ_{mul}$ and ${>^{ec}}_{mul}$: $\Pi(E_i, \hat{s}_i \simeq \hat{t}_i) >_{UKBC} \Pi(E_j, \hat{s}_j \simeq \hat{t}_j)$ if and only if $(\{\hat{s}_i, \hat{t}_i\}, \hat{s}_i \leftrightarrow^*_{E_i} \hat{t}_i) >_{uc} (\{\hat{s}_j, \hat{t}_j\}, \hat{s}_j \leftrightarrow^*_{E_j} \hat{t}_j)$. The proof of Lemma 3.1 is unaffected if $>_{UKBC}$ replaces $>_{UKB}$.

Lemma 3.3 *The* Cancellation *inference rules are proof-reducing.*

Proof: we show that if $(E_i; \hat{s}_i \simeq \hat{t}_i) \vdash_{UKB} (E_{i+1}; \hat{s}_i \simeq \hat{t}_i)$ is a Cancellation step, then if $\Pi(E_i, \hat{s} \simeq \hat{t}) \neq \Pi(E_{i+1}, \hat{s} \simeq \hat{t})$, that is the inference step affects the proof of $\hat{s} \simeq \hat{t}$, $\Pi(E_i, \hat{s} \simeq \hat{t}) {>^{ec}}_{mul} \Pi(E_{i+1}, \hat{s} \simeq \hat{t})$ holds.

- An application of the rule Cancellation 1 to an equation $f(p, u) \simeq f(q, v)$ affects any minimal proof in E_i which contains a step $s \leftrightarrow t$ such that $s = c[f(p, u)\tau]$, $t = c[f(q, v)\tau]$ and $\tau \trianglerighteq \sigma$, where $\trianglerighteq$ is the subsumption ordering and σ is the mgu such that $u\sigma = v\sigma$ of the application of Cancellation 1. The step $s \leftrightarrow_{f(p,u) \simeq f(q,v)} t$ has complexity $(s, f(p, u)\tau, f(p, u), t)$, if $s \succ t$. In the minimal proofs in E_{i+1} the step $s \leftrightarrow_{f(p,u) \simeq f(q,v)} t$ is replaced by a step $s \leftrightarrow_{p\sigma \simeq q\sigma} t$ justified by the new equation $p\sigma \simeq q\sigma$ generated by the application of Cancellation 1. The step $s \leftrightarrow_{p\sigma \simeq q\sigma} t$ has complexity $(s, p\tau, p\sigma, t)$. Since $f(p, u)\tau \blacktriangleright p\tau$, $\{(s, f(p, u)\tau, f(p, u), t)\} >^{ec} \{(s, p\tau, p\sigma, t)\}$ follows. A symmetric argument applies if $t \succ s$.

- An application of the rule Cancellation 2 to an equation $f(d_1, d_2) \simeq y$ affects any minimal proof in E_i which contains a step $s \leftrightarrow t$ such that $s = c[f(d_1, d_2)\tau]$, $t = c[y\tau]$ and $\tau \trianglerighteq \sigma$, where σ is $\{y \mapsto f(x, d_2)\}$. Since $y \in V(d_1)$, we have $f(d_1, d_2)\tau \succ y\tau$ by the subterm property and therefore $s \succ t$ by monotonicity, so that the step $s \leftrightarrow t$ has complexity $(s, f(d_1, d_2)\tau, f(d_1, d_2), t)$. In the minimal proofs in E_{i+1} the step $s \leftrightarrow t$ is replaced by a step $s \leftrightarrow_{d_1\sigma \simeq x} t$ justified by the new equation $d_1\sigma \simeq x$ generated by the application of Cancellation 2. The step $s \leftrightarrow_{d_1\sigma \simeq x} t$ has complexity $(s, d_1\tau, d_1\sigma, t)$. Since $f(d_1, d_2)\tau \blacktriangleright d_1\tau$, $\{(s, f(d_1, d_2)\tau, f(d_1, d_2), t)\} {>^{ec}}_{mul} \{(s, d_1\tau, d_1\sigma, t)\}$ follows.

- An application of the rule Cancellation 3 to two equations $f(p_1, q_1) \simeq r_1$ and $f(p_2, q_2) \simeq r_2$ affects any minimal proof in E_i which contains a subproof $s \leftrightarrow u \leftrightarrow t$ such that $s = c[f(p_1, q_1)\tau]$, $u = c[r_1\tau]$, $t = c[f(p_2, q_2)\tau]$ and $\tau \trianglerighteq \sigma$, where σ is the mgu such that $q_1\sigma = q_2\sigma$ and $r_1\sigma = r_2\sigma$ of the application of Cancellation 3. It follows that $q_1\tau = q_2\tau$ and $r_1\tau = r_2\tau$ too. The subproof $s \leftrightarrow u \leftrightarrow t$ is replaced in any minimal proof in E_{i+1} by a single step $s \leftrightarrow_{p_1\sigma \simeq p_2\sigma} t$ justified by the new equation $p_1\sigma \simeq p_2\sigma$ generated by the application of Cancellation 3.

  1. If $s \succ t \succ u$, the subproof $s \leftrightarrow u \leftrightarrow t$ has complexity $\{(s, f(p_1, q_1)\tau, f(p_1, q_1), u), (t, f(p_2, q_2)\tau, f(p_2, q_2), u)\}$ and the step $s \leftrightarrow_{p_1\sigma \simeq p_2\sigma} t$ has complexity $(s, p_1\tau, p_1\sigma, t)$. Since $f(p_1, q_1)\tau \blacktriangleright p_1\tau$, the result follows. A symmetric argument applies if $t \succ s \succ u$.
  2. If $s \succ u \succ t$, the subproof $s \leftrightarrow u \leftrightarrow t$ has complexity $\{(s, f(p_1, q_1)\tau, f(p_1, q_1), u), (u, r_1\tau, r_1, t)\}$ and the step $s \leftrightarrow_{p_1\sigma \simeq p_2\sigma} t$ has complexity $(s, p_1\tau, p_1\sigma, t)$. Since $f(p_1, q_1)\tau \blacktriangleright p_1\tau$, the result follows. A symmetric argument applies if $t \succ u \succ s$.

3. If $u \succ s \succ t$, the subproof $s \leftrightarrow u \leftrightarrow t$ has complexity $\{(u, r_1\tau, r_1, s), (u, r_1\tau, r_1, t)\}$ and the step $s \leftrightarrow_{p_1\sigma \simeq p_2\sigma} t$ has complexity $(s, p_1\tau, p_1\sigma, t)$. Since $u \succ s$, the result trivially follows. A symmetric argument applies if $u \succ t \succ s$. □

The UKB or AC-UKB procedure enriched with the inference rules for cancellation is complete [36]. Most of the experimental results reported in [1, 2, 14, 3, 5] are obtained by AC-UKB with the inference rules for cancellation.

3.3 Efficiency of the Unfailing Knuth-Bendix procedure

The UKB procedure is complete, but it is not very efficient in general. The main source of inefficiency is the Deduction inference rule, that is the forward reasoning component of UKB. All the backward reasoning steps are Simplification steps, which are strictly proof-reducing. On the other hand, a Deduction step is guaranteed to reduce the proof of some theorem, but not necessarily the proof of the target. The UKB procedure is inefficient because it generates many critical pairs which do not help in proving the target.

Therefore, our goal is to reduce the number of critical pairs generated or equivalently to perform less forward reasoning and more backward reasoning.

For the forward reasoning part, a possible approach to the problem consists in designing search plans which generate first the critical pairs that are estimated to be likely to reduce the proof of the target. Since the effect of a critical pair on the target cannot be completely determined a priori, such a search plan is based on *heuristical criteria* that measure how useful a critical pair is expected to be with respect to the task of simplifying the goal. Some examples of these heuristics are given in [3, 4].

For the backward reasoning part, we observe that if the target $\hat{s}_i \simeq \hat{t}_i$ is fully simplified with respect to E_i, $\hat{s}_i \simeq \hat{t}_i$ is minimal in the ordering $\succ_{mul}$ among all the ground equations E-equivalent to the input target $s_0 \simeq t_0$, where $E = \bigcup_{0 \leq j \leq i} E_j$. If a Simplification-first plan is adopted, UKB maintains a minimal target. Therefore, it could seem that no improvement can be obtained on the target side. However, we shall see that this is not the case.

The notion of minimal target is relative to the assumed partially ordered set (poset) of targets. If we assume the poset of ground equalities ordered by $\succ_{mul}$, $\hat{s}_i \simeq \hat{t}_i$ is minimal among the ground equations E-equivalent to the input target $s_0 \simeq t_0$. The situation changes if we assume as poset of targets the poset of disjunctions of ground equalities ordered by an ordering $\succ'_{mul}$ defined as follows: $N_1 \succ'_{mul} N_2$ if $min(N_1) \succ_{mul} min(N_2)$, where N_1 and N_2 are disjunctions of ground equalities and $min(N)$ is the smallest equality in N according to $\succ_{mul}$. Since the equalities are ground and the simplification ordering is total on ground, there is a smallest element in a disjunction and this ordering is well defined. Furthermore, the poset of equalities is embedded[1]

[1] Given two posets $\mathcal{P}_1 = (D_1, >_1)$ and $\mathcal{P}_2 = (D_2, >_2)$, an *embedding* $h: \mathcal{P}_1 \rightarrow \mathcal{P}_2$ is an injective function $h: D_1 \rightarrow D_2$ which preserves the ordering: for all $x, y \in D_1$, $x >_1 y$ implies $h(x) >_2 h(y)$. The function which maps a ground equality into the disjunction given by the ground equality itself is clearly an embedding of the poset of ground equalities into the poset of disjunctions of ground equalities, since the smallest element in a disjunction given by a single equality is the equality itself.

in the poset of disjunctions.

We show why the backward reasoning part of UKB is not guaranteed to compute a minimal target if the poset of disjunctions is assumed. Let $(E_i; \hat{s}_i \simeq \hat{t}_i)$ be the current stage in an UKB derivation and $l \simeq r$ be an un-orientable equation in E_i, such that $\hat{s}_i|u = l\sigma$ for some position u and substitution σ, but $\hat{s}_i \prec \hat{s}_i[r\sigma]_u$. In other words, l matches a subterm of $\hat{s}_i$ but Simplification does not apply because $\hat{s}_i$ would not be replaced by a smaller term. However, we assume that the target $\hat{s}_i[r\sigma]_u \simeq \hat{t}_i$ is generated nonetheless and that by simplification it reduces to an equation which is smaller than $\hat{s}_i \simeq \hat{t}_i$, that is $\hat{s}_i[r\sigma]_u \rightarrow^*_{E_i} \hat{s}'$, $\hat{t}_i \rightarrow^*_{E_i} \hat{t}'$ and $\{\hat{s}', \hat{t}'\} \prec_{mul} \{\hat{s}_i, \hat{t}_i\}$. If these conditions hold, we have that the disjunction $\hat{s}_i \simeq \hat{t}_i \vee \hat{s}' \simeq \hat{t}'$ is smaller than the disjunction given by $\hat{s}_i \simeq \hat{t}_i$ alone in the poset of disjunctions defined above. Therefore, if we assume the poset of disjunctions as posets of targets, it is not true that UKB maintains a minimal target.

The intuition behind the choice of considering disjunctions of equalities rather than equalities is that if we consider more than one target equality, we have a greater chance to find a short proof. In order to work on disjunctions of equalities, we need to add to the UKB procedure an expansion inference rule, so that the target is eventually expanded into a disjunction of ground equalities. Such an expansion inference rule must satisfy the relevance requirement, so that proving the validity of any of the equalities in the disjunction is equivalent to prove the input target $s_0 \simeq t_0$. Also, the application of such rule must be restricted, in order to avoid the generation of a high number of target equalities, which may slow down the search for a solution.

This new inference rule is superposition of an un-orientable equation onto a target equality $\hat{s} \simeq \hat{t}$ to generate a new target equality. A newly generated target equality is first simplified as much as possible and then it is kept only if it is smaller than $\hat{s} \simeq \hat{t}$:

Ordered saturation:

$$\frac{(E \cup \{l \simeq r\}; N \cup \{\hat{s} \simeq \hat{t}\})}{(E \cup \{l \simeq r\}; N \cup \{\hat{s} \simeq \hat{t}, \hat{s}' \simeq \hat{t}'\})} \quad \begin{array}{c} \hat{s}|u = l\sigma \qquad \hat{s}[r\sigma]_u \rightarrow^*_E \hat{s}' \qquad \hat{t} \rightarrow^*_E \hat{t}' \\ \{\hat{s}', \hat{t}'\} \prec_{mul} \{\hat{s}, \hat{t}\} \end{array}$$

Ordered saturation applies if $\hat{s} \prec \hat{s}[r\sigma]_u$, since if $\hat{s} \succ \hat{s}[r\sigma]_u$ holds, simplification would apply. The target equality $\hat{s}' \simeq \hat{t}'$ might have a shorter proof than the other target equalities. We do not know which one has the shortest proof. We keep all of them to broaden our chance of reaching the proof as soon as possible.

In addition, we need to modify the *Deletion* inference rule, since the computation halts successfully as soon as an equality in the disjunction is reduced to a trivial equality:

Deletion:

$$\frac{(E; N \cup \{\hat{s} \simeq \hat{s}\})}{(E; true)}$$

The procedure obtained by adding Ordered saturation to UKB and by modifying Deletion as above, is called the *Inequality Ordered-Saturation strategy* (IOS) [3]. A derivation by the IOS strategy has the form

$$(E_0; N_0) \vdash_{IOS} (E_1; N_1) \vdash_{IOS} \ldots \vdash_{IOS} (E_i; N_i) \vdash_{IOS} \ldots$$

where the set N_0 contains the initial goal $\hat{s}_0 \simeq \hat{t}_0$ and at stage i, N_i is the current set of target equalities. The derivation halts at stage k if N_k contains a target $\hat{s}_i \simeq \hat{t}_i$ such that $\hat{s}_i$ and $\hat{t}_i$ are identical and the clause in N_k reduces to *true*.

In order to show that the IOS strategy is a completion procedure, we assume a proof ordering $>_{IOS}$ defined as follows: $\Pi(E;N) >_{IOS} \Pi(E';N')$ if and only if $\Pi(E;min(N)) >_{UKB} \Pi(E';min(N'))$. In other words the proof of a disjunction is represented by the proof of the smallest target in the disjunction.

Lemma 3.4 *The* Ordered saturation *inference rule is proof-reducing.*

Proof: we show that if $(E_i;N_i) \vdash_{IOS} (E_i;N_{i+1})$ is an Ordered saturation step, then $\Pi(E_i,N_i) \geq_{IOS} \Pi(E_i,N_{i+1})$. Since $N_i \subset N_{i+1}$, $min(N_i) \succeq_{mul} min(N_{i+1})$ and the result follows. □

Theorem 3.3 *The Inequality Ordered-Saturation strategy is a completion procedure.*

Proof: it follows from Theorem 3.1 and Lemma 3.4. □

The IOS strategy has been implemented and observed to perform better than the UKB procedure [3]. In practice, few target equalities are kept, so that the overhead of handling them is negligible with respect to the advantage of keeping more than one target.

3.4 The S-strategy

The *S-strategy* [35] is an extension of the UKB procedure to the logic of equality and inequality. A presentation is a set of equations E_0 and a theorem φ is a sentence $\bar{Q}\bar{x}\ s_0 \simeq t_0 \vee \ldots \vee s_n \simeq t_n$, where $\bar{Q}\bar{x}$ is any sequence of quantifier-variable pairs. A theorem φ in this form is transformed into a target $N_0 = s_0 \simeq t_0 \vee \ldots \vee s_n \simeq t_n$, where all variables are implicitly existentially quantified, by replacing all the universally quantified variables by constants and by dropping the quantifiers. If φ is $\forall \bar{x} s_0 \simeq t_0$, N_0 is $\hat{s}_0 \simeq \hat{t}_0$ and the S-strategy reduces to the UKB procedure. A computation has the form

$$(E_0;N_0) \vdash_S (E_1;N_1) \vdash_S \ldots \vdash_S (E_i;N_i) \vdash_S \ldots$$

where $\forall i \geq 0$, E_i is a set of equalities and N_i is a disjunction of target equalities with existentially quantified variables. A derivation halts at stage k if N_k contains a target $s_i \simeq t_i$ whose sides are unifiable. The set of inference rules of UKB is modified as follows:

- *Presentation inference rules*:
  - *Simplification*:

$$\frac{(E \cup \{p \simeq q, l \simeq r\}; N)}{(E \cup \{p[r\sigma]_u \simeq q, l \simeq r\}; N)} \quad \begin{array}{ll} p|u = l\sigma & p \succ p[r\sigma]_u \\ p \mathrel{\dot{\triangleright}} l \vee q \succ p[r\sigma]_u & \end{array}$$

  - *Deduction*:

$$\frac{(E \cup \{p \simeq q, l \simeq r\}; N)}{(E \cup \{p \simeq q, l \simeq r, p[r]_u\sigma \simeq q\sigma\}; N)} \quad \begin{array}{ll} p|u \notin X & (p|u)\sigma = l\sigma \\ p\sigma \not\preceq q\sigma, p[r]_u\sigma & \end{array}$$

- *Deletion*:
$$\frac{(E \cup \{l \simeq l\}; N)}{(E; N)}$$
- *Functional subsumption*:
$$\frac{(E \cup \{p \simeq q, l \simeq r\}; N)}{(E \cup \{l \simeq r\}; N)} \quad (p \simeq q) \blacktriangleright (l \simeq r)$$

- *Target inference rules*:
  - *Simplification*:
$$\frac{(E \cup \{l \simeq r\}; N \cup \{s \simeq t\})}{(E \cup \{l \simeq r\}; N \cup \{s[r\sigma]_u \simeq t\})} \quad \begin{array}{l} s|u = l\sigma \\ s \succ s[r\sigma]_u \end{array}$$
  - *Deduction*:
$$\frac{(E \cup \{l \simeq r\}; N \cup \{s \simeq t\})}{(E \cup \{l \simeq r\}; N \cup \{s \simeq t, s[r]_u\sigma \simeq t\sigma\})} \quad \begin{array}{ll} s|u \notin X & (s|u)\sigma = l\sigma \\ s\sigma \not\preceq s[r]_u\sigma & \end{array}$$
  - *Deletion*:
$$\frac{(E; N \cup \{s \simeq t\})}{(E; true)} \quad s\sigma = t\sigma$$

The *Deduction* inference rule applies to both equalities and inequalities. In the second case no ordering based condition applies to the inequality. The *Deletion* rule for the target is modified because the target contains variables: a contradiction is detected when the two sides of a target equality unify.

In order to characterize the S-strategy as a completion procedure, we define the following ordering: $\Pi(E_i; N_i) >_S \Pi(E_{i+1}; N_{i+1})$ if and only if $\Pi(\hat{E}_i; \hat{s}_i \simeq \hat{t}_i) >_{UKB} \Pi(\hat{E}_{i+1}; \hat{s}_{i+1} \simeq \hat{t}_{i+1})$, where $\forall i \geq 0$, $\hat{E}_i \cup \{\hat{s}_i \neq \hat{t}_i\}$ is the smallest finite unsatisfiable set of ground instances of clauses in $E_i \cup \neg N_i$. We show that this ordering is well defined. First we show how the pair $(\hat{E}_i; \hat{s}_i \simeq \hat{t}_i)$ is defined. N_i is a theorem of E_i if and only if $E_i \cup \neg N_i$ is unsatisfiable, where N_i is a disjunction of equations $s_0 \simeq t_0 \vee \ldots \vee s_n \simeq t_n$ with existentially quantified variables and therefore $\neg N_i$ is a conjunction of inequalities $s_0 \neq t_0 \wedge \ldots \wedge s_n \neq t_n$ with universally quantified variables. By the Herbrand Theorem [18], the set $E_i \cup \neg N_i$ is unsatisfiable if and only if there is a finite subset of ground instances of the clauses in $E_i \cup \neg N_i$ which is unsatisfiable. Since $\neg N_i$ is a set of inequalities with universally quantified variables, an unsatisfiable ground instance of $E_i \cup \neg N_i$ needs to contain just one ground inequality: $\hat{E}_i \cup \{\hat{s}_i \neq \hat{t}_i\}$ is the smallest such set with respect to the ordering $\succ_{mul}$. Since $\succ$ is total on ground terms, there exists a smallest set.

The above definition of the ordering $>_S$ says that the complexity of the proof $\Pi(E_i; N_i)$ is measured by the complexity of the ground proof $\Pi(\hat{E}_i; \hat{s}_i \simeq \hat{t}_i)$ and that the impact of the inference steps on $\Pi(E_i; N_i)$ is measured by the impact of the inference steps on $\Pi(\hat{E}_i; \hat{s}_i \simeq \hat{t}_i)$. This approach is correct if to every inference step on $(E_i; N_i)$ corresponds an inference steps on $(\hat{E}_i; \hat{s}_i \simeq \hat{t}_i)$ and vice versa. In order to prove this, we need the following lemma, which rephrases for the S-strategy the *Paramodulation Lifting Lemma*. We recall that a ground substitution is E-irreducible if it does contain any pair $\{x \mapsto t\}$ such that t can be simplified by an equation in E.

Lemma 3.5 (Peterson 1983) [55], (Hsiang and Rusinowitch 1987) [37] *If σ is a ground, E-irreducible substitution, then for all inference rules f of S-strategy, if $(E\sigma; s\sigma \simeq t\sigma) \vdash_f (E'; s' \simeq t')$, then $(E; s \simeq t) \vdash_f (E''; s'' \simeq t'')$, where E' and $s' \simeq t'$ are instances of E'' and $s'' \simeq t''$ respectively.*

Lemma 3.6 *$(E_i; N_i) \vdash_S (E_{i+1}; N_{i+1})$ if and only if $(\hat{E}_i; \hat{s}_i \simeq \hat{t}_i) \vdash_S (\hat{E}_{i+1}; \hat{s}_{i+1} \simeq \hat{t}_{i+1})$.*

Proof:

$\Rightarrow$) An inference step on $(E_i; N_i)$ is trivially an inference step on $(\hat{E}_i; \hat{s}_i \simeq \hat{t}_i)$, since an inference step on non ground clauses is trivially an inference step on all their instances.

$\Leftarrow$) Since $(\hat{E}_i; \hat{s}_i \simeq \hat{t}_i)$ is minimal, $\hat{E}_i \subseteq E_i\sigma$ and $\hat{s}_i \simeq \hat{t}_i \in N_i\sigma$ for an E_i-irreducible substitution σ. Therefore, by Lemma 3.5, an inference step on $(\hat{E}_i; \hat{s}_i \simeq \hat{t}_i)$ is an inference step on $(E_i; N_i)$. □

We can finally state the following theorem:

Theorem 3.4 *The S-strategy is a completion procedure on the domain $\mathcal{T}$ of all ground equalities.*

Proof: monotonicity and relevance follow from the soundness of the inference rules. By the definition of the ordering $>_S$, the inference rules of S-strategy are proof-reducing if they are proof-reducing on ground proofs with respect to the ordering $>_{UKB}$. This follows from Lemma 3.1 and Lemma 3.2, since target Deduction is just target Simplification if the target is ground. □

If a *fair* search plan is provided, the S-strategy is is a semidecision procedure for theories in the logic of equality and inequality:

Theorem 3.5 (Hsiang and Rusinowitch 1987) [35] *A sentence $\bar{Q}\bar{x}\ s_0 \simeq t_0 \vee \ldots \vee s_n \simeq t_n$ is a theorem of an equational theory E if and only if the S-strategy derives* true *from $(E; s_0 \simeq t_0 \vee \ldots \vee s_n \simeq t_n)$.*

4 Semidecision procedures for disproving inductive theorems

The Knuth-Bendix completion procedure has been also applied to *disprove inductive theorem* in equational theories. This method has been called *inductionless induction*, *proof by consistency* or *proof by the lack of inconsistency* by several authors [53, 30, 40, 52, 42, 29, 47, 48, 10, 43]. Extensions of this method to Horn logic with equality are explored in [50].

First of all, we show that a completion procedure applied to disprove inductive theorems is a *semidecision procedure*. We denote by $G(S)$ the set of all ground terms on the signature of a presentation S and we use $Ran(\sigma)$ to represent the range of a substitution σ, so that a ground substitution is a substitution such that $Ran(\sigma) \subset G(S)$. A clause φ is an *inductive theorem* of S, written $S \models_{Ind} \varphi$, if and only if for all ground substitutions σ, $\varphi\sigma \in Th(S)$. We denote by $Ind(S)$ the set of all the inductive theorems of S, $Ind(S) = \{\varphi |\ S \models_{Ind} \varphi\}$, by $GTh(S)$ the set of all the ground theorems of S, $GTh(S) = \{\varphi |\ \varphi \in Th(S), \varphi\ ground\}$ and by $G(\varphi)$ the set of all the ground instances of φ, $G(\varphi) = \{\varphi\sigma |\ Ran(\sigma) \subset G(S)\}$.

The set $Ind(S)$ is not semidecidable. Even if we have a decision procedure for $G(\varphi) \cap GTh(S)$, we still cannot prove that φ is an inductive theorem, because the set $G(\varphi)$ is infinite. However, the complement problem, that is proving that φ is *not* an inductive theorem of S, is semidecidable in certain theories.

If $\varphi \notin Ind(S)$, then there is a ground instance $\varphi\sigma$ such that $\varphi\sigma \notin GTh(S)$. Therefore $GTh(S \cup \{\varphi\}) \neq GTh(S)$, since $\varphi\sigma \in GTh(S \cup \{\varphi\})$ for all ground instances $\varphi\sigma$. Thus, we can prove that φ is not an inductive theorem of S by proving the following target:

$$\Phi_0 = \exists\sigma\ Ran(\sigma) \subset G(S)\ \exists\psi \in S \cup \{\varphi\}\ \textit{such that}\ \psi\sigma \in GTh(S \cup \{\varphi\}) - GTh(S).$$

If there exists an oracle $\mathcal{O}$ to decide such target, a completion procedure $\mathcal{C} =< I_p, I_t; \Sigma; \mathcal{O} >$ equipped with the oracle $\mathcal{O}$ will be a semidecision procedure for disproving inductive theorems. A derivation has the form

$$(S \cup \{\varphi\}; \Phi_0) \vdash_{\mathcal{C}} (S_1; \Phi_1) \vdash_{\mathcal{C}} \ldots (S_i; \Phi_i) \vdash_{\mathcal{C}} \ldots,$$

where at each step the target is

$$\Phi_i = \exists\sigma\ Ran(\sigma) \subset G(S)\ \exists\psi \in S_i\ \textit{such that}\ \psi\sigma \in GTh(S_i) - GTh(S).$$

No inference step applies to the target: the procedure takes as input the presentation $S \cup \{\varphi\}$ given by the original presentation and the inductive conjecture and it proceeds by applying inference rules to the presentation until it obtains a presentation S_k such that the oracle applied to S_k answers positively and replaces Φ_k by *true*.

In the equational case, an oracle to decide Φ_i is available only under the assumption that the input set of equations E is ground confluent. Under this hypothesis, Φ_i is true if and only if there are two ground E-irreducible terms s and t such that $s_i\sigma \rightarrow_E^* s$, $t_i\sigma \rightarrow_E^* t$ and $s \simeq t \in GTh(E_i)$. Therefore, we can restrict our attention to ground E-irreducible terms.

A first oracle was given in [40] for equational presentations satisfying the *principle of definition*: the signature of E is given by the disjoint union of a set C of *constructors* and a set D of *defined symbols*, such that the set $T(C)$ of all ground constructor terms is *free* and all function symbols in D are *completely defined* on C, that is for all ground term $t \in T(F)$, there exists a unique ground constructor term $t' \in T(C)$ such that $t \leftrightarrow_E^* t'$.

A more general oracle was proposed in [42] for the Knuth-Bendix completion procedure and extended to the UKB procedure in [10]. This test is based on *ground reducibility*: a term t is *ground E-reducible* if for all ground substitutions σ, $t\sigma$ is E-reducible. Ground E-reducibility is decidable [56] only if E is a ground confluent rewrite system. Therefore, the test in [42, 10] applies only if the input presentation E is ground confluent and all its equations can be oriented into rewrite rules.

In order to characterize an inductive theorem proving strategy as a completion procedure, we define the proof of the target Φ_i as follows:

$$\Pi(S_i, \Phi_i) = \Pi(S_i, min\{\psi\sigma |\ \psi \in S_i, \psi\sigma \in GTh(S_i) - GTh(S)\}),$$

that is the proof of the target is the proof of the smallest ground instance of some clause in S_i which is a theorem in S_i but not in S.

In the equational case, a completion procedure which eventually generates a ground confluent set of equations, is able to reduce the proofs of all ground theorems and therefore the proof of the target. However, this is not necessary. Since the proof of the target is the proof of the smallest ground theorem which is not a theorem of the original presentation, we can restrict our attention to a smaller set of ground theorems:

Definition 4.1 (Fribourg 1986) [29] *Given a ground confluent presentation E, a set of substitutions H is* E-inductively complete *if for all ground substitutions ρ, there exist a substitution $\sigma \in H$ and a ground substitution τ such that $\rho \rightarrow^*_E \sigma\tau$.*

We denote by H_E one such set and by $\mathcal{IT}_E$ the domain of all the ground equations which are instances of substitutions in H_E, that is $\mathcal{IT}_E = \{(l \simeq r)\sigma\tau \mid \sigma \in H_E, (l \simeq r)\sigma\tau \text{ is ground}\}$. The proof of the target is the proof of the smallest ground theorem which is not a theorem of the original presentation. This smallest ground theorem is in $\mathcal{IT}_E$ and therefore reducing the proofs of the theorems in $\mathcal{IT}_E$ is sufficient to guarantee that the proof of the target is reduced, as was first proved in [29] for the application of Knuth-Bendix completion to disprove inductive theorems in equational theories:

Theorem 4.1 (Fribourg 1986) [29] *A completion procedure $\mathcal{C} =< I_p, I_t; \Sigma; \mathcal{O} >$ on the domain $\mathcal{IT}_E$, with complete inference rules and fair search plan is a semidecision procedure for the complement of $Ind(E)$ for all equational presentations E, for which the oracle $\mathcal{O}$ is computable.*

As a consequence, the Deduction inference rule of UKB can be restricted in such a way that a superposition between $l \simeq r$ and $p \simeq q$ at position u in $p \simeq q$ is performed only if the set of mgus $\{\sigma | l\sigma = (p|u)\sigma, l \simeq r \in E_i\}$ is E-inductively complete. The position u is called *completely superposable* in [29]. This result requires an algorithm to detect the completely superposable positions. An equivalent characterization is the following: a position u in p is completely superposable if for all ground instances $(p|u)\rho$ there is an equation $l \simeq r$ in E such that $(p|u)\rho = l\sigma$ and $l\sigma \succ r\sigma$. This shows that the problem of detecting completely superposable positions is basically an instance of the ground reducibility problem. However, if the presentation satisfies the principle of definition, a position u is completely superposable if $p|u$ is a term which has a defined symbol at the root and only constructor symbols and variables at the positions below the root. Therefore, the above theorem can be applied in practice to presentations satisfying the principle of definition.

5 Conclusions

We have given a new abstract framework for the study of Knuth-Bendix type completion procedures, which are regarded as *semidecision procedures* for theorem proving.

All the fundamental concepts are uniformly defined in terms of *proof reduction* with respect to a well founded proof ordering. In order to do this, we have given a new, more general notion of proof ordering, such that also proofs of different theorem can be compared.

A completion procedure is given by a set of *inference rules* and a *search plan*. We have emphasized the distinction between these two components throughout our work. This distinction is often overlooked in the literature, where most theorem proving strategies are presented by giving the set of inference rules only, whereas the search plan is what ultimately turns a set of inference rules into a procedure.

If the inference rules are *refutationally complete* and the search plan is *fair*, a completion procedure is a semidecision procedure for theorem proving. The key part of this result is the notion of *fairness*. Our definition of fairness is the first definition of fairness for completion procedures which addresses the theorem proving problem. It is new in three ways: it is *target oriented*, that is it keeps the theorem to be proved into consideration, it is explicitly stated as a property of the search plan and it is defined in terms of proof reduction, so that expansion inferences and contraction inferences are treated uniformly. We have also shown that the process of diproving inductive theorems by the so called *inductionless induction* method is a semidecision process.

In the second part of this work, we have presented some equational completion procedures based on Unfailing Knuth-Bendix completion, which include the AC-UKB procedure with Cancellation laws, the S-strategy and the Inequality Ordered Saturation strategy. These extensions of UKB had not been presented in a unified framework for completion before.

Directions for further research include the study of efficient, fair, but not uniformly fair, search plans and the full extension of this approach to completion procedures for Horn and first order logic with equality.

References

[1] S.Anantharaman and J.Mzali, Unfailing Completion modulo a set of equations, Technical Report, LRI, Université de Paris Sud, 1989.

[2] S.Anantharaman, J.Hsiang and J.Mzali, SbReve2: A Term Rewriting Laboratory with (AC)-Unfailing Completion, in N.Dershowitz (ed.), *Proceedings of the Third International Conference on Rewriting Techniques and Applications*, Chapel Hill, NC, USA, April 1989, Springer Verlag, Lecture Notes in Computer Science 355, 533–537, 1989.

[3] S.Anantharaman, J.Hsiang, Automated Proofs of the Moufang Identities in Alternative Rings, *Journal of Automated Reasoning*, Vol. 6, No. 1, 76–109, 1990.

[4] S.Anantharaman, N.Andrianarivelo, Heuristical Criteria in Refutational Theorem Proving, in A.Miola (ed.), *Proceedings of the Symposium on the Design and Implementation of Systems for Symbolic Computation*, 184–193, Capri, Italy, April 1990.

[5] S.Anantharaman, N.Andrianarivelo, M.P.Bonacina, J.Hsiang, SBR3: A Refutational Prover for Equational Theorems, to appear in M.Okada, S.Kaplan (eds.), *Proceedings of the Second International Workshop on Conditional and Typed Rewriting Systems*, Montreal, Canada, June 1990.

[6] L.Bachmair, N.Dershowitz, J.Hsiang, Orderings for Equational Proofs, in *Proceedings of the First Annual IEEE Symposium on Logic in Computer Science*, 346–357, Cambridge, MA, June 1986.

[7] L.Bachmair, N.Dershowitz, Completion for rewriting modulo a congruence, in P.Lescanne (ed.), *Proceedings of the Second International Conference on Rewriting Techniques and Applications*, Bordeaux, France, May 1987, Springer Verlag, Lecture Notes in Computer Science 256, 192–203, 1987.

[8] L.Bachmair, N.Dershowitz, Inference Rules for Rewrite-Based First-Order Theorem Proving, in *Proceedings of the Second Annual Symposium on Logic in Computer Science*, Ithaca, New York, June 1987.

[9] L.Bachmair, Proofs Methods for Equational Theories, Ph.D. thesis, Department of Computer Science, University of Illinois, Urbana, IL.,1987.

[10] L.Bachmair, Proof by consistency in equational theories, in *Proceedings of the Third Annual IEEE Symposium on Logic in Computer Science*, 228–233, Edinburgh, Scotland, July 1988.

[11] L.Bachmair, N.Dershowitz and D.A.Plaisted, Completion without failure, in H.Ait-Kaci, M.Nivat (eds.), *Resolution of Equations in Algebraic Structures*, Vol. II: Rewriting Techniques, 1–30, Academic Press, New York, 1989.

[12] L.Bachmair, H.Ganzinger, On Restrictions of Ordered Paramodulation with Simplification, in *Proceedings of the Tenth International Conference on Automated Deduction*, Kaiserslautern, Germany, July 1990.

[13] L.Bachmair, H.Ganzinger, Completion of First-Order Clauses with Equality by Strict Superposition, to appear in M.Okada, S.Kaplan (eds.), *Proceedings of the Second International Workshop on Conditional and Typed Rewriting Systems*, Montreal, Canada, June 1990.

[14] M.P.Bonacina, G.Sanna, KBlab: An Equational Theorem Prover for the Macintosh, in N.Dershowitz (ed.), *Proceedings of the Third International Conference on Rewriting Techniques and Applications*, Chapel Hill, NC, USA, April 1989, Springer Verlag, Lecture Notes in Computer Science 355, 548–550, 1989.

[15] M.P.Bonacina, J.Hsiang, Operational and Denotational Semantics of Rewrite Programs, to appear in *Proceedings of the North American Conference on Logic Programming*, Austin, TX, October 1990.

[16] M.P.Bonacina, J.Hsiang, On fairness of completion-based theorem proving strategies, Technical report, Department of Computer Science, SUNY at Stony Brook.

[17] M.P.Bonacina, J.Hsiang, The Knuth-Bendix-Huet theorem and its extensions, in preparation.

[18] C.L.Chang, R.C.Lee, *Symbolic Logic and Mechanical Theorem Proving*, Academic Press, New York, 1973.

[19] N.Dershowitz, Z.Manna, Proving termination with multisets orderings, *Communications of the ACM*, Vol. 22, No. 8, 465–476, August 1979.

[20] N.Dershowitz, N.A.Josephson, Logic Programming by Completion, in *Proceedings of the Second International Conference on Logic Programming*, 313–320, Uppsala, Sweden, 1984.

[21] N.Dershowitz, Computing with Rewrite Systems, *Information and Control*, Vol. 65, 122–157, 1985.

[22] N.Dershowitz, D.A.Plaisted, Logic Programming Cum Applicative Programming, in *Proceedings of the IEEE Symposium on Logic Programming*, 54–66, Boston, MA, 1985.

[23] N.Dershowitz, Termination of Rewriting, *Journal of Symbolic Computation*, Vol. 3, No. 1 & 2, 69–116, February/April 1987.

[24] N.Dershowitz, Completion and its Applications, in *Proceedings of Conference on Resolution of Equations in Algebraic Structures*, Lakeway, Texas, May 1987.

[25] N.Dershowitz, J.-P.Jouannaud, Rewrite Systems, Technical Report 478, LRI, Université de Paris Sud, April 1989 and Chapter 15 of Volume B of *Handbook of Theoretical Computer Science*, North-Holland, 1989.

[26] N.Dershowitz, J.-P.Jouannaud, Notations for Rewriting, Rapport de Recherche, LRI, Université de Paris Sud, January 1990.

[27] N.Dershowitz, A Maximal-Literal Unit Strategy for Horn Clauses, to appear in M.Okada, S.Kaplan (eds.), *Proceedings of the Second International Workshop on Conditional and Typed Rewriting Systems*, Montreal, Canada, June 1990.

[28] F.Fages, Associative-commutative unification, in R.Shostak (ed.), *Proceedings of the Seventh International Conference on Automated Deduction*, Napa Valley, CA, USA, 1984, Springer Verlag, Lecture Notes in Computer Science 170, 1984.

[29] L.Fribourg, A Strong Restriction to the Inductive Completion Procedure, in *Proceedins of the Thirteenth International Conference on Automata Languages and Programming*, Rennes, France, July 1986, Springer Verlag, Lecture Notes in Computer Science 226, 1986.

[30] J.A.Goguen, How to prove algebraic inductive hypotheses without induction, in W.Bibel and R.Kowalski (eds.), *Proceedings of the Fifth International Conference on Automated Deduction*, 356–373, Les Arcs, France, 1980, Springer Verlag, Lecture Notes in Computer Science 87, 1980.

[31] J.Hsiang, N.Dershowitz, Rewrite Methods for Clausal and Nonclausal Theorem Proving, in *Proceedings of the Tenth International Conference on Automata, Languages and Programming*, Barcelona, Spain, July 1983, Springer Verlag, Lecture Notes in Computer Science 154, 1983.

[32] J.Hsiang, Refutational Theorem Proving Using Term Rewriting Systems, *Artificial Intelligence*, Vol. 25, 255–300, 1985.

[33] J.Hsiang, M.Rusinowitch, A New Method for Establishing Refutational Completeness in Theorem Proving, in J.Siekmann (ed.), *Proceedings of the Eighth Conference on Automated Deduction*, Oxford, England, July 1986, Springer Verlag, Lecture Notes in Computer Science 230, 141–152, 1986.

[34] J.Hsiang, Rewrite Method for Theorem Proving in First Order Theories with Equality, *Journal of Symbolic Computation*, Vol. 3, 133–151, 1987.

[35] J.Hsiang, M.Rusinowitch, On word problems in equational theories, in Th.Ottman (ed.), *Proceedings of the Fourteenth International Conference on Automata, Languages and Programming*, Karlsruhe, Germany, July 1987, Springer Verlag, Lecture Notes in Computer Science 267, 54–71, 1987.

[36] J.Hsiang, M.Rusinowitch and K. Sakai, Complete Inference Rules for the Cancellation Laws, in *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milano, Italy, August 1987, 990–992, 1987.

[37] J.Hsiang, M.Rusinowitch, Proving Refutational Completeness of Theorem Proving Strategies: the Transfinite Semantic Tree Method, to appear in *Journal of the ACM*, 1990.

[38] G.Huet, Confluent reductions: abstract properties and applications to term rewriting systems, *Journal of the ACM*, Vol. 27, 797–821, 1980.

[39] G.Huet, A Complete Proof of Correctness of the Knuth-Bendix Completion Algorithm, *Journal of Computer and System Sciences*, Vol. 23, 11–21, 1981.

[40] G.Huet, J.M.Hullot, Proofs by Induction in Equational Theories with Constructors, Journal of Computer and System Sciences, Vol. 25, 239–266, 1982.

[41] J.-P.Jouannaud, C.Kirchner, Completion of a set of rules modulo a set of equations, *SIAM Journal of Computing*, Vol. 15, 1155–1194, November 1986.

[42] J.-P.Jouannaud, E.Kounalis, Proofs by induction in equational theories without constructors, in *Proceedings of the First Annual IEEE Symposium on Logic in Computer Science*, 358–366, Cambridge, MA, June 1986.

[43] J.-P.Jouannaud, E.Kounalis, Automatic proofs by induction in equational theories without constructors, *Information and Computation*, 1989.

[44] J.-P.Jouannaud, C.Kirchner, Solving Equations in Abstract Algebras: A Rule-Based Survey of Unification, Rapport de Recherche, LRI, Université de Paris Sud, November 1989.

[45] S.Kamin, J.-J.Lévy, Two generalizations of the recursive path ordering, Unpublished note, Department of Computer Science, University of Illinois, Urbana, Illinois, February 1980.

[46] D.Kapur and P.Narendran, An equational approach to theorem proving in first order predicate calculus, in *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, 1146–1153, Los Angeles, CA, August 1985.

[47] D.Kapur and D.R.Musser, Proof by consistency, *Artificial Intelligence*, Vol. 31, No. 2, 125–157, February 1987.

[48] D.Kapur, P.Narendran and H.Zhang, Proof by induction using test sets, in J.Siekmann (ed.), *Proceedings of the Eighth Conference on Automated Deduction*, Oxford, England, July 1986, Springer Verlag, Lecture Notes in Computer Science 230, 99–117, 1986.

[49] D.E.Knuth, P.Bendix, Simple Word Problems in Universal Algebras, in J.Leech (ed.), *Proceedings of the Conference on Computational Problems in Abstract Algebras*, Oxford, England, 1967, Pergamon Press, Oxford, 263–298, 1970.

[50] E.Kounalis, M.Rusinowitch, On Word Problems in Horn Theories, in E.Lusk, R.Overbeek (eds.), *Proceedings of the Ninth International Conference on Automated Deduction*, 527–537, Argonne, Illinois, May 1988, Springer Verlag, Lecture Notes in Computer Science 310, 1988.

[51] D.S.Lankford, Canonical inference, Memo ATP-32, Automatic Theorem Proving Project, University of Texas, Austin, TX, May 1975.

[52] D.S.Lankford, A simple explanation of inductionless induction, Technical report MTP-14, Mathematics Department, Louisiana Technical University, Ruston, LA, 1981.

[53] D.Musser, On proving inductive properties of abstract data types, in *Proceedings of the Seventh ACM Symposium on Principles of Programming Languages*, 154–162, Las Vegas, Nevada, 1980.

[54] G.E.Peterson, M.E.Stickel, Complete sets of reductions for some equational theories, *Journal of the ACM*, Vol. 28, No. 2, 233–264, 1981.

[55] G.E.Peterson, A Technique for Establishing Completeness Results in Theorem proving with Equality, *SIAM Journal of Computing*, Vol. 12, No. 1, 82–100, 1983.

[56] D.A.Plaisted, Semantic confluence tests and completion methods, *Information and Control*, Vol. 65, 182–215, 1985.

[57] M.Rusinowitch, Theorem-proving with resolution and superposition: an extension of Knuth and Bendix procedure as a complete set of inference rules, Thèse d'Etat, Université de Nancy, 1987.

[58] M.E.Stickel, Unification Algorithms for Artificial Intelligence Languages, Ph.D. thesis, Carnegie Mellon University, 1976.

[59] M.E.Stickel, A unification algorithm for associative-commutative functions, *Journal of the ACM*, Vol. 28, No. 3, 423–434, 1981.

[60] H.Zhang, D.Kapur, First Order Theorem Proving Using Conditional Rewrite RUles, in E.Lusk, R.Overbeek (eds.), *Proceedings of the Ninth International Conference on Automated Deduction*, 1–20, Argonne, Illinois, May 1988, Springer Verlag, Lecture Notes in Computer Science 310, 1988.