Towards a foundation of completion procedures as semidecision procedures

Maria Paola Bonacina
Department of Computer Science
University of Iowa
Iowa City, IA 52242-1419, USA

Jieh Hsiang
Department of Computer Science
National Taiwan University
Taipei, Taiwan

> "[Imam al-Mamun] has encouraged me to compose a short work on calculating by Completion and Reduction, ..."
>
> (al-Khwārizmī, The Compendious Book on Calculation by *al-jabr* and *al-muqabala*, IX century)

Abstract

Completion procedures, originated from the seminal work of Knuth and Bendix, are well-known as procedures for generating confluent rewrite systems, i.e. decision procedures for equational theories. In this paper we present a new abstract framework for the utilization of completion procedures as semidecision procedures for theorem proving. The key idea in our approach is that a semidecision process should be *target-oriented*, i.e. keep into account the target theorem to be proved. For the *inference rules* of a completion procedure, we present target-oriented schemes of *contraction* inference rules, i.e. inference rules that delete sentences which are *redundant* for proving the target. For the *search plan*, we give a target-oriented, definition of *fairness*, according to which not all critical pairs need to be considered. We prove that our notion of fairness, together with the *refutational completeness* of the inference rules, is sufficient for a completion procedure to be a semidecision procedure. By relaxing the requirement of considering all critical pairs, our target-oriented framework should be more suitable for designing efficient procedures for theorem proving. The generation of decision procedures is included as a special side-effect and all the results of the classical approach to completion are re-obtained in our framework. The application of completion to disprove inductive conjectures, i.e. the so called *inductionless induction* method, is also covered as a semidecision process. Finally, we present according to our framework, some equational completion procedures based on Unfailing Knuth-Bendix completion.

1 Introduction

The Knuth-Bendix completion procedure [51] computes a possibly infinite confluent rewrite system equivalent to a given set of equations [41]. 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 [53, 42]. 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 [38, 12], which produces 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 [56, 44, 11], Horn logic with equality [52, 29, 30], first order logic [34, 35, 48, 7], first order logic with equality [36, 37, 39, 40, 59, 62, 13, 15], inductive theorem proving in equational and Horn theories [43, 32, 45, 52] and logic programming [22, 23, 26, 17]. Surveys have been given in [25, 27].

Completion procedures have usually been regarded as procedures for generating confluent rewrite systems, which are by themselves decision procedures for the input theories. This view imposes serious and unnecessary limitation on the applicability of completion procedures, since few theories are decidable. In this paper we propose a different perspective, which treats completion procedures as theorem proving methods, that is, as semidecision procedures. From an operational point of view, they are used for proving individual target theorems rather than generating decision procedures.

1.1 Overview

The interpretation of completion as semidecision procedure appeared first in [42], where it was proved that if the procedure is fair, the limit of an unfailed Knuth-Bendix derivation is a confluent rewrite system. As a side-effect, if a theorem $s \simeq t$ is given to the procedure, it semidecides the validity of $s \simeq t$. The same theorem was obtained in a more general framework in [6]. This view of theorem proving as a side-effect is not satisfactory from a theorem proving perspective. Intuitively, in a theorem proving application one only wants to concentrate on deducing consequences which may contribute to a proof of the *target* theorem. Instead, a procedure which is guaranteed to generate eventually a confluent system must take into account all critical pairs which may lead to establishing the equivalence of any arbitrary theorems in the theory. Our motivation is to make possible the design of completion-based theorem proving strategies, which gain efficiency by generating fewer critical pairs. Thus, we reverse the traditional approach to completion procedures: we regard them as *semidecision procedures* with the generation of confluent systems as a potential side-effect.

The key idea in our approach is to consider a theorem proving derivation as a process of *target-oriented proof reduction.* Given a *target theorem* φ and a *presentation* of a theory, i.e. a set of axioms S, the process of proving φ from S can be characterized as a reduction, with respect to a *well-founded ordering*, of a proof of φ in S. Success is reached when the proof is *empty.* The intuition that proving a given theorem requires in most cases less work than generating a confluent system can now be formulated in terms of proof reduction: reducing one proof is conceivably a smaller task than reducing all the proofs. Therefore, a target-oriented completion procedure, i.e. a strategy which focuses on reducing the proof of the given target, should be more efficient as a theorem prover than a procedure which works blindly to reduce all the proofs. We investigate how a procedure can be made target-oriented both at the level of the *inference rules* and at the level of the *search plan* of a completion procedure.

The inference rules determine what can be derived from given data. They can be classified as *expansion* inference rules, that derive new consequences, and *contraction* inference rules, that delete or replace existing data. Expansion rules are generally sufficient for completeness, but contraction rules are critical for efficiency. Therefore we concentrate on contraction. We characterize the contraction steps in terms of target-oriented proof reduction and target-oriented *redundancy*. Then we define *refutational completeness* of the inference rules in terms of proof reduction: for all unsatisfiable inputs, there exist derivations which reduce a proof of the target to the empty proof.

The search plan chooses at each step of the derivation which inference rule to apply to which data. Therefore it determines the unique derivation that the procedure computes from a given input. A search plan is required to be *fair*. Our notion of fairness is radically different from the previous ones. In [42] and in all the following work on completion [6, 14, 59, 15], 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. In fact, the requirement of uniform fairness clashes with the goal of having an efficient search plan for theorem proving. For instance, we may have a problem where the target $s \simeq t$ is an equation on a signature F_1 and the input presentation E is the union of a set E_1 of equations on the signature F_1 and a set E_2 of equations on another signature F_2, disjoint from F_1. Such a phenomenon often occur in definitions of abstract data types, where the signature F_1 contains the constructors and a set of defined symbols, whereas the signature F_2 is another set of defined symbols. Intuitively, a derivation where no inference from E_2 is performed is fair. On the other hand, uniform fairness requires to compute critical pairs from the equations in E_2 as well.

Clearly, theorem proving would benefit from a definition of fairness which is weaker than uniform fairness. We provide such a new definition of fairness by using target-oriented proof reduction as for refutational completeness. Fairness means that whenever successful derivations exist, the search plan must ensure that the computed derivation is successful. We prove that if the inference rules are complete and the search plan is fair according to our definitions, the procedure is a *semidecision procedure*. By showing that fairness is sufficient for theorem proving, we prove the classical result in [42] from weaker, strictly theorem proving oriented hypotheses. No confluence property of the limit of the derivation is implied, since such properties are not necessary for theorem proving. The interpretation of completion procedures as generators of decision procedures, e.g. confluent systems, is also covered in our framework in the special case where the search plan is uniformly fair.

The paper is organized as follows. In Section 2 we give the basic definitions; we introduce expansion and contraction inference rules and search plans and we define proof orderings for theorem proving. Section 3 is devoted to target-oriented proof reduction and redundancy. The definition of completion procedure summarizes the concepts introduced so far. Section 4 contains the notions of refutational completeness, fairness, uniform fairness and the theorem showing that fairness, rather than uniform fairness, is sufficient for semidecision procedures. In Section 4.2, we consider the generation of decision procedures by uniformly fair derivations. In Section 5, we present some completion procedures for equational logic: we show that the basic *Unfailing Knuth-Bendix procedure* [38, 12] and some of its extensions, such as the *AC-UKB procedure* [56, 44, 11, 2] with *Cancellation laws* [39], the *S-strategy* [38] 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. In the last technical section we show how the so called *inductionless induction* method is covered by the semidecision concept as well: completion for inductionless induction [43] is a semidecision procedure for *disproving* inductive theorems. We conclude with some discussion and directions for future research.

Acknowledgements

The bulk of the work reported in this paper was done while both authors were with the Department of Computer Science of the State University of New York at Stony Brook. In addition to the support by the university and the department, research was also sponsored in part by grants CCR-8805734, INT-8715231 and CCR-8901322, funded by the National Science Foundation. The first author was also supported by Dottorato di ricerca in Informatica, Universitá degli Studi di Milano, Italy. This paper is a revised and significantly extended version of "Completion procedures as Semidecision procedures", in M.Okada and S.Kaplan (eds.), *Proceedings of the* 2^{nd} *CTRS Workshop*, held in Montréal, in June 1990, and available in Springer Verlag, Lecture Notes in Computer Science Vol. 516, pages 206–232, and "On fairness of completion-based theorem proving strategies", in R.V.Book (ed.), *Proceedings of the* 4^{th} *RTA Conference*, held in Como, Italy, in April 1991, and available in Springer Verlag, Lecture Notes in Computer Science Vol. 488, pages 348–360.

2 Preliminaries

In this section we present all the preliminary material for the construction of our framework. Section 2.1 recalls basic definitions in term rewriting systems and completion procedures with the notations of [27, 28]. Sections 2.2 and 2.3 covers inference rules, search plans, derivations and proof orderings for theorem proving. The main contributions of this section are the distinction between expansion and contraction inference rules, the role of the search plan, the notion of derivation with a target and the application of proof orderings to derivations with target.

2.1 Basic definitions

Given a finite set F of constant symbols and function symbols with their arities and a denumerable set X of variable symbols, $T(F, X)$ is the set of *terms* on F and X and $T(F)$ is the set of *ground* terms, i.e. without variables. $V(t)$ denotes the set of variables occurring in the term t. A term s is a *subterm* of a term t if s occurs in t. Subterm positions in a term are indicated by strings of natural numbers: the empty string λ denotes the root position, i.e. $t|\lambda = t$, and the string $i \cdot u$ denotes the position u in the i-th subterm of t, i.e. $f(t_1 \ldots t_n)|i \cdot u = t_i|u$. We write $t = c[s]$ to indicate that s is a subterm of t in the *context* c, $s = t|u$ to specify that s is the subterm of t at position u and $t[r]_u$ represents the term obtained by replacing $t|u$ by r.

A *substitution* σ is a set $\{x_1 \mapsto s_1 \ldots x_n \mapsto s_n\}$ such that

- $\forall i, j,\ i \neq j$ implies $x_i \neq x_j$ and
- $\forall i, j,\ x_i \notin V(s_j)$.

The *domain* and *range* of a substitution σ are the sets $Dom(\sigma) = \{x_1 \ldots x_n\}$ and $Ran(\sigma) = \bigcup_{j=1}^{n} V(s_j)$. A substitution σ is *ground* if $Ran(\sigma) = \emptyset$. A substitution σ applies to a term t as follows:

- $t\sigma = s$ if $t = x$ and $x \mapsto s \in \sigma$,
- $t\sigma = t$ if $t = x$ and $x \notin Dom(\sigma)$ or t is a constant and
- $t\sigma = f(t_1\sigma \ldots t_n\sigma)$, if $t = f(t_1 \ldots t_n)$.

Given two substitutions $\sigma = \{x_1 \mapsto s_1 \ldots x_n \mapsto s_n\}$ and $\rho = \{y_1 \mapsto r_1 \ldots y_m \mapsto r_m\}$ such that $Dom(\sigma) \cap Ran(\rho) = \emptyset$, their *composition* is the substitution $\sigma\rho = \{x_1 \mapsto s_1\rho \ldots x_n \mapsto s_n\rho\} \cup \{y_j \mapsto r_j | y_j \mapsto r_j \in \rho, y_j \notin Dom(\sigma)\}$. A term t is an *instance* of a term s if $t = s\sigma$ for some substitution σ.

An *ordering* $\succ$ is a transitive and irreflexive binary relation. An ordering is *total* if for every two distinct elements s and t in the ordered set, either $s \succ t$ or $t \succ s$; it is *partial* otherwise. An ordering is *well-founded* if there is no infinite chain $s_1 \succ s_2 \succ \ldots s_n \succ \ldots$. The basic orderings on terms are the *subterm ordering* $\trianglerighteq$, where $t \trianglerighteq s$ if $t = c[s]$, the *subsumption ordering* $\mathrel{\underline{\gtrdot}}$, where $t \mathrel{\underline{\gtrdot}} s$ if t is an instance of s, $t = s\sigma$, and the *encompassment ordering* $\mathrel{\underline{\rhd\!\!\!\cdot}}$, which is the composition of the subterm ordering and the subsumption ordering: $t \mathrel{\underline{\rhd\!\!\!\cdot}} s$ if $t = c[s\sigma]$. If $t \mathrel{\underline{\gtrdot}} s$ and $s \mathrel{\underline{\gtrdot}} t$, we say that s and t are *variants* or *equal up to a renaming of variables*, expressed as $s \stackrel{\bullet}{=} t$. We write $t \gtrdot s$, if $t \mathrel{\underline{\gtrdot}} s$ and $t \stackrel{\bullet}{\neq} s$, and $t \mathrel{\rhd\!\!\!\cdot} s$, if $t \mathrel{\underline{\rhd\!\!\!\cdot}} s$ and $s \stackrel{\bullet}{\neq} t$. They are called the *proper subsumption ordering* and the *encompassment ordering* respectively.

The subsumption ordering is extended to substitutions: $\sigma \mathrel{\underline{\gtrdot}} \theta$ if $\forall x \in Dom(\sigma), x\sigma = x\theta\rho$ for some substitution ρ. A substitution σ is a *unifier* of two terms s and t if $s\sigma = t\sigma$; it is a *most general unifier* (mgu) of s and t if σ is a unifier of s and t and for all unifiers ρ of s and t, $\rho \mathrel{\underline{\gtrdot}} \sigma$.

The following properties of orderings on terms are often needed:

- *monotonicity:* $s \succ t$ implies $c[s] \succ c[t]$ for all contexts c,
- *stability*: $s \succ t$ implies $s\sigma \succ t\sigma$ for all substitutions σ and
- *subterm property*: $c[s] \succ s$ for all terms s and contexts c.

A monotonic, stable and well-founded ordering is a *reduction ordering.* A monotonic and stable ordering with the subterm property is a *simplification ordering.* A simplification ordering is well-founded [21]. A simplification ordering which is total on the set of ground terms is called a *complete simplification ordering.* Some well-known simplification orderings are the *recursive path ordering* [20], the *lexicographic path ordering* [47] and the *Knuth-Bendix ordering* [51]. We refer to [24] for a survey of orderings and we recall here two techniques for constructing orderings, which will be used later. The *lexicographic extension* of given orderings $\succ_1 \ldots \succ_n$ is the ordering $\succ_{lex}$

such that $(t_1 \ldots t_n) \succ_{lex} (s_1 \ldots s_n)$ if and only if there exists an i, $1 \leq i \leq n$, such that $t_j = s_j$, $\forall j < i$ and $t_i \succ_i s_i$. The *multiset extension* of a given ordering $\succ$ is the ordering $\succ_{mul}$ on multisets of terms such that:

- $\{a\} \cup M \succ_{mul} \emptyset$, where $\emptyset$ is the empty multiset.
- $\{a\} \cup M \succ_{mul} \{a\} \cup N$ if $M \succ_{mul} N$.
- $\{a\} \cup M \succ_{mul} \{b\} \cup N$ if $a \succ b$ and $\{a\} \cup M \succ_{mul} N$.

The lexicographic and multiset extensions of well-founded orderings are well-founded [20].

An *equation* is an unordered pair of terms $l \simeq r$. A *rewrite rule* is an ordered pair of terms $l \rightarrow r$. A set of rewrite rules is called a *term rewriting system* or *rewrite system*. If $l \succ r$ for a reduction ordering $\succ$, then an equation $l \simeq r$ may be oriented into a rewrite rule $l \rightarrow r$. A rewrite system R defines a relation $\rightarrow_R$ on terms as follows: $s \rightarrow_R t$ if there are a rewrite rule $l \rightarrow r \in R$, a substitution σ and a position u such that $s|u = l\sigma$ and t is $s[r\sigma]_u$. The relation $\leftrightarrow_R$ is defined as the union $\rightarrow_R \cup \leftarrow_R$, and $\rightarrow_R^*$ and $\leftrightarrow_R^*$ are the transitive and reflexive closures of $\rightarrow_R$ and $\leftrightarrow_R$. For a set of equations E, $s \leftrightarrow_E t$ if there are an equation $l \simeq r \in E$, a substitution σ and a position u such that $s|u = l\sigma$ and t is $s[r\sigma]_u$; $s \rightarrow_E t$ if $s \leftrightarrow_E t$ and $s \succ t$ for a reduction ordering $\succ$. The closure $\leftrightarrow_E^*$ is the congruence defined by E on the set of terms. The equality $\leftrightarrow_E = \rightarrow_E \cup \leftarrow_E$ holds only if the ordering $\succ$ is total in every congruence class defined by E.

All the following definitions apply to both a rewrite system R and a set of equations E. The only difference is the way the relations $\rightarrow_R$ and $\rightarrow_E$ are defined, as shown above. A term s is in *E-normal form* or *E-irreducible*, if there is no term t such that $s \rightarrow_E t$. A set of equations E is *Church-Rosser*, if $s \leftrightarrow_E^* t$ implies $s \rightarrow_E^* \circ \leftarrow_E^* t$, *confluent*, if $s \leftarrow_E^* \circ \rightarrow_E^* t$ implies $s \rightarrow_E^* \circ \leftarrow_E^* t$, *locally confluent*, if $s \leftarrow_E \circ \rightarrow_E t$ implies $s \rightarrow_E^* \circ \leftarrow_E^* t$, *canonical*, if it is both confluent and *reduced*, that is for all $l \simeq r \in E$, l and r are in normal form with respect to $E - \{l \simeq r\}$.

If we add to the signature a finite set P of predicate symbols with their arities, we obtain $A(P, F, X)$ and $A(P, F)$, i.e. the sets of *atoms* and *ground atoms* on $< P, F, X >$. If P includes the equality predicate, an equation is an atom. A *literal* is an atom or a negated atom, a *clause* is a disjunction of literals, a *unit clause* is a clause made of one literal and a *Horn clause* is a clause with at most one positive literal. All variables in a clause are implicitly universally quantified. The definitions given for terms and substitutions extend to atoms, literals and clauses. In particular, a (complete) simplification ordering $\succ$ on terms and literals can be extended to equations, clauses and sets of clauses, as shown for instance in [40].

2.2 Inference rules and search plans

In this section we introduce 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. 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 [29]. 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:

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

Alternative schemes for inference rules, called *deduction* and *deletion*, are given in [15]. We further distinguish between inference rules which transform the presentation (*forward reasoning*) and inference rules which transform the target[1] (*backward reasoning*):

- *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*: $f\colon \dfrac{(S;\varphi)}{(S;\varphi')}$ where φ logically implies φ'.
  - *Contraction inference rules*: $f\colon \dfrac{(S;\varphi)}{(S;\varphi')}$ where φ does not logically imply φ'.

Example 2.1 Deduction *of a critical pair is an* expansion *inference rule on the presentation, since it adds to the given set a new equation:*

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

where E is a set of equations, σ is the most general unifier of the non-variable subterm $p|u$ and l, and $\succ$ is the assumed complete simplification ordering on terms. The target is an equational theorem $\forall \bar{x} s \simeq t$, which we write as $\hat{s} \simeq \hat{t}$ to denote that it contains only universally quantified variables and therefore can be regarded as a ground equality.

Simplification *of the target is a* contraction *inference rule:*

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

The inference rules are required to be *sound*:

[1] The target can be formulated as a set of sentences. For convenience of representation, we write the target as a single sentence.

Definition 2.1 *An inference step* $(S;\varphi) \vdash (S';\varphi')$ *is* sound *if* $Th(S') \subseteq Th(S)$, monotonic *if* $Th(S) \subseteq Th(S')$. *It is* relevant *if* $\varphi' \in Th(S')$ *if and only if* $\varphi \in Th(S)$.

Soundness ensures that a presentation inference step does not create new elements which are not true in the theory. Monotonicity guarantees that all theorems are preserved. 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. For an interesting example of expansion inference rule for the target, we refer to Section 5.3.

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. For instance, a *Simplification-first* search plan [38] is a search plan where Simplification has priority over expansion.

The inference rules and the search plan cooperate to generate a derivation from a given input. The input for a theorem proving strategy is a pair $(S;\varphi)$, where S is a presentation of the theory $Th(S) = \{\varphi | S \models \varphi\}$ and φ is 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 $\varphi_{i+1} \in Th(S_{i+1})$. 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 $\varphi_0 \in Th(S_0)$.

2.3 Proof orderings for theorem proving

In this section we apply *proof orderings* [6, 8] to describe a theorem proving derivation as a *target-oriented proof reduction* process.

A *proof ordering* is a monotonic, stable and well-founded ordering on proofs [6]. Proof orderings are defined in general starting from some ordering on the data involved in the proofs. Thus, we assume to have a complete simplification ordering $\succ$ on terms and literals. We prefer to have a simplification ordering, although a well-founded, monotonic and stable ordering total on ground terms would be sufficient. The following example [29] shows how to define a proof ordering starting from an ordering on terms:

Example 2.2 *Equational proofs can be represented as chains [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. A proof ordering to compare ground equational proofs 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* $\succ$*. Then we compare two proofs* $s \leftrightarrow^*_E t$ *and* $s \leftrightarrow^*_{E'} t$ *by the multiset extension* $>^e{}_{mul}$ *of* $>^e$*.*

Proof orderings were introduced in [6] to prove correctness of the Knuth-Bendix completion procedure as a procedure which generates 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, with no target. A confluent rewrite system is a presentation such that for all theorems $s \simeq t$ there is a *rewrite proof*, i.e. a proof in the form $s \rightarrow^* \circ \leftarrow^* t$. Therefore, correctness of Knuth-Bendix completion is proved by showing that *all* the proofs in the theory are eventually reduced to rewrite proofs during the derivation. Since such a derivation transforms *only* the presentation, with the purpose of reducing *all* the proofs, one needs to compare the proof of φ in S_i with the proof of φ in S_{i+1} for all the theorems φ in the theory. For this reason, proof orderings are applied in [6] to compare only proofs of the same theorem. A theorem proving derivation

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

also has the target and *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 proofs of different theorems 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 extended as follows:

Example 2.3 *We can compare any two ground equational proofs* $s \leftrightarrow^*_E t$ *and* $s' \leftrightarrow^*_{E'} t'$ *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 such that proofs of different theorems may be comparable. We assume that both the ordering on proofs and the ordering on terms and literals have a bottom element. For proofs, the minimum is the *empty proof*, which we denote by ε. For terms and literals, the minimum is a dummy element *true*, which represents the theorem whose proof is ε. Given a proof ordering $>_p$, we denote by $\Pi(S, \varphi)$ the set of all the *minimal proofs* of φ from S with respect to $>_p$. By assuming a proof ordering $>_p$, 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 a minimal proof of φ_0 in S_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 $\Pi(S_k, \varphi_k) = \{\varepsilon\}$ and φ_k is *true*. The introduction of the symbol *true* is not mere formality. For instance, in equational logic a theorem $s \simeq s$ is regarded as trivially true. However, it is not so in equational theorem proving, since a procedure needs to check that the two sides of the equation are identical before stating that the theorem is true. This requires an inference step and therefore a derivation that has reached the state $(S; s \simeq s)$ is not successful yet. Indeed, the proof of $s \simeq s$ is not empty. Thus we need the symbol *true* to indicate the success of a derivation.

3 Completion procedures

In this section we give the core of our framework. We start by characterizing the requirements for a derivation to be a process of *target-oriented proof reduction.* All the inferences are assumed to be monotonic and relevant. Intuitively, since the purpose of a derivation is to reach the bottom element in the proof ordering, an inference should not increase the complexity of proofs:

Definition 3.1 *An inference step* $(S; \varphi) \vdash (S'; \varphi')$ *is* proof-reducing on φ *if for all* $P \in \Pi(S, \varphi)$*, either* $P \in \Pi(S', \varphi')$ *or there exists a* $Q \in \Pi(S', \varphi')$ *such that* $P >_p Q$*. If the latter holds for some* $P \in \Pi(S, \varphi)$*, then the step is* strictly proof-reducing.
A target inference step $(S; \varphi) \vdash (S; \varphi')$ *is* (strictly) proof-reducing *if it is (strictly) proof-reducing on* φ*.*

In other words, every proof which is minimal at a certain stage of the derivation can be replaced only by a smaller proof.

Example 3.1 *Simplification of the target as given in Example 2.1 is strictly proof-reducing. If the target* $\hat{s} \simeq \hat{t}$ *is replaced by the target* $\hat{s}' \simeq \hat{t}'$ *because* $\hat{s}$ *is simplified to* $\hat{s}'$*, we have* $\hat{s} \succ \hat{s}'$*,* $\hat{t} = \hat{t}'$ *and therefore* $\{\hat{s}, \hat{t}\} \succ_{mul} \{\hat{s}', \hat{t}'\}$*. If we assume the proof ordering* $>_u$ *introduced in Example 2.3, it follows that* $\hat{s} \leftrightarrow_E^* \hat{t} >_u \hat{s}' \leftrightarrow_E^* \hat{t}'$*.*

For a presentation inference step we allow more flexibility, because an inference step on the presentation may not immediately reduce any proof of the target but still be necessary to decrease it eventually. The proof reduction effect of a presentation inference step needs to be checked on a larger set of theorems in the theory, not just on the given target. We call *domain*, denoted by $\mathcal{T}$, the set of sentences where the presentation inference rules are proof-reducing:

Definition 3.2 *A presentation inference step* $(S; \varphi) \vdash (S'; \varphi)$ *is* proof-reducing *on* $\mathcal{T}$ *if*

1. *either it is strictly proof-reducing on* φ
2. *or*
   (a) $\Pi(S, \varphi) = \Pi(S', \varphi)$,
   (b) $\forall \psi \in \mathcal{T}$*,* $(S; \psi) \vdash (S', \psi)$ *is proof-reducing on* ψ *and*
   (c) $\exists \psi \in \mathcal{T}$ *such that* $(S; \psi) \vdash (S', \psi)$ *is strictly proof-reducing on* ψ*.*

The first condition dictates that an inference step which reduces a proof of the target is proof-reducing, regardless of its effects on other theorems. On the other hand, an inference step which does not affect any proof of the target is proof-reducing, if it does not increase any proof and strictly decreases at least one. The domain $\mathcal{T}$ may vary according to individual completion procedures. 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. In principle, a procedure with a more restricted domain should be more efficient, because it would not spend time in reducing proofs of theorems that are not related to the given target.

Example 3.2 *Deduction of a critical pair as given in Example 2.1 is proof-reducing on the domain $\mathcal{T}$ of all ground equations. We assume the proof ordering $>_u$ introduced in Example 2.3. Given two equations $l \simeq r$ and $p \simeq q$ in E, a* critical overlap *of $l \simeq r$ and $p \simeq q$ is a 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, t is $c[q\tau]$, $(p|u)\tau = l\tau$ for some non-variable subterm $p|u$ of p and s is $c[p[r]_u\tau]$. 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. The proof $s \leftrightarrow_{p[r]_u\sigma\simeq q\sigma} t$, justified by the critical pair, is smaller than the proof $s \leftarrow_{l\simeq r} v \rightarrow_{p\simeq q} t$: since $v \succ s$ and $v \succ t$, and thus $\{(v,l,s),(v,p,t)\} >^e{}_{mul}\{(s,p[r]_u\sigma,t)\}$ (assuming, without loss of generality, that $s \succ t$). Therefore, every minimal proof which contains $s \leftarrow_{l\simeq r} v \rightarrow_{p\simeq q} t$ as a subproof is no longer minimal after the generation of the critical pair. Such a proof is replaced by the smaller proof where all occurrences of $s \leftarrow_{l\simeq r} v \rightarrow_{p\simeq q} t$ are replaced by $s \leftrightarrow_{p[r]_u\sigma\simeq q\sigma} t$: for all $\psi \in \mathcal{T}$, $\Pi(E \cup \{p[r]_u\sigma \simeq q\sigma\},\psi) = \Pi(E,\psi) - \{P[s \leftarrow_{l\simeq r} v \rightarrow_{p\simeq q} t]\} \cup \{P[s \leftrightarrow_{p[r]_u\sigma\simeq q\sigma} t]\}$. If a minimal proof of the target itself contains a critical overlap between $l \simeq r$ and $p \simeq q$, the Deduction step is strictly proof-reducing.*

The notion of proof reduction defined so far 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 need a notion of *redundancy*:

Definition 3.3 *A sentence φ is* redundant in S on ψ *if $\Pi(S,\psi) = \Pi(S \cup \{\varphi\},\psi)$; it is* redundant in S on domain $\mathcal{T}$ *if it is redundant on all $\psi \in \mathcal{T}$.*

A sentence is redundant in a presentation on a specific target, if adding it to the presentation does not affect any minimal proof of the target. If this holds on the entire domain, the sentence is said to be redundant on the domain.

Example 3.3 *An inference rule which deletes an equation without adding any is* Functional subsumption*:*

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

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 2.3. 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 $p \mathrel{\bullet\!\!\!\triangleright} l$ and $q \mathrel{\bullet\!\!\!\triangleright} r$.

Definition 3.4 *An inference step $(S;\varphi) \vdash (S';\varphi')$ is* reducing *on $\mathcal{T}$ (on φ) if either it is proof-reducing on $\mathcal{T}$ (on φ) or it deletes a sentence which is redundant in S on domain $\mathcal{T}$ (on φ).*
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 3.5 *A theorem proving strategy $\mathcal{C} =< I; \Sigma >$ 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:

- *soundness: $\forall i \geq 0$, $Th(S_{i+1}) \subseteq Th(S_i)$,*
- *relevance: $\forall i \geq 0$, $\varphi_i \in \mathcal{T}$ and $\varphi_{i+1} \in Th(S_{i+1})$ 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 $\mathcal{T}$ (on φ_i).*

The definition requires soundness and not monotonicity, because soundness and relevance together are sufficient for theorem proving. Monotonicity will be required only for the application of completion to the generation of confluent sets. *Reduction* is the fundamental property of 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.

3.1 Redundancy

In this section we study further the notion of *redundancy*. The interest in redundancy of data in a theorem proving derivation resides in the importance of contraction inference rules. Although contraction inference rules are necessary to make theorem proving feasible, only a few of them are known. The purpose of studying redundancy is to gain insight into how to design new and powerful contraction rules. A notion of redundant clauses appeared in [59] and in [15]. We show that redundant clauses according to these works are redundant in our sense. On the other hand, there are clauses which are intuitively redundant and redundant according to our definition, but not according to the definitions in [59] and [15].

Definition 3.6 (Rusinowitch 1988) [59] *A clause φ is* R-redundant *in a set S if there exists a clause $\psi \in S$ such that ψ properly subsumes φ, i.e. $\varphi \gtrdot \psi$, where $\gtrdot$ is the proper subsumption ordering on clauses.*

R-redundancy has been investigated in [60] in the context of proofs by resolution in first order logic. Very high numbers of R-redundant clauses may be generated in such derivations, resulting in waste of space to hold them and in waste of time to perform the subsumption test to detect them. Two techniques to limit the generation of R-redundant clauses are proposed in [60].

Definition 3.7 (Bachmair and Ganzinger 1990) [15] *A clause φ is* B-redundant *in a set S if there exists an ordering $>^d$ on clauses, which is monotonic, stable, well-founded and total on* *ground clauses, such that the following holds: for all ground instances $\varphi\sigma$ of φ, there are ground instances $\psi_1 \ldots \psi_n$ of clauses in S such that $\{\psi_1 \ldots \psi_n\} \models \varphi\sigma$ and $\forall j, 1 \leq j \leq n, \varphi\sigma >^d \psi_j$.*

Lemma 3.1 (Bachmair and Ganzinger 1990) [15] *R-redundant clauses are B-redundant.*

In our view, the intuition behind the notion of redundancy is that a clause φ is redundant in S if adding φ to S does not decrease any minimal proof in S (Definition 3.3). In fact our definition captures the meaning of Definition 3.7:

Theorem 3.1 *If a clause φ is B-redundant in S, then it is redundant on the domain of all ground clauses.*

Proof: for all ground clauses ψ, we regard any set $\{\psi_1 \ldots \psi_n\}$ of ground instances of clauses in S, such that $\{\psi_1 \ldots \psi_n\} \models \psi$, as a proof in S of ψ. Since the ordering $>^d$ assumed in the definition of B-redundancy is well-founded and total on ground clauses, its multiset extension ${>^d}_{mul}$ is also well-founded and total on multisets of ground clauses. Let the proof ordering $>_p$ be ${>^d}_{mul}$. Since ${>^d}_{mul}$ is total, the minimal proof in S of a ground clause ψ is unique. By slightly abusing our notation, we use $\Pi(S, \psi)$ to denote the unique minimal proof of ψ in S. Let φ be B-redundant in S. We show that $\Pi(S \cup \{\varphi\}, \psi) = \Pi(S, \psi)$ for all ground theorems ψ. Since $S \subset S \cup \{\varphi\}$, $\Pi(S \cup \{\varphi\}, \psi) \leq_p \Pi(S, \psi)$ trivially holds and therefore we simply have to show that $\Pi(S \cup \{\varphi\}, \psi) \not<_p \Pi(S, \psi)$. The proof is done by way of contradiction: if $\Pi(S \cup \{\varphi\}, \psi) <_p \Pi(S, \psi)$, then the smallest set of ground instances of clauses in $S \cup \{\varphi\}$ which logically entails ψ has the form $S' \cup \{\varphi\sigma_1 \ldots \varphi\sigma_k\}$ for some set S' of ground instances of clauses in S and some ground substitutions $\sigma_1 \ldots \sigma_k$. Since φ is B-redundant in S, for all $\varphi\sigma_i$, $1 \leq i \leq k$, there are ground instances $\{\psi^i_1 \ldots \psi^i_{n_i}\}$ of clauses in S such that $\{\psi^i_1 \ldots \psi^i_{n_i}\} \models \varphi\sigma_i$ and $\varphi\sigma_i >^d \psi^i_j$, $\forall j, 1 \leq j \leq n_i$. Therefore, $S' \cup \{\psi^i_1 \ldots \psi^i_{n_i}\}^k_{i=1} {<^d}_{mul} S' \cup \{\varphi\sigma_1 \ldots \varphi\sigma_k\}$ and $S' \cup \{\psi^i_1 \ldots \psi^i_n\}^k_{i=1} \models \psi$, that is $S' \cup \{\varphi\sigma_1 \ldots \varphi\sigma_k\}$ cannot be the smallest set entailing ψ. It follows that $\Pi(S \cup \{\varphi\}, \psi) = \Pi(S, \psi)$. □

On the other hand, there are cases where trivially redundant clauses are not B-redundant, whereas they are redundant according to our definition:

Example 3.4 *If $S = \{P, \neg R, R\}$, where P and R are ground atoms, P is intuitively redundant and it is redundant according to our Definition 3.3: the minimal proof of every ground theorem is given by $\{\neg R, R\}$, since $\{\neg R, R\}$ yields the empty clause and therefore any clause. However, if $R \succ P$ and thus $R >^d P$, then P is not B-redundant.*

This example shows that a notion of redundancy based on an ordering on clauses is not ideal, since different precedences on predicate symbols may be needed in order to characterize as redundant different clauses during a computation.

In our definition of completion, we have required that if a contraction step simply deletes a sentence, then the sentence deleted must be redundant. The following lemma shows that if a contraction step replaces a sentence by another, the replaced sentence must be redundant on the specific target if the step is strictly proof-reducing, otherwise it must be redundant on the entire domain:

Lemma 3.2 *If a contraction inference step $(S \cup \{\psi\}; \varphi) \vdash (S \cup \{\psi'\}; \varphi)$ is proof-reducing on $\mathcal{T}$ by Condition 1 in Definition 3.2, then ψ is redundant in $S \cup \{\psi'\}$ on φ; if it is proof-reducing by Condition 2 in Definition 3.2, then ψ is redundant in $S \cup \{\psi'\}$ on the domain $\mathcal{T}$.*

Proof: if ψ does not occur as an axiom in any proof $P \in \Pi(S \cup \{\psi\}, \varphi)$, then $\Pi(S \cup \{\psi\}, \varphi) = \Pi(S, \varphi)$, i.e. ψ is redundant on φ in S and therefore also in $S \cup \{\psi'\}$. If ψ is an axiom in some proof $P \in \Pi(S \cup \{\psi\}, \varphi)$, then $P \notin \Pi(S \cup \{\psi'\}, \varphi)$, since $\psi \notin S \cup \{\psi'\}$. By Condition 1 in Definition 3.2, there exists a $Q \in \Pi(S \cup \{\psi'\}, \varphi)$ such that $P >_p Q$. In other words, all the proofs of φ where ψ occurs as an axiom are replaced by smaller proofs in $\Pi(S \cup \{\psi'\}, \varphi)$. Adding ψ to $S \cup \{\psi'\}$ would not reduce any proof in $\Pi(S \cup \{\psi'\}, \varphi)$ and therefore ψ is redundant on φ in $S \cup \{\psi'\}$. By applying the same argument to all theorems in the domain we obtain the second part of the lemma. □

It follows that all sentences deleted by contraction steps are redundant. A similar relationship between deletion and B-redundancy holds according to the approach proposed in [15]. The main difference between our approach and that of [15] is that we have a notion of redundancy on the domain as well as one on the target. For derivations without target, e.g. derivations which generate a confluent system, we have soundness, monotonicity, proof-reduction on the domain and redundancy on the domain. For such derivations our approach is basically equivalent to that in [15]. For derivations with target, e.g. theorem proving derivations, we require soundness and relevance, but not monotonicity; we allow proof-reduction and redundancy on the target, in addition to proof-reduction and redundancy on the domain. By proof reduction and redundancy on the target, contraction steps may replace sentences which are not redundant on the whole domain, provided they are redundant on the specific target. In this way our definitions allow in principle very strong contraction inference rules.

4 Fairness and completeness

A theorem proving method is complete if, whenever φ_0 is a theorem of S_0, the derivation from $(S_0; \varphi_0)$ succeeds. Completeness involves both the inference rules and the search plan. First, it requires that if $\varphi_0 \in Th(S_0)$, 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 $(S_0; \varphi_0)$.

Intuitively, 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 $(S; true)$. We use the term "refutational completeness" for the inference rules to differentiate it from the completeness of the theorem proving strategy. Furthermore, "refutational" emphasizes that the goal is to prove a specific theorem. The following definition is an equivalent characterization of this concept in terms of proof reduction:

Definition 4.1 *A set I of inference rules is* refutationally complete *if whenever $\varphi \in Th(S)$ and $\Pi(S,\varphi) \neq \{\varepsilon\}$, $\forall P \in \Pi(S,\varphi)$ there exists a path $(S;\varphi)\vdash_I(S_1;\varphi_1)\vdash_I \ldots \vdash_I(S';\varphi')$ such that $P >_p Q$ for some $Q \in \Pi(S',\varphi')$.*

A set of inference rules is refutationally complete if it can reduce any non-empty proof of the target. 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. An 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 [14].

Given a completion procedure $\mathcal{C} =< I;\Sigma >$, 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 $(S_i;\varphi_i)$ may have 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 formalize this concept in terms of proof reduction:

Definition 4.2 *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 $\forall i \geq 0$, $\forall P \in \Pi(S_i,\varphi_i)$, 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 $P >_p Q$, for some $Q \in \Pi(S',\varphi')$, then there exists an $(S_j;\varphi_j)$, for some $j > i$, and an $R \in \Pi(S_j,\varphi_j)$ such that $Q \geq_p R$.*
A search plan Σ is fair *if all the derivations controlled by Σ are fair.*

In other words, if the inference rules can reduce a proof of the target at $(S_i;\varphi_i)$, a fair search plan guarantees that such a proof will be reduced at a later stage $(S_j;\varphi_j)$. This definition is target-oriented because it only requires that the proofs of the intended target are reduced. Actually it only requires that *one* proof of the target is reduced. If a proof of φ is reduced to ε at stage j, the set of minimal proofs of φ collapses to $\{\varepsilon\}$ and $P >_p \varepsilon$ for every proof P considered at all stages earlier than j. In theorem proving we are only interested in finding one proof of the target and therefore a search plan may trim the search space considerably and still be fair as long as it does not remove the possibility of finding any proof.

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

Theorem 4.1 *If a completion procedure $\mathcal{C}$ on domain $\mathcal{T}$ has refutationally complete inference rules and fair search plan, 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* $\Pi(S_i,\varphi_i) \neq \{\varepsilon\}$*, then* $\forall P \in \Pi(S_i,\varphi_i)$*, there exists an* (S_j,φ_j)*, for some* $j > i$*, such that* $P >_p R$ *for some* $R \in \Pi(S_j,\varphi_j)$*.*

Proof: if $\Pi(S_i,\varphi_i) \neq \{\varepsilon\}$, then by completeness of the inference rules, for all $P \in \Pi(S_i,\varphi_i)$ there exists a path $(S_i;\varphi_i)\vdash_I\ldots\vdash_I(S';\varphi')$ such that $P >_p Q$ for some $Q \in \Pi(S',\varphi')$. By fairness of the search plan, there exists an $(S_j;\varphi_j)$, for some $j > i$, and an $R \in \Pi(S_j,\varphi_j)$ such that $P >_p Q \geq_p R$. □

Corollary 4.1 *If a completion procedure* $\mathcal{C}$ *on domain* $\mathcal{T}$ *has refutationally complete inference rules and a fair search plan, then for all inputs* $(S_0;\varphi_0)$*, if* $\varphi_0 \in Th(S_0)$*, the derivation*

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

reaches a stage k*,* $k \geq 0$*, such that* φ_k *is the clause true.*

Proof: let P be any proof in $\Pi(S_0,\varphi_0)$. By Theorem 4.1 and the well-foundedness of $>_p$ the derivation reaches a stage k such that P has been reduced to ε. Then $\Pi(S_k,\varphi_k) = \{\varepsilon\}$ and φ_k is the clause *true*. □

4.1 Uniform fairness and saturated sets

In this subsection and in the next one we show how the classical results on completion as generation of confluent systems are incorporated in our framework.

We consider derivations without target and we assume that the monotonicity property (see Definition 2.1) is added to our definition of completion. The key difference between derivations with a theorem proving target and derivations without target is the fairness requirement. Our definition of fairness (Definition 4.2) is sufficient for theorem proving (Theorem 4.1), but it is not sufficient to guarantee that a confluent rewrite system is generated eventually, because it does not guarantee that all critical pairs are considered eventually. This demands a stronger fairness property, which we call *uniform fairness*. The first definition of uniform fairness appeared in [42], where it is required that the search plan sorts the rewrite rules by a well-founded ordering, in order to ensure that no rule is indefinitely postponed. We mention this very first notion of (uniform) fairness, because it states explicitly that fairness is a property of the search plan. We recall here a more recent definition. Let $I_e(S)$ be the set of sentences which can be generated in one expansion step from S and $S_\infty = \bigcup_{j\geq 0}\bigcap_{i\geq j} S_i$ be the *limit* of the derivation, i.e. the possibly infinite set of all the *persistent* sentences [42, 8]:

Definition 4.3 (Rusinowitch 1988) [59], (Bachmair and Ganzinger 1990) [15] *A derivation* $S_0\vdash_{\mathcal{C}} S_1\vdash_{\mathcal{C}}\ldots S_i\vdash_{\mathcal{C}}\ldots$ *is* uniformly fair *on domain* $\mathcal{T}$ *if* $\forall\varphi \in I_e(S_\infty)$ *there exists an* S_j *such that either* $\varphi \in S_j$ *or* φ *is redundant in* S_j *on domain* $\mathcal{T}$*.*

This definition of fairness generalizes previous definitions given in [42, 6, 8, 14]. As an example, a Knuth-Bendix derivation such that all critical pairs from persisting equations are eventually generated or subsumed or reduced to a common term is uniformly fair.

Fairness and uniform fairness are conceptually different. First, fairness is *target-oriented*, whereas uniform fairness is defined for a derivation without a target. In [59, 15], Definition 4.3 is applied to refutational theorem proving, where S_0 contains the negation of the target. In this case the only persisting clause is the empty clause $\Box$ and $I_e(\bigcup_{i\geq 0}\bigcap_{j\geq i} S_j) = \{\Box\}$. Then Definition 4.3 says that the limit of the derivation is the empty clause. A notion of fairness given in terms of $S_\infty = \{\Box\}$ does not help the design of search plans, because it simply re-states that the derivation should eventually succeed. It does not provide any hint on how a search plan should choose a successor at any given stage of the derivation. Second, the intuitive meaning of uniform fairness is to be fair to the inference rules, that is to apply all the inference rules to all the data. However, this is impossible in the presence of contraction rules: if a clause φ is deleted by a contraction step before an expansion rule f is applied to φ, the derivation is not fair to f. The problem was then to define fairness in such a way that the application of contraction rules is fair. This problem has been solved in the definition of uniform fairness by establishing that it is fair not to perform an expansion inference step if its premises are not persistent and it is fair to replace a clause φ by clauses which make it redundant. In theorem proving the idea of fairness is not to be fair to the inference rules, but to the target. Therefore the interaction of expansion and contraction rules is no longer an issue. All inference rules are treated uniformly by considering their effect with respect to the goal of reducing the proof of the target.

The following example illustrates the conditions which were proved sufficient for uniform fairness of an Unfailing Knuth-Bendix derivation in [8]. These conditions represent the most well known definition of (uniform) fairness for a completion procedure:

Example 4.1 *A derivation* $E_0 \vdash_{UKB} E_1 \vdash_{UKB} \ldots \vdash_{UKB} E_i \vdash_{UKB} \ldots$ *is uniformly fair if for all critical pairs* $g \simeq d \in I_e(E_\infty)$*,* $g \simeq d \in \bigcup_{i\geq 0} E_i$ *and* E_∞ *is reduced. The first condition alone is actually sufficient for uniform fairness: the application of contraction rules is allowed but not required. Since at any stage of the computation it is not known which equations are going to persist and which equations are going to be simplified, the above conditions for uniform fairness prescribe in practice to apply exhaustively all the inference rules of Unfailing Knuth-Bendix completion until none applies.*

The concept of uniform fairness leads to the following notion of *saturated* presentation:

Definition 4.4 (Kounalis and Rusinowitch 1988) [52], (Bachmair and Ganzinger 1990) [15] *A presentation* S *is* saturated *on the domain* $\mathcal{T}$ *of a completion procedure if and only if* $\forall\psi \in I_e(S)$*, either* $\psi \in S$ *or* ψ *is redundant in* S *on* $\mathcal{T}$*.*

In other words, no non-trivial consequences can be added to a saturated presentation. In the equational case, as remarked in [52], a set of equations is saturated if no divergent critical pairs can be deduced, or equivalently, the set is *locally confluent.* As in the definition of uniform fairness, the application of contraction inference rules is allowed but not required: for instance, a locally confluent equational presentation is not necessarily reduced.

If a derivation is uniformly fair, S_∞ is saturated. Since uniform fairness is defined in terms of redundancy and our notion of redundancy is more general than those in [59] and [15], we give a new proof of this result. First we prove the following lemma:

Lemma 4.1 *If a derivation $S_0 \vdash_{\mathcal{C}} S_1 \vdash_{\mathcal{C}} \ldots S_i \vdash_{\mathcal{C}} \ldots$ is* uniformly fair *on $\mathcal{T}$, then $\forall \psi \in \mathcal{T}$, $\forall i \geq 0$, if ψ is redundant in S_i on $\mathcal{T}$, ψ is also redundant in S_∞ on $\mathcal{T}$.*

Proof: if ψ is redundant in S_i on $\mathcal{T}$, ψ does not occur as an axiom in any minimal proof in S_i of theorems in $\mathcal{T}$. This also holds for ψ itself. In other words there exists at least a proof $P \in \Pi(S_i, \psi)$ which is smaller than the proof represented by ψ itself: $\psi >_p P$. Since there is no target, the derivation is proof-reducing by Condition 2 in Definition 3.2. It follows that either $P \in \Pi(S_\infty, \psi)$ or P is replaced by a proof $Q \in \Pi(S_\infty, \psi)$ such that $P >_p Q$. In both cases, there is a proof $R \in \Pi(S_\infty, \psi)$ such that $\psi >_p R$. By monotonicity and stability of $>_p$, $C[\psi\sigma] >_p C[R\sigma]$ for all proof contexts C and substitutions σ. In other words, ψ is not involved in any minimal proof in S_∞ of a theorem in $\mathcal{T}$, since any occurrence of ψ in a proof can be replaced by a proof of ψ smaller than ψ itself. It follows that ψ is redundant in S_∞ on $\mathcal{T}$. □

Theorem 4.2 (Kounalis and Rusinowitch 1988) [52], (Bachmair and Ganzinger 1990) [15] *If a derivation $S_0 \vdash_{\mathcal{C}} S_1 \vdash_{\mathcal{C}} \ldots S_i \vdash_{\mathcal{C}} \ldots$ is* uniformly fair on $\mathcal{T}$, *then S_∞ is* saturated *on $\mathcal{T}$.*

Proof: we show that for all $\varphi \in I_e(S_\infty)$, either $\varphi \in S_\infty$ or φ is redundant in S_∞ on $\mathcal{T}$. By uniform fairness of the derivation, there exists an S_j, for some $j \geq 0$, such that either $\varphi \in S_j$ or φ is redundant in S_j on $\mathcal{T}$. If φ is redundant in S_j, it is also redundant in S_∞ by Lemma 4.1. If $\varphi \in S_j$, then either φ is not deleted afterwards, that is $\varphi \in S_\infty$, or φ is deleted at some stage $i > j$. If φ is simply deleted, φ is redundant in S_i on $\mathcal{T}$ by Definition 3.5 of completion. If φ is replaced by another sentence, φ is redundant in S_{i+1} on $\mathcal{T}$ by Lemma 3.2. In both cases φ is redundant in S_∞ by Lemma 4.1. □

This theorem generalizes the following classical results:

Theorem 4.3 (Knuth and Bendix 1970) [51], (Huet 1981) [42], (Bachmair, Dershowitz and Hsiang 1986) [6] *If a derivation $E_0 \vdash_{KB} E_1 \vdash_{KB} \ldots E_i \vdash_{KB} \ldots$ by the Knuth-Bendix completion procedure does not fail (on a persistent unoriented equation) and is* uniformly fair *on the domain $\mathcal{T}$ of all equations, then E_∞ is a* confluent *term rewriting system.*

Theorem 4.4 (Hsiang and Rusinowitch 1987) [38], (Bachmair, Dershowitz and Plaisted 1989) [12] *If a derivation $E_0 \vdash_{UKB} E_1 \vdash_{UKB} \ldots E_i \vdash_{UKB} \ldots$ by the Unfailing Knuth-Bendix completion procedure is* uniformly fair *on the domain $\mathcal{T}$ of all ground equations, then E_∞ is a* ground confluent *set of equations.*

If E is ground confluent, $\hat{s} \leftrightarrow_E^* \hat{t}$ if and only if $\hat{s} \rightarrow_E^* \circ \leftarrow_E^* \hat{t}$ and therefore $E \models \forall \bar{x} s \simeq t$ can be decided by well-founded reduction by E. This introduces us to the topic of the next section.

4.2 Decision procedures

In this section we study the properties of derivations in a finite, saturated presentation. We shall show that, under appropriate hypotheses, a saturated presentation is a *decision procedure* for its theory. First, we define under which conditions a presentation is a decision procedure:

Definition 4.5 *Let $\mathcal{C}$ be complete on domain $\mathcal{T}$. A finite presentation S is a* decision procedure *for $Th(S) \cap \mathcal{T}$, if for all $\varphi_0 \in \mathcal{T}$ the derivation $(S;\varphi_0) \vdash_{\mathcal{C}} (S;\varphi_1) \vdash_{\mathcal{C}} \ldots \vdash_{\mathcal{C}} (S;\varphi_i) \vdash_{\mathcal{C}} \ldots$ halts at some stage k, $k > 0$, and $\varphi_k = true$ if and only if $\varphi_0 \in Th(S)$.*

The presentation S can be regarded as an algorithm, which, if interpreted by the procedure $\mathcal{C}$, decides the validity of sentences in $\mathcal{T}$ in the theory of S. Clearly, once termination of the derivation is ensured, the correctness of the result is a consequence of the completeness of the completion procedure. Therefore, the key property of a decision procedure is that all derivations are guaranteed to halt regardless of the truth of the given target. Sufficient conditions for the termination of derivations can be given, if it is possible to exclude the application of expansion inference rules. This is exactly where the assumption of having a saturated presentation plays a role:

Lemma 4.2 *If a presentation S is* saturated *on $\mathcal{T}$, then no expansion inference rule which is proof-reducing on $\mathcal{T}$ applies to S.*

Proof: if a proof-reducing expansion inference rule derives S' from S, then, there is a $\psi \in \mathcal{T}$ such that $P >_p Q$ for some $P \in \Pi(S,\psi)$ and $Q \in \Pi(S',\psi)$. Since S is saturated on $\mathcal{T}$, this is impossible. □

In other words, if the presentation is saturated, all derivations are made only of target inference steps and contraction steps on the presentation. Termination conditions for these kinds of inferences can be given by using well-founded orderings:

Definition 4.6 *A target inference step $(S;\varphi) \vdash (S;\varphi')$ is* target-reducing *if $\varphi \succ \varphi'$. A contraction inference step $(S;\varphi) \vdash (S';\varphi)$ is* data-reducing *if either it deletes a redundant sentence or it replaces a sentence ψ in S by a sentence ψ' such that $\psi \succ \psi'$.*

An inference rule is *target-reducing* (*data-reducing*) if all the steps where it is applied are target-reducing (data-reducing). For instance, the Simplification inference rule is target-reducing (data-reducing if applied to the presentation) as shown in Examples 2.1 and 3.1.

Lemma 4.3 *A derivation $(S_0;\varphi_0) \vdash_{\mathcal{C}} (S_1;\varphi_1) \vdash_{\mathcal{C}} \ldots \vdash_{\mathcal{C}} (S_i;\varphi_i) \vdash_{\mathcal{C}} \ldots$ where every step is either target-reducing or data-reducing is guaranteed to halt.*

Proof: let $>_r$ be the lexicographic combination of the multiset extension $\succ_{mul}$ and of $\succ$ itself. By the definition of target-reducing and data-reducing steps, we have that $\forall i \geq 0$, $(S_i;\varphi_i) >_r (S_{i+1};\varphi_{i+1})$. Since the ordering $>_r$ is well-founded, the derivation is guaranteed to halt. □

We can now prove that a saturated set is a decision procedure:

Theorem 4.5 *Let $\mathcal{C}$ be a complete completion procedure on domain $\mathcal{T}$, such that all its target inference rules are target-reducing and all its contraction inference rules on the presentation are data-reducing. Then a presentation S which is* saturated *on $\mathcal{T}$ is a* decision procedure *for $\mathcal{T} \cap Th(S)$.*

Proof: by Lemma 4.2, for all $\varphi_0 \in \mathcal{T}$ the derivation from $(S; \varphi_0)$ may contain only target inference steps and contraction steps on the presentation. By the hypotheses on the inference rules and Lemma 4.3, such a derivation is guaranteed to halt at some stage k. Either $\varphi_k = true$ or $\varphi_k \neq true$. By completeness of $\mathcal{C}$, φ_k is *true* if and only if $\varphi_0 \in Th(S)$. Therefore, S is a decision procedure for $Th(S) \cap \mathcal{T}$. □

If we also assume that no contraction rule applies to the presentation, then all derivations from S are made only of target-reducing inference steps. In equational logic this corresponds to assume that S is not only confluent, but also *reduced*, i.e. *canonical* [27]. Derivations made only of target inference steps are traditionally called *linear* [19]. Therefore, a saturated set such that no contraction rule applies to the presentation yields only linear and terminating derivations.

Finally, we can characterize a completion procedure as a *generator of decision procedures*:

Theorem 4.6 *Let $\mathcal{C} =< I; \Sigma >$ be a completion procedure on domain $\mathcal{T}$ such that*

- *the procedure satisfies the monotonicity property,*
- *I is refutationally complete,*
- *Σ is uniformly fair and*
- *all the target inference rules are target-reducing and all the contraction inference rules on the presentation are data-reducing.*

For all presentations S_0, if the limit S_∞ of the derivation $S_0 \vdash_{\mathcal{C}} S_1 \vdash_{\mathcal{C}} \ldots \vdash_{\mathcal{C}} S_i \vdash_{\mathcal{C}} \ldots$ is finite, then S_∞ is a decision procedure for $\mathcal{T} \cap Th(S_0)$.

Proof: by Theorem 4.2, S_∞ is saturated on $\mathcal{T}$. S_∞ is a decision procedure for $\mathcal{T} \cap Th(S_\infty)$ by Theorem 4.5. By monotonicity, $Th(S_\infty) = Th(S_0)$ and therefore S_∞ is a decision procedure for $\mathcal{T} \cap Th(S_0)$. □

This theorem generalizes the classical results for equational logic and their extensions to Horn logic with equality. In equational logic, the (Unfailing) Knuth-Bendix completion procedure generates a (ground) confluent presentation that can be used to decide the validity of theorems in the form $\forall \bar{x} s \simeq t$ by well-founded simplification. Extensions to Horn logic with equality have been studied in [52, 15, 30]. Given a complete completion procedure for Horn logic with equality, such as those in [52, 15, 30], the issue is how to guarantee that derivations in a saturated and reduced presentation are target-reducing and therefore terminating. Targets have the form $B_1 \wedge \ldots \wedge B_m$, where each B_i is a ground positive literal. In [52] and [15] the problem is solved by imposing special restrictions on the clauses in the saturated set:

Definition 4.7 (Kounalis and Rusinowitch 1988) [52] *A Horn clause $A : -B_1 \ldots B_n$ is* ground-preserving *if the following two conditions hold:*

- *all variables occurring in a negated literal also occur in A and*

- *if A is an equation $s \simeq t$, either it can be oriented into a rewrite rule or s and t have the same set of variables.*

The conditions given in [15] are slightly different, but the purpose is basically the same: the ground-preserving condition is designed to ensure that whenever an inference step is applied between a clause in the saturated set and a ground target, the newly generated target is ground as well. The resolution and paramodulation inference rules in [52] and [15] are *ordered*, i.e. they are restricted by a given complete simplification ordering on terms and literals in such a way that at each step a ground literal in the target is replaced by a set of smaller ground literals. Therefore ordered resolution and ordered paramodulation steps between a ground-preserving clause and a ground target are target-reducing. A saturated presentation containing only ground-preserving clauses is then a decision procedure by Lemma 4.2, Lemma 4.3 and Theorem 4.5:

Theorem 4.7 (Kounalis and Rusinowitch 1988) [52], (Bachmair and Ganzinger 1990) [15] *Let S be a presentation in Horn logic with equality such that S is saturated on the domain of ground clauses and all clauses in S are ground-preserving. Then S is a decision procedure for targets in the form $B_1 \wedge \ldots \wedge B_m$, where each B_i is a ground positive literal.*

The requirement that all clauses are ground-preserving is quite strong. For instance the Horn clause $T(x,y) \vee \neg R(x,z) \vee \neg R(z,y)$ in the definition of the transitive closure T of a relation R is not ground-preserving, because of the variable z. The restriction to ground-preserving clauses is resemblant of the restriction to oriented equations for Knuth-Bendix completion. This restriction is lifted in the Unfailing Knuth-Bendix procedure by assuming a complete simplification ordering on terms and by designing a simplification rule that applies oriented instances of equations as simplifiers. In this way, a "static" requirement on the presentation, that it contains only oriented equations, is replaced by a "dynamic" property of the inferences, that use only oriented instances of equations. A similar result has been obtained for Horn logic with equality in [30] by assuming a complete simplification ordering $\succ$ on terms and literals. There is no restriction on the clauses in the presentation, but the target inference rules are designed in such a way that only *decreasing* instances of clauses are applied:

Definition 4.8 (Dershowitz 1991) [30] *A Horn clause $l \simeq r : -p_1 \simeq q_1 \ldots p_n \simeq q_n$ is* decreasing *if for all ground substitutions σ, $l\sigma \succ r\sigma$, $l\sigma \succ p_i\sigma$, $l\sigma \succ q_i\sigma$, $1 \leq i \leq n$.*

Whenever a decreasing instance of a clause in the saturated presentation is applied to a ground target, the newly generated ground target is smaller, i.e. the derivation is target-reducing:

Theorem 4.8 (Dershowitz 1991) [30] *A presentation saturated on the domain of ground clauses in Horn logic with equality is a decision procedure for targets in the form $B_1 \wedge \ldots \wedge B_m$, where each B_i is a ground positive literal.*

The practical importance of the interpretation of completion procedures as generators of decision procedures is limited by the observation that few theories have a finite saturated presentation. For instance, the *Maximal Unit Strategy* of [29] is a complete method for Horn logic with equality.

(The name derives from the restriction that a unit clause resolves/paramodulates into a negative literal which is maximal among all the negative literals in the clause.) The method is strongly oriented toward forward reasoning, since it basically works by inferring facts from the given facts and implications. Therefore, the saturated set is infinite in most cases, since it contains all the true facts in the theory:

Theorem 4.9 *If S_∞ is the saturated limit of a derivation by the Maximal Unit Strategy, then every non-unit clause is redundant in S_∞.*

Proof: the assumed domain is the domain of ground clauses. The proof is done by way of contradiction. We assume that there are a ground target $B_1 \wedge \ldots \wedge B_m$ and a minimal proof $P \in \Pi(S_\infty, B_1 \wedge \ldots \wedge B_m)$ where non-unit clauses occur as axioms. Without loss of generality, we can assume that $m = 1$ and that a non unit clause $A :- C_1 \ldots C_n$ is the first clause applied in P, i.e. $B_1 = A\sigma$ for some ground substitution σ. Let $A :- C_1 \ldots C_n$ be the shortest clause that can be applied to B_1. This step generates the subgoal $C_1\sigma \ldots C_n\sigma$. Let $C_1\sigma$ be maximal in $C_1\sigma \ldots C_n\sigma$. By completeness of the unit strategy, there exists a unit clause $G \in S_\infty$ such that $G\rho = C_1\sigma\rho$ for some substitution ρ. The literals G and C_1 have a common instance and therefore there is an mgu ρ' such that $G\rho' = C_1\rho'$. Thus, the clause $A\rho' :- C_2\rho' \ldots C_n\rho'$ can be generated in one unit resolution step from G and $A :- C_1 \ldots C_n$. Since S_∞ has been saturated by the Maximal Unit Strategy, $A\rho' :- C_2\rho' \ldots C_n\rho'$ is in S_∞. Furthermore, there exists a substitution τ such that $A\rho'\tau = B_1$, since $\rho' \lessdot \sigma\rho$. It follows that $A\rho' :- C_2\rho' \ldots C_n\rho'$ can be applied at the place of $A :- C_1 \ldots C_n$ in P, contradicting the hypothesis that $A :- C_1 \ldots C_n$ is the shortest applicable clause. □

The Maximal Unit Strategy represents a rather extreme case. But it remains that most theories have infinite saturated presentations under most completion procedures. Therefore, the interpretation of completion procedures as semidecision procedures is the most useful one in practice.

5 Completion procedures in equational logic

In the second part of this work we present in the framework developed so far some Knuth-Bendix type completion procedures for equational logic: UKB, AC-UKB, UKB with Cancellation Laws, Inequality Ordered-Saturation strategy and S-strategy. This presentation is new, because these procedures had not been described before as procedures for theorem proving in a unified framework based on target-oriented proof reduction. For instance, the Cancellation Laws [39] and the S-strategy [38] appeared first in the transfinite-trees approach of [40], so that proof reduction was not applied to them. Similarly, this is the first presentation in terms of target-oriented proof reduction of the Inequality Ordered-Saturation strategy (IOS) [3, 4]. The latter is especially relevant, because IOS is a refinement of UKB based on the target-oriented philosophy.

5.1 Unfailing Knuth-Bendix completion

The *Unfailing Knuth-Bendix procedure* [38, 12] is a semidecision procedure for equational theories. A derivation by UKB has the form

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

and it succeeds at stage k if $\hat{s}_k$ and $\hat{t}_k$ are identical. 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*:
    $$\frac{(E \cup \{p \simeq q, l \simeq r\}; \hat{s} \simeq \hat{t})}{(E \cup \{p[r\sigma]_u \simeq q, l \simeq r\}; \hat{s} \simeq \hat{t})} \quad \begin{array}{ll} p|u = l\sigma & \qquad p \succ p[r\sigma]_u \\ p \mathrel{\triangleright\!\!\!\cdot\,} l \vee q \succ p[r\sigma]_u & \end{array}$$
  - *Deduction*:
    $$\frac{(E \cup \{p \simeq q, l \simeq r\}; \hat{s} \simeq \hat{t})}{(E \cup \{p \simeq q, l \simeq r, p[r]_u\sigma \simeq q\sigma\}; \hat{s} \simeq \hat{t})} \quad \begin{array}{ll} p|u \notin X & \qquad (p|u)\sigma = l\sigma \\ p\sigma \not\preceq q\sigma, p[r]_u\sigma & \end{array}$$
  - *Deletion*:
    $$\frac{(E \cup \{l \simeq l\}; \hat{s} \simeq \hat{t})}{(E; \hat{s} \simeq \hat{t})}$$
  - *Functional subsumption*:
    $$\frac{(E \cup \{p \simeq q, l \simeq r\}; \hat{s} \simeq \hat{t})}{(E \cup \{l \simeq r\}; \hat{s} \simeq \hat{t})} \quad (p \simeq q) \mathrel{\triangleright\!\!\!\cdot\,} (l \simeq r)$$
- *Target inference rules*:
  - *Simplification*:
    $$\frac{(E \cup \{l \simeq r\}; \hat{s} \simeq \hat{t})}{(E \cup \{l \simeq r\}; \hat{s}[r\sigma]_u \simeq \hat{t})} \quad \hat{s}|u = l\sigma \quad \hat{s} \succ \hat{s}[r\sigma]_u$$
  - *Deletion*:
    $$\frac{(E; \hat{s} \simeq \hat{s}\})}{(E; true)}$$

We have already presented some of these inference rules in Examples 2.1, 3.1, 3.2 and 3.3. In *Simplification*, the conditions $p \succ p[r\sigma]_u$ and $\hat{s} \succ \hat{s}[r\sigma]_u$ ensure that simplification is well-founded and together with $p \mathrel{\triangleright\!\!\!\cdot\,} l \vee q \succ p[r\sigma]_u$ that it is proof-reducing (see Lemma 5.1). In *Deduction*, a critical pair $p[r]_u\sigma \simeq q\sigma$ 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. This makes Deduction proof-reducing (see Example 3.2). Simplification is the most important among the above inference rules, because it reduces dramatically the number and the size of the generated equations. If Simplification is not applied, Deduction may rapidly saturates the memory space with equations, making impossible to reach a proof in reasonable time. Thus, a search plan for UKB should be a *Simplification-first* plan.

We characterize the UKB procedure as a completion procedure by using the ordering $>_u$ introduced in Example 2.3:

Lemma 5.1 *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 3.2.
- 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 $p \mathrel{\dot{\triangleright}} l$, we have
      - · if $t \succ v$, $\{(s, p, t)\} >^e{}_{mul} \{(s, l, v), (t, q, v)\}$ since $p \mathrel{\dot{\triangleright}} l$ and $s \succ t$,
      - · if $v \succ t$, $\{(s, p, t)\} >^e{}_{mul} \{(s, l, v), (v, p[r\sigma]_u, t)\}$ since $p \mathrel{\dot{\triangleright}} l$ and $s \succ v$;
    - ∗ if $p \stackrel{\bullet}{=} l$ 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 3.3. □

Lemma 5.2 *The target inference rules of the UKB procedure are strictly proof-reducing.*

Proof: the proof for Simplification was given in Example 3.1. For a Deletion step we have $\{\hat{s}, \hat{s}\} \succ_{mul} \{true\}$, since *true* is smaller than any term. □

We can then show that UKB is a completion procedure:

Theorem 5.1 *The Unfailing Knuth-Bendix procedure is a completion procedure on the domain $\mathcal{T}$ of all ground equations.*

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

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

has the soundness, relevance and reduction properties. Soundness and relevance were proved among others in [42, 6, 8]. Reduction follows from Lemma 5.1 and Lemma 5.2. □

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

Theorem 5.2 (Hsiang and Rusinowitch 1987) [38], (Bachmair, Dershowitz and Plaisted 1989) [12] *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 $(E; \hat{s} \simeq \hat{t})$.*

5.2 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.

The efficiency of the UKB strategy can be greatly improved if associativity and commutativity are built in the inference rules. The UKB procedure with associativity and commutativity built-in is called *AC-UKB* [2, 61, 56, 31, 44, 11, 27, 46]. If AC denotes 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, written $s =_{AC} t$. In the AC-UKB procedure, the inference rules of UKB 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 $\succ$ *commutes* with $=_{AC}$: 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: s matches a term t *modulo AC* if there is a substitution σ such that $s\sigma =_{AC} t$ and s and t unify *modulo AC* if there is a substitution σ such that $s\sigma =_{AC} t\sigma$. Finally, the strict encompassment ordering $\mathrel{\cdot\!\!\!\rhd}$ is replaced by the ordering $\mathrel{\cdot\!\!\!\rhd}_{AC}$, that is $s \mathrel{\cdot\!\!\!\rhd}_{AC} t$ if and only if $s \mathrel{\cdot\!\!\!\rhd} r$ and $r =_{AC} t$ for some term r. The set of inference rules of the UKB procedure is then modified as follows:

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

$$\frac{(E \cup \{p \simeq q, l \simeq r\}; \hat{s} \simeq \hat{t})}{(E \cup \{p[r\sigma]_u \simeq q, l \simeq r\}; \hat{s} \simeq \hat{t})} \quad \begin{array}{ll} p|u =_{AC} l\sigma & \qquad p \succ p[r\sigma]_u \\ p \mathrel{\cdot\!\!\!\rhd}_{AC} l \vee q \succ p[r\sigma]_u & \end{array}$$

  - *Deduction*:

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

  - *Extension*:

$$\frac{(E \cup \{f(p,q) \simeq r\}; \hat{s} \simeq \hat{t})}{(E \cup \{f(p,q) \simeq r, f(p,q,z) \simeq f(r,z)\}; \hat{s} \simeq \hat{t})} \quad f \textit{ is } AC \quad f(p,q) \not\preceq r$$

  - *Deletion*:

$$\frac{(E \cup \{l \simeq l\}; \hat{s} \simeq \hat{t})}{(E; \hat{s} \simeq \hat{t})}$$

  - *Functional subsumption*:

$$\frac{(E \cup \{p \simeq q, l \simeq r\}; \hat{s} \simeq \hat{t})}{(E \cup \{l \simeq r\}; \hat{s} \simeq \hat{t})} \quad (p \simeq q) \mathrel{\cdot\!\!\!\rhd}_{AC} (l \simeq r)$$

- *Target inference rules*:
  - *Simplification*:

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

– *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* [56]. 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 $f(p,f(q,z)) \simeq f(r,z)$, written in *flattened* form as $f(p,q,z) \simeq f(r,z)$. These critical pairs are called *extended rules*. Computing the extended rules is sufficient to ensure completeness of the AC-UKB procedure: no other critical pairs between E and AC need to be computed [56].

The UKB or AC-UKB procedure can be further improved by adding 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 [39]:

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:

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

Cancellation 3:

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

Cancellation 4:

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

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 $d_1\sigma \simeq x$. *Cancellation 4* is not necessary for the purpose of completeness, since the same effect can be obtained by a step of *Cancellation 1* with empty mgu followed by a step of Functional subsumption. It is added to improve the efficiency.

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 as proof ordering a slight modification of $>_u$, which we call $>_{uc}$: 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 $\succ$, $\mathrel{\triangleright\!\!\!\cdot\,}$, $\mathrel{\triangleright\!\!\!\cdot\,}$ and $\succ$. Proofs are compared by the lexicographic combination $>_{uc}$ of the multiset extensions $\succ_{mul}$ and ${>^{ec}}_{mul}$. The proof of Lemma 5.1 is unaffected if $>_{UKBC}$ replaces $>_{UKB}$.

Lemma 5.3 *The* Cancellation *inference rules are proof-reducing.*

Proof: we assume that $(E_i; \hat{s}_i \simeq \hat{t}_i) \vdash_{UKB} (E_{i+1}; \hat{s}_i \simeq \hat{t}_i)$ is a Cancellation step:

1. 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 $\tau \mathrel{\underline{\gtrdot}} \sigma$, where $\mathrel{\underline{\gtrdot}}$ 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 $f(p, u)\tau \mathrel{\triangleright\!\!\!\cdot\,} p\tau$, $(s, f(p, u)\tau, f(p, u), t) >^{ec} (s, p\tau, p\sigma, t)$ follows. A symmetric argument applies if $t \succ s$.

2. 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 $\tau \mathrel{\underline{\gtrdot}} \sigma$, 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 $f(d_1, d_2)\tau \mathrel{\triangleright\!\!\!\cdot\,} d_1\tau$, $(s, f(d_1, d_2)\tau, f(d_1, d_2), t) >^{ec} (s, d_1\tau, d_1\sigma, t)$ follows.

3. 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 $\tau \mathrel{\underline{\gtrdot}} \sigma$, 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 $r_1\tau = r_2\tau$. 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.

   (a) 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 $f(p_1, q_1)\tau \mathrel{\triangleright\!\!\!\cdot\,} p_1\tau$, the result follows. A symmetric argument applies if $t \succ s \succ u$.

   (b) 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 $f(p_1, q_1)\tau \mathrel{\triangleright\!\!\!\cdot\,} p_1\tau$, the result follows. A symmetric argument applies if $t \succ u \succ s$.

   (c) 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 $u \succ t \succ s$. □

Completeness of the inference rules for cancellation is proved in [39]. Most of the experimental results reported in [2, 1, 16, 3, 5] are obtained by AC-UKB with the inference rules for cancellation.

5.3 The Inequality Ordered-Saturation strategy

The UKB procedure is complete, but sometimes it is not sufficiently efficient. The main cause of inefficiency of UKB, from a theorem proving point of view, is that it often computes 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. Such search plans are 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 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 always maintains a minimal target. Therefore, it would seem that no improvement can be obtained on the target side. However, this is not the case. The notion of a minimal target is relative to the assumed partially ordered set (poset) of targets. If we assume that the poset of ground equalities is ordered by $\succ_{mul}$, then $\hat{s}_i \simeq \hat{t}_i$ 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 $\succ_{mul}$.

We show why the backward reasoning part of UKB is not guaranteed to compute a minimal target if the poset of disjunctions is used. 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 proving the input target $s_0 \simeq t_0$. Also, the application of such a rule must be restricted, in order to avoid the generation of too many 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 not greater than any already existing target:

Ordered saturation:

$$\frac{(E \cup \{l \simeq r\}; N \cup \{\hat{s} \simeq \hat{t}\})}{(E \cup \{l \simeq r\}; N \cup \{\hat{s} \simeq \hat{t}, \hat{s}' \simeq \hat{t}'\})} \quad \hat{s}|u = l\sigma \quad \begin{array}{l} \hat{s}[r\sigma]_u \rightarrow^*_E \hat{s}',\ \hat{t} \rightarrow^*_E \hat{t}' \\ \{\hat{s}', \hat{t}'\} \not\succeq_{mul} \{\hat{g}, \hat{d}\},\ \forall \hat{g} \simeq \hat{d} \in N \cup \{\hat{s} \simeq \hat{t}\} \end{array}$$

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. If the ordering $\succ$ is total on ground terms, the condition $\{\hat{s}', \hat{t}'\} \not\succeq_{mul} \{\hat{g}, \hat{d}\}$, $\forall \hat{g} \simeq \hat{d} \in N \cup \{\hat{s} \simeq \hat{t}\}$ becomes $\{\hat{s}', \hat{t}'\} \prec_{mul} \{\hat{g}, \hat{d}\}$, $\forall \hat{g} \simeq \hat{d} \in N \cup \{\hat{s} \simeq \hat{t}\}$. We have given the inference rule for the more general case: in fact, the ordering is not assumed to be total in [3], where a version of this inference rule first appeared. 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 succeeds 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 that the ordering $\succ$ is total on ground terms, coherently with the treatment of the other completion procedures for equational logic. Then, we order proofs as follows: the proof of a disjunction N is represented by the proof of the smallest equality in N, i.e. $min(N)$, and proofs of equalities are ordered by $>_u$.

Lemma 5.4 *The* Ordered saturation *inference rule under a total ordering on terms is strictly proof-reducing.*

Proof: if $(E_i; N_i) \vdash_{IOS} (E_i; N_{i+1})$ is an Ordered saturation step, then $N_i \subset N_{i+1}$ and therefore $min(N_i) \succeq_{mul} min(N_{i+1})$. Since the ordering $\succ$ is total on ground terms, we have $min(N_i) \succ_{mul} min(N_{i+1})$ and Ordered saturation is strictly proof-reducing. □

Theorem 5.3 *The Inequality Ordered-Saturation strategy is a completion procedure.*

Proof: it follows from Theorem 5.1 and Lemma 5.4. □

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.

5.4 The S-strategy

The *S-strategy* [38] 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 succeeds 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*:
  $$\frac{(E \cup \{p \simeq q, l \simeq r\}; N)}{(E \cup \{p[r\sigma]_u \simeq q, l \simeq r\}; N)} \quad \begin{array}{ll} p|u = l\sigma & \qquad\qquad p \succ p[r\sigma]_u \\ p \mathrel{\triangleright\!\!\!\cdot}\, l \vee q \succ p[r\sigma]_u & \end{array}$$
  - *Deduction*:
  $$\frac{(E \cup \{p \simeq q, l \simeq r\}; N)}{(E \cup \{p \simeq q, l \simeq r, p[r]_u\sigma \simeq q\sigma\}; N)} \quad \begin{array}{ll} \qquad p|u \notin X & \qquad (p|u)\sigma = l\sigma \\ p\sigma \not\preceq q\sigma, p[r]_u\sigma & \end{array}$$
  - *Deletion*:
  $$\frac{(E \cup \{l \simeq l\}; N)}{(E; N)}$$
  - *Functional subsumption*:
  $$\frac{(E \cup \{p \simeq q, l \simeq r\}; N)}{(E \cup \{l \simeq r\}; N)} \quad (p \simeq q) \mathrel{\triangleright\!\!\!\cdot}\, (l \simeq r)$$
- *Target inference rules*:
  - *Simplification*:
  $$\frac{(E \cup \{l \simeq r\}; N \cup \{s \simeq t\})}{(E \cup \{l \simeq r\}; N \cup \{s[r\sigma]_u \simeq t\})} \quad s|u = l\sigma \quad s \succ s[r\sigma]_u$$
  - *Deduction*:
  $$\frac{(E \cup \{l \simeq r\}; N \cup \{s \simeq t\})}{(E \cup \{l \simeq r\}; N \cup \{s \simeq t, s[r]_u\sigma \simeq t\sigma\})} \quad \begin{array}{ll} \quad s|u \notin X & \qquad (s|u)\sigma = l\sigma \\ s\sigma \not\preceq s[r]_u\sigma & \end{array}$$
  - *Deletion*:
  $$\frac{(E; N \cup \{s \simeq t\})}{(E; \mathit{true})} \quad s\sigma = t\sigma$$

The *Deduction* inference rule applies also to target 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 measure the complexity of proofs of disjunctions, we observe the following: a target N is a theorem of E if and only if $E \cup \neg N$ is unsatisfiable, where N is a disjunction of equations $s_0 \simeq t_0 \vee \ldots \vee s_n \simeq t_n$ with existentially quantified variables and therefore $\neg N$ 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 [19], the set $E \cup \neg N$ is unsatisfiable if and only if there is a finite set of ground instances of clauses in $E \cup \neg N$ which is unsatisfiable. Since $\neg N$ is a set of inequalities with universally quantified variables, an unsatisfiable set of ground instances of clauses in $E \cup \neg N$ needs to contain just one ground inequality: let $\hat{E} \cup \{\hat{s} \neq \hat{t}\}$ be the smallest such set with respect to the ordering $\succ_{mul}$. Since $\succ$ is total on ground terms, there exists a smallest set. Then the minimal proof of N in E is represented by the minimal ground equational proof of $\hat{s} \simeq \hat{t}$ in $\hat{E}$. Ground equational proofs are ordered by the ordering $>_u$. This approach is correct if to every inference step on $(\hat{E}_i; \hat{s}_i \simeq \hat{t}_i)$ corresponds an inference steps on $(E_i; N_i)$. This is proved by the *Paramodulation Lifting Lemma* for S-strategy. We recall that a ground substitution is E-irreducible if it does not contain any pair $\{x \mapsto t\}$ such that t can be simplified by an equation in E:

Lemma 5.5 (Peterson 1983) [57], (Hsiang and Rusinowitch 1987) [40] *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 ground instances of E'' and $s'' \simeq t''$, respectively.*

Since $\hat{E} \cup \{\hat{s} \simeq \hat{t}\}$ is the smallest unsatisfiable set, $\hat{E} \subseteq E\sigma$ and $\hat{s} \simeq \hat{t} \in N\sigma$ for an E-irreducible substitution σ. Therefore, Lemma 5.5 applies and to every inference step on $(\hat{E}; \hat{s} \simeq \hat{t})$ corresponds an inference step on $(E; N)$. We can finally state the following theorem:

Theorem 5.4 *The S-strategy is a completion procedure on the domain $\mathcal{T}$ of all ground equalities.*

Proof: soundness and relevance were proved in [38]. By the above discussion on the complexity of proofs of disjunctions, an inference step on $(E; N)$ is proof-reducing if it is proof-reducing on the minimal proof of $\hat{s} \simeq \hat{t}$ in $\hat{E}$. Thus, the inference rules of S-strategy are proof-reducing if they are proof-reducing on ground equational proofs with respect to the ordering $>_u$. This follows from Lemma 5.1 and Lemma 5.2, since Deduction on the target is just 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 5.5 (Hsiang and Rusinowitch 1987) [38] *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 $(E; s_0 \simeq t_0 \vee \ldots \vee s_n \simeq t_n)$.*

6 Semidecision procedures for disproving inductive theorems

The Knuth-Bendix completion procedure has also been applied to *disprove inductive theorems* in equational theories. This method has been called *inductionless induction*, *proof by consistency* or *proof by the lack of inconsistency* [55, 33, 43, 54, 32, 49, 50, 9, 45] and extensions to Horn logic with equality are explored in [52]. This application of completion has also been viewed traditionally as a special side-effect of the generation of confluent systems. In this section we show that a completion procedure for disproving inductive theorems is a *semidecision procedure*. This adds to the generality of our approach based on the semidecision concept, by showing that it covers also inductionless induction. Furthermore, it gives to inductionless induction first class status, showing that it is a semidecision procedure like all other completion procedures, rather than an almost accidental side-effect of generating confluent systems.

From a technical point of view, this section has two new contributions. The first one is the construction of the target of an inductionless induction derivation. Identifying the target is necessary to describe inductionless induction as a semidecision process. While it is not technically difficult, this task is not trivial, because the given inductive conjecture is being disproved, not proved, and therefore it is not the target. The second technical contribution is related to the concept of fairness. The classical results on inductionless induction assumed uniform fairness, so that proof reduction was applied to all the ground proofs. We identify the set of minimal ground proofs of the target in an inductionless induction derivation: such set is smaller than the set of all the ground proofs. Similar to refutational theorem proving, also inductionless induction simply requires that one of the proofs of the target be reduced. Therefore, fairness replaces uniform fairness and not all critical pairs need to be generated when disproving inductive theorems. A completion procedure for inductionless induction which does not compute all critical pairs was proposed in [32]. We conclude this section by re-formulating the result of [32] as a concrete instance of our general approach.

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, $G(\varphi) = \{\varphi\sigma|\ \sigma\ ground\}$. 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:

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

If there exists an oracle $\mathcal{O}$ to decide such target, a completion procedure $\mathcal{C} =< I; \Sigma; \mathcal{O} >$ 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

$$\Phi_i = \exists\sigma \text{ } ground \text{ } \exists\psi \in S_i \text{ } such \text{ } that \text{ } \psi\sigma \in GTh(S_i) - GTh(S).$$

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, the target is

$$\Phi_i = \exists\sigma \text{ } ground \text{ } \exists s_i \simeq t_i \in E_i \text{ } such \text{ } that \text{ } (s_i \simeq t_i)\sigma \in GTh(E_i) - GTh(E).$$

Oracles to decide Φ_i are known if the input set of equations E is ground confluent. Under this hypothesis, $(s_i \simeq t_i)\sigma \in GTh(E_i) - GTh(E)$ if and only if there are 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 [43] for equational presentations satisfying the *principle of definition*. The signature F of E is given by the disjoint union of a set C of *constructors* and a set D of *defined symbols*. 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 $t \leftrightarrow_E^* t'$. If a presentation E satisfies the principle of definition, the ground E-irreducible terms are the ground terms made only of constructor symbols. Therefore, Φ_i is true if and only if there are two ground constructor terms t_1 and t_2 such that $t_1 \leftrightarrow_{E_i}^* t_2$. The following inference rules implement this test [43]:

- *Disproof 1*:
  $$\frac{(E \cup \{f(t_1 \ldots t_n) \simeq g(s_1 \ldots s_n)\}; \Phi)}{(E \cup \{f(t_1 \ldots t_n) \simeq g(s_1 \ldots s_n)\}; true)} \quad f, g \in C, \ f \neq g$$
- *Disproof 2*:
  $$\frac{(E \cup \{f(t_1 \ldots t_n) \simeq x\}; \Phi)}{(E \cup \{f(t_1 \ldots t_n) \simeq x\}; true)} \quad f \in C$$
- *Decompose*:
  $$\frac{(E \cup \{f(t_1 \ldots t_n) \simeq f(s_1 \ldots s_n)\}; \Phi)}{(E \cup \{t_1 \simeq s_1 \ldots t_n \simeq s_n\}; \Phi')} \quad f \in C$$

where Φ' is Φ_i for $E_i = E \cup \{t_1 \simeq s_1 \ldots t_n \simeq s_n\}$. The two *Disproof* inference rules detect that equalities between constructor terms have been derived. By the principle of definition, the theory of E_0 does not include such equalities. They are a consequence of adding the inductive conjecture $s \simeq t$, which is therefore disproved. The *Decompose* rule is added for the purpose of efficiency. It replaces an equation $f(t_1 \ldots t_n) \simeq f(s_1 \ldots s_n)$, where f is a constructor, by the equations $t_1 \simeq s_1 \ldots t_n \simeq s_n$: since f is a constructor, by the principle of definition two terms $f(t_1 \ldots t_n)$ and $f(s_1 \ldots s_n)$ may be equal only if their arguments are equal.

Theorem 6.1 (Huet and Hullot 1982) [43], (Bachmair 1988) [9] *If E is a ground confluent equational presentation, satisfying the principle of definition, the Unfailing Knuth-Bendix completion procedure enriched with the inference rules* Decompose*,* Disproof 1 *and* Disproof 2 *is a semidecision procedure for the complement of $Ind(E)$.*

A more general oracle was proposed in [45] for the Knuth-Bendix completion procedure and extended to the UKB procedure in [9]. 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 [58] only if E is a ground confluent rewrite system. Therefore, the test in [45, 9] applies only if the input presentation E is ground confluent and all the input equations can be oriented into rewrite rules. We assume that E has these properties and we call it R. An equation $l \simeq r$ is *ground* R*-reducible* if for all ground substitutions σ, such that $l\sigma$ and $r\sigma$ are distinct, either $l\sigma$ or $r\sigma$ is R-reducible. If an equation $l \simeq r$ which is not ground R-reducible is derived from $R \cup \{s \simeq t\}$ at stage i, there is a ground instance $l\sigma \simeq r\sigma$ of the equation such that $l\sigma$ and $r\sigma$ are distinct and R-irreducible, but $l\sigma \simeq r\sigma \in GTh(E_i)$. This means that Φ_i is true and the inductive conjecture is disproved. The following inference rule implements this test [9]:

Disproof 3:

$$\frac{(E \cup \{l \simeq r\}; \Phi)}{(E \cup \{l \simeq r\}; true)} \; l \simeq r \; is \; not \; ground \; R - reducible$$

Theorem 6.2 (Jouannaud and Kounalis 1986) [45], (Bachmair 1988) [9] *If R is a ground confluent rewrite system, the Unfailing Knuth-Bendix completion procedure enriched with the inference rule* Disproof 3 *is a semidecision procedure for the complement of $Ind(R)$.*

Both Theorem 6.1 and Theorem 6.2 assume a uniformly fair search plan on the domain of all ground equations. The ground reducibility test is not a practical solution to the problem of inductive theorem proving, because its complexity is very high. Furthermore, most results about the UKB procedure for disproving inductive conjectures have been obtained in contexts where completion was considered a generator of confluent systems and the capability of disproving inductive conjectures was regarded as a side effect. This explains why a uniformly fair search plan was assumed. On the other hand, disproving an inductive conjecture is a semidecision process of a specific target. Therefore, only a proof of the target needs to be reduced. We define the set of the minimal proofs 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 a minimal proof of Φ_i is given by a minimal 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 a proof of the target is a 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 6.1 (Fribourg 1986) [32] *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 $\rho \rightarrow^*_E \sigma\tau$.*

For instance, if E includes the axioms $0+x \simeq x$ and $succ(x)+y \simeq succ(x+y)$, a set of substitutions H is E-inductively complete if it contains two substitutions σ_1 and σ_2 such that $\{x \mapsto 0\} \in \sigma_1$

and $\{x \mapsto succ(y)\} \in \sigma_2$. Indeed, all ground terms reduce either to 0 or to some term $succ^n(0)$, so that a set of substitutions which covers the instances $x \mapsto 0$ and $x \mapsto succ(y)$ cover all instances.

We are interested in *minimal* E-inductively complete sets of substitutions. All such sets are equivalent for our purposes, since they all have the property of covering all the ground substitutions. 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 $\mathcal{IT}_E = \{(l \simeq r)\sigma\tau \mid \sigma \in H_E, (l \simeq r)\sigma\tau \textit{ is ground}\}$. A minimal proof of the target is a minimal 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. This result was first proved in [32] for the application of Knuth-Bendix completion to disprove inductive theorems in equational theories:

Theorem 6.3 (Fribourg 1986) [32] *A completion procedure $\mathcal{C} =< I; \Sigma; \mathcal{O} >$ 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 considerably, while still preserving the completeness of UKB as semidecision procedure to disprove inductive conjectures [32]. At stage i of the derivation, superpositions on $p \simeq q$ at position u are performed only if the set of mgus $\{\sigma | l\sigma = (p|u)\sigma, l \simeq r \in E_i\}$ is E-inductively complete. A position u with this property is called *completely superposable* in [32]. Furthermore, for all equations $p \simeq q$, generated during the derivation, it is sufficient to perform superpositions on just one completely superposable position in $p \simeq q$. In other words, a search plan which selects just one completely superposable position in every equation is fair. These modifcations require 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 $l\sigma \succ r\sigma$. The problem of detecting completely superposable positions reduces to 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 restriction to completely superposable positions can be applied in practice to presentations satisfying the principle of definition.

7 Conclusions

We described an 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 *target-oriented proof reduction* with respect to a well-founded proof ordering.

A completion procedure is given by a set of *inference rules* and a *search plan.* The role of the search plan is often overlooked in the literature, where most theorem proving strategies are presented by giving a set of non-deterministic inference rules only and leaving the task of designing a suitable search plan to the implementation phase. This is not satisfactory, since the search plan is what ultimately turns a set of inference rules into a procedure. The actual performance of the prover depends heavily on the search plan. We tried to emphasize the role of the search plan throughout our work.

The key property of a search plan is *fairness*. Intuitively, fairness of a search strategy means that every inference step which needs to be considered will eventually be considered. In completion-based methods, this usually means resolving all potential critical pairs. In theorem proving, on the other hand, one is not interested in critical pairs which do not contribute to proving the target theorem. Thus, in theorem proving applications fairness does not require resolving all possibile conflicts but only those which may lead to a proof. Our definition of fairness is the first definition of fairness for completion procedures which incorporates this idea. By focusing on the given target, it makes possible to design fair search strategies which ignore the majority of possible critical pairs.

Correspondingly, at the inference rules level, we gave target-oriented notions of *contraction*, *redundancy* and *refutational completeness*. If the inference rules are *refutationally complete* and the search plan is *fair*, a completion procedure is a semidecision procedure for theorem proving. This result makes the interpretation of completion procedures as semidecision procedures independent from the interpretation as generators of confluent systems: a completion procedure can be a semidecision procedure without being a generator of confluent systems.

If the search plan is *uniformly fair*, a completion procedure generates a *saturated* presentation which, under additional hypotheses, may act as a *decision procedure* for the validity of the theorems in the theory. We provided a new, general formulation for these results, covering the classical theorems in equational logic and their extensions to Horn logic with equality in [52, 15, 30]. Most authors conceived completion as a *compilation* process to generate decision procedures. Theorem proving is then regarded as a two-phase process: first compile the given presentation into a finite saturated one and next prove theorems in the saturated presentation. We feel that this approach is not adequate in practice, since in most cases the saturated presentation is infinite. We prefer to interpret completion as a semidecision procedure, focusing on theorem proving as its main application.

We 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. We also showed that the process of disproving inductive theorems by the so called *inductionless induction* method is a semidecision process.

This work raises several open problems, which may be pursued by further research. At the logic level, one possible direction is the full extension of our approach to completion procedures for first order logic with equality. This would require the study of the structure of proofs and proof orderings in such larger logics. The most important challenge, however, is the design of search plans that are fair, but not uniformly fair. We posed the question of theorem proving without saturation and we provided a theoretical framework where theorem proving without saturation is possible, but we did not exhibit a new, concrete strategy which realizes our idea. Borrowing the terminology of algorithm analysis, replacing uniform fairness by fairness is analogous to lowering the known lower bound for a problem: the notion of fairness says how much work is *necessary* (and sufficient) to do theorem proving. Exhibiting a new, more efficient strategy is analogous to the complementary task of lowering the known upper bound for the problem.

After designing a fair but not uniformly fair search plan, one would like to be able to compare it to a uniformly fair search plan and show that the former is more efficient. Thus, another open problem is to find ways of analyzing and comparing different theorem proving strategies, or more precisely different search plans for a given inference system, a priori of implementation. This leads to the even broader question of providing a theory of *strategy analysis*. Classical algorithm analysis is not appropriate for analyzing theorem proving strategies such as those we considered. Indeed, algorithms analysis is for algorithms, whereas theorem proving strategies are semidecision procedures for a semidecidable problem.

Furthermore, even if we could restrict the analysis to a set of input theorems where a theorem proving strategy is guaranteed to halt, the application of algorithm analysis techniques would be far from obvious. Roughly speaking, classical algorithm analysis techniques compute the number of steps of an algorithm based on knowledge of the initial state of the data set, the final state of the data set and the algorithm. For instance, when analyzing a sorting algorithm, one knows that the final state is a sorted list. One difficulty in applying step counting techniques to theorem proving strategies is that the final state of the data set is unknown. Saying that the final state of the data set for a successful theorem proving derivation is the set containing only the empty clause does not help, because such final state is the same for all inputs, it is not related to any property of the input, e.g. size. From the point of view of logic, all unsatisfiable sets are equivalent and equivalent to the set containing only the empty clause, but they are not equivalent from the point of view of computational complexity. In theorem proving, the final state of the data set cannot be known a priori of execution; because if it were known, a proof would be known and the theorem proving problem would be solved already. Techniques designed for algorithms transforming data do not apply directly to strategies searching for solutions. The entire field of complexity of search is largely unexplored.

References

[1] 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.

[2] S.Anantharaman and J.Mzali, Unfailing Completion modulo a set of equations, Technical Report, LRI, Université de Paris Sud, 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 Sys-*

tems for Symbolic Computation, Capri, Italy, April 1990, Springer Verlag, Lecture Notes in Computer Science 429, 184–193, 1990.

[5] S.Anantharaman, M.P.Bonacina, An Application of the Theorem Prover SBR3 to Many-valued Logic, in M.Okada and S.Kaplan (eds.), *Proceedings of the Second International Workshop on Conditional and Typed Term Rewriting Systems*, Montréal, Canada, June 1990, Springer Verlag, Lecture Notes in Computer Science 516, 156–161, 1991.

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

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

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

[9] 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.

[10] L.Bachmair, N.Dershowitz, Critical Pair Criteria for Completion, *Journal of Symbolic Computation*, Vol. 6, No. 1, 1–18, August 1988.

[11] L.Bachmair, N.Dershowitz, Completion for rewriting modulo a congruence, *Theoretical Computer Science*, Vol. 67, No. 2 & 3, 173–202, October 1989.

[12] 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.

[13] L.Bachmair, H.Ganzinger, On Restrictions of Ordered Paramodulation with Simplification, in M.E.Stickel (ed.), *Proceedings of the Tenth International Conference on Automated Deduction*, Kaiserslautern, Germany, July 1990, Springer Verlag, Lecture Notes in Artificial Intelligence 449, 427–441, 1990.

[14] L.Bachmair, N.Dershowitz, Equational inference, canonical proofs and proof orderings, *Journal of the ACM*, Vol. 41, No. 2, 236–276, March 1994.

[15] L.Bachmair, H.Ganzinger, Completion of First-Order Clauses with Equality by Strict Superposition, in M.Okada and S.Kaplan (eds.), *Proceedings of the Second International Workshop on Conditional and Typed Term Rewriting Systems*, Montréal, Canada, June 1990, Springer Verlag, Lecture Notes in Computer Science 516, 162–180, 1991.

[16] 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.

[17] M.P.Bonacina, J.Hsiang, On Rewrite Programs: Semantics and Relationship with Prolog, *Journal of Logic Programming*, Vol. 14, No. 1&2, 155-180, October 1992.

[18] M.P.Bonacina, Sulla dimostrazione di teoremi per completamento, (in Italian, English version: On completion theorem proving), Thesis of "Dottorato di Ricerca", Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano, Milano, Italy, January 1991.

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

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

[21] N.Dershowitz, Orderings for term-rewriting systems, *Theoretical Computer Science*, Vol. 17, 279–301, 1982.

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

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

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

[25] N.Dershowitz, Completion and its Applications, in H.Aït-Kaci, M.Nivat (eds.), *Resolution of Equations in Algebraic Structures*, Vol. II: Rewriting Techniques, 31–86, Academic Press, New York, 1989.

[26] N.Dershowitz, D.A.Plaisted, Equational Programming, in J.E.Hayes, D.Michie and J.Richards (eds.), *Machine Intelligence 11: The logic and acquisition of knowledge*, Chapter 2, 21-56, Oxford Press, 1988.

[27] N.Dershowitz, J.-P.Jouannaud, Rewrite Systems, Chapter 15, Volume B, *Handbook of Theoretical Computer Science*, North-Holland, 1989.

[28] N.Dershowitz, J.-P.Jouannaud, Notations for Rewriting, Technical Report 478, LRI, Université de Paris Sud, January 1990.

[29] N.Dershowitz, A Maximal-Literal Unit Strategy for Horn Clauses, in M.Okada, S.Kaplan (eds.), *Proceedings of the Second International Workshop on Conditional and Typed Rewriting Systems*, Montréal, Canada, June 1990, Springer Verlag, Lecture Notes in Computer Science 516, 14–25, 1991.

[30] N.Dershowitz, Canonical Sets of Horn Clauses, in *Proceedings of the Eighteenth International Conference on Automata, Languages and Programming*, Madrid, Spain, July 1991, Springer Verlag, Lecture Notes in Computer Science, 1991.

[31] 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.

[32] L.Fribourg, A Strong Restriction to the Inductive Completion Procedure, *Journal of Symbolic Computation*, Vol. 8, No. 3, 253–276, September 1989.

[33] 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*, Les Arcs, France, 1980, Springer Verlag, Lecture Notes in Computer Science 87, 356–373, 1980.

[34] 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.

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

[36] 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.

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

[38] 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.

[39] 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.

[40] J.Hsiang, M.Rusinowitch, Proving Refutational Completeness of Theorem Proving Strategies: the Transfinite Semantic Tree Method, *Journal of the ACM*, Vol. 38, No. 3, 559–587, July 1991.

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

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

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

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

[45] J.-P.Jouannaud, E.Kounalis, Automatic proofs by induction in equational theories without constructors, *Information and Computation*, Vol. 82, No. 1, 1–33, July 1989.

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

[47] 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.

[48] D.Kapur, 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, California, August 1985.

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

[50] 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.

[51] 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.

[52] E.Kounalis, M.Rusinowitch, On Word Problems in Horn Theories, *Journal of Symbolic Computation*, Vol. 11, No. 1 & 2, 113–128, January/February 1991.

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

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

[55] 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.

[56] 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.

[57] 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.

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

[59] M.Rusinowitch, Theorem-proving with Resolution and Superposition, *Journal of Symbolic Computation*, Vol. 11, No. 1 & 2, 21–50, January/February 1991.

[60] R.Socher-Ambrosius, How to Avoid the Derivation of Redundant Clauses in Reasoning Systems, *Journal of Automated Reasoning*, Vol. 9, No. 1, 77–98, August 1992.

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

[62] 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.