Completion is an Instance of Abstract Canonical System Inference

Guillaume Burel, Claude Kirchner

Completion is an Instance of Abstract Canonical System Inference

Guillaume Burel[1] and Claude Kirchner[2]

[1] Ecole Normale Supérieure de Lyon & LORIA*
[2] INRIA & LORIA*

Abstract. Abstract canonical systems and inference (ACSI) were introduced to formalize the intuitive notions of good proof and good inference appearing typically in first-order logic or in Knuth-Bendix like completion procedures.
Since this abstract framework is intended to be generic, it is of fundamental interest to show its adequacy to represent the main systems of interest. This has been done for ground completion (where all equational axioms are ground) but was still an open question for the general completion process.
By showing that the standard completion is an instance of the ACSI framework we close the question. For this purpose, two proof representations, proof terms and proofs by replacement, are compared to built a proof ordering that provides an instantiation adapted to the abstract canonical system framework.

Classification: Logic in computer science, rewriting and deduction, completion, good proof, proof representation, canonicity.

1 Introduction

The notion of good proof is central in mathematics and crucial when mechanizing deduction, in particular for defining useful and efficient tactics in proof assistant and theorem provers. Motivated on one hand by this quest for *good proof* theory and on the other by the profound similarities between many proof search approaches, N. Dershowitz and C. Kirchner proposed in [17, 18] a general framework based on ordering the set of proofs. In this context the best proofs are simply the minimal one. Once one has defined what the best proofs are by the mean of a proof ordering, the next step is to obtain the best presentation of a theory, i.e. the set of axioms necessary for obtaining the best proofs for all the theory, but not containing anything useless.

To formalize this, the notion of *good inference* was introduced by M.P. Bonacina and N. Dershowitz [6]. Given a theory, its canonical presentation is defined as the set of the axioms needed to obtain the minimal proofs. It is general enough to produce all best proofs, leading to a notion of *saturation*, but

* UMR 7503 CNRS-INPL-INRIA-Nancy2-UHP

it does not contain any redundant informations, hence the notion of *contraction*. Presentations, i.e. sets of axioms, are then transformed using appropriate deduction mechanisms to produce this canonical presentation.

This leaded to the Abstract Canonical Systems and Inference (ACSI) generic framework presented in [18, 6].

The ACSI framework got its sources of inspiration from three related points. First, the early works on *Proof orderings* as introduced in [3] and [4] to prove the completeness of completion procedures *a la* Knuth-Bendix. Second, the developments about redundancy [24, 5] to focus on the important axioms to perform further inferences. Last but not least, by the completion procedure [31], central in most theorem proving tools where an equality predicate is used. This procedure has been refined, mainly for two purposes: to have a more specific and thus more efficient algorithm when dealing with particular cases, or to increase the efficiency although remaining general. For the first case, a revue of specific completion procedures for specific algebraic structures can be found in [33]. For the second case, completion has been extended to equational completion [25, 36, 28]; inductionless induction, initiated by J.A. Goguen [21] and D. Musser [35]; and ordered completion [32, 24, 4], to mention only a few. One important application of the completion procedure is rewrite based programming, either based on matching or on unification. The seminal work of J.A. Goguen on OBJ and its various incarnations [22] plays a preeminent role in this class of algebraic languages and has directly inspired CafeOBJ [20], ELAN [8] or Maude [14]. When the operational semantics of the language is based on unification, we find logic programming languages of the Prolog family, where EQLOG [23] is also a preeminent figure. Good syntheses about completion based rewrite programs can be found in [15, 7].

Several works intend to uniform this different completion procedures, and to make it a special case of a more general process. The notion of critical-pair completion procedure was introduced by [10] and covers not only standard completion, but also Buchberger algorithm for Gröbner basis [9, 42] and resolution [37]. Indeed, R. Bündgen shown that Buchberger's algorithm can be simulated by standard completion [11]. This concept of critical-pair completion was categorically formalized by K. Stokkermans [40]. Other generalizations can be found in works of M. Schorlemmer [39], M. Aiguier and D. Bahrami [1] or in the PhD of G. Struth [41], where standard completion, Buchberger's algorithm and resolution are shown to be special instantiation of a non-symmetric completion procedure.

But, even if initially motivated by these three points, the ACSI framework has been developed as a full stand alone theory. This theory provide important abstract results based on basic hypothesis on proofs and a few postulates.

Therefore, a main question remains: is this framework indeed useful? Does this theory allows to *uniformly* understand and prove the main properties of a proof system, centered around the appropriate ordering on proofs?

At the price of a slight generalization of two postulates, it is shown in [12], that good proofs in natural deduction are indeed the cut free proofs as soon as proofs are compared using the ordering induced by beta reduction over the sim-

ply typed lambda-terms. For ground completion, the adequacy of the framework has been shown in [16], leaving the more general question of standard completion open.

This paper proves the adequacy to the framework for the standard completion procedure, generalizing in a non trivial way the result of [16] and showing the usefulness of abstract canonical systems. This brings serious hopes that the ACSI framework is indeed well adapted and useful to uniformly understand and work with other algorithms, in particular all the ones based on critical-pair completion.

The next section will summarize the framework of abstract canonical systems, as defined in [18, 6], and briefly recall the standard completion. Section 3 deals with two representations of proofs in equational logic, namely as proof terms in the rewriting logic [34], and as proof by replacement [3]. We will show how to combine them to keep the tree structure of the first one, and the ordering associated with the second one, which is well adapted to prove the completeness of the standard completion. Finally, in Section 4, we will apply the abstract canonical systems framework to this proof representation to show the completeness of the standard completion. The proofs details are given in the Appendix.

2 Presentation

2.1 Abstract Canonical Systems

The results in this section are extracted from [18, 6], which should be consulted for motivations, details and proofs.

Let $\mathbb{A}$ be the set of all formulæ over some fixed vocabulary. Let $\mathbb{P}$ be the set of all proofs. These sets are linked by two functions: $[\cdot]^{Pm} : \mathbb{P} \to 2^{\mathbb{A}}$ gives the *premises* in a proof, and $[\cdot]_{Cl} : \mathbb{P} \to \mathbb{A}$ gives its *conclusion*. Both are extended to sets of proofs in the usual fashion. The set of proofs built using assumptions in $A \subseteq \mathbb{A}$ is noted by[3]

$$Pf(A) \quad \stackrel{!}{=} \quad \left\{p \in \mathbb{P} : [p]^{Pm} \subseteq A\right\} \ .$$

The framework proposed here is predicated on two *well-founded* partial orderings over $\mathbb{P}$: a *proof ordering* $>$ and a *subproof relation* $\rhd$. They are related by a monotonicity requirement (postulate E). We assume for convenience that the proof ordering only compares proofs with the same conclusion ($p > q \Rightarrow [p]_{Cl} = [q]_{Cl}$), rather than mention this condition each time we have cause to compare proofs.

We will use the term *presentation* to mean a set of formulæ, and *justification* to mean a set of proofs. We reserve the term *theory* for deductively closed presentations:

$$Th\, A \quad \stackrel{!}{=} \quad [Pf(A)]_{Cl} \quad = \quad \{[p]_{Cl} : p \in \mathbb{P},\ [p]^{Pm} \subseteq A\} \ .$$

Theories are monotonic:

[3] $\stackrel{!}{=}$ is used for definitions.

Proposition 1 (Monotonicity). *For all presentations A and B:*

$$A \subseteq B \Rightarrow Th\, A \subseteq Th\, B$$

Presentations A and B are *equivalent* ($A \equiv B$) if their theories are identical: $Th\, A = Th\, B$. In addition to this, we assume the two following postulates:

Postulate A (Reflexivity). *For all presentations A:*

$$A \subseteq Th\, A$$

Postulate B (Closure). *For all presentations A:*

$$Th\ Th\, A \subseteq Th\, A$$

We call a proof *trivial* when it proves only its unique assumption and has no subproofs other than itself, that is, if $[p]^{Pm} = \{[p]_{Cl}\}$ and $p \trianglerighteq q \Rightarrow p = q$, where $\trianglerighteq$ is the reflexive closure of the subproof ordering $\triangleright$. We denote by $\widehat{a}$ such a trivial proof of $a \in \mathbb{A}$ and by $\widehat{A}$ the set of trivial proofs of each $a \in A$.

We assume that proofs use their assumptions (postulate C), that subproofs don't use non-existent assumptions (postulate D), and that proof orderings are monotonic with respect to subproofs (postulate E):

Postulate C (Trivia). *For all proofs p and formulæ a:*

$$a \in [p]^{Pm} \Rightarrow p \trianglerighteq \widehat{a}$$

Postulate D (Subproofs Premises Monotonicity). *For all proofs p and q:*

$$p \trianglerighteq q \Rightarrow [p]^{Pm} \supseteq [q]^{Pm}$$

Postulate E (Replacement). *For all proofs p, q and r:*

$$p \triangleright q > r \Rightarrow \exists v \in Pf([p]^{Pm} \cup [r]^{Pm}).\ p > v \triangleright r$$

We make no other assumptions regarding proofs or their structure. As remarked in [6], the subproof relation essentially defines a tree structure over proof: a "leaf" is a proof with no subproofs but itself, and direct subproofs, i.e. subproofs that are not subproofs of another subproof, can be considered as "subtrees". These trees can be infinitely branching, but their height is finite because of the wellfoundedness of $\triangleright$.

The proof ordering $>$ is lifted to an ordering $\gtrsim$ over presentations:

$$A \gtrsim B \text{ if } A \equiv B \text{ and } \forall p \in Pf(A)\ \exists q \in Pf(B).\ p \geq q\ .$$

We define what a *normal-form proof* is, i.e. one of the minimal proofs of $Pf(Th\, A)$:

$$Nf(A) \stackrel{!}{=} \mu Pf(Th\, A) \stackrel{!}{=} \{p \in Pf(Th\, A)\ :\ \neg\exists q \in Pf(Th\, A).\ p > q\}\ .$$

The *canonical presentation* contains those formulæ that appear as assumptions of normal-form proofs:

$$A^\sharp \quad \stackrel{!}{=} \quad [Nf(A)]^{Pm} \ .$$

So, we will say that A is *canonical* if $A = A^\sharp$.

A presentation A is *saturated* if it supports all possible normal form proofs:

$$Pf(A) \supseteq Nf(A) \ .$$

The set of all *redundant formulæ* of a given presentation A will be denoted as follows:

$$Red\, A \quad \stackrel{!}{=} \quad \{r \in A \colon A \gtrsim A \setminus \{r\}\} \ .$$

and a presentation A is *contracted* if

$$Red\, A = \emptyset \ .$$

The following main result can then be derived [17]:

Theorem 1. *A presentation is canonical iff it is saturated and contracted.*

We now consider inference and deduction mechanisms. A *deduction mechanism* $\rightsquigarrow$ is a function from presentations to presentations and we call the relation $A \rightsquigarrow B$ a *deduction step*. A sequence of presentations $A_0 \rightsquigarrow A_1 \rightsquigarrow \cdots$ is called a *derivation*. The *result* of the derivation is, as usual, its *persisting* formulæ:

$$A_\infty \quad \stackrel{!}{=} \quad \liminf_{j\to\infty} A_j \ = \ \bigcup_{j>0} \bigcap_{i>j} A_i \ .$$

A deduction mechanism $\rightsquigarrow$ is *sound* if $A \rightsquigarrow B$ implies $Th\, B \subseteq Th\, A$. It is *adequate* if $A \rightsquigarrow B$ implies $Th\, A \subseteq Th\, B$. It is *good* if proofs only get better:

$$\rightsquigarrow \ \subseteq \ \gtrsim \ .$$

A derivation $A_0 \rightsquigarrow A_1 \rightsquigarrow \cdots$ is *good* if $A_i \gtrsim A_{i+1}$ for all i.

We now extend the notion of saturation and contraction to derivation:

- A derivation $\{A_i\}_i$ is *saturating* if A_∞ is saturated.
- It is *contracting* if A_∞ is contracted.
- It is *canonical* if both saturating and contracting.

A canonical derivation can be used to build the canonical presentation of the initial presentation:

Theorem 2. *A good derivation is canonical if and only if*

$$A_\infty = A_0^\sharp \ .$$

2.2 The Standard Completion

The standard completion algorithm was first introduced by Knuth and Bendix in [31], hence the name it is often called. Its correctness was first shown by Huet in [26], using a fairness hypothesis. We use here a presentation of this algorithm as inference rules (see Fig. 1), as can be found in [3]. For basics on rewritings and completions, we refer to [2, 29].

The Knuth-Bendix algorithm consists of 6 rules which apply to a couple E, R of a set of equational axioms and a set of rewriting rules. It takes a reduction ordering $\gg$ over terms as argument. The rules are presented in Fig. 1.

Deduce: If (s, t) is a critical pair of R

$$E, R \quad \rightsquigarrow \quad E \cup \{s = t\}, R$$

Orient: If $s \gg t$

$$E \cup \{s = t\}, R \quad \rightsquigarrow \quad E, R \cup \{s \to t\}$$

Delete:

$$E \cup \{s = s\}, R \quad \rightsquigarrow \quad E, R$$

Simplify: If $s \underset{R}{\longrightarrow} u$

$$E \cup \{s = t\}, R \quad \rightsquigarrow \quad E \cup \{u = t\}, R$$

Compose: If $t \underset{R}{\longrightarrow} u$

$$E, R \cup \{s \to t\} \quad \rightsquigarrow \quad E, R \cup \{s \to u\}$$

Collapse[a]**:** If $s \underset{v \to w \in R}{\longrightarrow} u$, and $s \blacktriangleright v$,

$$E, R \cup \{s \to t\} \quad \rightsquigarrow \quad E \cup \{u = t\}, R$$

Fig. 1. Standard Completion Inference Rules.

[a] $\blacktriangleright$ designate the encompassment ordering, $s \blacktriangleright t$ if a subterm of s in an instance of t but not vice versa.

Since [26], standard completion is associated with a fairness assumption (see [3, Lemma 2.8]): at the limit, all equations are oriented ($E_\infty = \emptyset$) and all persistent critical pairs coming from R_∞ are treated by **Deduce** at least once.

Because we work with terms with variables, the reduction ordering $\gg$ cannot be total, so that **Orient** may fail. Therefore, the standard completion algorithm may either:

- terminate with success and yield a terminating, confluent set of rules;
- terminate with failure; or

– not terminate.

Here, the completeness of the standard completion will only be shown using the ACSI framework for the first case.

3 Proof Representations

Our goal is now to use the ACSI framework to directly show that standard completion inference rules are correct and complete. We have therefore first to find the right order on proofs. We have two main choices that we are now defining and relating.

3.1 Proof Terms

Let us first consider the proof representation coming from the one used in rewriting logic (introduced by Meseguer [34], see also [30]). Consider a signature Σ, and a set of variable V. The set of terms built upon these signature and variables is noted $\mathcal{T}(\Sigma, V)$. Consider also a set of equational axioms E and a set of rewrite rules R based on this signature. To simplify the notations of proof terms, equational axioms and rewrite rules are represented by labels not appearing in the signature Σ. An equational axiom or a rewrite rule $(l, r) \in E \cup R$ will be also noted $(l(x_1, \ldots, x_n), r(x_1, \ldots, x_n))$ where $x_1, \ldots, x_n$ are the free variables of both sides. We consider the rules of the equational logic given in the Fig. 2. These inference rules define the *proof term* associated with a proof. The notation $\pi : t \longrightarrow t'$ means that π is a proof term—that could also be seen as a trace—showing that the term t can be rewritten to the term t'.

By definition, $\mathcal{T}(\Sigma, V)$ is plunged into the proof terms when they are formed with the rules **Reflexivity** and **Congruence**. Also, **Reflexivity** for $t \longrightarrow t$ is not essential because it can be replaced by a tree of **Congruence** isomorph to t. The proof terms associated are furthermore the same in both case: t. Notice that these proof terms are a restricted form of rho-terms [13].

Example 1. Consider the rewrite rules and equational axiom

$$\ell_1 : g(x) \longrightarrow d(x), \quad \ell_2 : s = t, \quad \ell_3 : l \longrightarrow r,$$

– r is a proof term of $r = r$,
– $f(\ell_1(\ell_2), (\ell_3; r)^{-1})$ is a proof term of $f(g(s), r) = f(d(t), l)$.

Some proof terms defined here are "essentially the same". For instance, the transitivity operator should be considered as associative, so that the proofs $(\pi_1; \pi_2); \pi_3$ and $\pi_1; (\pi_2; \pi_3)$ are equal. This can be done by quotienting the proof terms algebra by the congruence rules of Fig. 3. In particular, in proof terms, parallel rewriting can be combined in one term without transitivity. The **Parallel Moves Lemma** equivalence corresponds to the fact that this parallel rewriting can be decomposed by applying first the outermost rule, then the innermost, or conversely. (About the Parallel Moves Lemma, see for instance [27].)

Reflexivity:

$$\frac{}{t : t \longrightarrow t}$$

Congruence:

$$\frac{\pi_1 : t_1 \longrightarrow t'_1 \quad \dots \quad \pi_n : t_n \longrightarrow t'_n}{f(\pi_1, \dots, \pi_n) : f(t_1, \dots, t_n) \longrightarrow f(t'_1, \dots, t'_n)}$$

Replacement: For all rules or equational axioms $\ell = (g(x_1, \dots, x_n), d(x_1, \dots, x_n)) \in E \cup R$,

$$\frac{\pi_1 : t_1 \longrightarrow t'_1 \quad \dots \quad \pi_n : t_n \longrightarrow t'_n}{\ell(\pi_1, \dots, \pi_n) : g(t_1, \dots, t_n) \longrightarrow d(t'_1, \dots, t'_n)}$$

Transitivity:

$$\frac{\pi_1 : t_1 \longrightarrow t_2 \quad \pi_2 : t_2 \longrightarrow t_3}{\pi_1; \pi_2 : t_1 \longrightarrow t_3}$$

Symmetry:

$$\frac{\pi : t_1 \longrightarrow t_2}{\pi^{-1} : t_2 \longrightarrow t_1}$$

Fig. 2. Inference Rules for Equational Logic

Example 2. From the rules **Associativity**, **Identities** and **Inverse** we can deduce that the proofs $(\pi_1; \pi_2)^{-1}$ and $\pi_2^{-1}; \pi_1^{-1}$ are equivalent:

$$\begin{aligned}
(\pi_1; \pi_2)^{-1} &\equiv (\pi_1; \pi_2)^{-1}; t \\
&\equiv (\pi_1; \pi_2)^{-1}; \pi_1; \pi_1^{-1} \\
&\equiv (\pi_1; \pi_2)^{-1}; \pi_1; t'; \pi_1^{-1} \\
&\equiv (\pi_1; \pi_2)^{-1}; \pi_1; \pi_2; \pi_2^{-1}; \pi_1^{-1} \\
&\equiv t''; \pi_2^{-1}; \pi_1^{-1} \\
&\equiv \pi_2^{-1}; \pi_1^{-1} \ .
\end{aligned}$$

We similarly have $f(\pi_1, \dots, \pi_n)^{-1}$ equivalent to $f(\pi_1^{-1}, \dots, \pi_n^{-1})$, because

$$\begin{aligned}
f(\pi_1^{-1}, \dots, \pi_n^{-1}) &\equiv f(\pi_1^{-1}, \dots, \pi_n^{-1}); f(t_1, \dots, t_n) \\
&\equiv f(\pi_1^{-1}, \dots, \pi_n^{-1}); (f(\pi_1, \dots, \pi_n); f(\pi_1, \dots, \pi_n)^{-1}) \\
&\equiv (f(\pi_1^{-1}, \dots, \pi_n^{-1}); f(\pi_1, \dots, \pi_n)); f(\pi_1, \dots, \pi_n)^{-1} \\
&\equiv f(\pi_1^{-1}; \pi_1, \dots, \pi_n^{-1}; \pi_n); f(\pi_1, \dots, \pi_n)^{-1} \\
&\equiv f(t'_1, \dots, t'_n); f(\pi_1, \dots, \pi_n)^{-1} \\
&\equiv f(\pi_1, \dots, \pi_n)^{-1} \ .
\end{aligned}$$

3.2 Proofs by Replacement of Equal by Equal

This proof representation was introduced by [3] to prove the completeness of the Knuth-Bendix completion algorithm, using an ordering over such proofs that decreases for every completion step.

Associativity: For all proof terms π_1, π_2, π_3,

$$\pi_1; (\pi_2; \pi_3) \;\equiv\; (\pi_1; \pi_2); \pi_3$$

Identities: For all proof terms $\pi : t \longrightarrow t'$,

$$\pi; t' \;\equiv\; t; \pi \;\equiv\; \pi$$

Preservation of Composition: For all proof terms $\pi_1, \ldots, \pi_n, \pi'_1, \ldots, \pi'_n$, for all function symbols f,

$$f(\pi_1; \pi'_1, \ldots, \pi_n; \pi'_n) \;\equiv\; f(\pi_1, \ldots, \pi_n); f(\pi'_1, \ldots, \pi'_n)$$

Parallel Moves Lemma: For all rewrite rules or equational axiom $\ell = (g(x_1, \ldots, x_n), d(x_1, \ldots, x_n)) \in E \cup R$, for all proof terms $\pi_1 : t_1 \longrightarrow t'_1, \ldots, \pi_n : t_n \longrightarrow t'_n$,

$$\begin{aligned} \ell(\pi_1, \ldots, \pi_n) \;&\equiv\; \ell(t_1, \ldots, t_n); d(\pi_1, \ldots, \pi_n) \\ &\equiv\; g(\pi_1, \ldots, \pi_n); \ell(t'_1, \ldots, t'_n) \end{aligned}$$

Inverse: For all proof terms $\pi : t \longrightarrow t'$,

$$\begin{aligned} \pi; \pi^{-1} \;&\equiv\; t \\ \pi^{-1}; \pi \;&\equiv\; t' \end{aligned}$$

Fig. 3. Equivalence of Proof Terms

An *equational proof step* is an expression $s \underset{e}{\overset{p}{\longleftrightarrow}} t$ where s and t are terms, e is an equational axiom $u = v$, and p is a position of s such that $s_{|p} = \sigma(u)$ and $t = s[\sigma(v)]_p$ for some substitution σ.

An *equational proof* of $s_0 = t_n$ is any finite sequence of equational proof steps $\left(s_i \underset{e_i}{\overset{p_i}{\longleftrightarrow}} t_i\right)_{i \in \{0,\ldots,n\}}$ such that $t_i = s_{i+1}$ for all $i \in \{0, \ldots, n-1\}$. It is noted:

$$s_0 \underset{e_0}{\overset{p_0}{\longleftrightarrow}} s_1 \underset{e_1}{\overset{p_1}{\longleftrightarrow}} s_2 \cdots s_n \underset{e_n}{\overset{p_n}{\longleftrightarrow}} t_n \;\;.$$

A *rewrite proof step* is an expression $s \underset{\ell}{\overset{p}{\longrightarrow}} t$ or $t \underset{\ell}{\overset{p}{\longleftarrow}} s$ where s and t are terms, ℓ is a rewrite rule $u \to v$, and p is a position of s such that $s_{|p} = \sigma(u)$ and $t = s[\sigma(v)]_p$ for some substitution σ.

An *proof by replacement (of equal by equal)* of $s_0 = t_n$ is any finite sequence of equational proof steps and rewrite proof step $\left(s_i \underset{\ell_i}{\overset{p_i}{\leftrightarrows_i}} t_i\right)_{i \in \{0,\ldots,n\}}$ where $\leftrightarrows_i \in \{\longleftrightarrow, \longrightarrow, \longleftarrow\}$ for $i \in \{0, \ldots, n\}$ and such that $t_i = s_{i+1}$ for all $i \in \{0, \ldots, n-1\}$. It is noted:

$$s_0 \underset{\ell_0}{\overset{p_0}{\leftrightarrows_0}} s_1 \underset{\ell_1}{\overset{p_1}{\leftrightarrows_1}} s_2 \cdots s_n \underset{\ell_n}{\overset{p_n}{\leftrightarrows_n}} t_n \;\;.$$

Example 3. Consider the rewrite rules and equational axiom:

$$\ell_1 : g(x) \longrightarrow d(x), \quad \ell_2 : s = t, \quad \ell_3 : l \longrightarrow r,$$

- r is a proof by replacement of $r = r$ (empty sequence),
- $f(g(s), r) \xrightarrow[\ell_1]{1} f(d(s), r) \xleftrightarrow[\ell_2]{11} f(d(t), r) \xleftarrow[\ell_3]{2} f(d(t), l)$ is a proof by replacement of $f(g(s), r) = f(d(t), l)$.

3.3 From Proof Terms to Proofs by Replacement

In order to have a one to one correspondence between proof representations, we use the equivalence of proof terms defined in Fig. 3. We can refine them to the proof term rewrite system $\rightsquigarrow$ given in Fig. 4, in which $\pi, \pi', \pi_1, \ldots$ range over proof terms, $t, t', t_1, \ldots$ over Σ-terms, f, g, d over function symbols, ℓ over rules and equational axioms labels and i and k over $\{1, \ldots, n\}$.

Delete Useless Identities:

$$\left.\begin{array}{c} \pi; t' \\ t; \pi \end{array}\right\} \rightsquigarrow \pi$$

Sequentialization: If $\pi_k : t_k \longrightarrow t'_k$ and there exists $i \neq j \in \{1, \ldots, n\}$ such that $\pi_i \neq t_i$ and $\pi_j \neq t_j$,

$$f(\pi_1, \ldots, \pi_n) \rightsquigarrow f(\pi_1, t_2, \ldots, t_n); f(t'_1, \pi_2, \ldots, t_n); \ldots; f(t'_1, t'_2, \ldots, \pi_n)$$

Composition Shallowing: If $\pi_i : t_i \longrightarrow t'_i$ and $\pi'_i : t'_i \longrightarrow t''_i$,

$$f(t_1, \ldots, \pi_i; \pi'_i, \ldots, t_n) \rightsquigarrow f(t_1, \ldots, \pi_i, \ldots, t_n); f(t_1, \ldots, \pi'_i, \ldots, t_n)$$

Parallel Moves: If $\ell = (g(x_1, \ldots, x_n), d(x_1, \ldots, x_n))$, $\pi_1 : t_1 \longrightarrow t'_1, \ldots, \pi_n : t_n \longrightarrow t'_n$, and if there exists $i \in \{1, \ldots, n\}$ such that $\pi_i \neq t_i$,

$$\ell(\pi_1, \ldots, \pi_n) \rightsquigarrow \ell(t_1, \ldots, t_n); d(\pi_1, \ldots, \pi_n)$$

Delete Useless Inverses:

$$t^{-1} \rightsquigarrow t$$

Inverse Congruence: If $\pi_i : t_i \longrightarrow t'_i$,

$$f(t_1, \ldots, \pi_i^{-1}, \ldots, t_n) \rightsquigarrow f(t_1, \ldots, \pi_i, \ldots, t_n)^{-1}$$

Inverse Composition:

$$(\pi_1; \pi_2)^{-1} \rightsquigarrow \pi_2^{-1}; \pi_1^{-1}$$

Fig. 4. Rewrite System for Proof Terms

The associativity is still considered in the congruence, so that all proof terms rewrite rules must be considered modulo the associativity of ; which will be noted $\sim$. The class rewrite system that we consider will be therefore noted $\rightsquigarrow / \sim$. As it is linear, we can use the framework and results from [25].

We first prove that this rewrite system is included in the equivalence relation of Fig. 3.

Proposition 2 (Correctness). *For all proof terms π_1, π_2, if $\pi_1 \rightsquigarrow \pi_2$ then $\pi_1 \equiv \pi_2$.*

The converse is false: for instance $f(\ell_1, \ell_2) \equiv f(t_1, \ell_2); f(\ell_1, t'_2)$ but we do not have $f(\ell_1, \ell_2) \overset{*}{\underset{\rightsquigarrow}{\leftrightarrow}} f(t_1, \ell_2); f(\ell_1, t'_2)$.

Proposition 3 (Termination and Confluence). *The proof term rewrite system $\rightsquigarrow$ modulo $\sim$ is terminating and confluent modulo $\sim$.*

The proof terms rewrite system $\rightsquigarrow$ allow us to give a correspondence between proof terms and proofs by replacement of equal by equal: normal forms of proof terms correspond exactly to proofs by replacement. This fact is expressed in the following theorem, which is indeed a generalization of Lemma 3.6 in [34] for equational logic. We also have operationalized the way to construct the chain of "one-step sequential rewrites".

Theorem 3 (Correspondence between Proof Representations). *The normal form of a proof term π for the rewrite system $\rightsquigarrow$, noted* $\mathrm{nf}(\pi)$*, has the following form: For some $n \in \mathbb{N}$, some contexts $w_1[], \ldots, w_n[]$, some indices $i_1, \ldots, i_n \in \{-1, 1\}$, some rule labels $\ell_1, \ldots, \ell_n$ and some terms $t^1_1, \ldots, t^1_{m_1}, \ldots, t^n_1, \ldots, t^n_{m_n}$:*

$$\mathrm{nf}(\pi) = (w_1[\ell_1(t^1_1, \ldots, t^1_{m_1})])^{i_1}; \ldots; (w_n[\ell_n(t^n_1, \ldots, t^n_{m_n})])^{i_n}$$

where for all proof terms ν, ν^1 is a notation for ν.

Such a proof term correspond with the following proof by replacement of equal by equal:

$$w_1[g_1(t^1_1, \ldots, t^1_{m_1})] \underset{\ell_1}{\overset{p_1}{\leftrightarrows}}_1 w_1[d_1(t^1_1, \ldots, t^1_{m_1})] \underset{\ell_2}{\overset{p_2}{\leftrightarrows}}_2 \cdots \underset{\ell_n}{\overset{p_n}{\leftrightarrows}}_n w_n[d_n(t^n_1, \ldots, t^n_{m_n})]$$

where for all $j \in \{1, \ldots, n\}$ we have:

- $\ell_j = (g_j, d_j)$,
- p_j *is the position of* $[]$ *in* $w_j[]$,
- $\leftrightarrows_j = \longrightarrow$ *if* $i_j = 1$ *and* $\ell_j \in R$,
  $\longleftarrow$ *if* $i_j = -1$ *and* $\ell_j \in R$,
  $\longleftrightarrow$ *if* $\ell_j \in E$.
- *if* $j \neq n$, $w_j[d_j(t^j_1, \ldots, t^j_{m_j})] = w_{j+1}[g_{j+1}(t^{j+1}_1, \ldots, t^{j+1}_{m_{j+1}})]$.

Example 4. Consider $\pi = f(\ell_1(\ell_2), (\ell_3; r)^{-1})$ where $\ell_1 : g(x) \longrightarrow d(x)$, $\ell_2 : s = t$, $\ell_3 : l \longrightarrow r$, we have:

$$\begin{aligned}
\pi &\underset{\rightsquigarrow}{\longrightarrow} f(\ell_1(s); d(\ell_2), (\ell_3; r)^{-1}) && (\textbf{Parallel Moves})\\
&\underset{\rightsquigarrow}{\longrightarrow} f(\ell_1(s); d(\ell_2), r); f(d(t), (\ell_3; r)^{-1}) && (\textbf{Sequentialization})\\
&\underset{\rightsquigarrow}{\longrightarrow} f(\ell_1(s); d(\ell_2), r); f(d(t), r^{-1}; \ell_3^{-1}) && (\textbf{Inverse Composition})\\
&\underset{\rightsquigarrow}{\longrightarrow} f(\ell_1(s); d(\ell_2), r); f(d(t), r; \ell_3^{-1}) && (\textbf{Delete Useless Inverses})\\
&\underset{\rightsquigarrow}{\longrightarrow} f(\ell_1(s); d(\ell_2), r); f(d(t), \ell_3^{-1}) && (\textbf{Delete Useless Identities})\\
&\underset{\rightsquigarrow}{\longrightarrow} f(\ell_1(s), r); f(d(\ell_2), r); f(d(t), \ell_3^{-1}) && (\textbf{Composition Shallowing})\\
&\underset{\rightsquigarrow}{\longrightarrow} f(\ell_1(s), r); f(d(\ell_2), r); f(d(t), \ell_3)^{-1} && (\textbf{Inverse Congruence})
\end{aligned}$$

This last term is the normal form proof term, and it is equivalent to the proof by replacement $f(g(s), r) \xrightarrow[\ell_1]{1} f(d(s), r) \xleftarrow[\ell_2]{11} f(d(t), r) \xleftarrow[\ell_3]{2} f(d(t), l)$.

Due to this theorem, normal forms of proof terms can be considered in the following indifferently as proof terms or as proofs by replacement.

3.4 Proofs Ordering

The representation of Bachmair by the mean of proof by replacement was defined to introduce an order on proofs [3]: given a reduction ordering $\gg$, to each single proof steps $s \overset{p}{\underset{\ell}{\leftrightarrows}} t$ is associated a *cost*. The cost of an equational proof step $s \xleftarrow[u=v]{p} t$ is the triple $(\{s, t\}, u, t)$. The cost of a rewrite proof step $s \xrightarrow[u \to v]{p} t$ is $(\{s\}, u, t)$. Proof steps are compared with each other according to their cost, using the lexicographic combination of the multiset $\gg_{mult}$ extension of the reduction ordering over terms in the first component, the encompassment ordering $\blacktriangleright$ on the second component, and the reduction ordering $\gg$ on the last component. Proofs are compared as multisets of their proof steps. For two proofs by replacement p, q, we will write $p >_{rep} q$ if p is greater than q for such an ordering.

Using theorem 3, we can translate Bachmair's proof ordering to proof terms:

Definition 1 (Bachmair's Ordering on Proof Terms).
For all proof terms π_1, π_2, we say that $\pi_1 >_B \pi_2$ iff

$$\mathrm{nf}(\pi_1) >_{rep} \mathrm{nf}(\pi_2) \ .$$

Example 5. Suppose we have $\Sigma = \{f^1, a^0, b^0, c^0\}$ where the exponents of function symbols denote their arity, and a precedence $f > a > b > c$.

Consider $\pi_1 = f(\ell_1^{-1}; \ell_2)$ and $\pi_2 = f(\ell_3)$ where $\ell_1 = a \longrightarrow b$, $\ell_2 = a \longrightarrow c$ and $\ell_3 = b = c$, and suppose $a > b > c$.

We have $\mathrm{nf}(\pi_1) = f(b) \xleftarrow[\ell_1]{1} f(a) \xrightarrow[\ell_2]{1} f(c)$ and $\mathrm{nf}(\pi_2) = f(b) \xrightarrow[\ell_3]{1} f(c)$. The cost of $\mathrm{nf}(\pi_1)$ is $\{(\{f(a)\}, a, f(b)), (\{f(a)\}, a, f(c))\}$, the cost of $\mathrm{nf}(\pi_2)$ is $\{(\{f(b), f(c)\}, b, f(c))\}$, so $\mathrm{nf}(\pi_1) >_{rep} \mathrm{nf}(\pi_2)$ and $\pi_1 >_B \pi_2$.

As we can see, the way we define the ordering over proofs is not trivial. The question remains if we could have defined it more directly, without using the representation as proof by replacement. The following statement give a beginning of answer: we cannot hope to extend an RPO on Σ-terms to a RPO[4] $>_{rpo}$ on proof terms so that $>_B$ and $>_{rpo}$ coincide for the normal forms of proof terms:

Counter-example 6. With the same hypothesis as in Example 5, let $\ell_f = f(a) \longrightarrow c$ and $\ell_b = b \longrightarrow c$.

We now want to extend the precedence to ℓ_f and ℓ_b in order to extend the RPO to proof terms. If we have $\ell_f < \ell_b$, $f(a) \xrightarrow[\ell_f]{\epsilon} c >_{rep} b \xrightarrow[\ell_b]{\epsilon} c$ but $\ell_f <_{rpo} \ell_b$.

If we suppose $f > \ell_f > \ell_b$ we have $f(a) \xrightarrow[\ell_f]{\epsilon} c >_{rep} f(b) \xrightarrow[\ell_b]{1} f(c)$ but $\ell_f <_{rpo} f(\ell_b)$.

If we suppose $\ell_f > \ell_b$ and $\ell_f > f$, then $f(f(b)) \xrightarrow[\ell_b]{11} f(f(c)) >_{rep} f(a) \xrightarrow[\ell_f]{\epsilon} c$ but $f(f(\ell_b)) <_{rpo} \ell_f$.

Such an extension is therefore impossible, there is no extension of $>_{rpo}$ on proof terms such that for all proof terms π_1, π_2, we have $\mathrm{nf}(\pi_1) >_{rpo} \mathrm{nf}(\pi_2)$ if and only if $\mathrm{nf}(\pi_1) >_B \mathrm{nf}(\pi_2)$.

In other words, the ordering we defined above can *not* be defined as a RPO over proof terms.

In the following, proofs will be represented by proof terms, the proof ordering $>$ between them will be the ordering $>_B$ restricted to proofs with the same conclusion, and the subproof relation $\rhd$ will be the subterm relation.

4 Standard Completion is an Instance of Abstract Canonical System

4.1 Adequacy to the Postulates

Adequacy to postulates A, B, C and D comes from the tree structure of the proof terms representation.

Postulate E is not trivially verified, because of the definition of the ordering as translation of an ordering over proof by replacement. Nevertheless:

Theorem 4 (Postulate E for Equational Proofs). *For all contexts $w[]$, for all proof terms q, r:*

$$q > r \text{ implies } w[q] > w[r] \ .$$

The deduction mechanism $\rightsquigarrow$ used here will be of course the standard completion. We now show that it has the required properties.

[4] Or better an ordering compatible with associativity, such as the *AC*-RPO [38].

4.2 Standard Completion is Sound and Adequate

This is shown in [3, Lemma 2.1]: if $E, R \rightsquigarrow E', R'$, then $\overset{*}{\underset{E \cup R}{\longleftrightarrow}}$ and $\overset{*}{\underset{E' \cup R'}{\longleftrightarrow}}$ are the same. To prove this, one has simply to verify it for each inference rule of standard completion.

4.3 Standard Completion is Good

This is shown in [3, Lemma 2.5, 2.6]: if $E, R \rightsquigarrow E', R'$, then proofs in E, R can be transformed to proofs in E', R' using following rules:

$$
\begin{array}{rcll}
s \underset{E}{\longleftrightarrow} t & \twoheadrightarrow & s \underset{R'}{\longrightarrow} t & (\textbf{Orient}) \\
s \underset{E}{\longleftrightarrow} t & \twoheadrightarrow & s \underset{R'}{\longrightarrow} u \underset{E'}{\longleftrightarrow} t & (\textbf{Simplify}) \\
s \underset{E}{\longleftrightarrow} s & \twoheadrightarrow & s & (\textbf{Delete}) \\
s \underset{R}{\longleftarrow} u \underset{R}{\longrightarrow} t & \twoheadrightarrow & s \underset{E'}{\longleftrightarrow} t & (\textbf{Deduce}) \\
s \underset{R}{\longleftarrow} u \underset{R}{\longrightarrow} t & \twoheadrightarrow & s \overset{*}{\underset{R'}{\longrightarrow}} v \overset{*}{\underset{R'}{\longleftarrow}} t & \\
s \underset{R}{\longrightarrow} t & \twoheadrightarrow & s \underset{R'}{\longrightarrow} v \underset{R'}{\longleftarrow} t & (\textbf{Compose}) \\
s \underset{R}{\longrightarrow} t & \twoheadrightarrow & s \underset{R'}{\longrightarrow} v \underset{E'}{\longleftrightarrow} t & (\textbf{Collapse})
\end{array}
$$

We have $\underset{\twoheadrightarrow}{\longrightarrow} \subseteq >$, so these proofs become indeed better.

4.4 Standard Completion is Canonical

We can now show the following theorem:

Theorem 5 (Completeness of Standard Completion). *Standard completion results—at the limit, when it terminates without failure—in the canonical, Church-Rosser basis.*

Proof. We can show $R_\infty = E_0^\sharp$, and because standard completion is good we can use Theorem 2.

Remark 1. When standard completion does not terminate, we can show that $E_0^\sharp = R_\infty^\sharp \subseteq R_\infty$. Consequently, the resulting set R_∞ is then *saturated*, but it is not necessarily *contracted*.

This shows that the standard completion is an instance of the framework of the abstract canonical systems, when we choose the convenient proof representation.

5 Conclusion

We presented a proof that standard completion can be seen as an instance of the abstract canonical systems and inference framework. This led us to make precise the relation between different equational proof representations. The first one, proof terms as presented in [34], is convenient to consider proofs as terms, with a subterm relation and substitutions. The other one, initiated in [3], is well adapted to the study of the completeness of the standard completion procedure. We presented a way to pass from one representation to another by the mean of the proof term rewrite rules presented in Fig. 4. Thanks to this, we extended the ordering introduced with the proof by replacement to the proof terms and thus combine the advantages of both representations. This therefore positively answer to the question whether the abstract canonical systems, centered in a quite general way around the notion of proof ordering, are indeed the right framework to uniformly prove the completeness of completion.

We plan now to understand how the results we have presented here can be extended to other completion procedures. Bachmair introduced another proof ordering to prove the completeness of the completion modulo [3], so that the generalization seems rather natural. We plan also to look at other kinds of deduction mechanisms, such as Buchberger's algorithm or resolution. For this, we may show that Struth's non-symmetric completion [41], which subsumes both procedures, is also an instance of the ACSI framework.

Furthermore, proof terms as presented by [34, 30] are specific terms of the rewriting calculus [13] [`http://rho.loria.fr`]. The link between the completion procedure and the sequent systems mentioned above can probably be found here and be related to Dowek's work proving that confluent rewrite rules can be linked with **Cut**-free proofs of some sequent systems [19].

Acknowledgments This paper benefited greatly from suggestions, discussions and the enthusiasm of Nachum Dershowitz. We thank also Georg Struth for his useful comments and the anonymous referees for their careful reading and constructive suggestions.

References

[1] M. Aiguier and D. Bahrami. Structures for abstract rewriting. *Journal of Automated Reasoning*, 2006. To appear.
[2] F. Baader and T. Nipkow. *Term Rewriting and all That*. Cambridge University Press, 1998.
[3] L. Bachmair. *Proof methods for equational theories*. PhD thesis, University of Illinois, Urbana-Champaign, (Ill., USA), 1987. Revised version, August 1988.
[4] L. Bachmair and N. Dershowitz. Equational inference, canonical proofs, and proof orderings. *Journal of Association for Computing Machinery*, 41(2):236–276, 1994.
[5] L. Bachmair and H. Ganzinger. Resolution theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 2, pages 19–99. Elsevier Science, 2001.

[6] M. P. Bonacina and N. Dershowitz. Abstract Canonical Inference. *ACM Transactions on Computational Logic*, 2006. To appear.
[7] M. P. Bonacina and J. Hsiang. On rewrite programs: semantics and relationship with Prolog. *Journal of Logic Programming*, 14(1 & 2):155–180, October 1992.
[8] P. Borovansky, C. Kirchner, H. Kirchner, and P.-E. Moreau. ELAN from a rewriting logic point of view. *Theoretical Computer Science*, 2(285):155–185, July 2002.
[9] B. Buchberger. *An algorithm for finding a basis for the residue class ring of a zero-dimensional polynomial ideal*. PhD thesis, University of Inssbruck (Austria), 1965. (in German).
[10] B. Buchberger. A critical-pair/completion algorithm for finitely generated ideals in rings. In E. Börger, G. Hasenjaeger, and D. Rödding, editors, *Proceedings of Logic and Machines: Decision problems and Complexity*, volume 171 of *Lecture Notes in Computer Science*, pages 137–161. Springer-Verlag, 1983.
[11] R. Bündgen. Simulating Buchberger's algorithm by Knuth-Bendix completion. In R. V. Book, editor, *Rewriting Techniques and Applications: Proc.of the 4th International Conference RTA-91*, pages 386–397. Springer, Berlin, Heidelberg, 1991.
[12] G. Burel. Systèmes Canoniques Abstraits : Application à la Déduction Naturelle et à la Complétion. Master's thesis, Université Denis Diderot – Paris 7, 2005.
[13] H. Cirstea and C. Kirchner. The rewriting calculus — Part I *and* II. *Logic Journal of the Interest Group in Pure and Applied Logics*, 9(3):427–498, May 2001.
[14] M. Clavel, S. Eker, P. Lincoln, and J. Meseguer. Principles of Maude. In J. Meseguer, editor, *Proceedings of the first international workshop on rewriting logic*, volume 4, Asilomar (California), September 1996. Electronic Notes in Theoretical Computer Science.
[15] N. Dershowitz. Computing with rewrite systems. *Information and Control*, 65(2/3):122–157, 1985.
[16] N. Dershowitz. Canonicity. *Electronic Notes in Theoretical Computer Science*, 86(1), June 2003.
[17] N. Dershowitz and C. Kirchner. Abstract saturation-based inference. In P. Kolaitis, editor, *Proceedings of LICS 2003*, Ottawa, Ontario, June 2003. ieee.
[18] N. Dershowitz and C. Kirchner. Abstract Canonical Presentations. *Theoretical Computer Science*, 357:53–69, 2006.
[19] G. Dowek. Confluence as a cut elimination property. In R. Nieuwenhuis, editor, *RTA*, volume 2706 of *Lecture Notes in Computer Science*, pages 2–13. Springer, 2003.
[20] K. Futatsugi and A. Nakagawa. An overview of CAFE specification environment – an algebraic approach for creating, verifying, and maintaining formal specifications over networks. In *Proceedings of the 1st IEEE Int. Conference on Formal Engineering Methods*, 1997.
[21] J. A. Goguen. How to prove algebraic inductive hypotheses without induction, with applications to the correctness of data type implementation. In W. Bibel and R. Kowalski, editors, *Proceedings 5th International Conference on Automated Deduction, Les Arcs (France)*, volume 87 of *Lecture Notes in Computer Science*, pages 356–373. Springer-Verlag, 1980.
[22] J. A. Goguen and G. Malcolm, editors. *Software Engineering with OBJ: algebraic specification in practice*, volume 2 of *Advances in Formal Methods*. Kluwer Academic Publishers, Boston, 2000.
[23] J. A. Goguen and J. Meseguer. EQLOG: Equality, types, and generic modules for logic programming. In D. DeGroot and G. Lindstrom, editors, *Logic Programming:*

Functions, Relations, and Equations, pages 295–363. Prentice-Hall, Englewood Cliffs, NJ, 1986.

[24] J. Hsiang and M. Rusinowitch. Proving refutational completeness of theorem proving strategies: The transfinite semantic tree method. *Journal of the ACM*, 38(3):559–587, July 1991.

[25] G. Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *Journal of the ACM*, 27(4):797–821, 1980.

[26] G. Huet. A complete proof of correctness of the Knuth–Bendix completion algorithm. *Journal of Computer and System Sciences*, 23(1):11–21, August 1981.

[27] G. Huet and J.-J. Lévy. Computations in orthogonal rewriting systems, I. In J.-L. Lassez and G. Plotkin, editors, *Computational Logic*, chapter 11, pages 395–414. The MIT press, 1991.

[28] J.-P. Jouannaud and H. Kirchner. Completion of a set of rules modulo a set of equations. *SIAM Journal of Computing*, 15(4):1155–1194, 1986.

[29] C. Kirchner and H. Kirchner. Rewriting, solving, proving. A preliminary version of a book available at `www.loria.fr/~ckirchne/rsp.ps.gz`, 1999.

[30] C. Kirchner, H. Kirchner, and M. Vittek. Designing constraint logic programming languages using computational systems. In P. Van Hentenryck and V. Saraswat, editors, *Principles and Practice of Constraint Programming. The Newport Papers.*, chapter 8, pages 131–158. The MIT press, 1995.

[31] D. E. Knuth and P. B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, Oxford, 1970.

[32] D. Lankford. Canonical inference. Technical report, Louisiana Tech. University, 1975.

[33] P. Le Chenadec. *Canonical Forms in Finitely Presented Algebras*. John Wiley & Sons, 1986.

[34] J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96(1):73–155, 1992.

[35] D. Musser. On proving inductive properties of abstract data types. In *Proceedings, Symposium on Principles of Programming Languages*, volume 7. Association for Computing Machinery, 1980.

[36] G. Peterson and M. Stickel. Complete sets of reductions for some equational theories. *Journal of the ACM*, 28:233–264, 1981.

[37] J. A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12:23–41, 1965.

[38] A. Rubio and R. Nieuwenhuis. A total AC-compatible ordering based on RPO. *Theoretical Computer Science*, 142(2):209–227, 1995.

[39] W. M. Schorlemmer. Rewriting logic as a logic of special relations. *Electr. Notes Theor. Comput. Sci.*, 15, 1998.

[40] K. Stokkermans. A categorical formulation for critical-pair/completion procedures. In M. Rusinowitch and J.-L. Remy, editors, *CTRS*, volume 656 of *Lecture Notes in Computer Science*, pages 328–342. Springer, 1992.

[41] G. Struth. *Canonical Transformations in Algebra, Universal Algebra and Logic*. Dissertation, Institut für Informatik, Universität des Saarlandes, Saarbrücken, Germany, June 1998.

[42] F. Winkler. Knuth-Bendix procedure and Buchberger algorithm - A synthesis. In *Proceedings of the ACM-SIGSAM 1989 International Symposium on Symbolic and Algebraic Computation*, pages 55–67, Portland (Oregon, USA), 1989. ACM Press.

A Proofs for Section 3 and 4

A.1 From Proof Terms to Proof by Replacement

To prove the termination of $\rightsquigarrow / \sim$, we need a reduction ordering compatible with associativity. We consider only associativity here, although most of the existing works use associativity and commutativity. Therefore, we need the following lemma.

Lemma 1. *If $A \subseteq B$ then $>$ is B-compatible implies $>$ is A-compatible.*

Proof. Just notice that $s' \xleftarrow[A]{*} s > t \xleftarrow[A]{*} t'$ implies $s' \xleftarrow[B]{*} s > t \xleftarrow[B]{*} t'$.

We can therefore use the AC-RPO ordering: a total AC-compatible simplification ordering on ground terms is defined in [38], as an extension of the RPO. To compare terms, they are interpreted using flattening and interpretation rules. As we consider here that the associative commutative symbols have the lowest precedence, we do not need the interpretation rules, and we will only present the flattening rules: terms are reduced using a set of rules

$$f(x_1, \ldots, x_n, f(y_1, \ldots, y_r), z_1, \ldots, z_m) \to f(x_1, \ldots, x_n, y_1, \ldots, y_r, z_1, \ldots, z_m) \tag{1}$$

for all AC-symbols f with $n + m \geq 1$ and $r \geq 2$. Such a rewrite system is terminating as shown in [38].

For all terms t, let $snf(t)$ denote the *set of normal forms* of t using rules (1).

Given a precedence $>$ on function symbols, let $>_{rpo}$ denote the recursive path ordering with precedence $>$ where AC function symbols have multiset status and other symbols have lexicographic status.

If $f(s_1, \ldots, s_n)$ is the normal form of a term s rewriting by (1) only at topmost position, then $tf(s) \stackrel{!}{=} (s_1, \ldots, s_n)$.

Definition 2 (AC-*RPO*). *For all terms s, t, $s >_{AC-rpo} t$ if:*

- $\forall t' \in snf(t)\ \exists s' \in snf(s),\ s' >_{AC-rpo} t'$ ***or***
- $\forall t' \in snf(t)\ \exists s' \in snf(s),\ s' \geq_{rpo} t'$ ***and*** $tf(s) = f(s_1, \ldots, s_m)$ *and* $tf(t) = (t_1, \ldots, t_n)$ ***and***
  - *if the head of s is AC then* $\{s_1, \ldots, s_m\} >_{AC-rpo_{mult}} \{t_1, \ldots, t_n\}$ ***or***
  - *if the head of s is not AC then* $(s_1, \ldots, s_m) >_{AC-rpo_{lex}} (t_1, \ldots, t_n)$.

Proposition 4 ([38]). *The AC-RPO is an AC-compatible simplification ordering which is total for non AC-equivalent ground terms.*

We define a precedence $>$ such that for all function symbols f and for all rule labels ℓ we have $\ell > f > \cdot^{-1} > \ ;\ .$ The AC-RPO built with this precedence will be noted $\succ$.

To show termination, we also need the following lemma:

Lemma 2. *For all proof terms $\pi : t \longrightarrow t'$, we have $\pi \succeq t$ and $\pi \succeq t'$.*

Proof. By induction on the structure of the proof term π.

For **Reflexivity**, $\pi = t = t'$.

For **Congruence**, $\pi = f(\pi_1, \ldots, \pi_n)$, $t = f(t_1, \ldots, t_n)$ and $t' = f(t'_1, \ldots, t'_n)$. By induction hypothesis, for all $i \in \{1, \ldots, n\}$, we have $\pi_i \succeq t_i, t'_i$. Furthermore, π is not reducible on the top position using rules (1), so that $snf(\pi) = \{f(\pi'_1, \ldots, \pi'_n) \ : \ \forall i,\ \pi'_i \in snf(\pi_i)\}$, whereas t and t' are not reducible. Consequently, by definition of an AC-RPO, $\pi \succeq t, t'$.

For **Replacement**, $\pi = \ell(\pi_1, \ldots, \pi_n)$, $t = g(t_1, \ldots, t_n)$ and $t' = d(t'_1, \ldots, t'_n)$ where $\ell = (g, d) \in E \cup R$. With the same arguments than for **Congruence**, we can conclude that $\pi \succeq t, t'$ (recall that $\ell > g, d$).

For **Transitivity**, $\pi = \pi_1; \pi_2$ where $\pi_1 : t \longrightarrow t''$ and $\pi_2 : t'' \longrightarrow t'$. By induction hypothesis, $\pi_1 \succeq t$ and $\pi_2 \succeq t'$. As $\succ$ is a simplification ordering, $\pi \succ \pi_1, \pi_2 \succeq t, t'$.

For **Symmetry**, $\pi = {\pi'}^{-1}$ where $\pi' : t' \longrightarrow t$. By induction hypothesis and because $\succ$ is a simplification ordering, $\pi \succ \pi' \succeq t', t$.

Proposition 5 (Termination). *The rewrite system $\rightsquigarrow$ of Fig. 4 modulo $\sim$ is terminating for ground proof terms.*

Proof. We can show that $\rightsquigarrow \subseteq \succ$, thus proving the termination of $\rightsquigarrow / \sim$:

For **Delete Useless Identities**, it comes from the fact that $\succ$ is a simplification ordering.

For **Sequentialization**, rules (1) are not applicable on the left side whereas they lead on the right side to $;(f(\pi_1, t_2, \ldots, t_n), f(t'_1, \pi_2, \ldots, t_n), \ldots, f(t'_1, t'_2, \ldots, \pi_n))$. We have $f >;$, thus by definition of a RPO, we must then prove that for all $i \in \{1, \ldots, n\}$ we have $f(\pi_1, \ldots, \pi_n) \succ_{RPO} f(t'_1, \ldots, t'_{i-1}, \pi_i, t_{i+1}, \ldots, t_n)$, i.e. $(\pi_1, \ldots, \pi_n) \succ^{lex}_{RPO} (t'_1, \ldots, t'_{i-1}, \pi_i, t_{i+1}, \ldots, t_n)$. By hypothesis there exists at least a $j \in \{1, \ldots, n\} \setminus \{i\}$ such that $\pi_j \neq t_j$, so we can conclude with the preceding lemma.

For **Composition Shallowing**, both sides are not reducible using rules (1). We have $f >;$, thus we have to show: $f(t_1, \ldots, \pi_i; \pi'_i, \ldots, t_n) \succ_{RPO} f(t_1, \ldots, \pi_i, \ldots, t_n)$ and $f(t_1, \ldots, \pi_i; \pi'_i, \ldots, t_n) \succ_{RPO} f(t_1, \ldots, \pi'_i, \ldots, t_n)$. Both comparisons hold by definition of a RPO.

For **Parallel Moves**, both sides are not reducible using rules (1). We have $\ell >;$, thus we have to prove that $\ell(\pi_1, \ldots, \pi_n) \succ_{RPO} \ell(t_1, \ldots, t_n)$ and $\ell(\pi_1, \ldots, \pi_n) \succ_{RPO} d(\pi_1, \ldots, \pi_n)$. The first comparison holds because of the lemma and because there exists a $i \in \{1, \ldots, n\}$ such that $\pi_i \neq t_i$; the second one holds because $\ell > d$.

For **Delete Useless Inverses**, this comes from the fact that $\succ$ is a simplification ordering.

For **Inverse Congruence**, both sides are not reducible using rules (1), therefore this is a consequence of $f > \cdot^{-1}$.

For **Inverse Composition**, both sides are not reducible using rules (1), therefore this is a consequence of $\cdot^{-1} >;$.

We can also prove confluence:

Proposition 6 (Confluence). *The rewrite system $\rightsquigarrow$ is confluent modulo $\sim$ on ground proof terms.*

Proof. The class rewrite system is linear and terminating, so we just have to check that the critical pairs are confluent [25].

For $\underset{R}{\longleftarrow} \circ \underset{R}{\longrightarrow}$, it is easy to check for most of the critical pairs that they are confluent. We only detail the most problematic one. For two possible applications of **Sequentialization**, we have for instance $f(g(\nu_1,\dots,\nu_m),\pi_1,\dots,\pi_n)$ that can be rewritten to $f(g(\nu_1,\dots,\nu_m),t_1,\dots,t_n);f(g(s_1,\dots,s_m),\pi_1,\dots,t_n);\dots;$ $f(g(s_1,\dots,s_m),t'_1,\dots,\pi_n)$ and to $f(g(\nu_1,\dots,s_m);\dots;g(s'_1,\dots,\nu_m),\pi_1,\dots,\pi_n)$. Both of them reduce to $f(g(\nu_1,\dots,s_m);\dots;g(s'_1,\dots,\nu_m),t_1,\dots,t_n);$ $f(g(s_1,\dots,s_m),\pi_1,\dots,t_n);\dots;f(g(s_1,\dots,s_m),t'_1,\dots,\pi_n)$.

For $\underset{R}{\longleftarrow} \circ \underset{A}{\longleftrightarrow}$, the only rules that can interfere with $\sim$ are **Delete Useless Identities**, **Composition Shallowing** and **Inverse Composition**. We can check that all critical pairs are confluent.

Theorem 6 (Correspondence between Proof Representations). *The normal form of a proof term π for the rewrite system $\rightsquigarrow$, noted* $\mathrm{nf}(\pi)$*, has the following form: For some $n \in \mathbb{N}$, some contexts $w_1[],\dots,w_n[]$, some indices $i_1,\dots,i_n \in \{-1,1\}$, some rule labels $\ell_1,\dots,\ell_n$ and some terms $t^1_1,\dots,t^1_{m_1},\dots,t^n_1,\dots,t^n_{m_n}$:*

$$\mathrm{nf}(\pi) = (w_1[\ell_1(t^1_1,\dots,t^1_{m_1})])^{i_1};\dots;(w_n[\ell_n(t^n_1,\dots,t^n_{m_n})])^{i_n}$$

where ν^1 is a notation for ν.

We will denote by $\mathrm{nf}(\pi)$ *the normal form of a proof term π.*

Such a proof term correspond with the following proof by replacement of equal by equal:

$$w_1[g_1(t^1_1,\dots,t^1_{m_1})] \underset{\ell_1}{\overset{p_1}{\leftrightarrows}}_1 w_1[d_1(t^1_1,\dots,t^1_{m_1})] \underset{\ell_2}{\overset{p_2}{\leftrightarrows}}_2 \cdots \underset{\ell_n}{\overset{p_n}{\leftrightarrows}}_n w_n[d_n(t^n_1,\dots,t^n_{m_n})]$$

where for all $j \in \{1,\dots,n\}$ we have:

- $\ell_j = (g_j, d_j)$,
- p_j *is the position of* $[]$ *in* $w_j[]$,
- $\leftrightarrows_j = \begin{array}{l} \longrightarrow \text{ if } i_j = 1 \text{ and } \ell_j \in R, \\ \longleftarrow \text{ if } i_j = -1 \text{ and } \ell_j \in R, \\ \longleftrightarrow \text{ if } \ell_j \in E. \end{array}$
- *if* $j \neq n$, $w_j[d_j(t^j_1,\dots,t^j_{m_j})] = w_{j+1}[g_{j+1}(t^{j+1}_1,\dots,t^{j+1}_{m_{j+1}})]$.

Proof. We first have to check that proof terms in that form are indeed irreducible by $\rightsquigarrow$, what is left to the reader.

Then, suppose that we have an irreducible proof term. Because **Sequentialization** cannot be applied, there is at most one ; under all function symbols.

Because **Composition Shallowing** cannot be applied, there are no ; under all function symbols. Because **Inverse Congruence** and **Inverse Composition** cannot be applied, $\cdot^{-1}$ is applied between ; and function symbols. Irreducible proof term are therefore application of ; over eventually $\cdot^{-1}$ over base terms composed of function symbols and rule labels.

Because **Delete Useless Identities** and **Delete Useless Inverse** cannot be applied, there is a least one non-trivial proof (i.e a proof with a label in it) in each of these base terms. Because **Sequentialization** cannot be applied, there is at most one non-trivial proof in each of them. Because **Parallel Moves** cannot be applied, the subterms of the labels are Σ-terms. Consequently, each base term contains one and only one rule label, applied to Σ-terms.

A.2 Adequacy to the Postulates

Postulate A: The proof of $(u, v) \in E \cup R$ labeled by ℓ is $\ell(x_1, \ldots, x_n)$ where $x_1, \ldots, x_n$ are the free variables of (u, v).

Postulate B: We can replace the assumption $\ell(\pi_1, \ldots, \pi_n)$ of something proved by its proof where the free variables are replaced by the proofs $\pi_1, \ldots, \pi_n$.

Postulate C and D: These postulates hold because of the tree structure of proofs.

Postulate E: This one does not trivially hold. We first show the following lemma:

Lemma 3. *For all function symbols f of arity $n + 1$, for all proof terms $\pi_1, \ldots, \pi_n$, q and r:*

$$q > r \text{ implies } f(\pi_1, \ldots, q, \ldots, \pi_n) > f(\pi_1, \ldots, r, \ldots, \pi_n) \ .$$

Proof. Suppose $q > r$, thus by definition $\mathrm{nf}(q) >_{rep} \mathrm{nf}(r)$. To compare $f(\pi_1, \ldots, q, \ldots, \pi_n)$ and $f(\pi_1, \ldots, r, \ldots, \pi_n)$, we have to transform them to proof by replacement. As $\underset{\rightsquigarrow/\sim}{\longrightarrow}$ is Church-Rosser, the way it is applied does not matter.

We have

$$\begin{aligned}
&f(\pi_1, \ldots, q, \ldots, \pi_n) \\
&\underset{\rightsquigarrow}{\longrightarrow}^* f(\pi_1, t_2, \ldots, t_n); \ldots; f(t'_1, \ldots, q, \ldots, t_n); \ldots; f(t'_1, \ldots, \pi_n) \\
&\underset{\rightsquigarrow}{\longrightarrow}^* f(\pi_1, t_2, \ldots, t_n); \ldots; \underline{f(t'_1, \ldots, \mathrm{nf}(q), \ldots, t_n)}; \ldots; f(t'_1, \ldots, \pi_n)
\end{aligned}$$

Then, if $\mathrm{nf}(q)$ contains ; the underlined term will be split by **Composition Shallowing**. If it contains $^{-1}$ the rule **Inverse Congruence** will be applied. Some terms outside the underline corresponding to identity will be removed by **Delete Useless Identities**, and the normal form will look like:

$$f(\pi_1, t_2, \ldots, t_n); \ldots; \underline{f(t'_1, \ldots, q_1, \ldots, t_n)^{i_1}; \ldots; f(t'_1, \ldots, q_m, \ldots, t_n)^{i_m}}; \ldots; f(t'_1, \ldots, \pi_n)$$

with $\mathrm{nf}(q) = q_1^{i_1}; \ldots; q_m^{i_m}$.

The same will apply with r, and therefore, to compare the initial proofs, we just have to compare the costs of the underlined terms.

The cost of $\mathrm{nf}(q)$ will look like $\{(\{s_1\}, u_1, h_1), \ldots, (\{s_m\}, u_m, h_m)\}$. Then the cost of $f(t'_1, \ldots, q_1, \ldots, t_n)^{i_1}; \ldots; f(t'_1, \ldots, q_m, \ldots, t_n)^{i_m}$ will be:

$$\left\{ \begin{array}{l} (\{f(t'_1, \ldots, s_1, \ldots, t_n)\}, u_1, f(t'_1, \ldots, h_1, \ldots, t_n)), \ldots, \\ (\{f(t'_1, \ldots, s_m, \ldots, t_n)\}, u_m, f(t'_1, \ldots, h, m, \ldots, t_n)) \end{array} \right\} .$$

For $\mathrm{nf}(r)$ they will be respectively $\{(\{g_1\}, v_1, d_1), \ldots, (\{g_p\}, v_p, d_p)\}$ and:

$$\left\{ \begin{array}{l} (\{f(t'_1, \ldots, g_1, \ldots, t_n)\}, v_1, f(t'_1, \ldots, d_1, \ldots, t_n)), \ldots, \\ (\{f(t'_1, \ldots, g_p, \ldots, t_n)\}, v_p, f(t'_1, \ldots, d_p, \ldots, t_n)) \end{array} \right\} .$$

$\gg$, which is used to compare the first and the third components of each part of the cost, is a reduction ordering, so that $\mathrm{nf}(q) >_{rep} \mathrm{nf}(r)$ implies for instance $f(t'_1, \ldots, q_1, \ldots, t_n)^{i_1}; \ldots; f(t'_1, \ldots, q_m, \ldots, t_n)^{i_m} >_{rep} f(t'_1, \ldots, r_1, \ldots, t_n)^{i_1}; \ldots; f(t'_1, \ldots, r_p, \ldots, t_n)^{i_p}$.

The same is true for labels:

Lemma 4. *For all rule labels ℓ, for all proof terms $\pi_1, \ldots, \pi_n$, q and r:*

$$q > r \text{ implies } \ell(\pi_1, \ldots, q, \ldots, \pi_n) > \ell(\pi_1, \ldots, r, \ldots, \pi_n)$$

Proof. $\ell(\pi_1, \ldots, q, \ldots, \pi_n)$ and $\ell(\pi_1, \ldots, r, \ldots, \pi_n)$ can be reduced by **Parallel Moves** to $\ell(t_1, \ldots, t_n); d(\pi_1, \ldots, q, \ldots, \pi_n)$ and $\ell(t_1, \ldots, t_n); d(\pi_1, \ldots, r, \ldots, \pi_n)$. We can therefore conclude using the preceding lemma.

This allows us to show

Theorem 7 (Postulate E for Equational Proofs). *For all proof terms p, r, for all position i of p:*

$$p_{|i} > r \text{ implies } p > p[r]_i \ .$$

Proof. This is proved by induction on i. For $i = \epsilon$ this is trivial. For $i \neq \epsilon$, by induction hypothesis, the result holds for the subproofs of p. For the head of p:

- for **Symmetry**, it is trivial;
- for **Transitivity**, it comes from the fact that equational proofs are compared as the multiset of their equational proof steps;
- for **Congruence**, it comes from lemma 3;
- for **Replacement**, it comes from lemma 4.

A.3 Standard Completion is Canonical

Remember that by fairness assumption, $E_\infty = \emptyset$.

Lemma 5. *For all standard completion derivations $(E_i, R_i)_i$:*

$$E_0^\sharp \ \subseteq \ R_\infty \ .$$

Proof. By contradiction, suppose there is $(a, b) \in E_0^\sharp \setminus R_\infty$, labeled ℓ. Because completion is adequate, there exists $p \in \mu Pf(R_\infty)$ proving $a = b$. Because $a = b \in E_0^\sharp$, $\ell(x_1, \ldots, x_n) \in Nf(E_0) = Nf(R_\infty)$ where $(x_i)_i$ are the free variables of ℓ, so that

$$p > \ell(x_1, \ldots, x_n)$$

- If there are no peak in $\mathrm{nf}(p)$, then $\mathrm{nf}(p)$ is a valley proof, and it is easy to show that it is smaller than $\ell(x_1, \ldots, x_n)$, which is a contradiction with the preceding comparison.
- If there is a parallel peak, for instance $s[c, e] \xleftarrow[\ell_1]{i} s[d, e] \xrightarrow[\ell_2]{j} s[d, f]$, then the proof by replacement where this peak is replaced by $s[c, e] \xrightarrow[\ell_2]{j} s[c, f] \xleftarrow[\ell_1]{i} s[d, f]$ is smaller, thus leading to a contradiction with the minimality of p in $Pf(R_\infty)$.
- If there is a critical peak, then by fairness assumption there is some step k where this critical peak is treated by **Deduce**. The proof of the conclusion of the critical peak at the step $k + 1$ is therefore smaller. Because standard completion is good, it can only go smaller, so that at the limit we can find by replacement of the critical peak by this proof a smaller proof of $a = b$, thus leading to a contradiction with the minimality of p in $Pf(R_\infty)$.

Lemma 6. *For all standard completion derivations $(E_i, R_i)_i$ which* terminate without failure*:*

$$R_\infty \subseteq E_0^\sharp \ .$$

Proof. By contradiction, suppose there is $(a, b) \in R_\infty \setminus E_0^\sharp$, labeled by ℓ. Then there exists a proof $p \in \mu Pf(E_0^\sharp)$ such that $\ell(x_1, \ldots, x_n) > p$ where $x_1, \ldots, x_n$ are the free variables of ℓ.

Rules comes from orientation of equational axioms through **Orient**, so that $a \gg b$. The cost of $\ell(x_1, \ldots, x_n)$ is then $\{(\{a\}, a, b)\}$. Consider the leftmost step of $\mathrm{nf}(p)$. It is of the form $a \overset{i}{\underset{(c,d)}{\leftrightarrows}} a[d]_i$ where $c = a_{|i}$. If it is $a \xleftarrow[d \to c]{i} a[d]_i$ then the cost of this proof step would be $\{(\{a[d]_i\}, d, a)\}$, which is then greater than $\{(\{a\}, a, b)\}$, thus leading to a contradiction with the fact that $\ell(x_1, \ldots, x_n) > p$. If $a \xleftrightarrow[c=d]{i} a[d]_i$ then the cost of this proof step would be $\{(\{a, a[d]_i\}, c, a[d]_i)\}$, which is then greater than $\{(\{a\}, a, b)\}$, thus leading to a contradiction with the fact that $\ell(x_1, \ldots, x_n) > p$. If it is $a \xrightarrow[c \to d]{i} a[d]_i$ then there is a critical pair $(b, a[d]_i)$ in R_∞ (we just proved that $E_0^\sharp \subseteq R_\infty$). The fairness assumption will therefore apply, and therefore **Deduce** will produce the equational axiom $b = a[d]_i$, which will be oriented, and $a \longrightarrow b \in R_\infty$ will be simplified through **Compose** or **Collapse**. Because $a \longrightarrow b$ is persisting, it must be generated once again, thus contradicting the termination of the completion.

Theorem 8 (Completeness of Standard Completion). *Standard completion results — at the limit, when it terminates without failure — in the canonical, Church-Rosser basis.*

Proof. There is nothing more to prove, because we have $R_\infty = E_0^\sharp$, and standard completion is good so we can use Theorem 2.