Proving weak termination also provides the right way to terminate - Extended version -

Olivier Fissore, Isabelle Gnaedig, Hélène Kirchner

Proving weak termination also provides the right way to terminate

Olivier Fissore, Isabelle Gnaedig, Hélène Kirchner
LORIA-INRIA & LORIA-CNRS
BP 239 F-54506 Vandœuvre-lès-Nancy Cedex
Phone: + 33 3 83 58 17 00
Fax: + 33 3 83 27 83 19
e-mail: fissore@loria.fr, gnaedig@loria.fr, Helene.Kirchner@loria.fr

May 7, 2004

Abstract

From an inductive method for proving weak innermost termination of rule-based programs, we automatically infer, for each successful proof, a finite strategy for data evaluation. The proof principle uses an explicit induction on the termination property, to prove that any input data has at least one finite evaluation. For that, we observe proof trees built from the rewrite system, schematizing the innermost derivations of any ground term. These proof trees are issued from patterns representing sets of ground terms. They are built using two mechanisms, namely abstraction, introducing variables that represent ground terms in normal form, and narrowing, schematizing rewriting on ground terms. From the proof trees, we extract for any given ground term, a rewriting strategy to compute one of its normal form, even if the ground term admits infinite rewriting derivations.

1 Introducing the problem

In the context of programming in general, termination is a key property that warrants the existence of a result for every evaluation of a program. For rule-based programs, written in languages like ASF+SDF [17], Maude [3], Cafe-OBJ [10], or ELAN [2], data evaluation consists in exploring rewriting derivations of an input term. Strong termination, expressing that every rewriting derivation terminates, often does not hold. When for any term, there is at least one terminating derivation, the rewrite system is said to be weakly terminating. In the context of programming, this is an interesting property, since then data evaluation consists in finding *one* irreducible form. In languages like ELAN [2], strategies can express that the result of the program evaluation on a data is *one of its possible* finite evaluations, or *the first* one. Weak termination then warrants a result for such strategies. In this paper, we tackle the innermost weak termination problem. We focus on the innermost strategy, consisting in rewriting always at the lowest possible positions, since it is most often used as a built-in mechanism in evaluation of rule-based languages and functional languages.

Analyzing termination allows choosing the good evaluation strategy. Indeed, if the program is strongly terminating, a depth-first evaluation can be used, while if the program is only weakly terminating, a breadth-first algorithm, often much more costly, is necessary in general. In the second case, if there is a way to find terminating branches, the breadth-first technique can be avoided, which yields then a considerable gain for program executions. This is what we propose.

Obviously, specific methods for proving strong termination of rewriting under strategies give a way to prove weak termination for standard rewriting (i.e. rewriting without any strategy), provided normal forms for rewriting with these strategies are also normal forms for standard rewriting. Let us cite [1] and [11] for the innermost strategy, [9] for the outermost strategy, and [7, 18] for local strategies on operators. Here, we are more specific: we consider directly the weak innermost termination problem, i.e. we prove that among all innermost rewriting derivations starting from any term, there is at least one finite derivation. Like the previously cited methods, the approach presented here also gives a way to prove weak termination of standard rewriting. But to our knowledge, it is the only approach able to handle term rewriting systems (TRSs in short) that are not strongly but only weakly innermost terminating. Moreover, the method is *constructive* in the sense that in establishing weak innermost termination of a TRS, it gives the strategy to follow to obtain one of the finite derivations.

The weak termination property has been studied in several situations: first, weak termination can imply strong termination [14]. J. Goubault-Larrecq proposed a proof of weak termination of typed Lambda-Sigma calculi in [13]. B. Gramlich established conditions on TRSs for the property to be preserved by the union operation on TRSs [15].

As an example, let us consider the following TRS where f and p can be seen as programs working on integers , and which doesn't strongly terminate, even with the innermost strategy:

$$\begin{aligned} f(x) &\to p(s(x)) && (1)\\ f(x) &\to p(s(s(x))) && (2)\\ p(s(s(x))) &\to p(x) && (3)\\ p(0) &\to 0 && (4)\\ p(s(0)) &\to f(0). && (5) \end{aligned}$$

Given an integer n, $f(n)$ innermost evaluates either into $p(s(n))$ or into $p(s(s(n)))$. A particularity of p is that, given an integer m, the rewriting derivation starting from $p(m)$ innermost terminates if m is even, and may not terminate if m is odd. Therefore we have at least one evaluation of $f(n)$ that innermost terminates, and one that does not, whatever the integer n.
For instance, considering $n = s(0)$, the following two innermost derivations are possible :

$$f(s(0)) \xrightarrow{(6)} p(s(s(0))) \xrightarrow{(8)} p(0) \xrightarrow{(9)} 0.$$

$$f(s(0)) \xrightarrow{(7)} p(s(s(s(0)))) \xrightarrow{(8)} p(s(0)) \xrightarrow{(10)} f(0) \xrightarrow{(6)} p(s(0)) \longrightarrow \dots$$

We first propose in this paper a method based on an explicit induction on the termination property, to prove that every element t of a given set of terms T weakly innermost terminates i.e., there is at least one finite innermost rewriting derivation starting from t. For that, we observe proof trees built from the rewrite system, and schematizing the innermost rewriting derivations of any ground term. These proof trees are issued from patterns $g(x_1, \dots, x_m)$ where g is a defined function symbol, and are built using two mechanisms, namely abstraction, introducing variables that represent ground terms in normal form, and narrowing, schematizing rewriting on ground terms.

Directly using the termination notion on terms has also been proposed in [12], for inductively proving well-foundedness of binary relations, among which path orderings. The approach differs from ours in that it works on general relations, that can then be used on TRSs, whereas we directly handle the termination proof of a given TRS.

When all proof trees have a successful branch for all ground instances of the patterns, the weak innermost termination property of the rewrite system is proved. Then from these successful branches, a normalizing strategy can be inferred for any ground term. We show how to extract the relevant information from the proof trees to guide the innermost normalization process. To some extent, our method has similarities with [16], where an automaton is built for normalization according to a needed-redex strategy in the case of orthogonal rewrite systems.

Proving weak termination of a program and deducing a normalizing strategy can be achieved at *compile-time*. Then, to evaluate a data at *run-time* with no risk of non-termination, it suffices to follow the strategy, which states which rule to apply and at which position in the term at each step of the normalization process. Henceforth, evaluation at run-time is made very efficient, since it always leads to a result, i.e. an irreducible term.

In Section 2, the background is presented. Section 3 introduces the basic concepts of the inductive proof mechanism. In Section 4, our method is formally described with inference rules and a strategy to apply them. Finally, in Section 5, a strategy is proposed to reach an innermost normal form from a given term, using information of the proof establishing weak termination.

2 Notations

We assume that the reader is familiar with the basic definitions and notations of term rewriting given for instance in [6]. $\mathcal{T}(\mathcal{F},\mathcal{X})$ is the set of terms built from a given finite set $\mathcal{F}$ of function symbols having an arity $n \in \mathbb{N}$, and a set $\mathcal{X}$ of variables denoted $x, y \ldots$. $\mathcal{T}(\mathcal{F})$ is the set of ground terms (without variables). The terms composed of a symbol of arity 0 are called constants; $\mathcal{C}$ is the set of constants of $\mathcal{F}$. Positions in a term are represented as sequences of integers. The empty sequence ϵ denotes the top position. The notation $t|_p$ stands for the subterm of t at position p. To emphasize that u contains subterms t_j at positions $j \in \{i_1..i_k\}$, we write $u[t_j]_{j\in\{i_1..i_k\}}$.

A substitution is an assignment from $\mathcal{X}$ to $\mathcal{T}(\mathcal{F},\mathcal{X})$, written $\sigma = (x \mapsto t)\ldots(y \mapsto u)$. It uniquely extends to an endomorphism of $\mathcal{T}(\mathcal{F},\mathcal{X})$. We identify a substitution $\sigma = (x \mapsto t)\ldots(y \mapsto u)$ with the finite conjunction of equations $(x = t) \wedge \ldots \wedge (y = u)$. The result of applying σ to a term $t \in \mathcal{T}(\mathcal{F},\mathcal{X})$ is written $\sigma(t)$ or σt. The domain of σ, denoted $Dom(\sigma)$ is the finite subset of $\mathcal{X}$ such that $\sigma x \neq x$. The range of σ, denoted $Ran(\sigma)$, is such that $Ran(\sigma) \cap Dom(\sigma) = \emptyset$, and defined by $Ran(\sigma) = \bigcup_{x\in Dom(\sigma)} Var(\sigma x)$. A ground substitution or instantiation is an assignment from $\mathcal{X}$ to $\mathcal{T}(\mathcal{F})$. The composition of substitutions σ_1 followed by σ_2 is denoted $\sigma_2\sigma_1$. Given two substitutions σ_1 and σ_2, we write $\sigma_1 \leq \sigma_2$ iff there exists a substitution μ such that $\sigma_2 = \mu\sigma_1$. We denote λ the empty substitution.

Given a set $\mathcal{R}$ of rewrite rules or term rewriting system on $\mathcal{T}(\mathcal{F},\mathcal{X})$, a function symbol in $\mathcal{F}$ is called a constructor if it does not occur in $\mathcal{R}$ at the top position of the left-hand side of a rule, and is called a defined function symbol otherwise. The set of constructors of $\mathcal{F}$ for $\mathcal{R}$ is denoted by $\mathcal{C}ons_{\mathcal{R}}$, the set of defined function symbols of $\mathcal{F}$ for $\mathcal{R}$ is denoted by $\mathcal{D}ef_{\mathcal{R}}$ ($\mathcal{R}$ is omitted when there is no ambiguity). The rewriting relation induced by $\mathcal{R}$ is denoted by $\rightarrow_{\mathcal{R}}$ ($\rightarrow$ if there is no ambiguity on $\mathcal{R}$). We note $s \rightarrow_{p,l\rightarrow r,\sigma} t$ (or $s \rightarrow^{p,l\rightarrow r,\sigma} t$ where either p or $l \rightarrow r$ or σ may be omitted) if s rewrites into t at position p with the rule $l \rightarrow r$ and the substitution σ, i.e. $s = s[l\sigma]_p$ and $t = s[r\sigma]_p$. The transitive (resp. reflexive transitive) closure of the rewriting relation induced by $\mathcal{R}$ is denoted by $\rightarrow^+_R$ (resp. $\rightarrow^*_R$). Given a term t, we call normal form of t, denoted by $t{\downarrow}$, any irreducible term, if it exists, such that $t \rightarrow^*_R t{\downarrow}$. For innermost rewriting, we replace $\rightarrow$ by $\rightarrow^{Inn}$.

An ordering $\succ$ on $\mathcal{T}(\mathcal{F},\mathcal{X})$ is said to be noetherian iff there is no infinite decreasing derivation (or chain) for this ordering. It is $\mathcal{F}$-stable iff for any pair of terms t, t' of $\mathcal{T}(\mathcal{F},\mathcal{X})$, for any context $f(\ldots\ \ldots)$, $t \succ t'$ implies $f(\ldots t \ldots) \succ f(\ldots t' \ldots)$. It has the subterm property iff for any t of $\mathcal{T}(\mathcal{F},\mathcal{X})$, $f(\ldots t \ldots) \succ t$. Notice that, for $\mathcal{F}$ and $\mathcal{X}$ finite, if $\succ$ is $\mathcal{F}$-stable and has the subterm property, then it is noetherian [5]. If, in addition, $\succ$ is stable by substitution (for any substitution σ, any pair of terms $t, t' \in \mathcal{T}(\mathcal{F},\mathcal{X})$, $t \succ t'$ implies $\sigma t \succ \sigma t'$), then it is called a simplification ordering. Let t be a term of $\mathcal{T}(\mathcal{F})$; like for standard rewriting, we say that t weakly (resp. strongly) (innermost) terminates if and only if at least one (resp. every) (innermost) rewriting derivation starting from t is finite. Obviously, strong (innermost) termination implies weak (innermost) termination. An innermost rewriting normal form of t is also denoted by $t{\downarrow}$.

3 Induction and constraints

For proving that the terms t of $\mathcal{T}(\mathcal{F})$ weakly innermost terminate, we proceed by induction on $\mathcal{T}(\mathcal{F})$ with a noetherian ordering $\succ$, assuming that for any t' such that $t \succ t'$, t' weakly innermost terminates. The main intuition is to observe the rewriting derivation tree starting from any ground term $t \in \mathcal{T}(\mathcal{F})$ which is any instance of a term $g(x_1,\ldots,x_m) \in \mathcal{T}(\mathcal{F},\mathcal{X})$, for some defined function symbol $g \in \mathcal{D}ef$, and variables $x_1, \ldots, x_m$. Proving weak innermost termination on ground terms amounts to prove that all these rewriting derivation trees have at least one finite branch.

Each rewriting derivation tree is simulated, using a lifting mechanism, by a proof tree developed from $g(x_1, \ldots, x_m)$ on $\mathcal{T}(\mathcal{F}, \mathcal{X})$, for every $g \in \mathcal{D}ef$, by alternatively using two main concepts, namely narrowing and abstraction. More precisely, narrowing schematizes all innermost rewriting possibilities of terms. The abstraction process simulates the innermost normalization of subterms in the derivations. It consists in replacing these subterms by special variables, denoting one of their possible innermost normal forms, without computing them. This abstraction step is performed on subterms that can be assumed weakly innermost terminating by induction hypothesis.

The schematization of ground rewriting derivation trees is achieved through constraints. The nodes of the developed proof trees are composed of a current term of $\mathcal{T}(\mathcal{F}, \mathcal{X})$, and a set of ground substitutions represented by a constraint progressively built along the successive abstraction and narrowing steps. Each node in an abstract tree schematizes a set of ground terms: all ground instances of the current term, that are solutions of the constraint.

The constraint is in fact composed of two kinds of formulas: ordering constraints, set to warrant the validity of the inductive steps, and abstraction constraints combined to narrowing substitutions, which effectively define the relevant sets of ground terms.

As said previously, we consider any term of $\mathcal{T}(\mathcal{F})$ as a ground instance of a term t of $\mathcal{T}(\mathcal{F}, \mathcal{X})$ occurring in a proof tree issued from a reference term t_{ref}. Using the termination induction hypothesis on $\mathcal{T}(\mathcal{F})$ naturally leads us to simulate the rewriting relation by two mechanisms:

- first, some subterms t_j of the current term t of the proof tree are supposed to have at least one innermost terminating ground instance, by induction hypothesis, if $\theta t_{ref} \succ \theta t_j$ for the induction ordering $\succ$ and for every θ solution of the constraint associated to t. They are replaced in t by *abstraction variables* X_j representing respectively one of their innermost normal forms $t_j\!\downarrow$. Reasoning by induction allows us to only suppose the existence of the $t_j\!\downarrow$ *without explicitly computing them*;

- second, innermost narrowing the resulting term $u = t[X_j]_{j \in \{i_1, \ldots, i_p\}}$ (where $i_1, \ldots, i_p$ are the positions of the abstracted subterm t_j in t), according to the possible instances of the X_j into terms v. This corresponds to innermost rewriting ground instances of u (characterized by the constraint associated to u) with all possible rewrite rules.

  In general, the narrowing step of u is not unique. We obviously have to consider all terms v such that θu innermost rewrites into θv, which corresponds to considering all innermost narrowing steps from u.

Then the weak innermost termination problem of the ground instances of t is reduced to the weak innermost termination problem of the ground instances of v. If $\theta t_{ref} \succ \theta v$ for every ground substitution θ solution of the constraint associated to v, by induction hypothesis, θv is supposed to be weakly innermost terminating. Else, the process is iterated on v, until getting a term t' such that either $\theta t_{ref} \succ \theta t'$, or $\theta t'$ is irreducible.

We now introduce some concepts to formalize and automate the technique sketched above.

3.1 Ordering constraints and abstraction

The induction ordering $\succ$ is constrained along the proof by imposing constraints between terms that must be comparable, each time the induction hypothesis is used in the abstraction mechanism. As we are working with a lifting mechanism on the proof trees with terms of $\mathcal{T}(\mathcal{F}, \mathcal{X})$, we directly work with an ordering $\succ_{\mathcal{P}}$ on $\mathcal{T}(\mathcal{F}, \mathcal{X})$ such that $t \succ_{\mathcal{P}} u$ induces $\theta t \succ \theta u$, for every θ solution of the constraint associated to u.

So inequalities of the form $t > u_1, \ldots, u_m$ are accumulated, which are called *ordering constraints*. Any ordering $\succ_{\mathcal{P}}$ on $\mathcal{T}(\mathcal{F}, \mathcal{X})$ satisfying them and which is stable by substitution fulfills the previous requirements on ground terms. The ordering $\succ_{\mathcal{P}}$, defined on $\mathcal{T}(\mathcal{F}, \mathcal{X})$, can then be seen as an extension of the induction ordering $\succ$, defined on $\mathcal{T}(\mathcal{F})$. For convenience, $\succ_{\mathcal{P}}$ will also be written $\succ$.

It is important to remark that, for establishing the inductive termination proof, it is sufficient to decide whether there exists such an ordering.

Definition 1 An ordering constraint *is a pair of terms of* $\mathcal{T}(\mathcal{F}, \mathcal{X})$ *noted* $(t > t')$. *It is said to be* satisfiable *if there exists an ordering* $\succ$, *such that for every instantiation* θ *whose domain contains* $\mathcal{V}ar(t) \cup \mathcal{V}ar(t')$, *we have* $\theta t \succ \theta t'$. *We say that* $\succ$ *satisfies* $(t > t')$.

A conjunction C *of ordering constraints is satisfiable if there exists an ordering satisfying all conjuncts. The empty conjunction, always satisfied, is denoted by* $\top$.

Satisfiability of a constraint C of this form is undecidable. But a sufficient condition for an ordering $\succ$ to satisfy C is that $\succ$ is stable by substitution and $t \succ t'$ for any constraint $t > t'$ of C.

Other constraints are introduced by the abstraction mechanism. To abstract a term u at positions $i_1, \ldots, i_p$, where the $u|_j$ are supposed to have a normal form $u|_j\downarrow$, we replace the $u|_j$ by abstraction variables X_j representing respectively one of their possible innermost normal forms. Let us define these special variables more formally.

Definition 2 *Let* $\mathcal{N}$ *be a set of new variables disjoint from* $\mathcal{X}$. *Symbols of* $\mathcal{N}$ *are called* NF-variables. *Substitutions and instantiations are extended to* $\mathcal{T}(\mathcal{F}, \mathcal{X} \cup \mathcal{N})$ *in the following way. Let* $X \in \mathcal{N}$; *for any substitution* σ *(resp. instantiation* θ*) such that* $X \in Dom(\sigma)$, σX *(resp.* θX*) is in normal form, and then* $\mathcal{V}ar(\sigma X) \subseteq \mathcal{N}$.

Definition 3 (term abstraction) *The term* $u[t_j]_{j \in \{i_1, \ldots, i_p\}}$ *is said to be* abstracted *into* u' *(called* abstraction *of* u*) at positions* $\{i_1, \ldots, i_p\}$ *iff* $u' = u[X_j]_{j \in \{i_1, \ldots, i_p\}}$, *where the* $X_j, j \in \{i_1, \ldots, i_p\}$ *are fresh distinct NF-variables.*

Weak termination on $\mathcal{T}(\mathcal{F})$ is proved by reasoning on terms with abstraction variables, i.e. on terms of $\mathcal{T}(\mathcal{F}, \mathcal{X} \cup \mathcal{N})$. Ordering constraints are extended to pairs of terms of $\mathcal{T}(\mathcal{F}, \mathcal{X} \cup \mathcal{N})$. When subterms t_i are abstracted by X_i, we state constraints on abstraction variables, called *abstraction constraints* to express that their instances can only be normal forms of the corresponding instances of t_i. Initially, they are of the form $t\downarrow = X$ where $t \in \mathcal{T}(\mathcal{F}, \mathcal{X} \cup \mathcal{N})$, and $X \in \mathcal{N}$, but we will see later how they are combined with the substitutions used for the narrowing process.

3.2 Narrowing

After abstraction of the current term t into $t[X_i]_{p_i}$, we test whether the possible ground instances of $t[X_i]_{p_i}$ are reducible, according to the possible values of the instances of the X_i. This is achieved by innermost narrowing $t[X_i]_{p_i}$.

To schematize innermost rewriting on ground terms, we need to refine the usual notion of narrowing. In fact, with the usual innermost narrowing relation, if a position p in a term t is a narrowing position, no suffix position of p can be a narrowing position too. However, if we consider ground instances of t, we can have rewriting positions p for some instances, and p' for some other instances, such that p' is a suffix of p. So, when using the narrowing relation to schematize innermost rewriting of ground instances of t, the narrowing positions p to consider depend on a set of ground instances of t, which is defined by excluding the ground instances of t that would be narrowable at some suffix position of p. For instance, with the TRS $R = \{g(a) \to a, f(g(x)) \to b\}$, the innermost narrowing positions of the term $f(g(X))$ are 1 with the narrowing substitution $\sigma = (X = a)$, and ϵ with any σ such that $\sigma X \neq a$.

The innermost narrowing steps applying to a given term t are computed in the following way. We look at every position p of t such that $t|_p$ unifies with the left-hand side of a rule using a substitution σ. The position p is an innermost narrowing position of t, iff there is no suffix position p' of t such that $\sigma t|_{p'}$ unifies with a left-hand side of rule. Then we look for every suffix position p' of p in t such that $\sigma t|_{p'}$ narrows with some substitution σ' and some rule $l' \to r'$, and we set a constraint to exclude these substitutions. So the substitutions used to narrow a term have in general to satisfy a set of disequalities coming from the negation of other substitutions. We then need the following notations and definitions.

Let σ be a substitution on $\mathcal{T}(\mathcal{F}, \mathcal{X} \cup \mathcal{N})$. In the following, we identify σ with the equality formula $\bigwedge_i (x_i = t_i)$, with $x_i \in \mathcal{X} \cup \mathcal{N}$, $t_i \in \mathcal{T}(\mathcal{F}, \mathcal{X} \cup \mathcal{N})$. Similarly, we call *negation* $\overline{\sigma}$ of the substitution σ the formula $\bigvee_i (x_i \neq t_i)$.

Definition 4 *A substitution σ is said* to satisfy *a constraint $\bigwedge_j \bigvee_{i_j} (x_{i_j} \neq t_{i_j})$, iff for all ground instantiation θ, $\bigwedge_j \bigvee_{i_j} (\theta\sigma x_{i_j} \neq \theta\sigma t_{i_j})$. A* constrained substitution *σ is a formula $\sigma_0 \wedge \bigwedge_j \bigvee_{i_j} (x_{i_j} \neq t_{i_j})$, where σ_0 is a substitution, and $\bigwedge_j \bigvee_{i_j} (x_{i_j} \neq t_{i_j})$ the constraint to be satisfied by σ_0.*

Definition 5 (innermost narrowing) *A term $t \in \mathcal{T}(\mathcal{F}, \mathcal{X} \cup \mathcal{N})$* innermost narrows *into a term $t' \in \mathcal{T}(\mathcal{F}, \mathcal{X} \cup \mathcal{N})$ at the non-variable position p, using the rule $l \rightarrow r \in \mathcal{R}$ with the constrained substitution $\sigma = \sigma_0 \wedge \bigwedge_{j \in [1..k]} \overline{\sigma_j}$,*
which is written $t \rightsquigarrow^{Inn}_{p, l \rightarrow r, \sigma} t'$ iff

$$\sigma_0(l) = \sigma_0(t|_p) \text{ and } t' = \sigma_0(t[r]_p)$$

where σ_0 is the most general unifier of t and l at position p, and $\sigma_j, j \in [1..k]$ are all most general unifiers of $\sigma_0 t$ and a left-hand side of rule of $\mathcal{R}$, at suffix positions of p.

A few remarks can be made on the choice of variables and on the domain of the substitutions generated during the proof process. It is always assumed that there is no variable in common between the rule and the term, i.e. that $Var(l) \cap Var(t) = \emptyset$. This requirement of disjoint variables is easily fulfilled by an appropriate renaming of variables in the rules when narrowing is performed. Observe that for the most general unifier σ_0 used in the above definition, $Dom(\sigma_0) \subseteq Var(l) \cup Var(t)$ and we can choose $Ran(\sigma_0) \cap (Var(l) \cup Var(t)) = \emptyset$, thus introducing in the range of σ_0 only fresh variables. Moreover, narrowing is only performed on terms t of $\mathcal{T}(\mathcal{F}, \mathcal{N})$, since an abstracting step is first applied on the reference terms, of the form $g(x_1, \ldots, x_m)$, replacing $x_1, \ldots, x_m \in \mathcal{X}$ by $X_1, \ldots, X_m \in \mathcal{N}$. Then from Definition 2 we infer that in the most general unifiers σ_0 produced during the proof process, the variables of $Ran(\sigma_0)$ are only NF-variables.

Notice also that we are interested in the narrowing substitution applied to the current term t, but not in its definition on the variables of the left-hand side of the rule. So the narrowing substitutions we consider are restricted to the variables of the narrowed term t.

3.3 Cumulating constraints

Abstraction constraints have to be combined with the narrowing constrained substitutions to characterize the ground terms schematized by the proof trees. A narrowing step is applied to a current term u if the most general unifier σ_0 effectively corresponds to a rewriting step of ground instances of u, i.e. if σ_0 is *compatible* with the abstraction constrained formula A associated to u (i.e. $\sigma_0 A$ is satisfiable). So the narrowing constraint attached to the narrowing step is added to the abstraction constraints initially of the form $t\!\downarrow = X$. This motivates the introduction of abstraction constrained formulas.

Definition 6 *An* abstraction constrained formula *(ACF in short) is a formula $\bigwedge_i (t_i\!\downarrow = t'_i) \wedge \bigwedge_j \bigvee_{k_j} (u_{k_j} \neq v_{k_j})$, where $t_i, t'_i, u_{k_j}, v_{k_j} \in \mathcal{T}(\mathcal{F}, \mathcal{X} \cup \mathcal{N})$.*

Definition 7 *An* abstraction constrained formula *$A = \bigwedge_i (t_i\!\downarrow = t'_i) \wedge \bigwedge_j \bigvee_{k_j} (u_{k_j} \neq v_{k_j})$ is* satisfiable *iff there exists at least one instantiation θ such that $\bigwedge_i (\theta t_i\!\downarrow = \theta t'_i) \wedge \bigwedge_j \bigvee_{k_j} (\theta u_{k_j} \neq \theta v_{k_j})$. The instantiation θ is then said to satisfy the ACF A and is called solution of A.*

Applying a constrained substitution $\sigma = \sigma_0 \wedge \bigwedge_i \bigvee_{j_i} (x_{j_i} \neq t_{j_i})$ to an ACF A is done by applying σ_0 and adding the constraint part to A.

Definition 8 *Let $A = \bigwedge_i (t_i\!\downarrow = t'_i) \wedge \bigwedge_j \bigvee_{k_j} (u_{k_j} \neq v_{k_j})$ be an ACF and $\sigma = \sigma_0 \wedge \bigwedge_l \bigvee_{m_l} (x_{m_l} \neq w_{m_l})$ a constrained substitution. The application of the constrained substitution σ to the ACF A and noted σA is the ACF $(\bigwedge_i \sigma_0 t_i\!\downarrow = \sigma_0 t'_i) \wedge \bigwedge_j \bigvee_{k_j} (\sigma_0 u_{k_j} \neq \sigma_0 v_{k_j}) \wedge \bigwedge_l \bigvee_{m_l} (x_{m_l} \neq w_{m_l})$.*

An ACF A is attached to each term u in the proof trees; its solutions characterize the interesting ground instances of this term, that are the θu such that θ is a solution of A. When A has no solution, the current node of the proof tree doesn't represent any ground term. Such nodes are then irrelevant for the weak termination proof. So we have the choice between generating only the relevant nodes of the proof tree, by testing satisfiability of A at each step, or stopping the proof on a branch on an irrelevant node, by testing unsatisfiability of A.

Checking satisfiability of A is in general undecidable. The disequality part of an ACF is a particular instance of a disunification problem (a quantifier free equational formula, qfef in short), whose satisfiability has been addressed in [4], that provides rules to transform any disunification problem into a solved form. The satisfiability of such a qfef has also been addressed in [20], where a polynomial solving algorithm is proposed. Testing satisfiability of the equational part of an ACF is undecidable in general, but sufficient conditions can be given, relying on a characterization of normal forms.

Unsatisfiability of A is also undecidable in general, but simple sufficient conditions can be used, very often applicable in practice. They rely on reducibility, unifiability, narrowing and constructor tests, and and are given in the next subsection.

So both satisfiability and unsatisfiability checks need to use sufficient conditions. But in the first case, the proof process stops with failure as soon as satisfiability of A cannot be proved. In the second one, it can go on, until A is proved to be unsatisfiable, or until other stopping conditions are fulfilled. In the approach followed below, narrowing and abstraction are applied without checking the satisfiability of abstraction constraints, and the process stops as soon as they are detected to be unsatisfiable.

Remark that, as in our proof process, we establish weak termination of the ground instances solutions of the ACF A attached to each term u in the proof trees, we only have to apply our induction mechanism on these instances. Then the ordering constraints have in fact not to be satisfied for every instantiation, but only for the instantiations verifying the formula A (see proof of Theorem 1). As we will see on examples, abstraction constraints in A can be crucial to find an ordering satisfying the ordering constraints.

3.4 Testing unsatisfiability of abstraction constraints: sufficient conditions

According to Definition 7, an ACF $A = \bigwedge_i (t_i\!\downarrow\, = t'_i) \bigwedge_j \bigvee_{k_j} (x_{k_j} \neq t_{k_j})$ is unsatisfiable if in particular a conjunct $t_i\!\downarrow\, = t'_i$ is unsatisfiable, i.e. is such that $\theta t'_i$ is not an innermost normal form of θt_i for any ground substitution θ. Here we highlight four automatable cases of unsatisfiability of an abstraction constraint $t\!\downarrow\, = t'$:

Case 1: $t\!\downarrow\, = t'$, with t' reducible. Indeed, in this case, any ground instance of t' is reducible, and hence cannot be a normal form.

Case 2: $t\!\downarrow\, = t' \wedge \ldots \wedge t'\!\downarrow\, = t''$, with t' and t'' not unifiable. Indeed, any ground substitution θ satisfying the above conjunction is such that (1) $\theta t\!\downarrow\, = \theta t'$ and (2) $\theta t'\!\downarrow\, = \theta t''$. In particular, (1) implies that $\theta t'$ is in normal form and hence (2) imposes $\theta t' = \theta t''$, which is impossible if t' and t'' are not unifiable.

Case 3: $t\!\downarrow\, = t'$ with $top(t) \in \mathcal{C}ons$ and $top(t) \neq top(t')$. Indeed, if the top symbol of t is a constructor s, then any (innermost) normal form of any ground instance of t is of the form $s(u)$, where u is a ground term in normal form. The above constraint is therefore unsatisfiable if the top symbol of t' is g, for some $g \neq s$.

Case 4: $t\!\downarrow\, = t'$ with $t, t' \in \mathcal{T}(\mathcal{F}, \mathcal{N})$ not unifiable and $\bigwedge_{t \leadsto^{inn} v} v\!\downarrow\, = t'$ unsatisfiable. This criterion is of interest if the unsatisfiability of each conjunct $v\!\downarrow\, = t'$ can be shown with one of the four criteria we present here.

Then, to show unsatisfiability of an abstraction constraint in practice, we try in sequence the above four criteria from the less costly (case (3), that uses a simple syntactic comparison) to the most one (case (4), based on narrowing). The order in which criteria are tried is then (3), (1), (2), (4).

Example: Let us consider a system $\mathcal{R}$ on a signature containing the symbols $add : 2, 0 : 0, s : 1$, in which the addition is defined, as usual, by the two rules $add(x, 0) \to x, add(x, s(y)) \to s(add(x, y))$.
Then the abstraction constraint $plus(X, s(0))\downarrow = 0$ can bo shown unsatisfiable by using (4) and then (3). Indeed, the term $plus(X, s(0))$ can only be innermost narrowed into $s(plus(X, 0))$, and then the abstraction constraint $s(plus(X, 0))\downarrow = 0$ is shown unsatisfiable by (2), since $s \neq 0$.

4 Inference rules for inductive termination proofs

We are now ready to describe the different steps of our mechanism on a term t, with initial empty constraints conjunctions A, C. It consists in iterating the following steps.

- The first step abstracts the current term u at given positions $i_1, \ldots, i_p$. The constraints $t > u|_{i_1}, \ldots, u|_{i_p}$ are set, allowing to suppose, by induction, the existence of irreducible forms for $u|_{i_1}, \ldots, u|_{i_p}$. Then, $u|_{i_1}, \ldots, u|_{i_p}$ are abstracted into abstraction variables $X_{i_1}, \ldots, X_{i_p}$. The abstraction constraint $u|_{i_1}\downarrow = X_{i_1}, \ldots, u|_{i_p}\downarrow = X_{i_p}$ is added to the ACF A. We call that step the *abstract* step.

  The abstraction positions are chosen so that the abstraction mechanism captures the greatest possible number of rewriting steps: we abstract all greatest possible subterms of $u = f(u_1, \ldots, u_m)$. More concretely, we try to abstract $u_1, \ldots, u_m$ and, for each $u_i = g(v_1, \ldots, v_n)$ that cannot be abstracted, we try to abstract $v_1, \ldots, v_n$, and so on. In the worst case, we are driven to abstract leaves of the term, which are either NF-variables, that do not need to be abstracted, or constants.

  Note also that it is not useful to abstract non narrowable subterms. Indeed, in this case, by lifting lemma (see proof of Theorem 1) and by Definition 2, every instance of these subterms is in normal form, hence abstracting them is not needed.

- The second step innermost narrows the resulting term in one step with all possible rewrite rules of the rewrite system $\mathcal{R}$, and all possible substitutions σ, into terms v, according to Definition 5. This step is a branching step, creating as many states as narrowing possibilities. The substitution σ is integrated to A, according to Definition 8. This is the *narrow* step.

- We then have a *stop* step halting the proof process on the current branch of the proof tree, when A is detected to be unsatisfiable, or when the ground instances of the current term can be stated weakly innermost terminating. This can happen when the whole current term u can be abstracted (i.e. when the induction hypothesis applies on it), or if the current term u is not narrowable.

The previously presented steps are performed by inference rules that transform 3-tuples (T, A, C) where T is a set of terms of $\mathcal{T}(\mathcal{F}, \mathcal{X} \cup \mathcal{N})$, containing the current term whose weak innermost termination has to be proved: this is either a singleton or the empty set, A is an ACF and C is a conjunction of ordering constraints stated by the abstract steps.

Before giving the corresponding inference rules, let us notice that the inductive reasoning can be completed in the following way. When the induction hypothesis cannot be applied on a term u, it is sometimes possible to prove weak innermost termination of every ground instance of u by another way. Let $WEAK{-}TERMIN(u)$ be a predicate that is true iff every ground instance of u weakly innermost terminates. In the first (resp third) previous step of the induction reasoning, we then associate the alternative predicate $WEAK{-}TERMIN(u|_{i_j})$ (resp $WEAK{-}TERMIN(u)$) to the condition $t > u|_{i_j}$ (resp. $t > u$).

For establishing that $WEAK{-}TERMIN(u)$ is true, in some cases, the notion of usable rules can be used. Given a TRS $\mathcal{R}$ and a term $u \in \mathcal{T}(\mathcal{F}, \mathcal{X} \cup \mathcal{N})$, we determine the only rewrite rules

Table 1: Inference rules for the weak innermost termination proof

Abstract: $$\frac{\{u\},\ \ A,\ \ C}{\{u'\},\ A \wedge u|_{i_1}\downarrow = X_{i_1} \ldots \wedge u|_{i_p}\downarrow = X_{i_p},\ C \wedge H_C(u|_{i_1}) \ldots \wedge H_C(u|_{i_p})}$$

where u is abstracted into u' at the positions $i_1, \ldots, i_p \neq \epsilon$

if $\mathrm{C} \wedge \mathrm{H_C}(\mathrm{u}|_{\mathrm{i_1}}) \ldots \wedge \mathrm{H_C}(\mathrm{u}|_{\mathrm{i_p}})$ is satisfiable

where $H_C(u|_{i_j}) = \begin{cases} true & \text{if } WEAK\!-\!TERMIN(u|_{i_j}) \\ t_{ref} > u|_{i_j} & \text{otherwise.} \end{cases}$

Narrow: $$\frac{\{u\},\ \ A,\ \ C}{\{v\},\ \ \sigma A,\ \ C}$$

if $u \rightsquigarrow_{\sigma}^{Inn} v$

Stop: $$\frac{\{u\},\ \ A,\ \ C}{\emptyset,\ \ A,\ \ C \wedge H_C(u)}$$

if $(C \wedge H_C(u))$ is satisfiable or **A is unsatisfiable**

where $H_C(u) = \begin{cases} true & \text{if } WEAK\!-\!TERMIN(u) \text{ or } \textbf{A is unsatisfiable} \\ t_{ref} > u & \text{otherwise.} \end{cases}$

that are likely to apply to any of its ground instances, for the standard rewriting relation, until its ground normal form is reached, if it exists. Proving weak innermost termination of any ground instance of u then comes down to proving weak innermost termination of its usable rules, which is often much easier than proving weak innermost termination of the whole TRS. In particular, our inductive proof process itself can be applied. Usable rules are even sometimes strongly terminating, which can be ensured with the classical ordering based termination methods. This approach is fully developed in [11].

The termination proof procedure is described by the set of rules given in Table 1. These rules must be applied with a specific strategy S on the initial pairs $(\{t_{ref} = g(x_1, \ldots, x_m)\}, \top, \top)$, where g is a defined symbol.

Before giving the strategy, let us first point out a few properties of these rules: **Narrow** is a non-deterministic rule that can produce several results. **Narrow** is applied after **Abstract** with all possible narrowing substitutions and all possible rewrite rules. According to the innermost strategy, **Abstract** has to apply first, which is possible on $g(x_1, \ldots, x_m)$ if the induction ordering is assumed to have the subterm property.

The strategy S used to control the rules is: (**Abstract**; dk(**Narrow**); **Stop**) * where ";" denotes the sequential application of rules, "dk" the application of a rule in all possible ways and "*" the iterative application of a strategy, until it is not possible anymore. The process stops if no inference rule applies anymore.

There are two cases for the behavior of the termination proof procedure. The strategy applied to the initial state $(\{t_{ref}\}, \top, \top)$ terminates if the rules do not apply anymore and all states are of the form $(\emptyset, A, C)$. Otherwise, the strategy does not terminate if there is an infinite number of applications of **Abstract** and **Narrow**.

A branch of the derivation tree is said to be successful if it is ended by an application of **Stop**, i.e. if its final state is successful i.e. of the form $(\emptyset, A, C)$.

Thus, the inductive weak termination proof is successful if there is at least one successful branch corresponding to each possible ground term. Let us develop this point.

In fact, branching, produced by **Narrow**, can generate different states with narrowing substitutions $\sigma_1, \ldots \sigma_n$. These substitutions can be compared according to the subsumption ordering. For σ_i and σ_j, three situations may occur: σ_i is strictly less general than σ_j, σ_i and σ_j are equal up to a renaming, or else σ_i and σ_j are incomparable.

States corresponding to substitutions that are more general than other ones then represent a set of ground instances that contains the other ones. So, for proving weak termination for all ground instances at a branching point, it is sufficient to prove weak termination only for the most general instances.

Note that the ignored states may modelize different rewriting steps than those we consider (at different positions, with different rewrite rules). So for the considered instances, if a "most general state" doesn't exclusively give rise to successful branches, we lose the possibility to test whether the other branches are successful. In practice, this case rarely occurs and the gain is greater in avoiding to consider redundant subsets of instances.

Let us illustrate our purpose by the small example $\{f(a) \to b, f(g(x)) \to c, f(g(a)) \to f(g(a))\}$, where a, b, c are constants. Applying the inference rules on $f(x)$, we get:

$f(x)$	$A = \top$	$C = \top$		
Abstract				
$f(X)$	$A = (x\downarrow = X)$	$C = (f(x) > x)$		
Narrow				
b	$A = (x\downarrow = a)$	$C = (f(x) > x)$	$\sigma_1 = (X = a)$	$(branch\ 1)$
c	$A = (x\downarrow = g(X'))$	$C = (f(x) > x)$	$\sigma_2 = (X = g(X'))$	$(branch\ 2)$
$f(g(a))$	$A = (x\downarrow = g(a))$	$C = (f(x) > x)$	$\sigma_3 = (X = g(a))$	$(branch\ 3)$

Narrow here produces one branch with the substitution $\sigma_1 = (X = a)$, one with the substitution $\sigma_2 = (X = g(X'))$ and one with the substitution $\sigma_3 = (X = g(a))$. The first narrowing branch with σ_1 modelizes the rewriting of the ground instances of $f(X)$ satisfying the substitution $\sigma = (X = a)$, i.e. the term $f(a)$. The second branch with σ_2 represents all ground instances of $f(X)$ satisfying the substitution $\sigma = (X = g(X'))$, i.e. all possible ground instances of $f(g(X'))$. The third one with σ_3 represents all ground instances of $f(X)$ satisfying the substitution $\sigma = (X = g(a))$, i.e. the term of $f(g(a))$. As the second branch represents ground instances, that are not represented by the other branches, we have to develop it for the termination proof. On the contrary, the third one is useless, and we can suppress it.

Therefore, for proving weak termination of all ground instances of $f(x)$, it will be enough to prove weak innermost termination from the state $(\{b\},\ A = (x\downarrow = a), C = (f(x) > x))$, and from the state $(\{c\}, A = (x\downarrow = g(X')), C = (f(x) > x))$. We then have:

Stop(*twice*)		
$\emptyset$	$A = (x\downarrow = a)$	$C = (f(x) > x)$
$\emptyset$	$A = (x\downarrow = g(X'))$	$C = (f(x) > x)$
$f(g(a))$	$A = (x\downarrow = g(a))$	$C = (f(x) > x)$

which ends the weak termination proof. The third branch, which gives an infinite succession of **Abstract** and **Narrow** from $(f(g(a)), A = (x\downarrow = g(a)), C = (f(x) > x))$ is useless in the proof.

This example shows how at each branching point, one can prune some branches. Let us formalize that now.

A branching node in a proof tree is a state, on which the Narrow rule applies. Let Σ be the set of narrowing substitutions (possibly with different rewrite rules) at a given branching node. Let Σ_0 be the reduced set from Σ such that $\sigma \in \Sigma_0$ iff $\sigma \in \Sigma$ and $\nexists\, \sigma' \in \Sigma$ such that $\sigma > \sigma'$ on $(Dom(\sigma) \setminus Var(l)) \cup (Dom(\sigma') \setminus Var(l'))$, where l and l' are the left-hand sides of rules respectively used to produce the narrowing substitutions σ and σ'. The set Σ_0 may yet contain equivalent substitutions which are marked as such. So for any two substitutions in Σ_0, either they are equivalent, or they are incomparable.

A proof tree is weakly successful if it is reduced to a state of the form $(\emptyset, A, C)$, or if at each branching node:

- for each class of equivalent substitutions, there exists at least one weakly successful subtree corresponding to a substitution in this class,
- all subtrees corresponding to incomparable substitutions are weakly successful.

So the strategy S can be optimized as follows: at each branching point of a proof tree, with set of substitutions Σ, we only develop the subtrees corresponding to Σ_0. Moreover, given two subtrees corresponding to equivalent substitutions, as soon as one of them is weakly successful, the other one is cut.

We write $SUCCESS(g, \succ)$ if the proof tree obtained by application on $(\{g(x_1, \ldots, x_m)\}, \top, \top)$, with the strategy S, of the inference rules whose conditions are satisfied by an ordering $\succ$, is weakly successful.

Theorem 1 *Let $\mathcal{R}$ be a TRS on a set $\mathcal{F}$ of symbols. If there exists an $\mathcal{F}$-stable ordering $\succ$ having the subterm property, such that for each defined symbol g, we have $SUCCESS(g, \succ)$, then every term of $\mathcal{T}(\mathcal{F})$ weakly innermost terminates.*

An important point is that the ordering $\succ$ has to be the same for all $g(x_1, \ldots, x_m) \in \mathcal{D}ef$.

A formal description with a complete set of inference rules for describing the subtree cut process is given in Section 7.

5 Finding a good derivation chain

As said previously, establishing weak termination of an undeterministic evaluation process warrants a result if a breadth-first strategy is adopted for this process. But such a strategy is in general very costly, and it is much better to have hints about the terminating derivations to compute them directly with a depth-first mechanism.

Our proof process, as it simulates the rewriting mechanism, gives complete information on a terminating rewriting branch. It allows extracting the exact application of rewrite rules that yields a normal form. To rewrite a term, it is enough to follow the rewriting scheme simulated by abstraction and narrowing in the proof trees.

We now formalize the use of the proof trees to compute a normal form for any term.

Definition 9 *Let $\mathcal{R}$ be a TRS proved weakly terminating with Theorem 1. The strategy tree ST_f associated to $f \in \mathcal{D}ef_{\mathcal{R}}$ is the proof tree obtained from the initial state $(\{f(x_1, \ldots, x_m)\}, \top, \top)$.*

Remark that although we suppress redundant branches when the strategy trees are developed, we can get redundant branches at some nodes, i.e. branches representing subsets of ground instances of other branches. This is the case when we get two final states in the same time. Let us give an illustrative example of this fact.

Example: Let us consider the following rewrite system $\mathcal{R}$, built on $\mathcal{F} = \{g : 1, h : 1, s : 1, a : 0, b : 0\}$:

$$\begin{array}{ll} g(s(x)) & \to h(x) \\ g(s(s(x))) & \to g(s(s(s(x)))) \\ h(x) & \to a \\ h(x) & \to b \end{array}$$

and let us develop the proof tree for the symbol g :

$$
\begin{tikzcd}
 & g(x_1) \arrow[d, "Abstract"] & \\
 & g(X_1) \arrow[d, "Narrow", "\sigma=(X_1=s(X_2))"'] & \\
 & h(X_2) \arrow[dl, "Narrow"', "\sigma=Id"] \arrow[dr, "Narrow", "\sigma=Id"'] & \\
a \arrow[d, "Stop"'] & & b \arrow[d, "Stop"'] \\
\emptyset & & \emptyset
\end{tikzcd}
$$

Note that $g(X_1)$ is also narrowable into $g(s(s(s(X_3))))$ with the substitution $\sigma' = (X_1 = s(s(X_3)))$. However, since σ' is strictly less general than the other narrowing substitution $\sigma = (X_1 = s(X_2))$, the narrowing branch with σ' is useless, hence not generated.

The two branches of the proof tree are successful, with same depth, and the instances covered by the branch obtained with one substitution Id are obviously also covered by the branch obtained with the other substitution Id.

The breadth first strategy, developping the same breadth nodes simultaneously, does not allow to detect that the branch with a is redundant, before the branch with b has been proved successful. So both branches appear in the strategy tree.

The normalizing process following the strategy trees is not necessarily deterministic. Several successful branches representing common ground terms can be generated because of the breadth first proof strategy, before the **Cut** process suppresses the possible redundant ones. This does not matter anyway, since one can choose any successful branch to reach a normal form.

Definition 10 *Let $\mathcal{R}$ be a TRS proved weakly terminating with Theorem 1. Let $ST = \{ST_f | f \in \mathcal{D}ef_{\mathcal{R}}\}$ be the set of strategy trees of $\mathcal{R}$ and $s = f(s_1, \ldots, s_m) \in \mathcal{T}(\mathcal{F})$. Normalizing s with respect to ST into $norm_{ST}(s)$ is defined in the following way:*

- *if $f \in \mathcal{C}ons_{\mathcal{R}}$, then $norm_{ST}(f(s_1, \ldots, s_n)) = f(norm_{ST}(s_1), \ldots, norm_{ST}(s_n))$,*
- *if $f \in \mathcal{D}ef_{\mathcal{R}}$, then normalizing s with respect to ST into $norm_{ST}(s)$ is performed by following the steps in the strategy tree ST_f of f, where $t = g(t_1, \ldots, t_n)$ is any term of the transformation chain of t with respect to ST and $u = g(u_1, \ldots, u_n)$ is the corresponding term in ST_f:*
  - *if the step is* **Abstract***, and abstracts u at positions $i_1, \ldots, i_p$, then $t \mapsto t[t'_1]_{i_1} \ldots [t'_p]_{i_p}$,*
    where $t'_j = \begin{cases} t|_{i_j}\downarrow & \text{if } WEAK\!-\!TERMIN(u|_{i_j}) \\ norm_{ST}(t|_{i_j}) & \text{otherwise,} \end{cases}$
  - *if the step is* **Narrow** *with $g(u_1, \ldots, u_n) \rightsquigarrow^{Inn}_{p,l\to r,\sigma} u'$, then $g(t_1, \ldots, t_n) \mapsto t'$*
    where t' is defined by $g(t_1, \ldots, t_n) \to^{Inn}_{p,l\to r,\mu} t' = \mu u'$, where $\theta = \mu\sigma[Var(g(u_1, \ldots, u_n)]$
    with $g(t_1, \ldots, t_n) = \theta g(u_1, \ldots, u_n)$ if μ exists,
    $t' = g(t_1, \ldots, t_n)$ otherwise, and in this case, the normalizing process stops,
  - *if the step is* **Stop***, then $g(t_1, \ldots, t_n) \mapsto t'$,*
    where $t' = \begin{cases} g(t_1, \ldots, t_n)\downarrow & \text{if } WEAK\!-\!TERMIN(g(u_1, \ldots, u_n)) \\ norm_{ST}(g(t_1, \ldots, t_n)) & \text{otherwise.} \end{cases}$

Given a TRS $\mathcal{R}$, the previous definition assumes that if the predicate $WEAK\!-\!TERMIN$ has been used to prove termination of a particular term t during the termination proof of $\mathcal{R}$, one is able to find a normalizing strategy for t. A simple sufficient condition is that t is proved strongly terminating, which can be established in most cases, like for $WEAK-TERMIN$, with the usable rules. Under this assumption, the following theorem holds.

Theorem 2 *Let $\mathcal{R}$ be a TRS proved weakly terminating with Theorem 1 and ST its set of strategy trees. Then for any term $t \in \mathcal{T}(\mathcal{F})$, $norm_{ST}(t)$ is an innermost normal form of t for $\mathcal{R}$.*

Let us come back to the TRS $\mathcal{R}$ presented in the introduction, built on $\mathcal{F} = \{f : 1, p : 1, s : 1, 0 : 0\}$. Let us first prove that every ground term t of $\mathcal{T}(\mathcal{F})$ can be innermost normalized with $\mathcal{R}$, and then infer from this proof a strategy allowing normalization of any ground term of $\mathcal{T}(\mathcal{F})$.

Since the only defined symbols of $\mathcal{R}$ are f and p, we have to apply the inference rules to $f(x_1)$ and to $p(x_1)$.

In the following, we give the states of the proof trees together with the position they have in these trees. For readability, we do not write the development of branches that are going to be cut. However, we will $\boxed{\text{highlight}}$ the first state of such branches when it is generated, will then write it in *italic* until highlighting it again when the branches initiated by this state are cut. Moreover, for readability, we will write in **bold** the states on which the next inference rule applies.

Let us apply the inference rules on $f(x_1)$.

ϵ	$f(x_1)$	$A = \top$	$C = \top$	
Abstract				
1	$f(X_1)$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	
Narrow				
1.1	$p(s(X_1))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
$\boxed{1.2}$	$p(s(s(X_1)))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$

In the following, we show that the branch starting from the state 1.1 is successful, which allows to cut the branch starting from 1.2. Formally, both branches should be developped in parallel, but for readability, we do not develop the latter branch here.

Narrow				
1.1.1	$p(X_2)$	$A = (x_1{\downarrow} = s(X_2))$	$C = (f(x_1) > x_1)$	$\sigma = (X_1 = s(X_2))$
1.1.2	$f(0)$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = (X_1 = 0)$
1.2	$p(s(s(X_1)))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	
Stop				
1.1.1.1	$\emptyset$	$A = (x_1{\downarrow} = s(X_2))$	$C = (f(x_1) > x_1, p(X_2))$	
1.1.2	$f(0)$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	
1.2	$p(s(s(X_1)))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	
Narrow				
1.1.1.1	$\emptyset$	$A = (x_1{\downarrow} = s(X_2))$	$C = (f(x_1) > x_1, p(X_2))$	
$\boxed{1.1.2.1}$	$p(s(0))$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.1.2.2	$p(s(s(0)))$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.2	$p(s(s(X_1)))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	

Stop applies on $p(X_2)$ with any simplification ordering with the precedence $f \succ_{\mathcal{F}} p$. Indeed, assuming that any ground term is greater than any of its normal forms, we get from $A = (x_1{\downarrow} = s(X_2))$ that any ground substitution satisfying A is such that $\theta x_1 \succeq \theta s(X_2) \succ \theta X_2$, and hence $\theta f(x_1) \succ \theta p(X_2)$ with the latter simplification ordering.

The above assumption according to which any ground term is greater than any of its normal forms can be made here, taking advantage of the fact that the studied system is well-covered, which implies that any ground term in normal form is composed of constructor symbols only. In this case, by choosing for the induction ordering a precedence-based ordering with defined symbols greater than constructor symbols, we get the desired property.

In the following, we show that the branch starting from the state 1.1.2.2 is successful, which allows to cut the branch starting from 1.1.2.1. For readability, as previously, we do not develop this latter branch.

Narrow				
1.1.1.1	$\emptyset$	$A = (x_1{\downarrow} = s(X_2))$	$C = (f(x_1) > x_1, p(X_2))$	
1.1.2.1	$p(s(0))$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	
1.1.2.2.1	$p(0)$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.2	$p(s(s(X_1)))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	
Narrow				
1.1.1.1	$\emptyset$	$A = (x_1{\downarrow} = s(X_2))$	$C = (f(x_1) > x_1, p(X_2))$	
1.1.2.1	$p(s(0))$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	
1.1.2.2.1.1	0	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.2	$p(s(s(X_1)))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	
Stop				
1.1.1.1	$\emptyset$	$A = (x_1{\downarrow} = s(X_2))$	$C = (f(x_1) > x_1, p(X_2))$	
$\boxed{\mathit{1.1.2.1}}$	$p(s(0))$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	
1.1.2.2.1.1.1	$\emptyset$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	
$\boxed{\mathit{1.2}}$	$p(s(s(X_1)))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	

We can now cut the subtree starting from the state 1.1.2.1, since the subtree starting from the state 1.1.2.2, generated with the same narrowing substitution Id, is successful.
We can now cut the subtree starting from the state 1.2, since the subtree starting from the state 1.1, generated with the same narrowing substitution Id, is successful. We finally get a successful tree for the symbol f.

Let us now apply the inference rules on $p(x_1)$.

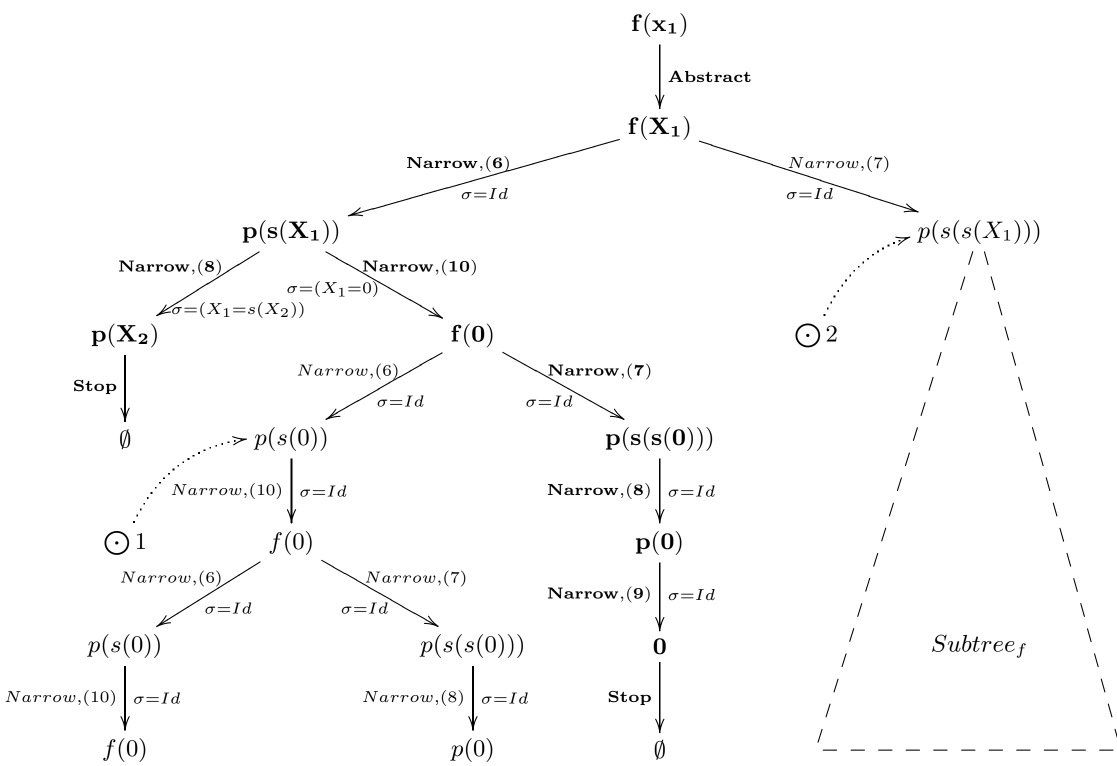
ϵ	$p(x_1)$	$A = \top$	$C = \top$	
Abstract				
1	$p(X_1)$	$A = (x_1{\downarrow} = X_1)$	$C = (p(x_1) > x_1)$	
Narrow				
1.1	0	$A = (x_1{\downarrow} = 0)$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = 0)$
1.2	$f(0)$	$A = (x_1{\downarrow} = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = s(0))$
1.3	$p(X_2)$	$A = (x_1{\downarrow} = s(s(X_2)))$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = s(s(X_2))$
Stop	(*on two branches*)			
1.1.1	$\emptyset$	$A = (x_1{\downarrow} = 0)$	$C = (p(x_1) > x_1)$	
1.2	$f(0)$	$A = (x_1{\downarrow} = s(0))$	$C = (p(x_1) > x_1)$	
1.3.1	$\emptyset$	$A = (x_1{\downarrow} = s(s(X_2)))$	$C = (p(x_1) > x_1, p(X_2))$	
Narrow				
1.1.1	$\emptyset$	$A = (x_1{\downarrow} = 0)$	$C = (p(x_1) > x_1)$	
$\boxed{1.2.1}$	$p(s(0))$	$A = (x_1{\downarrow} = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = Id$
1.2.2	$p(s(s(0)))$	$A = (x_1{\downarrow} = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = Id$
1.3.1	$\emptyset$	$A = (x_1{\downarrow} = s(s(X_2)))$	$C = (p(x_1) > x_1, p(X_2))$	

In the following, we show that the branch starting from the state 1.2.2 is successful, which allows to cut the branch starting from the state 1.2.1, which we hence do not develop, as previously, for readability.

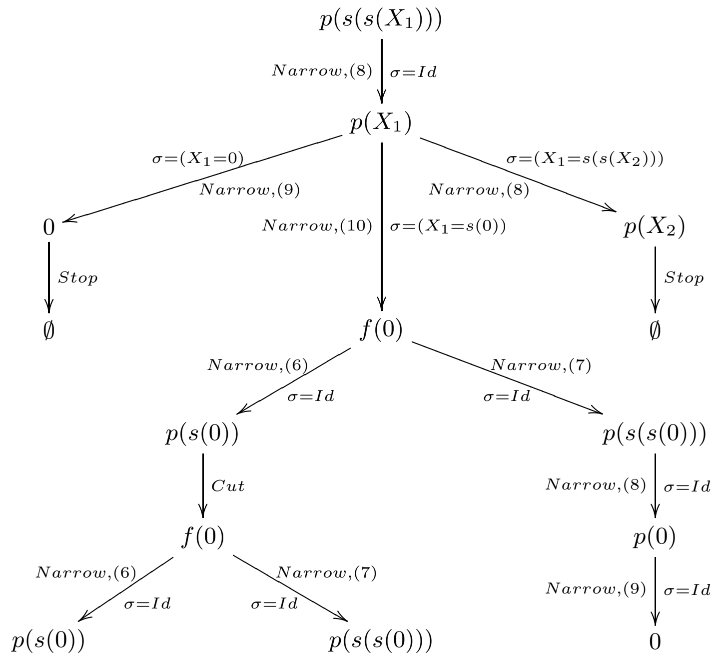
Narrow				
1.1.1	$\emptyset$	$A = (x_1\downarrow = 0)$	$C = (p(x_1) > x_1)$	
1.2.1	$p(s(0))$	$A = (x_1\downarrow = s(0))$	$C = (p(x_1) > x_1)$	
1.2.2.1	$p(0)$	$A = (x_1\downarrow = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = Id$
1.3.1	$\emptyset$	$A = (x_1\downarrow = s(s(X_2)))$	$C = (p(x_1) > x_1, p(X_2))$	
Narrow				
1.1.1	$\emptyset$	$A = (x_1\downarrow = 0)$	$C = (p(x_1) > x_1)$	
1.2.1	$p(s(0))$	$A = (x_1\downarrow = s(0))$	$C = (p(x_1) > x_1)$	
1.2.2.1.1	0	$A = (x_1\downarrow = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = Id$
1.3.1	$\emptyset$	$A = (x_1\downarrow = s(s(X_2)))$	$C = (p(x_1) > x_1, p(X_2))$	
Stop				
1.1	$\emptyset$	$A = (x_1\downarrow = 0)$	$C = (p(x_1) > x_1)$	
$\boxed{\textit{1.2.1}}$	$p(s(0))$	$A = (x_1\downarrow = s(0))$	$C = (p(x_1) > x_1)$	
1.2.2.1.1.1	$\emptyset$	$A = (x_1\downarrow = s(0))$	$C = (p(x_1) > x_1)$	
1.3	$\emptyset$	$A = (x_1\downarrow = s(s(X_2)))$	$C = (p(x_1) > x_1, p(X_2))$	

We can now cut the subtree starting from the state 1.2.1, since the subtree starting from the state 1.2.2, generated with the same narrowing substitution Id, is successful. We finally get a successful tree for the symbol f. We draw it below. When **Narrow** applies, we specify the narrowing substitution in the drawing and, in parentheses, the rewrite rule number used for narrowing.

The proof derivation tree obtained by applying the inference rules to $f(x_1)$ is the following.

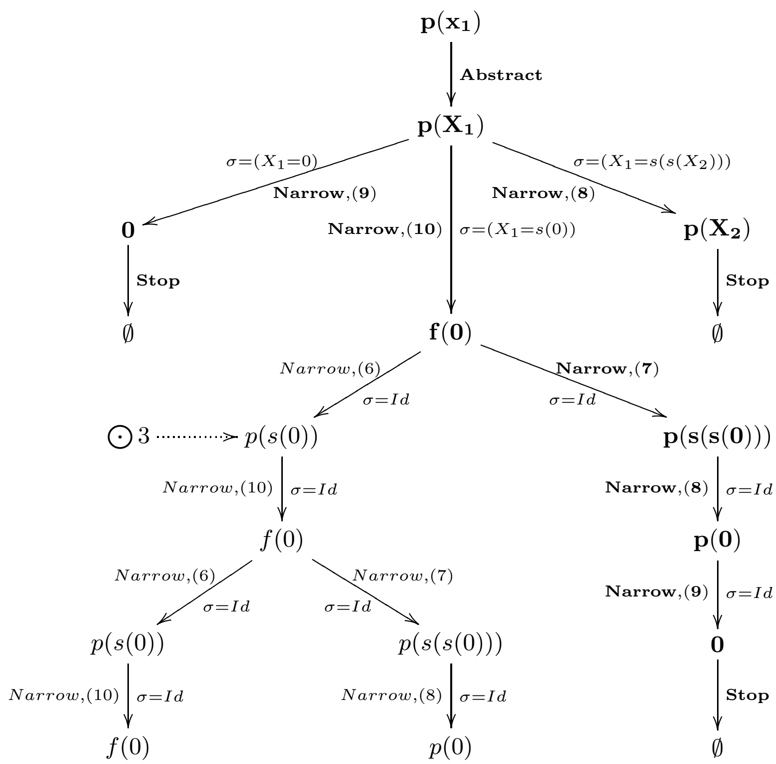


where $Subtree_f$ is the following subtree starting from $p(s(s(X_1)))$, as deep as the subtree starting from $p(s(X_1))$ on the left:



The subtree $\odot$ 1 is cut as soon as the second subtree generated on the right from $\mathbf{f(0)}$ with the same substitution Id is successful. Then the subtree $\odot$ 2, denoted $Subtree_f$, can be cut, since the subtree on the left generated from $\mathbf{f(X_1)}$ with the same substitution Id becomes successful.

The proof tree obtained by applying the inference rules to $p(x_1)$ is the following.



In the above proof tree, the subtree $\odot$ 3 is cut when the second subtree generated on the right from $\mathbf{f(0)}$ with the same substitution Id is successful.

The final proof trees are bold. Since they are both successful, $\mathcal{R}$ is proved weakly innermost terminating on the ground term algebra.

We can now infer from these trees a strategy normalizing any ground term t, according to Definition 10. As an example, let us use the strategy to normalize the term $f(s(s(s(0))))$ following the steps of ST_f.

(Step 1 in $\mathbf{ST_f}$: Abstract) The first step is **Abstract** at position 1, and then we get $f(s(s(s(0)))) \mapsto f(norm_{ST}(s(s(s(0)))))$. Since s is a constructor, we have :
$norm_{ST}(s(s(s(0)))) = s(norm_{ST}(s(s(0)))) = s(s(norm_{ST}(s(0)))) = s(s(s(norm_{ST}(0))))$.
Since 0 is a constant constructor, we get $norm_{ST}(0) = 0$, and finally $norm_{ST}(s(s(s(0)))) = s(s(s(0)))$. We are now on $f(X_1)$ in ST_f, with the current term $f(s(s(s(0))))$ in the derivation.

(Step 2 in $\mathbf{ST_f}$: Narrow) The second step is **Narrow** at the top position, with rule (6). The narrowing substitution σ is such that our current term is a ground instance of $\sigma f(X_1)$. So $f(s(s(s(0)))) \longrightarrow^{\epsilon,(6)} p(s(s(s(s(0)))))$.

(Step 3 in $\mathbf{ST_f}$: Narrow) The third step is **Narrow** at the top position, with rules (8) and (10). For (10), there is no narrowing substitution σ such that our current term $p(s(s(s(s(0)))))$ is a ground instance of $\sigma p(s(X_1))$. For (8), however, this is the case. So we rewrite our current term in the derivation with (8). We get : $p(s(s(s(s(0))))) \longrightarrow^{\epsilon,(8)} p(s(s(0)))$.

(Step 4 in $\mathbf{ST_f}$: Stop) The current step in the tree is **Stop** thanks to the induction hypothesis, and then we get $norm_{ST}(p(s(s(0))))$. We now have to follow ST_p to evaluate $norm_{ST}(p(s(s(0))))$.

(Step 1 in $\mathbf{ST_p}$: Abstract) Since the first step of ST_p is **Abstract** at position 1, we get $p(s(s(0))) \mapsto p(norm_{ST}(s(s(0))))$. Reasoning as previously, we have $norm_{ST}(s(s(0))) = s(s(0))$.

(Step 2 in $\mathbf{ST_p}$: Narrow) The second step is **Narrow** at the top position with rules $(8), (9), (10)$. The only rule such that the narrowing substitution σ is such that our current term $p(s(s(0)))$ is a ground instance of $\sigma p(X_i)$ is the rule (8), and then we get : $p(s(s(0))) \longrightarrow^{\epsilon,(8)} p(0)$.

(Step 3 in $\mathbf{ST_p}$: Stop) The current step in ST_p is **Stop** thanks to the induction hypothesis, and then we get $norm_{ST}(p(0))$. Once again, we have to follow ST_p to evaluate $norm_{ST}(p(0))$.

(Step 1 in $\mathbf{ST_p}$: Abstract) The first step is **Abstract** at position 1, and then we get $p(0) \mapsto p(norm_{ST}(0))$. Since 0 is a constant constructor, we have $norm_{ST}(0) = 0$.

(Step 2 in $\mathbf{ST_p}$: Narrow) The second step is **Narrow** at the top position with rules $(8), (9), (10)$. The only possible narrowing substitution is the one of the rule (9), and then we get : $p(0) \longrightarrow^{\epsilon,(9)} 0$.

(Step 3 in $\mathbf{ST_p}$: Stop) The current step is **Stop** on a ground term in normal form, which ends the normalizing process on 0, which hence is a normal form of $f(s(s(s(0))))$.

6 Another example

Let us consider the following TRS:

$$
\begin{aligned}
f(g(x), s(0)) &\rightarrow f(g(x), g(x)) && (6)\\
f(g(x), s(y)) &\rightarrow f(h(x, y), s(0)) && (7)\\
g(s(x)) &\rightarrow s(g(x)) && (8)\\
g(0) &\rightarrow 0 && (9)\\
h(x, y) &\rightarrow g(x). && (10)
\end{aligned}
$$

Obviously, because of the rule (2), $\mathcal{R}$ is not terminating, nor even innermost terminating. For instance, the following infinite sequence is possible in $\mathcal{R}$: $\underline{f(g(f(0,0)),s(0))} \rightarrow^{(2)} f(\underline{h(f(0,0),0)},s(0)) \rightarrow^{(5)} \underline{f(g(f(0,0)),s(0))}\ldots$ However, $\mathcal{R}$ is weakly innermost terminating ; in particular, the cycle above can be avoided by using the rule (1) instead of (2).

We prove weak innermost termination of $\mathcal{R}$ on $\mathcal{T}(\mathcal{F})$ with $\mathcal{F} = \{f:2, h:2, g:1, s:1, 0:0\}$. Since the defined symbols of $\mathcal{R}$ are f, g and h, we have to apply the inference rules to $f(x_1,x_2)$, $g(x_1)$ and $h(x_1,x_2)$.

Applying the inference rules to $(f(x_1,x_2),\top,\top)$, we get:

ϵ	$f(x_1,x_2)$	$A=\top$ $C=\top$
Abstract		
1	$f(X_1,X_2)$	$A=(x_1{\downarrow}=X_1 \wedge x_2{\downarrow}=X_2)$ $C=(f(x_1,x_2)>x_1,x_2)$
Narrow		
1.1	$f(h(X_3,X_4),s(0))$	$A=(x_1{\downarrow}=g(X_3)\wedge x_2{\downarrow}=s(X_4))\wedge(X_3\neq s(X_5)\wedge X_3\neq 0)$ $C=(f(x_1,x_2)>x_1,x_2)$ $\sigma=(X_1=g(X_3)\wedge X_2=s(X_4))\wedge(X_3\neq s(X_5)\wedge X_3\neq 0)$
Abstract		
1.1.1	$f(X_6,s(0))$	$A=(x_1{\downarrow}=g(X_3)\wedge x_2{\downarrow}=s(X_4)\wedge h(X_3,X_4){\downarrow}=X_6)$ $\wedge(X_3\neq s(X_5)\wedge X_3\neq 0)$ $C=(f(x_1,x_2)>x_1,x_2)$

Note that the term of the state 1 could also be narrowed into $f(g(X_3'),g(X_3'))$, with rule (1) and the narrowing substitution $\sigma' = (X_1 = g(X_3') \wedge X_2 = s(0)) \wedge (X_3' \neq s(X_5') \wedge X_3' \neq 0)$. But this substitution is strictly less general than the narrowing substitution σ used above, and then, according to the strategy S, this narrowing possibility is not considered for proving weak termination. More formally, we have $\Sigma = \{\sigma,\sigma'\}$ and, since $\sigma' > \sigma$, $\Sigma_0 = \{\sigma\}$. We then have to develop only the subtrees corresponding to Σ_0.

The second **Abstract** applies on the subterm $h(X_3,X_4)$ thanks to the $WEAK\!-\!TERMIN$ predicate and the usable rules. Indeed, the usable rules of $h(X_3,X_4)$ consist of the system $\{h(x,y)\rightarrow g(x), g(s(x))\rightarrow s(g(x)), g(0)\rightarrow 0\}$, orientable with any path ordering with the precedence $h \succ_{\mathcal{F}} g \succ_{\mathcal{F}} s$. Let us go on the proof process.

Narrow		
1.1.1.1	$f(g(X_7),g(X_7))$	$A=(x_1{\downarrow}=g(X_3)\wedge x_2{\downarrow}=s(X_4)\wedge h(X_3,X_4){\downarrow}=g(X_7))$ $\wedge(X_3\neq s(X_5)\wedge X_3\neq 0\wedge X_7\neq s(X_8)\wedge X_7\neq 0)$ $C=(f(x_1,x_2)>x_1,x_2)$ $\sigma=(X_6=g(X_7))\wedge(X_7\neq s(X_8)\wedge X_7\neq 0)$
1.1.1.2	$f(h(X_9,0),s(0))$	$A=(x_1{\downarrow}=g(X_3)\wedge x_2{\downarrow}=s(X_4)\wedge h(X_3,X_4){\downarrow}=g(X_9))$ $\wedge(X_3\neq s(X_5)\wedge X_3\neq 0\wedge X_9\neq s(X_{10})\wedge X_9\neq 0)$ $C=(f(x_1,x_2)>x_1,x_2)$ $\sigma=(X_6=g(X_9))\wedge(X_9\neq s(X_{10})\wedge X_9\neq 0)$
Abstract	(*on the two states*)	
1.1.1.1.1	$f(X_{11},X_{12})$	$A=(x_1{\downarrow}=g(X_3)\wedge x_2{\downarrow}=s(X_4)\wedge h(X_3,X_4){\downarrow}=g(X_7)$ $\wedge g(X_7){\downarrow}=X_{11}\wedge g(X_7){\downarrow}=X_{12})$ $\wedge(X_3\neq s(X_5)\wedge X_3\neq 0\wedge X_7\neq s(X_8)\wedge X_7\neq 0)$ $C=(f(x_1,x_2)>x_1,x_2)$
1.1.1.2.1	$f(X_{13},s(0))$	$A=(x_1{\downarrow}=g(X_3)\wedge x_2{\downarrow}=s(X_4)\wedge h(X_3,X_4){\downarrow}=g(X_9)$ $\wedge h(X_9,0){\downarrow}=X_{13})\wedge(X_3\neq s(X_5)\wedge X_3\neq 0\wedge X_9\neq s(X_{10})\wedge X_9\neq 0)$ $C=(f(x_1,x_2)>x_1,x_2)$

Narrow	(*on the two states*)	
1.1.1.1.1.1	$f(h(X_{14},X_{15}),s(0))$	$A=(x_1{\downarrow}=g(X_3)\wedge x_2{\downarrow}=s(X_4)\wedge \underline{h(X_3,X_4){\downarrow}=g(X_7)}$ $\wedge\, g(X_7){\downarrow}=g(X_{14})\wedge \underline{g(X_7){\downarrow}=s(X_{15})})\wedge(X_3\neq s(X_5)$ $\wedge\, X_3\neq 0\wedge X_7\neq s(X_8)\wedge X_7\neq 0\wedge X_{14}\neq s(X_{16})\wedge X_{14}\neq 0)$ $C=(f(x_1,x_2)>x_1,x_2)$ $\sigma=(X_{11}=g(X_{14})\wedge X_{12}=s(X_{15}))\wedge(X_{14}\neq s(X_{16})\wedge X_{14}\neq 0)$
1.1.1.2.1.1	$f(g(X_{17}),g(X_{17}))$	$A=(x_1{\downarrow}=g(X_3)\wedge x_2{\downarrow}=s(X_4)\wedge h(X_3,X_4){\downarrow}=g(X_9)$ $\wedge\, h(X_9,0){\downarrow}=g(X_{17}))\wedge(X_3\neq s(X_5)\wedge X_3\neq 0$ $\wedge\, X_9\neq s(X_{10})\wedge X_9\neq 0\wedge X_{17}\neq s(X_{18})\wedge X_{17}\neq 0)$ $C=(f(x_1,x_2)>x_1,x_2)$ $\sigma=(X_{13}=g(X_{17}))\wedge(X_{17}\neq s(X_{18})\wedge X_{17}\neq 0)$
1.1.1.2.1.2	$f(h(X_{19},0),s(0))$	$A=(x_1{\downarrow}=g(X_3)\wedge x_2{\downarrow}=s(X_4)\wedge h(X_3,X_4){\downarrow}=g(X_9)$ $\wedge\, h(X_9,0){\downarrow}=g(X_{19}))\wedge(X_3\neq s(X_5)\wedge X_3\neq 0$ $\wedge\, X_9\neq s(X_{10})\wedge X_9\neq 0\wedge X_{19}\neq s(X_{20})\wedge X_{19}\neq 0)$ $C=(f(x_1,x_2)>x_1,x_2)$ $\sigma=(X_{13}=g(X_{19}))\wedge(X_{19}\neq s(X_{20})\wedge X_{19}\neq 0)$
Stop		
1.1.1.1.1.1.1	$\emptyset$	$A=(x_1{\downarrow}=g(X_3)\wedge x_2{\downarrow}=s(X_4)\wedge h(X_3,X_4){\downarrow}=g(X_7)$ $\wedge\, g(X_7){\downarrow}=g(X_{14})\wedge g(X_7){\downarrow}=s(X_{15}))\wedge(X_3\neq s(X_5)\wedge X_3\neq 0$ $\wedge\, X_7\neq s(X_8)\wedge X_7\neq 0\wedge X_{14}\neq s(X_{16})\wedge X_{14}\neq 0)$ $C=(f(x_1,x_2)>x_1,x_2)$

Stop applies on the state 1.1.1.1.1.1 because the abstraction constrained formula of this state is not satisfiable. For readibility, we have underlined the conjuncts of the formula that make it unsatisfiable. The reasoning is the following: let us assume the existence of a ground substitution θ satisfying the formula. We have in particular $\theta h(X_3,X_4){\downarrow}=\theta g(X_7)$ and $\theta g(X_7){\downarrow}=\theta s(X_{15})$. The first equality requires $\theta g(X_7)$ to be in normal form, while the second equality implies that $\theta g(X_7)$ can be rewritten. Therefore such a ground substitution θ cannot exist, and hence the constraint is not satisfiable. We have automatable sufficient conditions to detect such cases of unsatisfiability [**?**].

The branch starting from the state 1.1.1.1, generated from the state 1.1.1 with the narrowing substitution $\sigma=(X_6=g(X_7))\wedge(X_7\neq s(X_8)\wedge X_7\neq 0)$ is now successful. Hence the branch starting from the state 1.1.1.2, generated from the state 1.1.1 with the narrowing substitution $\sigma=(X_6=g(X_9))\wedge(X_9\neq s(X_{10})\wedge X_9\neq 0)$ can be cut, since both substitutions are equivalent.

Applying the inference rules to $(g(x_1),\top,\top)$, we get:

ϵ	$g(x_1)$	$A=\top$	$C=\top$	
Abstract				
1	$g(X_1)$	$A=(x_1{\downarrow}=X_1)$	$C=(g(x_1)>x_1)$	
Narrow				
1.1	$s(g(X_2))$	$A=(x_1{\downarrow}=s(X_2))$	$C=(g(x_1)>x_1)$	$\sigma=(X_1=s(X_2))$
1.2	0	$A=(x_1{\downarrow}=0)$	$C=(g(x_1)>x_1)$	$\sigma=(X_1=0)$
Stop				
1.1.1	$\emptyset$	$A=(x_1{\downarrow}=s(X_2))$	$C=(g(x_1)>x_1)$	
1.2.1	$\emptyset$	$A=(x_1{\downarrow}=0)$	$C=(g(x_1)>x_1)$	

Stop applies to the state 1.1 thanks to the $WEAK\!-\!TERMIN$ predicate and the usable rules. Indeed, the usable rules of $s(g(X_2))$ consist of the system $\{g(s(x))\to s(g(x)), g(0)\to 0\}$, orientable with any path ordering with the precedence $g\succ_{\mathcal{F}} s$. Since the term 0 is in normal form, **Stop** also applies to the state 1.2.

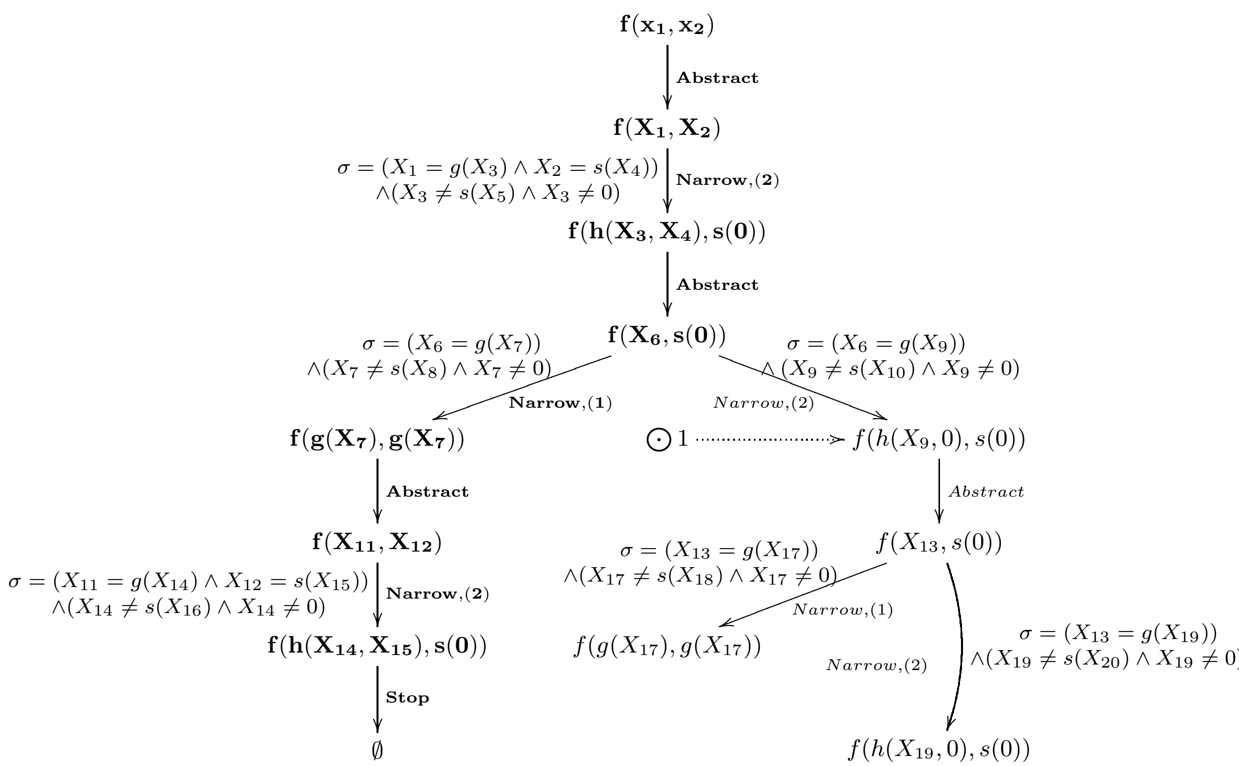
Applying the inference rules to $(h(x_1, x_2), \top, \top)$, we get :

ϵ	$h(x_1, x_2)$	$A = \top$	$C = \top$	
Abstract				
1	$h(X_1, X_2)$	$A = (x_1\downarrow = X_1 \wedge x_2\downarrow = X_2)$	$C = (f(x_1, x_2) > x_1, x_2)$	
Narrow				
1.1	$g(X_1)$	$A = (x_1\downarrow = X_1 \wedge x_2\downarrow = X_2)$	$C = (f(x_1, x_2) > x_1, x_2)$	$\sigma = Id$
Stop				
1.1.1	$\emptyset$	$A = (x_1\downarrow = X_1 \wedge x_2\downarrow = X_2)$	$C = (f(x_1, x_2) > x_1, x_2)$	

Stop applies to the state 1.1 thanks to the $WEAK\!-\!TERMIN$ predicate and the usable rules. Indeed, the usable rules of $g(X_1)$ consist of the system $\{g(s(x)) \rightarrow s(g(x)), g(0) \rightarrow 0\}$, orientable with any path ordering with the precedence $g \succ_{\mathcal{F}} s$.

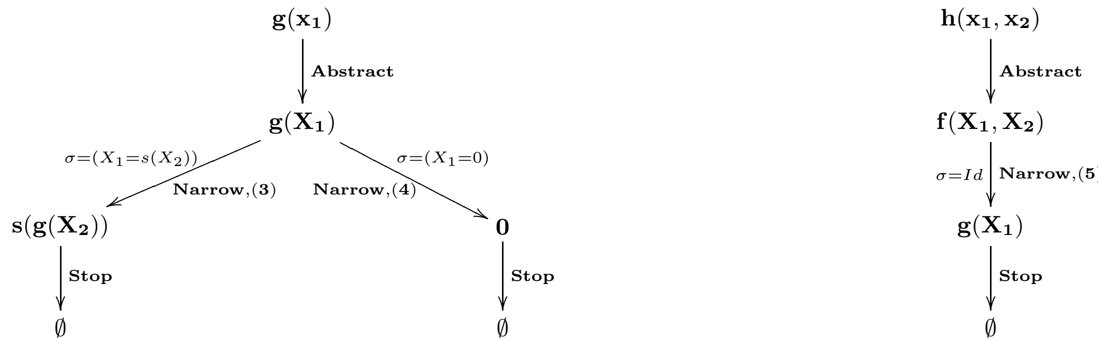
Note that any simplification ordering holds for satisfying all ordering constraints.

The proof tree obtained by applying the inference rules to $f(x_1, x_2)$ is the schematized as follows.



The subtree $\odot$ 1 is cut as soon as the subtree generated on the left from $\mathbf{f(X_6, s(0))}$ with the same substitution (up to a renaming) $\sigma = (X_6 = g(X_7)) \wedge (X_7 \neq s(X_8) \wedge X_7 \neq 0)$ is successful.

The proof trees obtained by applying the inference rules to $g(x_1)$ and $h(x_1, x_2)$ are schematized as follows.



The final proof trees are **bold**. Since they are all successful, $\mathcal{R}$ is proved weakly innermost terminating on the ground term algebra.

We can now infer from these trees a strategy normalizing any ground term t, according to Definition 10. As an example, let us use the strategy to normalize the term $f(g(f(0,0)),s(0))$ following the steps of ST_f. To improve readibility when jumping from a proof tree to an other one, we mark some steps by capital letters.

$\odot\, A$ **(Step 1 in $\mathbf{ST_f}$: Abstract)** The first step is **Abstract** at positions 1 and 2 by application of the induction hypothesis, and then we get $f(g(f(0,0)),s(0)) \mapsto f(norm_{ST}(g(f(0,0))), norm_{ST}(s(0))$. Since s is a constructor, we have $norm_{ST}(s(0)) = s(norm_{ST}(0))$. Since 0 is a constructor constant, we have $norm_{ST}(0) = 0$, and finally $norm_{ST}(s(0)) = s(0)$. We now have to compute $norm_{ST}(g(f(0,0)))$, by following the steps of ST_g.

$\odot\, B$ **(Step 1 in $\mathbf{ST_g}$: Abstract)** The first step is **Abstract** at position 1 by application of the induction hypothesis, and then we get $g(f(0,0)) \mapsto g(norm_{ST}(f(0,0)))$. To compute $norm_{ST}(f(0,0))$, we have to follow the steps of ST_f.

$\odot\, C$ **(Step 1 in $\mathbf{ST_f}$: Abstract)** The first step is **Abstract** at positions 1 and 2 by application of the induction hypothesis, and then we get $f(0,0) \mapsto f(norm_{ST}(0), norm_{ST}(0))$. Since 0 is a constant constructor, we have $norm_{ST}(0) = 0$, and then $f(0,0) \mapsto f(0,0)$.

(Step 2 in $\mathbf{ST_f}$: Narrow) The second step is **Narrow** at the top position, with rule (2). The narrowing substitution σ is such that our current term $f(0,0)$ is not a ground instance of $\sigma f(X_1,X_2)$. Therefore $f(0,0) \mapsto f(0,0)$, and finally $norm_{ST}(f(0,0)) = f(0,0)$.

$\odot\, B$ (continued) **(Step 2 in $\mathbf{ST_g}$: Narrow)** Our current term is $g(f(0,0))$, and the second step of ST_g is **Narrow** at the top position, with rules (3) and (4). None of the narrowing substitutions σ is such that our current term $g(f(0,0))$ is a ground instance of $\sigma g(X_1)$. Therefore $g(f(0,0)) \mapsto g(f(0,0))$, and finally $norm_{ST}(g(f(0,0))) = g(f(0,0))$.

$\odot\, A$ (continued) **(Step 2 in $\mathbf{ST_f}$: Narrow)** Our current term is $f(g(f(0,0)),s(0))$, and the current step in $\mathbf{ST_f}$ is **Narrow** at the top position with rule (2). The narrowing substitution σ is such that our current term is a ground instance of $\sigma f(X_1,X_2)$. So $f(g(f(0,0)),s(0)) \rightarrow^{\epsilon,(2)} f(h(f(0,0),0),s(0))$.

(Step 3 in $\mathbf{ST_f}$: Abstract) The current step in the proof tree is **Abstract** at position 1 thanks to the $WEAK\!-\!TERMIN$ predicate, and more precisely thanks to the usable rules which give a strong terminating system. Then we have $h(f(0,0),0) \mapsto h(f(0,0),0)\!\downarrow$, and it suffices to rewrite $h(f(0,0),0)$ as long as a normal form is reached, which is guaranteed by the termination of the usable rules. Here we have $h(f(0,0),0) \rightarrow^{\epsilon,(5)} g(f(0,0))$. Finally we get $f(h(f(0,0),0),s(0)) \mapsto f(g(f(0,0)),s(0))$.

(Step 4 in $\mathbf{ST_f}$: Narrow) The current step in the tree is **Narrow** at the top position with rule (1). The narrowing substitution σ is such that our current term is a ground instance of $\sigma f(X_6,s(0))$. So $f(g(f(0,0)),s(0)) \rightarrow^{\epsilon,(1)} f(g(f(0,0)),g(f(0,0)))$.

(Step 5 in $\mathbf{ST_f}$: Abstract) The current step in the tree is **Abstract** at positions 1 and 2 thanks to the $WEAK\!-\!TERMIN$ predicate, and then $f(g(f(0,0)),g(f(0,0))) \mapsto f(g(f(0,0))\!\downarrow,$

$g(f(0,0))\downarrow)$. Since $g(f(0,0))$ is in normal form, we get $f(g(f(0,0)), g(f(0,0))) \mapsto f(g(f(0,0)), g(f(0,0)))$.

(Step 6 in $\mathbf{ST_f}$: Narrow) The current step of ST_f is **Narrow** at the top position, with rule (2). The narrowing substitution σ is such that our current term is a not a ground instance of $\sigma f(X_{11}, X_{12})$. Therefore the normalizing process stops on $f(g(f(0,0)), g(f(0,0)))$, which hence is a normal form of $f(g(f(0,0)), s(0))$.

7 Mechanizing the cut process

To manage the cut process in the proof tree, the structure on which the inference rules applies has to be modified. Our inference rules will not apply on states $(\{u\}, A, C)$ anymore but on the proof tree, whose nodes will be states of the form $s = (\{u\}, A, C, \sigma)$, where σ is the substitution used in the last narrowing step performed to obtain the state s in the proof process. These rules are given in Table 2.

We will use the following operator on trees:

$$T \sqsupset_p s : \text{ the tree } T \text{ containing the state } s \text{ at position } p,$$

and the operation:

$$T \sqsupset_p (s \wedge_{child} s') : \text{ the state } s' \text{ is added as a child of } s \text{ in the tree } T.$$

The strategy $S\!-\!CUT$ to apply the rules of Table 2 is:

$$(\ \textbf{Abstract} \ ; \text{dk}(\textbf{Narrow}) \ ; \textbf{Stop} \ ; ((\textbf{Shorten})^* \ ; (\textbf{Cut})^*)^* \ )^*.$$

where, as previously, at each branching point of a proof tree, with set of substitutions Σ, we only develop the subtrees corresponding to Σ_0.

We write $SUCCESS\!-\!CUT(g, \succ)$ if the proof tree obtained by application, with the strategy $S\!-\!CUT$ on $(\{g(x_1, \ldots, x_m)\}, \top, \top, \lambda)$, of the inference rules of Table 2 whose conditions are satisfied by an ordering $\succ$, is reduced to a state of the form $(\emptyset, \top, \top, \lambda)$.

With these new inference rules, Definition 10 and Theorem 2 still hold but the definition of a strategy tree becomes:

Definition 11 *Let $\mathcal{R}$ be a TRS proved weakly terminating with the strategy $S\!-\!CUT$. The strategy tree ST_f associated to $f \in \mathcal{D}ef_R$ is the tree obtained by taking all branches built during the generation of the proof tree of f, by using only the termination proving rules, and ending with a state $(\emptyset, A, C)$.*

Theorem 3 *Let $\mathcal{R}$ be a TRS on a set $\mathcal{F}$ of symbols. If there exists an $\mathcal{F}$-stable ordering $\succ$ having the subterm property, such that for each defined symbol g, we have $SUCCESS\!-\!CUT(g, \succ)$, then any term of $\mathcal{T}(\mathcal{F})$ weakly innermost terminates.*

Let us now make precise the full development of the example presented in the introduction of the paper, with detailed applications of the inference rules. For readability, we do not write the development of branches that are going to be cut.

Let us apply the inference rules on $f(x_1)$.

ϵ	$f(x_1)$	$A = \top$	$C = \top$	$\sigma = \lambda$
Abstract				
1	$f(X_1)$	$A = (x_1\downarrow = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = \lambda$
Narrow				
1.1	$p(s(X_1))$	$A = (x_1\downarrow = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.2	$p(s(s(X_1)))$	$A = (x_1\downarrow = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$

In the following, we show that the state 1.1 is transformed into a successful state, which allows to cut the branch 1.2. For readability, we do not show the development of the latter branch here.

Table 2: Inference rules for weak innermost termination with the cut process

Termination proving rules:

Abstract: $$\frac{T \sqsupset_l s = (\{u\}, \;\; A, \;\; C, \;\; \sigma)}{T \sqsupset_l [s \wedge_{child} (\{u'\}, \;\; A \wedge \bigwedge_{j=i_1}^{i_p} u|_j{\downarrow} = X_j, \; C \wedge \bigwedge_{j=i_1}^{i_p} H_C(u|_j), \;\; \sigma)]}$$

where u is abstracted into u' at the positions $i_1, \ldots, i_p$

if $\bigwedge_{j=i_1}^{i_p} H_C(u|_j)$ is satisfiable

Narrow: $$\frac{T \sqsupset_l (\{u\}, \;\; A, \;\; C, \;\; \mu)}{T \sqsupset_l [(\{u\}, \;\; A, \;\; C, \;\; \mu) \wedge_{child} (\{v\}, \;\; \sigma A, \;\; C, \;\; \sigma)]}$$ if $u \rightsquigarrow_{\sigma}^{Inn} v$.

Stop: $$\frac{T \sqsupset_l (\{u\}, \;\; A, \;\; C, \;\; \sigma)}{T \sqsupset_l (\emptyset, \;\; A, \;\; C \wedge H_C(u), \;\; \sigma)}$$ if $C \wedge H_C(u)$ is satisfiable or A is unsatisfiable

Redundancy suppressing rules:

Shorten: $$\frac{T \sqsupset_p s = (\{u\}, \;\; A, \;\; C, \; \sigma)}{T \sqsupset_p (\emptyset, \;\; A, \;\; C, \; \sigma)}$$ if every child i of s in T is of the form $(\emptyset, A_i, C_i, \mu_i)$

Cut: $$\frac{[T \sqsupset_{p.k} (\emptyset, \;\; A, \;\; C, \; \sigma)] \sqsupset_{p.m} (\{u'\}, \;\; A', \;\; C, \; \mu)}{[T \sqsupset_{p.k} (\emptyset, \;\; A, \;\; C, \; \sigma)] \sqsupset_{p.m} (\emptyset, \;\; A', \;\; C, \; \mu)}$$ if μ is an instance of σ

where $H_C(u) = \begin{cases} true & \text{if } WEAK{-}TERMIN(u) \text{ or } \mathbf{A \text{ is unsatisfiable}} \\ t_{ref} > u & \text{otherwise.} \end{cases}$

In the first five rules, l is a leaf position in T.

Narrow

1.1.1	$p(X_2)$	$A = (x_1{\downarrow} = s(X_2))$	$C = (f(x_1) > x_1)$	$\sigma = (X_1 = s(X_2))$
1.1.2	$f(0)$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = (X_1 = 0)$
1.2	$p(s(s(X_1)))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$

Stop

1.1.1	$\emptyset$	$A = (x_1{\downarrow} = s(X_2))$	$C = (f(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(X_2))$
1.1.2	$f(0)$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = (X_1 = 0)$
1.2	$p(s(s(X_1)))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$

Narrow

1.1.1	$\emptyset$	$A = (x_1{\downarrow} = s(X_2))$	$C = (f(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(X_2))$
1.1.2.1	$p(s(0))$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.1.2.2	$p(s(s(0)))$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.2	$p(s(s(X_1)))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$

Stop applies on $p(X_2)$ with any simplification ordering with the precedence $f \succ_{\mathcal{F}} p$. Indeed, assuming that any term is greater than its normal form, we get from $A = (x_1{\downarrow} = s(X_2))$ that $x_1 \succeq s(X_2) \succ X_2$, and hence $f(x_1) \succ p(X_2)$ with the latter simplification ordering.

In the following, we show that the state 1.1.2.2 is transformed into a successful state, which allows to cut the branch 1.1.2.1. For readability, we do not show the development of this branch here.

Narrow

1.1.1	$\emptyset$	$A = (x_1{\downarrow} = s(X_2))$	$C = (f(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(X_2))$
1.1.2.1	$p(s(0))$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.1.2.2.1	$p(0)$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.2	$p(s(s(X_1)))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$

Narrow

1.1.1	$\emptyset$	$A = (x_1{\downarrow} = s(X_2))$	$C = (f(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(X_2))$
1.1.2.1	$p(s(0))$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.1.2.2.1.1	0	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.2	$p(s(s(X_1)))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$

Stop

1.1.1	$\emptyset$	$A = (x_1{\downarrow} = s(X_2))$	$C = (f(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(X_2))$
1.1.2.1	$p(s(0))$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.1.2.2.1.1	$\emptyset$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.2	$p(s(s(X_1)))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$

Shorten

1.1.1	$\emptyset$	$A = (x_1{\downarrow} = s(X_2))$	$C = (f(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(X_2))$
1.1.2.1	$p(s(0))$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.1.2.2.1	$\emptyset$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.2	$p(s(s(X_1)))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$

Shorten

1.1.1	$\emptyset$	$A = (x_1{\downarrow} = s(X_2))$	$C = (f(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(X_2))$
1.1.2.1	$p(s(0))$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.1.2.2	$\emptyset$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.2	$p(s(s(X_1)))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$

Cut

1.1.1	$\emptyset$	$A = (x_1{\downarrow} = s(X_2))$	$C = (f(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(X_2))$
1.1.2.1	$\emptyset$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.1.2.2	$\emptyset$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.2	$p(s(s(X_1)))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$

Shorten				
1.1.1	$\emptyset$	$A = (x_1{\downarrow} = s(X_2))$	$C = (f(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(X_2))$
1.1.2	$\emptyset$	$A = (x_1{\downarrow} = 0)$	$C = (f(x_1) > x_1)$	$\sigma = (X_1 = 0)$
1.2	$p(s(s(X_1)))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
Shorten				
1.1	$\emptyset$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.2	$p(s(s(X_1)))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$

Cut				
1.1	$\emptyset$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.2	$\emptyset$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
Shorten				
1	$\emptyset$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = \lambda$
Shorten				
ϵ	$\emptyset$	$A = \top$	$C = \top$	$\sigma = \lambda$

Let us now apply the inference rules on $p(x_1)$.

ϵ	$p(x_1)$	$A = \top$	$C = \top$	$\sigma = \lambda$
Abstract				
1	$p(X_1)$	$A = (x_1{\downarrow} = X_1)$	$C = (p(x_1) > x_1)$	$\sigma = \lambda$
Narrow				
1.1	0	$A = (x_1{\downarrow} = 0)$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = 0)$
1.2	$f(0)$	$A = (x_1{\downarrow} = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = s(0))$
1.3	$p(X_2)$	$A = (x_1{\downarrow} = s(s(X_2)))$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = s(s(X_2))$
Stop	(*on two branches*)			
1.1	$\emptyset$	$A = (x_1{\downarrow} = 0)$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = 0)$
1.2	$f(0)$	$A = (x_1{\downarrow} = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = s(0))$
1.3	$\emptyset$	$A = (x_1{\downarrow} = s(s(X_2)))$	$C = (p(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(s(X_2))$
Narrow				
1.1	$\emptyset$	$A = (x_1{\downarrow} = 0)$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = 0)$
1.2.1	$p(s(0))$	$A = (x_1{\downarrow} = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = Id$
1.2.2	$p(s(s(0)))$	$A = (x_1{\downarrow} = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = Id$
1.3	$\emptyset$	$A = (x_1{\downarrow} = s(s(X_2)))$	$C = (p(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(s(X_2))$

In the following, we show that the state 1.2.2 is transformed into a successful state, which allows to cut the branch 1.2.1, which we hence do not develop here, for readability.

Narrow				
1.1	$\emptyset$	$A = (x_1{\downarrow} = 0)$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = 0)$
1.2.1	$p(s(0))$	$A = (x_1{\downarrow} = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = Id$
1.2.2.1	$p(0)$	$A = (x_1{\downarrow} = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = Id$
1.3	$\emptyset$	$A = (x_1{\downarrow} = s(s(X_2)))$	$C = (p(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(s(X_2))$
Narrow				
1.1	$\emptyset$	$A = (x_1{\downarrow} = 0)$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = 0)$
1.2.1	$p(s(0))$	$A = (x_1{\downarrow} = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = Id$
1.2.2.1.1	0	$A = (x_1{\downarrow} = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = Id$
1.3	$\emptyset$	$A = (x_1{\downarrow} = s(s(X_2)))$	$C = (p(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(s(X_2))$
Stop				
1.1	$\emptyset$	$A = (x_1{\downarrow} = 0)$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = 0)$
1.2.1	$p(s(0))$	$A = (x_1{\downarrow} = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = Id$
1.2.2.1.1	$\emptyset$	$A = (x_1{\downarrow} = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = Id$
1.3	$\emptyset$	$A = (x_1{\downarrow} = s(s(X_2)))$	$C = (p(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(s(X_2))$

Shorten				
1.1.1	$\emptyset$	$A = (x_1\!\downarrow = s(X_2))$	$C = (f(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(X_2))$
1.1.2	$\emptyset$	$A = (x_1\!\downarrow = 0)$	$C = (f(x_1) > x_1)$	$\sigma = (X_1 = 0)$
1.2	$p(s(s(X_1)))$	$A = (x_1\!\downarrow = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
Shorten				
1.1	$\emptyset$	$A = (x_1\!\downarrow = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.2	$p(s(s(X_1)))$	$A = (x_1\!\downarrow = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
Cut				
1.1	$\emptyset$	$A = (x_1\!\downarrow = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
1.2	$\emptyset$	$A = (x_1\!\downarrow = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
Shorten				
1	$\emptyset$	$A = (x_1\!\downarrow = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = \lambda$
Shorten				
ϵ	$\emptyset$	$A = \top$	$C = \top$	$\sigma = \lambda$

Let us now apply the inference rules on $p(x_1)$.

ϵ	$p(x_1)$	$A = \top$	$C = \top$	$\sigma = \lambda$
Abstract				
1	$p(X_1)$	$A = (x_1\!\downarrow = X_1)$	$C = (p(x_1) > x_1)$	$\sigma = \lambda$
Narrow				
1.1	0	$A = (x_1\!\downarrow = 0)$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = 0)$
1.2	$f(0)$	$A = (x_1\!\downarrow = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = s(0))$
1.3	$p(X_2)$	$A = (x_1\!\downarrow = s(s(X_2)))$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = s(s(X_2))$
Stop	(*on two branches*)			
1.1	$\emptyset$	$A = (x_1\!\downarrow = 0)$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = 0)$
1.2	$f(0)$	$A = (x_1\!\downarrow = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = s(0))$
1.3	$\emptyset$	$A = (x_1\!\downarrow = s(s(X_2)))$	$C = (p(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(s(X_2))$
Narrow				
1.1	$\emptyset$	$A = (x_1\!\downarrow = 0)$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = 0)$
1.2.1	$p(s(0))$	$A = (x_1\!\downarrow = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = Id$
1.2.2	$p(s(s(0)))$	$A = (x_1\!\downarrow = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = Id$
1.3	$\emptyset$	$A = (x_1\!\downarrow = s(s(X_2)))$	$C = (p(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(s(X_2))$
Shorten				
1.1	$\emptyset$	$A = (x_1\!\downarrow = 0)$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = 0)$
1.2.1	$p(s(0))$	$A = (x_1\!\downarrow = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = Id$
1.2.2.1	$\emptyset$	$A = (x_1\!\downarrow = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = Id$
1.3	$\emptyset$	$A = (x_1\!\downarrow = s(s(X_2)))$	$C = (p(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(s(X_2))$
Shorten				
1.1	$\emptyset$	$A = (x_1\!\downarrow = 0)$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = 0)$
1.2.1	$p(s(0))$	$A = (x_1\!\downarrow = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = Id$
1.2.2	$\emptyset$	$A = (x_1\!\downarrow = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = Id$
1.3	$\emptyset$	$A = (x_1\!\downarrow = s(s(X_2)))$	$C = (p(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(s(X_2))$
Cut				
1.1	$\emptyset$	$A = (x_1\!\downarrow = 0)$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = 0)$
1.2.1	$\emptyset$	$A = (x_1\!\downarrow = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = Id$
1.2.2	$\emptyset$	$A = (x_1\!\downarrow = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = Id$
1.3	$\emptyset$	$A = (x_1\!\downarrow = s(s(X_2)))$	$C = (p(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(s(X_2))$
Shorten				
1.1	$\emptyset$	$A = (x_1\!\downarrow = 0)$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = 0)$
1.2	$\emptyset$	$A = (x_1\!\downarrow = s(0))$	$C = (p(x_1) > x_1)$	$\sigma = (X_1 = s(0))$
1.3	$\emptyset$	$A = (x_1\!\downarrow = s(s(X_2)))$	$C = (p(x_1) > x_1, p(X_2))$	$\sigma = (X_1 = s(s(X_2))$
Shorten				
1	$\emptyset$	$A = (x_1\!\downarrow = X_1)$	$C = (p(x_1) > x_1)$	$\sigma = \lambda$
Shorten				
ϵ	$\emptyset$	$A = \top$	$C = \top$	$\sigma = \lambda$

8 Conclusion and perspectives

In this paper, we have proposed a method to prove weak innermost termination of term rewriting systems by explicit induction on the termination property. Our method works on the ground term algebra using as induction relation an $\mathcal{F}$-stable ordering having the subterm property. The general proof principle relies on the simple idea that for establishing weak innermost termination of a ground term t, it is enough to suppose that subterms of t are smaller than t for this ordering, and that rewriting the context leads to at least one terminating chain. Iterating this process until a non-reducible context is obtained establishes weak innermost termination of t. Up to our knowledge, this is the first method proposed to ensure weak termination of rewriting systems that are not strongly innermost terminating.

From the proof of weak termination of a given TRS, we can extract for any given ground term, a rewriting strategy to compute one of its normal form, even if the ground term admits infinite rewriting derivations. This leads to the idea of a preprocessor of TRSs, at compile time, that would build the proof trees for each defined symbol of the signature. Then at evaluation time, the execution would be guided as described in Section 5 to obtain a normal form of any given term. This can then be very useful in the context of rule-based programming, to choose an efficient strategy for undeterministic and weakly terminating programs.

The important point to automate our proof principle is the satisfaction of the constraints at each step of the proof. On many examples, this is immediate: as the ordering constraints only express the subterm property, they are trivially satisfied by any simplification ordering. Otherwise, we can use automatic ordering constraint solvers. As for abstraction constraints, they can be managed with an unsatisfiability test, for which simple sufficient conditions exist, that are automated. Thus, in general, weak termination proof can be completely automatic. We are now studying the implementation of our technique in CARIBOO, a toolbox for proving termination under strategies [8].

As in our approach, the rewriting strategy is explicitly handled in the proof principle, the method should be applicable easily to other strategies, especially to outermost strategy, and to local strategies on operators.

References

[1] T. Arts and J. Giesl. Proving innermost normalisation automatically. In *Proceedings 8th Conference on Rewriting Techniques and Applications, Sitges (Spain)*, volume 1232 of *Lecture Notes in Computer Science*, pages 157–171. Springer-Verlag, 1997.

[2] Peter Borovanský, Claude Kirchner, Hélène Kirchner, Pierre-Etienne Moreau, and Christophe Ringeissen. An overview of ELAN. In Claude Kirchner and Hélène Kirchner, editors, *Proceedings of the second International Workshop on Rewriting Logic and Applications*, volume 15, `http://www.elsevier.nl/locate/entcs/volume15.html`, Pont-à-Mousson (France), September 1998. Electronic Notes in Theoretical Computer Science. Report LORIA 98-R-316.

[3] Manuel Clavel, Francisco Durán, Steven Eker, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, and José F. Quesada. Maude: Specification and programming in rewriting logic. *Theoretical Computer Science*, 285:187–243, 2002.

[4] H. Comon. Disunification: a survey. In Jean-Louis Lassez and G. Plotkin, editors, *Computational Logic. Essays in honor of Alan Robinson*, chapter 9, pages 322–359. The MIT press, Cambridge (MA, USA), 1991.

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

[6] Nachum Dershowitz and Jean-Pierre Jouannaud. *Handbook of Theoretical Computer Science*, volume B, chapter 6: Rewrite Systems, pages 244–320. Elsevier Science Publishers B. V. (North-Holland), 1990. Also as: Research report 478, LRI.

[7] O. Fissore, I. Gnaedig, and H. Kirchner. Termination of rewriting with local strategies. In M. P. Bonacina and B. Gramlich, editors, *Selected papers of the 4th International Workshop on Strategies in Automated Deduction*, volume 58 of *Electronic Notes in Theoretical Computer Science*. Elsevier Science Publishers B. V. (North-Holland), 2001.

[8] O. Fissore, I. Gnaedig, and H. Kirchner. CARIBOO : An induction based proof tool for termination with strategies. In *Proceedings of the Fourth International Conference on Principles and Practice of Declarative Programming*, pages 62–73, Pittsburgh (USA), October 2002. ACM Press.

[9] O. Fissore, I. Gnaedig, and H. Kirchner. Outermost ground termination. In *Proceedings of the Fourth International Workshop on Rewriting Logic and Its Applications*, volume 71 of *Electronic Notes in Theoretical Computer Science*, Pisa, Italy, September 2002. Elsevier Science Publishers B. V. (North-Holland).

[10] 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.

[11] I. Gnaedig, H. Kirchner, and O. Fissore. Induction for innermost and outermost ground termination. Technical Report A01-R-178, LORIA, Nancy (France), September 2001.

[12] Goubault-Larreck. Well-founded recursive relations. In *Proc. 15th Int. Workshop Computer Science Logic (CSL'2001)*, volume 2142 of *Lecture Notes in Computer Science*, Paris, 2001. Springer-Verlag.

[13] J. Goubault-Larrecq. A proof of weak termination of typed lambda-sigma-calculi. In *Proceedings of the TYPES'96 Workshop*, volume 1512 of *Lecture Notes in Computer Science*, Aussois (France), 1998. Springer-Verlag.

[14] Bernhard Gramlich. Relating innermost, weak, uniform and modular termination of term rewriting systems. In Andrei Voronkov, editor, *Proceedings of the 3rd International Conference on Logic Programming and Automated Reasoning (LPAR'92)*, volume 624 of *Lecture Notes in Computer Science*, pages 285–296, St. Petersburg, Russia, July 1992. Springer-Verlag.

[15] Bernhard Gramlich. On termination and confluence properties of disjoint and constructor-sharing conditional rewrite systems. *Theoretical Computer Science*, 165(1):97–131, September 1996.

[16] 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.

[17] P. Klint. A meta-environment for generating programming environments. *ACM Transactions on Software Engineering and Methodology*, 2:176–201, 1993.

[18] S. Lucas. Termination of OBJ programs with positive local strategies. In M. van der Brand and R. Verma, editors, *Proc. of 2nd International Workshop on Rule-Based Programming, RULE'01*, pages 64–78, Firenze, Italy, September 2001.

[19] A. Middeldorp and E. Hamoen. Completeness results for basic narrowing. *Applicable Algebra in Engineering, Communication and Computation*, 5(3 & 4):213–253, 1994.

[20] C. G. Nelson and D. C. Oppen. Fast decision procedures based on congruence closure. *Journal of the ACM*, 27(2):356–364, 1980.

Appendix

A Proof of Theorem 1

Theorem 1 *Let $\mathcal{R}$ be a TRS on a set $\mathcal{F}$ of symbols. If there exists an $\mathcal{F}$-stable ordering $\succ$ having the subterm property, such that for each defined symbol g, we have $SUCCESS(g, \succ)$, then any term of $\mathcal{T}(\mathcal{F})$ weakly innermost terminates.*

In Theorem 1, we use the relation between innermost rewriting and innermost narrowing, expressed in the following lifting lemma, whose proof needs the following two propositions (the first one is obvious).

Proposition 1 *Let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and σ a substitution of $\mathcal{T}(\mathcal{F}, \mathcal{X})$. Then $Var(\sigma t) = (Var(t) - Dom(\sigma)) \cup Ran(\sigma_{Var(t)})$.*

Proposition 2 *Suppose we have substitutions σ, μ, ν and sets A, B of variables such that $(B - Dom(\sigma)) \cup Ran(\sigma) \subseteq A$. If $\mu = \nu[A]$ then $\mu\sigma = \nu\sigma[B]$.*

Proof. Let us consider $(\mu\sigma)_B$, which can be divided as follows: $(\mu\sigma)_B = (\mu\sigma)_{B \cap Dom(\sigma)} \cup (\mu\sigma)_{B-Dom(\sigma)}$.
For $x \in B \cap Dom(\sigma)$, we have $\mathcal{V}ar(\sigma x) \subseteq Ran(\sigma)$, and then $(\mu\sigma)x = \mu(\sigma x) = \mu_{Ran(\sigma)}(\sigma x) = (\mu_{Ran(\sigma)}\sigma)x$. Therefore $(\mu\sigma)_{B \cap Dom(\sigma)} = (\mu_{Ran(\sigma)}\sigma)_{B \cap Dom(\sigma)}$.
For $x \in B - Dom(\sigma)$, we have $\sigma x = x$, and then $(\mu\sigma)x = \mu(\sigma x) = \mu x$. Therefore we have $(\mu\sigma)_{B-Dom(\sigma)} = \mu_{B-Dom(\sigma)}$. Henceforth we get $(\mu\sigma)_B = (\mu_{Ran(\sigma)}\sigma)_{B \cap Dom(\sigma)} \cup \mu_{B-Dom(\sigma)}$.
By a similar reasoning, we get $(\nu\sigma)_B = (\nu_{Ran(\sigma)}\sigma)_{B \cap Dom(\sigma)} \cup \nu_{B-Dom(\sigma)}$.
By hypothesis, we have $Ran(\sigma) \subseteq A$ and $\mu = \nu[A]$. Then we can infer $\mu_{Ran(\sigma)} = \nu_{Ran(\sigma)}$. Likewise, since $B - Dom(\sigma) \subseteq A$, we have $\mu_{B-Dom(\sigma)} = \nu_{B-Dom(\sigma)}$.
Then we have $(\mu\sigma)_B = (\mu_{Ran(\sigma)}\sigma)_{B \cap Dom(\sigma)} \cup \mu_{B-Dom(\sigma)} = (\nu_{Ran(\sigma)}\sigma)_{B \cap Dom(\sigma)} \cup \nu_{B-Dom(\sigma)} = (\nu\sigma)_B$ with the assumptions used in the second equality. Therefore $(\mu\sigma) = (\nu\sigma)[B]$.

Lemma 1 *Let $\mathcal{R}$ be a TRS. Let $s \in \mathcal{T}(\mathcal{F}, \mathcal{N})$, α a normalized ground substitution, and $\mathcal{Y} \subseteq \mathcal{N}$ a set of variables such that $Var(s) \cup Dom(\alpha) \subseteq \mathcal{Y}$. If $\alpha s \rightarrow^{Inn}_{p,l \to r} t'$, then there exist a term $s' \in \mathcal{T}(\mathcal{F}, \mathcal{N})$ and substitutions β, σ such that:*

1. $s \rightsquigarrow^{Inn}_{p,l \to r,\sigma} s'$,
2. $\beta s' = t'$,
3. $\beta\sigma = \alpha[\mathcal{Y}]$,
4. β *is normalized.*

Proof. The proof is an adaptation from the proof of the classical lifting lemma in [19].

To show the point 1., we will need to fulfill the conditions of the innermost narrowing definition, given in Definition 5. In the following, we assume that $\mathcal{V}ar(\mathcal{Y}) \cap \mathcal{V}ar(l) = \emptyset$ for every $l \to r \in \mathcal{R}$. If $\alpha s \rightarrow^{Inn}_{p,l \to r} t'$, then there exists a substitution τ such that $Dom(\tau) \subseteq \mathcal{V}ar(l)$ and $(\alpha s)|_p = \tau l$. Moreover, since α is normalized, p is a non variable position of s and we have $(\alpha s)|_p = \alpha(s|_p)$. Denoting $\mu = \alpha \wedge \tau$, we have:

$$\begin{aligned} \mu(s|_p) &= \alpha(s|_p) && \text{for } Dom(\tau) \subseteq \mathcal{V}ar(l) \text{ and } \mathcal{V}ar(l) \cap \mathcal{V}ar(s) = \emptyset \\ &= \tau l && \text{by definition of } \tau \\ &= \mu l && \text{for } Dom(\alpha) \subseteq \mathcal{Y} \text{ and } \mathcal{Y} \cap \mathcal{V}ar(l) = \emptyset, \end{aligned}$$

and therefore $s|_p$ and l are unifiable. Let us note σ_0 the most general unifier of $s|_p$ and l, and $s' = \sigma_0(s[r]_p)$. The conditions 1. and 2. of Definition 5 are verified, and we are now going to tackle the last point.

Let us suppose that there exist a rule $l' \to r' \in \mathcal{R}$, a position p' strict suffix of p and a substitution σ_i such that $\sigma_i(\sigma_0(s|_{p'})) = \sigma_i l'$. This means that $\sigma_i(\sigma_0(s|_{p'}))$ is reducible, and then, since σ_0 is more general than μ and $\mu = \alpha[\mathcal{V}ar(s)]$, that $\sigma_i(\alpha(s|_{p'}))$ is reducible.
Moreover, since α is a ground instantiation, we have $\sigma_i\alpha(s|_{p'}) = \alpha(s|_{p'})$. Then we get that $\alpha s|_{p'}$ is reducible, which is impossible by definition of innermost rewriting since αs innermost rewrites at position p. Therefore σ_0 satisfies $\bigwedge_{i\in[1..k]} \overline{\sigma_i}$ for all most general unifiers σ_i of s and a left-hand side of rule of $\mathcal{R}$ at strict suffix positions of p.

Therefore, denoting $\sigma = \sigma_0 \wedge \bigwedge_{i\in[1..k]} \overline{\sigma_i}$, we get, by definition: $s \rightsquigarrow^{Inn}_{[p,l\to r,\sigma]} s'$, and then the point 1. of the current lemma holds.

Since σ is more general than μ, there exists a substitution ρ such that $\rho\sigma = \mu$. Let $\mathcal{Y}_1 = (\mathcal{Y} - Dom(\sigma)) \cup Ran(\sigma)$. We define $\beta = \rho_{\mathcal{Y}_1}$. Clearly $Dom(\beta) \subseteq \mathcal{Y}_1$.
We now show that $\mathcal{V}ar(s') \subseteq \mathcal{Y}_1$, by the following reasoning:

- since $s' = \sigma(s[r]_p)$, we have $\mathcal{V}ar(s') = \mathcal{V}ar(\sigma(s[r]_p))$;
- the rule $l \to r$ is such that $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$, therefore we have $\mathcal{V}ar(\sigma(s[r]_p)) \subseteq \mathcal{V}ar(\sigma(s[l]_p))$, and then, thanks to the previous point, $\mathcal{V}ar(s') \subseteq \mathcal{V}ar(\sigma(s[l]_p))$;
- since $\sigma(s[l]_p) = \sigma s[\sigma l]_p$ and since σ unifies l and $s|_p$, we get $\sigma(s[l]_p) = \sigma s[\sigma(s|_p)]_p = \sigma s[s|_p]_p = \sigma s$ and, thanks to the previous point: $\mathcal{V}ar(s') \subseteq \mathcal{V}ar(\sigma s)$;
- according to Proposition 1, we have $\mathcal{V}ar(\sigma(s)) = (\mathcal{V}ar(s) - Dom(\sigma)) \cup Ran(\sigma_{\mathcal{V}ar(s)})$; by hypothesis, $\mathcal{V}ar(s) \subseteq \mathcal{Y}$. Moreover, since $Ran(\sigma_{\mathcal{V}ar(s)}) \subseteq Ran(\sigma)$, we have $\mathcal{V}ar(\sigma(s)) \subseteq (\mathcal{Y} - Dom(\sigma)) \cup Ran(\sigma)$, that is $\mathcal{V}ar(\sigma s) \subseteq \mathcal{Y}_1$. Therefore, with the previous point, we get $Var(s') \subseteq \mathcal{Y}_1$.

From $Dom(\beta) \subseteq \mathcal{Y}_1$ and $Var(s') \subseteq \mathcal{Y}_1$, we infer $Dom(\beta) \cup Var(s') \subseteq \mathcal{Y}_1$.

We are now going to demonstrate the point 2., that is $\beta s' = t'$.
Since $\beta = \rho_{\mathcal{Y}_1}$, we have $\beta = \rho[\mathcal{Y}_1]$. Since $Var(s') \subseteq \mathcal{Y}_1$, we get $\beta s' = \rho s'$. Since $s' = \sigma(s[r]_p)$, we have $\rho s' = \rho\sigma(s[r]_p) = \mu(s[r]_p) = \mu s[\mu r]_p$.
We have $Dom(\tau) \subseteq \mathcal{V}ar(l)$ and $\mathcal{Y} \cap \mathcal{V}ar(l) = \emptyset$, then we have $\mathcal{Y} \cap Dom(\tau) = \emptyset$. Therefore, from $\mu = \alpha \cup \tau$, we get $\mu = \alpha[\mathcal{Y}]$. Since $\mathcal{V}ar(s) \subseteq \mathcal{Y}$, we get $\mu s = \alpha s$.
Likewise, by hypothesis we have $Dom(\alpha) \subseteq \mathcal{Y}$, $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$ and $\mathcal{Y} \cap \mathcal{V}ar(l) = \emptyset$, then we get $Var(r) \cap Dom(\alpha) = \emptyset$, and then we have $\mu = \tau[Var(r)]$, and therefore $\mu r = \tau r$.
From $\mu s = \alpha s$ and $\mu r = \tau r$ we get $\mu s[\mu r]_p = \alpha s[\tau r]_p$. Since, by hypothesis, $\alpha s \to_p t'$, with $\tau l = (\alpha s)|_p$, then $\alpha s[\tau r]_p = t'$. Finally, we get $\beta s' = t'$ (2).

Next we show that $\beta\sigma = \alpha[\mathcal{Y}]$ (point 3. of the current lemma). Reminding that $\mathcal{Y}_1 = (\mathcal{Y} - Dom(\sigma)) \cup Ran(\sigma)$, Proposition 2 (with the notations A for $\mathcal{Y}_1$, B for $\mathcal{Y}$, μ for β, ν for ρ and σ for σ) yields $\beta\sigma = \rho\sigma[\mathcal{Y}]$. We already noticed that $\mu = \alpha[\mathcal{Y}]$. Linking these two equalities via the equation $\rho\sigma = \mu$ yields $\beta\sigma = \alpha[\mathcal{Y}]$ (3).

Finally, since β is defined on $\mathcal{Y}_1 \subseteq \mathcal{N}$, β is necessarily normalized.

A.1 Proof of Theorem 1

Proof. We prove by induction on $\mathcal{T}(\mathcal{F})$ that any ground instance $\theta f(x_1, \ldots, x_m)$ of any term $f(x_1, \ldots, x_m) \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ weakly innermost terminates. The induction ordering is constrained along the proof. At the beginning, it has at least to be $\mathcal{F}$-stable and to have the subterm property, which ensures its noetherianity. Such an ordering always exists on $\mathcal{T}(\mathcal{F})$ (for instance the embedding relation). Let us denote it $\succ$.

By subterm property of $\succ$, we have $\theta f(x_1, \ldots, x_m) = f(\theta x_1, \ldots, \theta x_m) \succ \theta x_1, \ldots, \theta x_m$. Then, by induction hypothesis, let us suppose that the $\theta x_1, \ldots, \theta x_m$ weakly innermost terminate. Let $\theta x_1\!\downarrow, \ldots, \theta x_m\!\downarrow$ be any of their normal forms. It remains to prove that $f(\theta x_1\!\downarrow, \ldots, \theta x_m\!\downarrow)$ weakly innermost terminates.

If f is a constructor, then $f(\theta x_1\!\downarrow, \ldots, \theta x_m\!\downarrow)$ is irreducible for the innermost rewriting relation, and therefore, weakly innermost terminating.

If f is not a constructor, let us denote it g and prove that $g(\theta x_1, \dots, \theta x_m)$ weakly innermost terminates for any θ satisfying $(A_0 = \top, C_0 = \top)$, if application of the inference rules with the strategy S on $(\{g(x_1, \dots, x_m)\}, \top, \top)$, terminates on a weakly successful proof tree. Let us denote $g(x_1, \dots, x_m)$ by t_{ref} in the sequel of the proof.

To each step of the procedure characterized by a state $s = (\{t\}, A, C)$, we associate the set of ground terms $G = \{\alpha t \mid \alpha \text{ satisfies } A\}$, that is the set of ground instances represented by s. Inference rules **Abstract** and **Narrow** transform $(\{t\}, A, C)$ into $(\{t'\}, A', C')$ to which is associated $G' = \{\beta t' \mid \beta \text{ satisfies } A'\}$. We then prove the following result: if for all $\beta t' \in G'$, $\beta t'$ weakly innermost terminates, then any αt in G weakly innermost terminates. In the following, we take into account the cases where G or G' is empty, since the satisfiability of the ACF is not checked when applying inference rules.

- Either **Abstract** is applied, so the current term t becomes $t' = t[X_{i_1}]_{i_1} \dots [X_{i_p}]_{i_p}$, where $i_1, \dots, i_p$ are the abstraction positions. For each α such that $\alpha(t)$ is in G, we prove that there exists a β such that $\beta(t')$ is in G' and such that weak innermost termination of $\beta(t')$ implies weak innermost termination of $\alpha(t)$.

  If $G = \emptyset$, then we also have $G' = \emptyset$. Indeed, if A is unsatisfiable, then the constraint A', which is a conjunction of constraints containing A, is also unsatisfiable.

  Let us assume $G' \neq \emptyset$. According to the condition of application of **Abstract** at positions $i_1, \dots, i_j$, each term $t|_{i_j}$ is such that:

  - either $WEAK{-}TERMIN(t|_{i_j})$ is true, and then by definition of the predicate $WEAK{-}TERMIN$, $\alpha t|_{i_j}$ weakly innermost terminates;
  - or $C \wedge t_{ref} > t|_{i_j}$ is satisfiable and then, by induction hypothesis, $\alpha t|_{i_j}$ weakly innermost terminates.

  Then let us define $\beta = \alpha \cup \bigcup_{j \in \{1,\dots,p\}} X_{i_j} = \alpha t_{i_j}\downarrow$. Clearly β satisfies A', and $\beta t' = \alpha t[\alpha t|_{i_1}\downarrow]_{i_1} \dots [\alpha t|_{i_p}\downarrow]_{i_p}$. Therefore, $G' \neq \emptyset$ and weak innermost termination of all $\beta t'$, for all possible normal forms $\alpha t|_{i_1}\downarrow, \dots, \alpha t|_{i_p}\downarrow$ of $\alpha t|_{i_1}, \dots, \alpha t|_{i_p}$, implies weak innermost termination of αt.

- Or **Narrow** is applied on $(\{t = f(u_1, \dots, u_m)\}, A, C)$.

  If $G = \emptyset$, then we also have $G' = \emptyset$. Indeed, if the constraint A is unsatisfiable, then in particular $A' = \sigma A$ is unsatisfiable, for any substitution σ.

  Let us assume $G \neq \emptyset$. For any α satisfying A, either $\alpha f(u_1, \dots, u_m)$ is irreducible, and so weakly innermost terminates, or $\alpha f(u_1, \dots, u_m)$ is innermost reducible. Since **Narrow** is applied with the set Σ_0 of substitutions, that, by definition of Σ_0, covers all possible instances $\alpha f(u_1, \dots, u_m)$ of $f(u_1, \dots, u_m)$, for each given α and by Lemma 1, there is a narrowing step $f(u_1, \dots, u_m) \leadsto^{Inn}_{\sigma,p,l\to r} v$ corresponding to some rewriting step $\alpha f(u_1, \dots, u_m) \to^{inn}_{p,l\to r} t'$, such that there exists a substitution β such that $Dom(\beta) = \mathcal{V}ar(v)$, $\beta\sigma = \alpha$ and $t' = \beta v$. By definition of weak innermost termination, weak innermost termination of βv implies weak innermost termination of $\alpha f(u_1, \dots, u_m)$.

  On the other hand, on variables of $f(u_1, \dots, u_m)$, i.e. variables introduced by the previous application of **Abstract**, we have $\alpha = \beta\sigma$. In addition, the domain of β, that is $Var(v)$, can be extended to the variables of A by setting $\beta x = \alpha x$ for $x \in Var(A) \setminus Var(f(u_1, \dots, u_m))$. Thus, since α satisfies A, β satisfies σA.

Let us now prove that the ground instances satisfying A of each term t removed from the set $\mathcal{T} = \{t\}$ containing the current term of the state during the application of the rules, weakly innermost terminate.
The only rule removing terms from $\mathcal{T}$ is **Stop**. When **Stop** is applied and removes t from $\mathcal{T}$:

- either $WEAK{-}TERMIN(t)$ and then αt innermost terminates and hence weakly innermost terminates for any ground substitution α;

- or $C \wedge t_{ref} > t$ is satisfiable and then, by induction hypothesis, αt weakly innermost terminates;
- or A is unsatisfiable and then G is empty, hence the property is trivially true.

The strategy S also cuts subtrees whose root represents the same set of ground instances as the root of a successful subtree. As these successful subtrees modelize some possible terminating rewriting chains of the given sets of instances, then by definition of weak termination, it is sufficient to ensure the property for the given instances.

As the process is initialized with $\{t_{ref}\}$ and a constraint $A = \top$ satisfiable by any ground substitution, we get that $g(\theta x_1, \ldots, \theta x_m)$ is weakly innermost terminating, for any $t_{ref} = g(x_1, \ldots, x_m)$, and any ground instance θ.

Moreover, as the terms $f(x_1, \ldots, x_m)$, where f is a constructor, are also weakly innermost terminating for any ground instances $\theta x_1, \ldots, \theta x_m$, then any term of $\mathcal{T}(\mathcal{F})$ is weakly innermost terminating.

B Proof of Theorem 2

We proceed by crossed proof with the following lemma.

Lemma 2 *Let $s = f(s_1, \ldots, s_m) \in \mathcal{T}(\mathcal{F})$ and $t \mapsto t'$ be any step of a transformation chain of s with respect to ST until $norm_{ST}(s)$. If this step is determined relatively to a step $u \hookrightarrow u'$ of the strategy tree ST_f, then t is a ground instance of u.*

Proof. Let $s = f(s_1, \ldots, s_m) \in \mathcal{T}(\mathcal{F})$ be a term and $t \mapsto t'$ any step of a transformation chain of s with respect to ST until $norm_{ST}(s)$. The proof is made by induction on the length l of the transformation chain of s with respect to ST.

Let $l = 1$.

- if $f \in \mathcal{C}ons_{\mathcal{R}}$, by Definition 10, there is nothing to prove since ST_f is not used for normalizing the whole term s.
- if $f = g \in \mathcal{D}ef_R$, then by definition of the strategy S, the first step of each proof tree of ST is **Abstract**, or **Narrow** if f is a defined constant. As $l = 1$, this should also be the last step of the branch in the proof tree, which is impossible by definition of S.

Let $l = n \geq 1$.

Let $s' = g(s'_1, \ldots, s'_n)$ be the term obtained with a transformation chain of length n from s with respect to ST. Let the property be true for $l = n$, i.e. for any step $t \mapsto t'$ of the transformation chain of s with respect to ST until s', determined relatively to a step $u \hookrightarrow u'$ of the strategy tree ST_f, then t is a ground instance of u.

Let $s' \mapsto s''$ be the n+1-th step of the given transformation chain of s with respect to ST, determined relatively to a step $u' \hookrightarrow u''$ of the strategy tree ST_f. We prove that s'' is a ground instance of u''.

Let $s^0 \mapsto s'$ be the n-th step of the given transformation chain of s with respect to ST, determined relatively to a step $u^0 \hookrightarrow u'$ of the strategy tree ST_f. By hypothesis, s' is a ground instance of u'.

- if $u' \hookrightarrow u''$ is a step **Abstract**, then u'' is of the form $u'[X_1]_{i_1} \ldots [X_p]_{i_p}$, where $i_1, \ldots, i_p$ are the abstraction positions of u' and the X_j, for $j \in \{1, \ldots, p\}$, are new abstraction variables.

  By Definition 10, $s' \mapsto s'[s''_1]_{i_1} \ldots [s''_p]_{i_p}$, where the s''_j are $norm_{ST}(s'|_{i_j})$ if we do not have $WEAK\text{-}TERMIN(u'|_{i_j})$, and $s'|_{i_j}\downarrow$ otherwise:

– In the latter case, $s'|_{i_j}\downarrow$ is a ground instance of X_j ; let us note that $s'|_{i_j}\downarrow$ exists since by induction hypothesis $s'|_{i_j}$ is an instance of $u'|_{i_j}$ and we have $WEAK\text{-}TERMIN(u'_{i_j})$;

– In the former case, by Theorem 2 on each $s'|_{i_j}$ strict subterm of s', we have $norm_{ST}(s'|_{i_j})$ in normal form, hence a ground instance of X_j.

Therefore s'' is a ground instance of u''.

- if $u' \hookrightarrow u''$ is a step **Narrow**, by definition 10, either:
  - $s' \rightarrow^{p,l\rightarrow r,\mu} s'' = \mu u''$, where the used branch of the step **Narrow** (always existing since **Narrow**, with the strategy S, by definition of Σ_0 which covers all possible reducible instances of s') is such that $u' \rightsquigarrow^{Inn}_{p,l\rightarrow r,\sigma} u''$, with $\theta = \mu\sigma[Var(u')]$ and $s' = \theta u'$. In this case s'' is a ground instance of u''.
  - Or s' is irreducible and $s'' = s'$ and there is no narrowing step in the tree to follow, so there is no term u'' and then, there is nothing to prove.
- if $u' \hookrightarrow u''$ is a step **Stop**, then u'' does not exist, since the state in ST_f after this step is a state of the form $(\emptyset, A, C)$. Then there is nothing to prove.

Theorem 2. *Let $\mathcal{R}$ be a TRS proved weakly terminating with Theorem 1 and ST its set of strategy trees. Then reducing any term $t \in \mathcal{T}(\mathcal{F})$ with respect to ST leads to an irreducible term, which is an innermost normal form of t for $\mathcal{R}$.*

Proof. Let $s = f(s_1, \ldots, s_m) \in \mathcal{T}(\mathcal{F})$ be a term to be reduced with respect to ST. We first prove if a transformation chain of s with respect to ST gives a term t, t is also an innermost reduced form of s for $\mathcal{R}$. We then show that the last term of any transformation chain of s with respect to ST is irreducible for $\mathcal{R}$. This term eventually exists, since the normalizing process always stops, for ST is a set of successful trees, whose branches only have a finite number of states.

Let us show that if a transformation chain of s with respect to ST gives a term t, t is also an innermost reduced form of s for $\mathcal{R}$, by induction on the length l of the transformation chain of s with respect to ST. For that, we use an induction on $\mathcal{T}(\mathcal{F})$ and the property : $norm_{ST}(s)$ is an innermost normal form of s for $\mathcal{R}$, with an ordering $\succ$ satisfying all constraints generated by S in proving that $\mathcal{R}$ is weakly innermost terminating. We choose for that the ordering used in Theorem 1, for proving weak innermost termination of $\mathcal{R}$. Such an ordering exists since, by hypothesis, $\mathcal{R}$ is a TRS proved weakly terminating with the strategy S.

Let $l = 1$.

- if $f \in \mathcal{C}ons_{\mathcal{R}}$, by Definition 10, $norm_{ST}(f(s_1, \ldots, s_m)) = f(norm_{ST}(s_1), \ldots, norm_{ST}(s_n))$. With the second induction, we have $s = f(s_1, \ldots, s_m) \succ s_1, \ldots, s_m$, and then, by induction hypothesis, $norm_{ST}(s_1), \ldots, norm_{ST}(s_n)$ are respective normal forms of $s_1, \ldots, s_m$ for $\mathcal{R}$. Then, by definition of the innermost strategy, $f(norm_{ST}(s_1), \ldots, norm_{ST}(s_n))$ is an innermost reduced form of s for $\mathcal{R}$, and as f is a constructor symbol, is in normal form.

- if $f = g \in \mathcal{D}ef_R$, then by definition of the strategy S, the first step of each proof tree of ST is **Abstract** (or **Narrow** if $f \in \mathcal{C}$). As $l = 1$, this should also be the last step of the branch in the proof tree, which is impossible by definition of S.

Let $l = n \geq 1$.

Let $t = g(t_1, \ldots, t_n)$ be the term obtained with a transformation chain of length n from s with respect to ST. Let the property be true for $l = n$, i.e. t is an innermost reduced form of s with $\mathcal{R}$. We prove it for $l = n + 1$.

- If the current step in the proof tree of f is **Abstract**, by Definition 10, $t \mapsto t' = t[t'_1]_{i_1} \ldots [t'_p]_{i_p}$, where $i_1, \ldots, i_p$ are the abstraction positions and the t'_j are $norm_{ST}(t|_{i_j})$ if we do not have $WEAK\text{-}TERMIN(u_{|}i_j)$, and $t|_{i_j}\downarrow$ otherwise.

For $j \in \{1, \dots, p\}$ such that $WEAK\!-\!TERMIN(u|_{i_j})$, any ground instance of $u|_{i_j}$ is weakly innermost terminating. By Lemma 2, $t|_{i_j}$ strict subterm of t is a ground instance of $u|_{i_j}$, hence weakly innermost terminating, and $t'_j = t|_{i_j}\!\downarrow$ is an innermost normal form of $t|_{i_j}$.
For $j \in \{1, \dots, p\}$ such that we do not have $WEAK\!-\!TERMIN(u|_{i_j})$, induction hypothesis of the second induction applies to $u|_{i_j}$. By Lemma 2, $t|_{i_j}$ is a ground instance of $u|_{i_j}$, and then, by induction hypothesis, $norm_{ST}(t|_{i_j})$ is an innermost normal form of $t|_{i_j}$.
Then, by definition of the innermost strategy, the term $t[t'_1]_{i_1} \dots [t'_p]_{i_p}$ is an innermost reduced form of t for $\mathcal{R}$. Since, by induction hypothesis (with the first induction), t is an innermost reduced form of s for $\mathcal{R}$, we conclude that t' is an innermost reduced form of s for $\mathcal{R}$.

- If the current step in the proof tree of f is **Narrow**, by Definition 10, $t \rightarrow^{p,l\rightarrow r,\mu} \mu t'$, where the used branch of the step **Narrow** (always existing since **Narrow**, with the strategy S, by definition of Σ_0 which covers all possible reducible instances of t) is such that $u = g(u_1, \dots, u_n) \rightsquigarrow^{Inn}_{p,l\rightarrow r,\sigma} s'$, with $\theta = \mu\sigma[Var(u)]$ and $g(t_1, \dots, t_n) = \theta g(u_1, \dots, u_n)$.

  By definition of the rule **Narrow**, the narrowing step is innermost and, by Lemma 1, the reduction $g(t_1, \dots, t_n) \rightarrow^{p,l\rightarrow r,\mu} \mu t'$ is innermost, and hence $\mu t'$ is an innermost reduced form of t. Thus, reasoning as in the previous case, as t is an innermost reduced form of s with $\mathcal{R}$, we conclude that $\mu t'$ is an innermost reduced form of s with $\mathcal{R}$.

  If the latter rewriting is not possible (this is the case where μ does not exist), then t is already in normal form. The process stops, and the property is trivially satisfied.

- If the current step in the proof tree of f is **Stop**, then by Definition 10, $t = g(t_1, \dots, t_m) \mapsto t'$ where $t' = norm_{ST}(t)$ if we do not have $WEAK\!-\!TERMIN(u)$ and $t' = t\!\downarrow$ otherwise.
  In the first case, induction hypothesis of the second induction applies on u ; by Lemma 2, t is an instance of u, and then, by induction hypothesis on t, $norm_{ST}(t)$ is an innermost normal form of t for R.
  In the second case, any instance of u is weakly innermost terminating ; by Lemma 2, t is an instance of u and then $t' = t\!\downarrow$ is an innermost normal form of t.

This last step, together with the special case in **Narrow**, is the step where the normalizing process with respect to ST stops, in the case where $l = n > 1$. As shown previously, this step stops with an irreducible form for $\mathcal{R}$. For the special case in **Narrow**, t is already in normal form, which ends the proof.

C Proof of Theorem 3.

Theorem 3 *Let $\mathcal{R}$ be a TRS on a set $\mathcal{F}$ of symbols. If there exists an $\mathcal{F}$-stable ordering $\succ$ having the subterm property, such that for each defined symbol g, we have $SUCCESS\!-\!CUT(g, \succ)$, then any term of $\mathcal{T}(\mathcal{F})$ weakly innermost terminates.*

Proof. We prove by induction on $\mathcal{T}(\mathcal{F})$ that any ground instance $\theta f(x_1, \dots, x_m)$ of any term $f(x_1, \dots, x_m) \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ weakly innermost terminates. The induction ordering is constrained along the proof. At the beginning, it has at least to be $\mathcal{F}$-stable and to have the subterm property, which ensures its noetherianity. Such an ordering always exists on $\mathcal{T}(\mathcal{F})$ (for instance the embedding relation). Let us denote it $\succ$.

By subterm property of $\succ$, we have $\theta f(x_1, \dots, x_m) = f(\theta x_1, \dots, \theta x_m) \succ \theta x_1, \dots, \theta x_m$. Then, by induction hypothesis, let us suppose that the $\theta x_1, \dots, \theta x_m$ weakly innermost terminate. Let $\theta x_1\!\downarrow, \dots, \theta x_m\!\downarrow$ be any of their normal forms. It remains to prove that $f(\theta x_1\!\downarrow, \dots, \theta x_m\!\downarrow)$ weakly innermost terminates.

If f is a constructor, then $f(\theta x_1\!\downarrow, \dots, \theta x_m\!\downarrow)$ is irreducible for the innermost rewriting relation, and therefore, weakly innermost terminating.

If f is not a constructor, let us denote it g and prove that $g(\theta x_1, \ldots, \theta x_m)$ weakly innermost terminates for any θ satisfying $(A_0 = \top, C_0 = \top)$, if application of the inference rules on $\Lambda \sqsupset_p (\{g(x_1, \ldots, x_m)\}, \top, \top, \lambda)$, terminates on a proof tree reduced to a state $(\emptyset, A_p, C_p, \lambda)$. Let us denote $g(x_1, \ldots, x_m)$ by t_{ref} in the sequel of the proof.

To each step of the procedure characterized by a state $s = (\{t\}, A, C, \sigma)$, we associate the set of ground terms $G = \{\alpha t \mid \alpha \text{ satisfies } A\}$, that is the set of ground instances represented by s. Inference rules **Abstract** and **Narrow** transform $(\{t\}, A, C, \sigma)$ into $(\{t'\}, A', C', \mu)$ to which is associated $G' = \{\beta t' \mid \beta \text{ satisfies } A'\}$. We then prove the following result: if for all $\beta t' \in G'$, $\beta t'$ weakly innermost terminates, then any αt in G weakly innermost terminates. In the following, we take into account the cases where G or G' is empty, since the satisfiability of the ACF is not checked when applying inference rules.

- Either **Abstract** is applied, so the current term t becomes $t' = t[X_{i_1}]_{i_1} \ldots [X_{i_p}]_{i_p}$, where $i_1, \ldots, i_p$ are the abstraction positions of t. For each α such that $\alpha(t)$ is in G, we prove that there exists a β such that $\beta(t')$ is in G' and such that weak innermost termination of $\beta(t')$ implies weak innermost termination of $\alpha(t)$.

  If $G = \emptyset$, then we also have $G' = \emptyset$. Indeed, if A is unsatisfiable, then the constraint A', which is a conjunction of constraints containing A, is also unsatisfiable.

  Let us assume $G' \neq \emptyset$. According to the condition of application of **Abstract** at positions $i_1, \ldots, i_j$, each term $t|_{i_j}$ is such that:

  - either $WEAK{-}TERMIN(t|_{i_j})$ is true, and then by definition of the predicate $WEAK{-}TERMIN$, $\alpha t|_{i_j}$ weakly innermost terminates;
  - or $C \wedge t_{ref} > t|_{i_j}$ is satisfiable and then, by induction hypothesis, $\alpha t|_{i_j}$ weakly innermost terminates.

  Then let us define $\beta = \alpha \cup \bigcup_{j \in \{1,\ldots,p\}} X_{i_j} = \alpha t_{i_j}\downarrow$. Clearly β satisfies A', and $\beta t' = \alpha t[\alpha t|_{i_1}\downarrow]_{i_1} \ldots [\alpha t|_{i_p}\downarrow]_{i_p}$. Therefore, $G' \neq \emptyset$ and weak innermost termination of all $\beta t'$, for all possible normal forms $\alpha t|_{i_1}\downarrow, \ldots, \alpha t|_{i_p}\downarrow$ of $\alpha t|_{i_1}, \ldots, \alpha t|_{i_p}$, implies weak innermost termination of αt.

- Or **Narrow** is applied on $(\{t = f(u_1, \ldots, u_m)\}, A, C, \lambda)$.

  If $G = \emptyset$, then we also have $G' = \emptyset$. Indeed, if the constraint A is unsatisfiable, then in particular $A' = \sigma A$ is unsatisfiable, for any substitution σ.

  Let us assume $G \neq \emptyset$. For any α satisfying A, either $\alpha f(u_1, \ldots, u_m)$ is irreducible, and so weakly innermost terminates, or $\alpha f(u_1, \ldots, u_m)$ is innermost reducible. Since **Narrow** is applied with the set Σ_0 of substitutions, that, by definition of Σ_0, covers all possible instances $\alpha f(u_1, \ldots, u_m)$ of $f(u_1, \ldots, u_m)$, for each given α and by Lemma 1, there is a narrowing step $f(u_1, \ldots, u_m) \leadsto^{Inn}_{\sigma,p,l\to r} v$ corresponding to some rewriting step $\alpha f(u_1, \ldots, u_m) \to^{inn}_{p,l\to r} t'$, such that there exists a substitution β such that $Dom(\beta) = \mathcal{V}ar(v)$, $\beta\sigma = \alpha$ and $t' = \beta v$. By definition of weak innermost termination, weak innermost termination of βv implies weak innermost termination of $\alpha f(u_1, \ldots, u_m)$.

  On the other hand, on variables of $f(u_1, \ldots, u_m)$, i.e. variables introduced by the previous application of **Abstract**, we have $\alpha = \beta\sigma$. In addition, the domain of β, that is $Var(v)$, can be extended to the variables of A by setting $\beta x = \alpha x$ for $x \in Var(A) \setminus Var(f(u_1, \ldots, u_m))$. Thus, since α satisfies A, β satisfies σA.

Let us now prove that the ground instances satisfying A of each term t removed from the set $\mathcal{T} = \{t\}$ containing the current term of the state during the application of the rules, weakly innermost terminate.
The only rule removing terms from $\mathcal{T}$ is **Stop**. When **Stop** is applied and removes t from $\mathcal{T}$:

- either $WEAK{-}TERMIN(t)$ and then αt innermost terminates and hence weakly innermost terminates for any ground substitution α;

- or $C \wedge t_{ref} > t$ is satisfiable and then, by induction hypothesis, αt weakly innermost terminates;
- or A is unsatisfiable and then G is empty, hence the property is trivially true.

Let us now study the case of the rule **Shorten**. The only rules **Abstract** and **Narrow** generate children of a final state of the current proof tree. As proved above, the weak innermost termination of the ground instances represented by each of the children of a state implies the weak innermost termination of the ground instances represented by that state. The rule **Shorten** also removes a current term s of $\mathcal{T}$. By definition, it applies only if all children of the state s are of the form $(\emptyset, A, C, \mu)$. As proved previously in considering the rule **Stop**, and below in considering the rule **Cut**, a state $(\emptyset, A, C, \mu)$ generated from $s = (\{t\}, A, C, \mu)$ implies that the ground instances represented by s are weakly innermost terminating.

The last rule removing terms from $\mathcal{T}$ is **Cut**. By definition of this rule, $s = (\{t\}, A, C, \mu)$ is replaced by $(\emptyset, A, C, \mu)$ if s is the child of a state r having another child s' whose ground instances are weakly innermost terminating, and such that the ground instances of r that rewrite into the ground instances of s' contain the ground instances of r that rewrite into the ground instances of s. The ground instances of s' are weakly innermost terminating, and by definition of the weak innermost termination, the ground instances of s are weakly innermost terminating.

As the process is initialized with $\{t_{ref}\}$ and a constraint $A = \top$ satisfiable by any ground substitution, we get that $g(\theta x_1, \ldots, \theta x_m)$ is weakly innermost terminating, for any $t_{ref} = g(x_1, \ldots, x_m)$, and any ground instance θ.

Moreover, as the terms $f(x_1, \ldots, x_m)$, where f is a constructor, are also weakly innermost terminating for any ground instances $\theta x_1, \ldots, \theta x_m$, then any term of $\mathcal{T}(\mathcal{F})$ is weakly innermost terminating.

D Additional examples

In this section, we give extra examples, together with, at the beginning, the full development of the example given in the introduction of the paper. For each example, we detail in Section D.1 the proof trees without the rules **Cut** and **Shorten**, by directly cutting useless branches, and in Section D.2 we detail the operational way of pruning useless branches with the Cut and Shorten mechanism.

D.1 Examples without the Cut and Shorten mechanism

Example 1:

Let $\mathcal{R}$ be the following TRS, which is weakly innermost terminating, but not strongly innermost terminating:

$$\begin{array}{l} f(a) \rightarrow f(b) \\ f(x) \rightarrow x \\ b \rightarrow a. \end{array}$$

We prove its weak termination on $\mathcal{T}(\mathcal{F})$ with $\mathcal{F} = \{f : 1, a : 0, b : 0\}$ as follows.

Applying the rules on b, we get:

ϵ	b	$A = \top$	$C = \top$	
Narrow				
1	a	$A = \top$	$C = \top$	$\sigma = Id$
Stop				
1.1	$\emptyset$	$A = \top$	$C = \top$	

Applying the rules on $f(x)$, we get:

ϵ	$f(x_1)$	$A = \top$	$C = \top$	
Abstract				
1	$f(X_1)$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	
Narrow				
1.1	X_1	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
Stop				
1.1.1	$\emptyset$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	

The set of ordering constraints C is satisfiable by any $\mathcal{F}$-stable ordering having the subterm property. Since all proof trees are successful, weak innermost termination of $\mathcal{R}$ is ensured.

Example 2:
Let us consider the following rewrite system $\mathcal{R}$, which is not innermost terminating but is weakly innermost terminating.

$$
\begin{array}{ll}
f(x) & \to g(x) \\
f(a) & \to f(b) \\
b & \to a \\
g(x) & \to x \\
g(x) & \to f(g(x))
\end{array}
$$

Applying the rules on b, we get:

ϵ	b	$A = \top$	$C = \top$	
Narrow				
1	a	$A = \top$	$C = \top$	$\sigma = Id$
Stop				
1.1	$\emptyset$	$A = \top$	$C = \top$	

Applying the inference rules on $g(x_1)$, we get:

ϵ	$g(x_1)$	$A = \top$	$C = \top$	
Abstract				
1	$g(X_1)$	$A = (x_1{\downarrow} = X_1)$	$C = (g(x_1) > x_1)$	
Narrow				
1.1	X_1	$A = (x_1{\downarrow} = X_1)$	$C = (g(x_1) > x_1)$	$\sigma = Id$
$\boxed{1.2}$	$f(g(X_1))$	$A = (x_1{\downarrow} = X_1)$	$C = (g(x_1) > x_1)$	$\sigma = Id$
Stop				
1.1.1	$\emptyset$	$A = (x_1{\downarrow} = X_1)$	$C = (g(x_1) > x_1)$	
$\boxed{\mathit{1.2}}$	$f(g(X_1))$	$A = (x_1{\downarrow} = X_1)$	$C = (g(x_1) > x_1)$	

We can now cut the subtree starting from the state 1.2, since the subtree starting from the state 1.1, generated with the same narrowing substitution Id, is successful. The ordering constraints of C are satisfied by any $\mathcal{F}$-stable ordering having the subterm property. We finally get a successful tree for the symbol g.

Applying the inference rules on $f(x_1)$, we get:

ϵ	$f(x_1)$	$A = \top$	$C = \top$	
Abstract				
1	$f(X_1)$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	
Narrow				
1.1	$g(X_1)$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
Narrow				
1.1.1	X_1	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
$\boxed{1.1.2}$	$f(g(X_1))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
Stop				
1.1.1.1	$\emptyset$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	
$\boxed{\textit{1.1.2}}$	$f(g(X_1))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	

We can now cut the subtree starting from the state 1.1.2, since the subtree starting from the state 1.1.1, generated with the same narrowing substitution Id, is successful. We finally get a successful tree for the symbol f.

Example 3:

Let us consider the following rewrite system $\mathcal{R}$, which is not innermost terminating but is weakly innermost terminating.

$$
\begin{array}{ll}
h(0,x) & \rightarrow f(0,x,x) \\
f(0,1,x) & \rightarrow h(x,x) \\
g(x,y) & \rightarrow x \\
g(x,y) & \rightarrow g(g(x,y),y).
\end{array}
$$

Applying the inference rules on $h(x_1, x_2)$, we get:

ϵ	$h(x_1, x_2)$	$A = \top$	$C = \top$	
Abstract				
1	$h(X_1, X_2)$	$A = (x_1{\downarrow} = X_1 \wedge x_2{\downarrow} = X_2)$	$C = (h(x_1, x_2) > x_1, x_2)$	
Narrow				
1.1	$f(0, X_2, X_2)$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = X_2)$	$C = (h(x_1, x_2) > x_1, x_2)$	$\sigma = (X_1 = 0)$
Narrow				
1.1.1	$h(1,1)$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = 1)$	$C = (h(x_1, x_2) > x_1, x_2)$	$\sigma = (X_2 = 1)$
Stop				
1.1.1.1	$\emptyset$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = 1)$	$C = (h(x_1, x_2) > x_1, x_2)$	

Appying the inference rules on $f(x_1, x_2, x_3)$, we get:

ϵ	$f(x_1, x_2, x_3)$	$A = \top$	$C = \top$
Abstract			
1	$f(X_1, X_2, X_3)$	$A = (x_1{\downarrow} = X_1 \wedge x_2{\downarrow} = X_2 \wedge x_3{\downarrow} = X_3)$	$C = (f(x_1, x_2, x_3) > x_1, x_2, x_3)$
Narrow			
1.1	$h(X_3, X_3)$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = 1 \wedge x_3{\downarrow} = X_3)$	$C = (f(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = 0 \wedge X_2 = 1)$
Narrow			
1.1.1	$f(0,0,0)$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = 1 \wedge x_3{\downarrow} = 0)$	$C = (f(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_3 = 0)$
Stop			
1.1.1.1	$\emptyset$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = 1 \wedge x_3{\downarrow} = 0)$	$C = (f(x_1, x_2, x_3) > x_1, x_2, x_3)$

Applying the rules on $g(x_1, x_2)$, we get:

ϵ	$g(x_1, x_2)$	$A = \top$	$C = \top$	
Abstract				
1	$g(X_1, X_2)$	$A = (x_1{\downarrow} = X_1 \wedge x_2{\downarrow} = X_2)$	$C = (g(x_1, x_2) > x_1, x_2)$	
Narrow				
1.1	X_1	$A = (x_1{\downarrow} = X_1 \wedge x_2{\downarrow} = X_2)$	$C = (g(x_1, x_2) > x_1, x_2)$	$\sigma = Id$
$\boxed{1.2}$	$g(g(X_1, X_2), X_2)$	$A = (x_1{\downarrow} = X_1 \wedge x_2{\downarrow} = X_2)$	$C = (g(x_1, x_2) > x_1, x_2)$	$\sigma = Id$
Stop				
1.1.1	$\emptyset$	$A = (x_1{\downarrow} = X_1 \wedge x_2{\downarrow} = X_2)$	$C = (g(x_1, x_2) > x_1, x_2)$	
$\boxed{\mathit{1.2}}$	$g(g(X', Y'), Y')$	$A = (x_1{\downarrow} = X_1 \wedge x_2{\downarrow} = X_2)$	$C = (g(x_1, x_2) > x_1, x_2)$	

We can now cut the subtree starting from the state 1.2, since the subtree starting from the state 1.1, generated with the same narrowing substitution Id, is successful. We finally get a successful tree for the symbol g.

Finally, all proof trees are auccessful, and weak innermost termination of $\mathcal{R}$ is hence proved.

Example 4:

Let us consider the following rewrite system $\mathcal{R}$, which is not innermost terminating but is weakly innermost terminating.

$$\begin{array}{ll} f(x) & \rightarrow g(f(x)) \\ f(x) & \rightarrow a \\ g(x) & \rightarrow f(x). \end{array}$$

Applying the inference rules on $f(x_1)$, we get:

ϵ	$f(x_1)$	$A = \top$	$C = \top$	
Abstract				
1	$f(X_1)$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	
Narrow				
1.1	a	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
$\boxed{1.2}$	$g(f(X_1))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
Stop				
1.1.1	$\emptyset$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	
$\boxed{\mathit{1.2}}$	$g(f(X_1))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	

We can now cut the subtree starting from the state 1.2, since the subtree starting from the state 1.1, generated with the same narrowing substitution Id, is successful. We finally get a successful tree for the symbol f.

Applying the inference rules on $g(x_1)$, we get:

ϵ	$g(x_1)$	$A = \top$	$C = \top$	
Abstract				
1	$g(X_1)$	$A = (x_1{\downarrow} = X_1)$	$C = (g(x_1) > x_1)$	
Narrow				
1.1	$f(X_1)$	$A = (x_1{\downarrow} = X_1)$	$C = (g(x_1) > x_1)$	$\sigma = Id$
Narrow				
1.1.1	a	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
$\boxed{1.1.2}$	$g(f(X_1))$	$A = (x_1{\downarrow} = X_1)$	$C = (g(x_1) > x_1)$	$\sigma = Id$
Stop				
1.1.1.1	$\emptyset$	$A = (x_1{\downarrow} = X_1)$	$C = (g(x_1) > x_1)$	
$\boxed{\mathit{1.1.2}}$	$g(f(X_1))$	$A = (x_1{\downarrow} = X_1)$	$C = (g(x_1) > x_1)$	

We can now cut the subtree starting from the state 1.1.2, since the subtree starting from the state 1.1.1, generated with the same narrowing substitution Id, is successful. We finally get a successful tree for the symbol g.

Since all proof trees are successful, $\mathcal{R}$ is weakly innermost terminating.

Example 5:

Let us consider the following rewrite system $\mathcal{R}$, which is not innermost terminating but is weakly innermost terminating.

$$\begin{array}{ll} f(h(x)) & \to f(g(x)) \\ f(x) & \to i(x) \\ g(x) & \to h(x). \end{array}$$

We prove the weak innermost termination of this system on $\mathcal{T}(\mathcal{F}, \mathcal{X})$, where $F = \{f : 1, g : 1, h : 1, i : 1, a : 0\}$.

Applying the inference rules on $f(x_1)$, we get:

ϵ	$f(x_1)$	$A = \top$	$C = \top$	
Abstract				
1	$f(X_1)$	$A = (x_1\downarrow = X_1)$	$C = (f(x_1) > x_1)$	
Narrow				
1.1	$i(X_1)$	$A = (x_1\downarrow = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
Stop				
1.1.1	$\emptyset$	$A = (x_1\downarrow = X_1)$	$C = (f(x_1) > x_1)$	

Applying the inference rules on $g(x_1)$, we get:

ϵ	$g(x_1)$	$A = \top$	$C = \top$	
Abstract				
1	$g(X_1)$	$A = (x_1\downarrow = X_1)$	$C = (g(x_1) > x_1)$	
Narrow				
1.1	$h(X_1)$	$A = (x_1\downarrow = X_1)$	$C = (g(x_1) > x_1)$	$\sigma = Id$
Stop				
1.1.1	$\emptyset$	$A = (x_1\downarrow = X_1)$	$C = (g(x_1) > x_1)$	

All proof trees are successful, hence $\mathcal{R}$ is weakly innermost terminating.

Example: 6

Let us consider the following rewrite system $\mathcal{R}$, which is not innermost terminating but is weakly innermost terminating.

$$\begin{array}{ll} f(g(x)) & \to f(h(x)) \\ f(h(x)) & \to f(g(x)) \\ f(x) & \to i(x). \end{array}$$

We prove the weak innermost termination of this system on $\mathcal{T}(\mathcal{F}, \mathcal{X})$, where $F = \{f : 1, g : 1, h : 1, i : 1, a : 0\}$.

Applying the inference rules on $f(x_1)$, we get:

ϵ	$f(x_1)$	$A = \top$	$C = \top$	
Abstract				
1	$f(X_1)$	$A = (x_1\downarrow = X_1)$	$C = (f(x_1) > x_1)$	
Narrow				
1.1	$i(X_1)$	$A = (x_1\downarrow = X_1)$	$C = (f(x_1) > x_1)$	$\sigma = Id$
Stop				
1.1.1	$\emptyset$	$A = (x_1\downarrow = X_1)$	$C = (f(x_1) > x_1)$	

Since the proof tree of the only defined symbol f is successful, $\mathcal{R}$ is weakly innermost terminating.

Example 7:

Let us consider the following rewrite system $\mathcal{R}$, which is neither terminating nor innermost terminating but weakly innermost terminating.

$$
\begin{array}{lll}
quot(0, s(y), s(z)) & \to 0 & (1) \\
quot(s(x), s(y), z) & \to quot(x, y, z) & (2) \\
quot(x, 0, s(z)) & \to s(quot(x, s(z), s(z))) & (3) \\
quot(0, y, 0) & \to quot(y, 0, 0) & (4) \\
quot(s(x), 0, 0) & \to s(x) & (5) \\
quot(0, 0, 0) & \to 0 & (6)
\end{array}
$$

Applying the rules on $quot(x_1, x_2, x_3)$, we get :

ϵ	$quot(x_1, x_2, x_3)$	$A = \top$ $C = \top$
Abstract		
1	$quot(X_1, X_2, X_3)$	$A = (x_1{\downarrow} = X_1 \wedge x_2{\downarrow} = X_2 \wedge x_3{\downarrow} = X_3)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$
Narrow		
1.1	0	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = s(X_4) \wedge x_3{\downarrow} = s(X_5))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = 0 \wedge X_2 = s(X_4) \wedge X_3 = s(X_5))$
1.2	$quot(X_6, X_7, X_3)$	$A = (x_1{\downarrow} = s(X_6) \wedge x_2{\downarrow} = s(X_7) \wedge x_3{\downarrow} = X_3)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = s(X_6) \wedge X_2 = s(X_7))$
1.3	$s(quot(X_1, s(X_8), s(X_8)))$	$A = (x_1{\downarrow} = X_1 \wedge x_2{\downarrow} = 0 \wedge x_3{\downarrow} = s(X_8))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_2 = 0 \wedge X_3 = s(X_8))$
1.4	$quot(X_2, 0, 0)$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = X_2 \wedge x_3{\downarrow} = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = 0 \wedge X_3 = 0)$
1.5	$s(X_1)$	$A = (x_1{\downarrow} = s(X_9) \wedge x_2{\downarrow} = 0 \wedge x_3{\downarrow} = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = s(X_9) \wedge X_2 = 0 \wedge X_3 = 0)$

A sixth narrowing branch could be obtained with the substitution $(X_1 = 0 \wedge X_2 = 0 \wedge X_3 = 0)$. However, this substitution is strictly less general than the substitution $(X_1 = 0 \wedge X_3 = 0)$ used to generate the state 1.4, hence the corresponding branch is not built.

Stop (*on three branches*)

1.1.1	$\emptyset$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = s(X_4) \wedge x_3{\downarrow} = s(X_5))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$
1.2.1	$\emptyset$	$A = (x_1{\downarrow} = s(X_6) \wedge x_2{\downarrow} = s(X_7) \wedge x_3{\downarrow} = X_3)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3, quot(X_6, X_7, X_3))$
1.3	$s(quot(X_1, s(X_8), s(X_8)))$	$A = (x_1{\downarrow} = X_1 \wedge x_2{\downarrow} = 0 \wedge x_3{\downarrow} = s(X_8))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$
1.4	$quot(X_2, 0, 0)$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = X_2 \wedge x_3{\downarrow} = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$
1.5.1	$\emptyset$	$A = (x_1{\downarrow} = s(X_9) \wedge x_2{\downarrow} = 0 \wedge x_3{\downarrow} = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$

Stop applies on states 1.1 and 1.5 because each term t of these states is not narrowable and contains only NF-variable (if any), which ensures that any ground instance of t is in normal form.

Stop also applies on state 1.2 if we choose for the induction ordering a simplification ordering based on the precedence $quot \succ_{\mathcal{F}} s, 0$. Indeed, $\mathcal{R}$ is well-covered, i.e. any normal form in $\mathcal{R}$ is composed only of constructor symbols. With the precedence chosen above, we then get $\theta(t) \succeq \theta(t{\downarrow})$, for any term t and ground instance θ with $Dom(\theta) \subseteq \mathcal{V}ar(t)$. Under these assumptions, for any θ satisfying $A = (x_1\downarrow = s(X_6) \wedge x_2\downarrow = s(X_7) \wedge x_3\downarrow = X_3)$, we have $\theta x_1 \succeq \theta s(X_6)$, $\theta x_2 \succeq \theta s(X_7)$ and $\theta x_3 \succeq \theta X_3$. By subterm property of the ordering, and because it is stable by instanciation, we get $\theta s(X_6) \succ \theta X_6$ and, by transitivity, $\theta x_1 \succ \theta X_6$. Likewise, we get $\theta x_2 \succ \theta X_7$. By $\mathcal{F}$-stability of the ordering, we finally have $\theta quot(x_1, x_2, x_3) \succ \theta quot(X_6, X_7, X_3)$, for any substitution θ satisfying A. Therefore the induction hypothesis applies on $quot(X_6, X_7, X_3)$.

Narrow

1.1.1	$\emptyset$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = s(X_4) \wedge x_3\downarrow = s(X_5))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$
1.2.1	$\emptyset$	$A = (x_1\downarrow = s(X_6) \wedge x_2\downarrow = s(X_7) \wedge x_3\downarrow = X_3)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3, quot(X_6, X_7, X_3))$
1.3.1	$s(0)$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = 0 \wedge x_3\downarrow = s(X_8))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = 0)$
1.3.2	$s(quot(X_9, X_8, s(X_8)))$	$A = (x_1\downarrow = s(X_9) \wedge x_2\downarrow = 0 \wedge x_3\downarrow = s(X_8))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = s(X_9))$
$\boxed{1.4.1}$	$quot(0, 0, 0)$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = 0 \wedge x_3\downarrow = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_2 = 0)$
1.4.2	$s(X_{10})$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = s(X_{10}) \wedge x_3\downarrow = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_2 = s(X_{10}))$
1.4.3	0	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = 0 \wedge x_3\downarrow = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_2 = 0)$
1.5.1	$\emptyset$	$A = (x_1\downarrow = s(X_9) \wedge x_2\downarrow = 0 \wedge x_3\downarrow = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$

Here, the *narrow* step applies because the *abstract* step does nothing, since there is no possible abstraction position.
Indeed, on the term $t = s(quot(X_1, s(X_8), s(X_8)))$ of the state 1.3, the only position candidate for abstraction is 1 (since for any suffix position p we have ground instances $t|_p$ in normal form), and $t|_1 = quot(X_1, s(X_8), s(X_8))$ cannot be abstracted here.
As for the term $t = quot(X_2, 0, 0)$ of the state 1.4, each of its subterm $t|_i$ is such that its ground instances are in normal form, and then does not need to be abstracted.

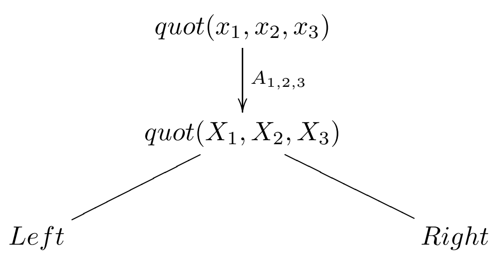
According to the strategy S of application of inference rules, **Narrow** applies and yields the new states 1.3.1, 1.3.2, 1.4.1, 1.4.2 and 1.4.3.

Stop	(*on four branches*)	
1.1.1	$\emptyset$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = s(X_4) \wedge x_3\downarrow = s(X_5))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$
1.2.1	$\emptyset$	$A = (x_1\downarrow = s(X_6) \wedge x_2\downarrow = s(X_7) \wedge x_3\downarrow = X_3)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3, quot(X_6, X_7, X_3))$
1.3.1.1	$\emptyset$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = 0 \wedge x_3\downarrow = s(X_8))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$
1.3.2.1	$\emptyset$	$A = (x_1\downarrow = s(X_9) \wedge x_2\downarrow = 0 \wedge x_3\downarrow = s(X_8))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3, s(quot(X_9, X_8, s(X_8))))$
$\boxed{1.4.1}$	$quot(0, 0, 0)$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = 0 \wedge x_3\downarrow = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$
1.4.2.1	$\emptyset$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = s(X_{10}) \wedge x_3\downarrow = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$
1.4.3.1	$\emptyset$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = 0 \wedge x_3\downarrow = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$
1.5.1	$\emptyset$	$A = (x_1\downarrow = s(X_9) \wedge x_2\downarrow = 0 \wedge x_3\downarrow = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$

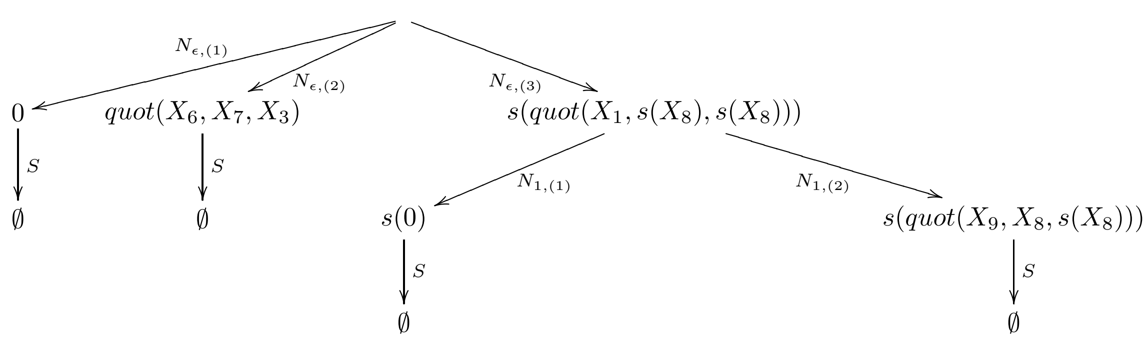
Stop applies on states 1.3.1 and 1.4.2 because each term t of these states is such that any of its ground instance is in normal form. **Stop** applies on the state 1.3.2 using the same reasoning as the one we got to apply **Stop** on the state 1.2 above.

The branch starting from the state 1.4.1 can be cut, because the narrowing substitution $\sigma = (X_2 = 0)$ generating this state is the same as the narrowing substitution generating the successful branch starting from the state 1.4.3. From now on, we have only successful states and hence the proof tree is successful.

By Theorem 1, we then get that $\mathcal{R}$ is terminating. Moreover, we can extract from the proof above the following strategy tree for symbol *quot*, where we denote $A_{i_1,\ldots,i_p}$ for the abstraction at positions $i_1, \ldots, i_p$, $N_{p,(i)}$ for innermost narrowing at position p with rule (i) and S for a **Stop** application.



where $Left$ and $Right$, respectively, are the trees



and

$$
\begin{tikzcd}
 & & & {} \arrow[dl, "N_{\epsilon,(4)}"] \arrow[dr, "N_{\epsilon,(5)}"'] & \\
 & & quot(X_2, 0, 0) \arrow[dll, "N_{\epsilon,(5)}"] \arrow[dr, "N_{\epsilon,(6)}"'] & & s(X_1) \arrow[d, "S"] \\
s(X_{10}) \arrow[d, "S"] & & & 0 \arrow[d, "S"] & \emptyset \\
\emptyset & & & \emptyset &
\end{tikzcd}
$$

Let us use the above strategy tree for normalizing the term $quot(s^5(0), s^5(0), s^2(0))$, where we use the notation $t \mapsto t' /*rule*/$ for the normalizing step from t to t' following the corresponding *rule* in the strategy tree :

$$
\begin{array}{llr}
q(s^5(0), s^5(0), s^2(0)) & \mapsto q(s^5(0), s^5(0), s^2(0)) & /*A_{1,2,3}*/ \\
 & \mapsto q(s^4(0), s^4(0), s^2(0)) & /*N_{\epsilon,(2)}*/ \\
 & \mapsto norm_{ST}(q(s^4(0), s^4(0), s^2(0))) & /*S*/ \\
q(s^4(0), s^4(0), s^2(0)) & \mapsto q(s^4(0), s^4(0), s^2(0)) & /*A_{1,2,3}*/ \\
 & \mapsto q(s^3(0), s^3(0), s^2(0)) & /*N_{\epsilon,(2)}*/ \\
 & \mapsto norm_{ST}(q(s^3(0), s^3(0), s^2(0))) & /*S*/ \\
q(s^3(0), s^3(0), s^2(0)) & \mapsto q(s^3(0), s^3(0), s^2(0)) & /*A_{1,2,3}*/ \\
 & \mapsto q(s^2(0), s^2(0), s^2(0)) & /*N_{\epsilon,(2)}*/ \\
 & \mapsto norm_{ST}(q(s^2(0), s^2(0), s^2(0))) & /*S*/ \\
q(s^2(0), s^2(0), s^2(0)) & \mapsto q(s^2(0), s^2(0), s^2(0)) & /*A_{1,2,3}*/ \\
 & \mapsto q(s(0), s(0), s^2(0)) & /*N_{\epsilon,(2)}*/ \\
 & \mapsto norm_{ST}(q(s(0), s(0), s^2(0))) & /*S*/ \\
q(s(0), s(0), s^2(0)) & \mapsto q(s(0), s(0), s^2(0)) & /*A_{1,2,3}*/ \\
 & \mapsto q(0, 0, s^2(0)) & /*N_{\epsilon,(2)}*/ \\
 & \mapsto norm_{ST}(q(0, 0, s^2(0))) & /*S*/ \\
q(0, 0, s^2(0)) & \mapsto q(0, 0, s^2(0)) & /*A_{1,2,3}*/ \\
 & \mapsto s(q(0, s^2(0), s^2(0))) & /*N_{\epsilon,(3)}*/ \\
 & \mapsto s(0) & /*N_{1,(1)}*/ \\
 & \mapsto s(0) & /*S*/
\end{array}
$$

We finally got that $s(0)$ is an innermost normal form of $quot(s^5(0), s^5(0), s^2(0))$.

D.2 Examples with the Cut and Shorten mechanism

We present in this section the same examples than those developped in the previous section, now proved weakly innermost terminating with the rules of Table 2.

Example 1:

Let R be the following TRS, which is weakly innermost terminating, but not strongly innermost terminating:

$$
\begin{array}{l}
f(a) \to f(b) \\
f(x) \to x \\
b \to a.
\end{array}
$$

We prove its weak termination on $\mathcal{T}(\mathcal{F})$ with $\mathcal{F} = \{f : 1, a : 0, b : 0\}$ as follows.

Applying the rules on b, we get:

ϵ	b	$A = \top$	$C = \top$	$\sigma = \lambda$
Narrow				
1	a	$A = \top$	$C = \top$	$\sigma = Id$
Stop				
1	$\emptyset$	$A = \top$	$C = \top$	$\sigma = Id$
Shorten				
ϵ	$\emptyset$	$A = \top$	$C = \top$	$\sigma = \lambda$

Applying the rules on $f(x)$, we get:

ϵ	$f(x)$	$A = \top$	$C = \top$	$\sigma = \lambda$
Abstract				
1	$f(X)$	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = \lambda$
Narrow				
1.1	X	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = Id$
Stop				
1.1	$\emptyset$	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = Id$
Shorten				
1	$\emptyset$	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = \lambda$
Shorten				
ϵ	$\emptyset$	$A = \top$	$C = \top$	$\sigma = \lambda$

The ordering constraints of C are satisfied by any $\mathcal{F}$-stable ordering having the subterm property.

Example 2:
Let us consider the following rewrite system $\mathcal{R}$, which is not innermost terminating but is weakly innermost terminating.

$$\begin{array}{ll} f(x) & \rightarrow g(x) \\ f(a) & \rightarrow f(b) \\ b & \rightarrow a \\ g(x) & \rightarrow x \\ g(x) & \rightarrow f(g(x)) \end{array}$$

Applying the rules on b, we get:

ϵ	b	$A = \top$	$C = \top$	$\sigma = \lambda$
Narrow				
1	a	$A = \top$	$C = \top$	$\sigma = Id$
Stop				
1	$\emptyset$	$A = \top$	$C = \top$	$\sigma = Id$
Shorten				
ϵ	$\emptyset$	$A = \top$	$C = \top$	$\sigma = \lambda$

Applying the inference rules on $g(x_1)$, we get:

ϵ	$g(x_1)$	$A = \top$	$C = \top, \quad \sigma = \lambda$
Abstract			
1	$g(X_1)$	$A = (x_1{\downarrow} = X_1)$	$C = (g(x_1) > x_1), \quad \sigma = \lambda$
Narrow			
1.1	X_1	$A = (x_1{\downarrow} = X_1)$	$C = (g(x_1) > x_1), \quad \sigma = Id$
1.2	$f(g(X_1))$	$A = (x_1{\downarrow} = X_1)$	$C = (g(x_1) > x_1), \quad \sigma = Id$
Stop			
1.1	$\emptyset$	$A = (x_1{\downarrow} = X_1)$	$C = (g(x_1) > x_1), \quad \sigma = Id$
1.2	$f(g(X_1))$	$A = (x_1{\downarrow} = X_1)$	$C = (g(x_1) > x_1), \quad \sigma = Id$
Cut			
1.1	$\emptyset$	$A = (x_1{\downarrow} = X_1)$	$C = (g(x_1) > x_1), \quad \sigma = Id$
1.2	$\emptyset$	$A = (x_1{\downarrow} = X_1)$	$C = (g(x_1) > x_1), \quad \sigma = Id$
Shorten			
1	$\emptyset$	$A = (x_1{\downarrow} = X_1)$	$C = (g(x_1) > x_1), \quad \sigma = \lambda$
Shorten			
ϵ	$\emptyset$	$C = \top, \quad \sigma = \lambda$	

The ordering constraints of C are satisfied by any $\mathcal{F}$-stable ordering having the subterm property.

Applying the inference rules on $f(x_1)$, we get:

ϵ	$f(x_1)$	$A = \top$	$C = \top, \quad \sigma = \lambda$
Abstract			
1	$f(X_1)$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1), \quad \sigma = \lambda$
Narrow			
1.1	$g(X_1)$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1), \quad \sigma = Id$
Narrow			
1.1.1	X_1	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1), \quad \sigma = Id$
1.1.2	$f(g(X_1))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1), \quad \sigma = Id$
Stop			
1.1.1	$\emptyset$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1), \quad \sigma = Id$
1.1.2	$f(g(X_1))$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1), \quad \sigma = Id$
Cut			
1.1.1	$\emptyset$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1), \quad \sigma = Id$
1.1.2	$\emptyset$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1), \quad \sigma = Id$
Shorten			
1.1	$\emptyset$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1), \quad \sigma = Id$
Shorten			
1	$\emptyset$	$A = (x_1{\downarrow} = X_1)$	$C = (f(x_1) > x_1), \quad \sigma = \lambda$
Shorten			
ϵ	$\emptyset$	$A = \top$	$C = \top, \quad \sigma = \lambda$

Example 3:

Let us consider the following rewrite system $\mathcal{R}$, which is not innermost terminating but is weakly innermost terminating.

$$\begin{array}{ll} h(0,x) & \rightarrow f(0,x,x) \\ f(0,1,x) & \rightarrow h(x,x) \\ g(x,y) & \rightarrow x \\ g(x,y) & \rightarrow g(g(x,y),y). \end{array}$$

Applying the inference rules on $h(x_1, x_2)$, we get:

ϵ	$h(x_1, x_2)$	$A = \top$	$C = \top$	$\sigma = \lambda$
Abstract				
1	$h(X_1, X_2)$	$A = (x_1{\downarrow} = X_1 \wedge x_2{\downarrow} = X_2)$	$C = (h(x_1, x_2) > x_1, x_2)$	$\sigma = \lambda$
Narrow				
1.1	$f(0, X_2, X_2)$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = X_2)$	$C = (h(x_1, x_2) > x_1, x_2)$	$\sigma = (X_1 = 0)$
Narrow				
1.1.1	$h(1, 1)$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = 1)$	$C = (h(x_1, x_2) > x_1, x_2)$	$\sigma = (X_2 = 1)$
Stop				
1.1.1	$\emptyset$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = 1)$	$C = (h(x_1, x_2) > x_1, x_2)$	$\sigma = (X_2 = 1)$
Shorten				
1.1	$\emptyset$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = X_2)$	$C = (h(x_1, x_2) > x_1, x_2)$	$\sigma = (X_1 = 0)$
Shorten				
1	$\emptyset$	$A = (x_1{\downarrow} = X_1 \wedge x_2{\downarrow} = X_2)$	$C = (h(x_1, x_2) > x_1, x_2))$	$\sigma = \lambda$
Shorten				
ϵ	$\emptyset$	$A = \top$	$C = \top$	$\sigma = \lambda$

Appying the inference rules on $f(x_1, x_2, x_3)$, we get:

ϵ	$f(x_1, x_2, x_3)$	$A = \top$	$C = \top$ $\sigma = \lambda$
Abstract			
1	$f(X_1, X_2, X_3)$	$A = (x_1{\downarrow} = X_1 \wedge x_2{\downarrow} = X_2 \wedge x_3{\downarrow} = X_3)$	$C = (f(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = \lambda$
Narrow			
1.1	$h(X_3, X_3)$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = 1 \wedge x_3{\downarrow} = X_3)$	$C = (f(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = 0 \wedge X_2 = 1)$
Narrow			
1.1.1	$f(0, 0, 0)$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = 1 \wedge x_3{\downarrow} = 0)$	$C = (f(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_3 = 0)$
Stop			
1.1.1	$\emptyset$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = 1 \wedge x_3{\downarrow} = 0)$	$C = (f(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_3 = 0)$
Shorten			
1.1	$\emptyset$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = 1 \wedge x_3{\downarrow} = X_3)$	$C = (f(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = 0 \wedge X_2 = 1)$
Shorten			
1	$\emptyset$	$A = (x_1{\downarrow} = X_1 \wedge x_2{\downarrow} = X_2 \wedge x_3{\downarrow} = X_3)$	$C = (f(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = \lambda$
Shorten			
ϵ	$\emptyset$	$A = \top$	$C = \top$ $\sigma = \lambda$

Applying the rules on $g(x_1, x_2)$, we get:

ϵ	$g(x_1, x_2)$	$A = \top$	$C = \top$	$\sigma = \lambda$
Abstract				
1	$g(X_1, X_2)$	$A = (x_1\downarrow = X_1 \wedge x_2\downarrow = X_2)$	$C = (g(x_1, x_2) > x_1, x_2)$	$\sigma = \lambda$
Narrow				
1.1	X_1	$A = (x_1\downarrow = X_1 \wedge x_2\downarrow = X_2)$	$C = (g(x_1, x_2) > x_1, x_2)$	$\sigma = Id$
1.2	$g(g(X_1, X_2), X_2)$	$A = (x_1\downarrow = X_1 \wedge x_2\downarrow = X_2)$	$C = (g(x_1, x_2) > x_1, x_2)$	$\sigma = Id$
Stop				
1.1	$\emptyset$	$A = (x_1\downarrow = X_1 \wedge x_2\downarrow = X_2)$	$C = (g(x_1, x_2) > x_1, x_2)$	$\sigma = Id$
1.2	$g(g(X', Y'), Y')$	$A = (x_1\downarrow = X_1 \wedge x_2\downarrow = X_2)$	$C = (g(x_1, x_2) > x_1, x_2)$	$\sigma = Id$
Cut				
1.1	$\emptyset$	$A = (x_1\downarrow = X_1 \wedge x_2\downarrow = X_2)$	$C = (g(x_1, x_2) > x_1, x_2)$	$\sigma = Id$
1.2	$\emptyset$	$A = (x_1\downarrow = X_1 \wedge x_2\downarrow = X_2)$	$C = (g(x_1, x_2) > x_1, x_2)$	$\sigma = Id$
Shorten				
1	$\emptyset$	$A = (x_1\downarrow = X_1 \wedge x_2\downarrow = X_2)$	$C = (g(x_1, x_2) > x_1, x_2)$	$\sigma = \lambda$
Shorten				
ϵ	$\emptyset$	$A = \top$	$C = \top$	$\sigma = \lambda$

Example 4:
Let us consider the following rewrite system $\mathcal{R}$, which is not innermost terminating but is weakly innermost terminating.

$$\begin{array}{ll} f(x) & \rightarrow g(f(x)) \\ f(x) & \rightarrow a \\ g(x) & \rightarrow f(x). \end{array}$$

Applying the inference rules on $f(x)$, we get:

ϵ	$f(x)$	$A = \top$	$C = \top$	$\sigma = \lambda$
Abstract				
1	$f(X)$	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = \lambda$
Narrow				
1.1	a	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = Id$
1.2	$g(f(X))$	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = Id$
Stop				
1.1	$\emptyset$	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = Id$
1.2	$g(f(X))$	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = Id$
Cut				
1.1	$\emptyset$	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = Id$
1.2	$\emptyset$	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = Id$
Shorten				
1	$\emptyset$	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = \lambda$
Shorten				
ϵ	$\emptyset$	$A = \top$	$C = \top$	$\sigma = \lambda$

Applying the inference rules on $g(x)$, we get:

ϵ	$g(x)$	$A = \top$	$C = \top$	$\sigma = \lambda$
Abstract				
1	$g(X)$	$A = (x\downarrow = X)$	$C = (g(x) > x)$	$\sigma = \lambda$
Narrow				
1.1	$f(X)$	$A = (x\downarrow = X)$	$C = (g(x) > x)$	$\sigma = Id$
Narrow				
1.1.1	a	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = Id$
1.1.2	$g(f(X))$	$A = (x\downarrow = X)$	$C = (g(x) > x)$	$\sigma = Id$
Stop				
1.1.1	$\emptyset$	$A = (x\downarrow = X)$	$C = (g(x) > x)$	$\sigma = Id$
1.1.2	$g(f(X))$	$A = (x\downarrow = X)$	$C = (g(x) > x)$	$\sigma = Id$
Cut				
1.1.1	$\emptyset$	$A = (x\downarrow = X)$	$C = (g(x) > x)$	$\sigma = Id$
1.1.2	$\emptyset$	$A = (x\downarrow = X)$	$C = (g(x) > x)$	$\sigma = Id$
Shorten				
1.1	$\emptyset$	$A = (x\downarrow = X)$	$C = (g(x) > x)$	$\sigma = Id$
Shorten				
1	$\emptyset$	$A = (x\downarrow = X)$	$C = (g(x) > x)$	$\sigma = \lambda$
Shorten				
ϵ	$\emptyset$	$A = \top$	$C = \top$	$\sigma = \lambda$

Example 5:

Let us consider the following rewrite system $\mathcal{R}$, which is not innermost terminating but is weakly innermost terminating.

$$\begin{array}{ll} f(h(x)) & \rightarrow f(g(x)) \\ f(x) & \rightarrow i(x) \\ g(x) & \rightarrow h(x). \end{array}$$

We prove the weak innermost termination of this system on $\mathcal{T}(\mathcal{F}, \mathcal{X})$, where $\mathcal{F} = \{f : 1, g : 1, h : 1, i : 1, a : 0\}$.

Applying the inference rules on $f(x)$, we get:

ϵ	$f(x)$	$A = \top$	$C = \top$	$\sigma = \lambda$
Abstract				
1	$f(X)$	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = \lambda$
Narrow				
1.1	$i(X)$	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = Id$
Stop				
1.1	$\emptyset$	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = Id$
Shorten				
1	$\emptyset$	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = \lambda$
Shorten				
ϵ	$\emptyset$	$A = \top$	$C = \top$	$\sigma = \lambda$

Applying the inference rules on $g(x_1)$, we get:

ϵ	$g(x_1)$	$A = \top$	$C = \top$	$\sigma = \lambda$
Abstract				
1	$g(X_1)$	$A = (x_1\downarrow = X_1)$	$C = (g(x_1) > x_1)$	$\sigma = \lambda$
Narrow				
1.1	$h(X_1)$	$A = (x_1\downarrow = X_1)$	$C = (g(x_1) > x_1)$	$\sigma = Id$
Stop				
1.1	$\emptyset$	$A = (x_1\downarrow = X_1)$	$C = (g(x_1) > x_1)$	$\sigma = Id$
Shorten				
1	$\emptyset$	$A = (x_1\downarrow = X_1)$	$C = (g(x_1) > x_1)$	$\sigma = \lambda$
Shorten				
ϵ	$\emptyset$	$A = (x_1\downarrow = X_1)$	$C = (g(x_1) > x_1)$	$\sigma = \lambda$

Example 6:

Let us consider the following rewrite system $\mathcal{R}$, which is not innermost terminating but is weakly innermost terminating.

$$\begin{array}{ll} f(g(x)) & \rightarrow f(h(x)) \\ f(h(x)) & \rightarrow f(g(x)) \\ f(x) & \rightarrow i(x). \end{array}$$

We prove the weak innermost termination of this system on $\mathcal{T}(\mathcal{F}, \mathcal{X})$, where $\mathcal{F} = \{f : 1, g : 1, h : 1, i : 1, a : 0\}$.

Applying the inference rules on $f(x)$, we get:

ϵ	$f(x)$	$A = \top$	$C = \top$	$\sigma = \lambda$
Abstract				
1	$f(X)$	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = \lambda$
Narrow				
1.1	$i(X)$	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = Id$
Stop				
1.1	$\emptyset$	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = Id$
Shorten				
1	$\emptyset$	$A = (x\downarrow = X)$	$C = (f(x) > x)$	$\sigma = \lambda$
Shorten				
ϵ	$\emptyset$	$A = \top$	$C = \top$	$\sigma = \lambda$

Example 7:

Let us consider the following rewrite system $\mathcal{R}$, which is neither terminating nor innermost terminating but weakly innermost terminating.

$$\begin{array}{lll} quot(0, s(y), s(z)) & \rightarrow 0 & (1) \\ quot(s(x), s(y), z) & \rightarrow quot(x, y, z) & (2) \\ quot(x, 0, s(z)) & \rightarrow s(quot(x, s(z), s(z))) & (3) \\ quot(0, y, 0) & \rightarrow quot(y, 0, 0) & (4) \\ quot(s(x), 0, 0) & \rightarrow s(x) & (5) \\ quot(0, 0, 0) & \rightarrow 0 & (6) \end{array}$$

The only constant of $\mathcal{R}$ is the constructor symbol 0, hence terminating. Applying the rules on $quot(x_1, x_2, x_3)$, we get :

ϵ	$quot(x_1, x_2, x_3)$	$A = \top$ $C = \top$ $\sigma = \lambda$
Abstract		
1	$quot(X_1, X_2, X_3)$	$A = (x_1{\downarrow} = X_1 \wedge x_2{\downarrow} = X_2 \wedge x_3{\downarrow} = X_3)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = \lambda$
Narrow		
1.1	0	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = s(X_4) \wedge x_3{\downarrow} = s(X_5))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = 0 \wedge X_2 = s(X_4) \wedge X_3 = s(X_5))$
1.2	$quot(X_6, X_7, X_3)$	$A = (x_1{\downarrow} = s(X_6) \wedge x_2{\downarrow} = s(X_7) \wedge x_3{\downarrow} = X_3)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = s(X_6) \wedge X_2 = s(X_7))$
1.3	$s(quot(X_1, s(X_8), s(X_8)))$	$A = (x_1{\downarrow} = X_1 \wedge x_2{\downarrow} = 0 \wedge x_3{\downarrow} = s(X_8))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_2 = 0 \wedge X_3 = s(X_8))$
1.4	$quot(X_2, 0, 0)$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = X_2 \wedge x_3{\downarrow} = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = 0 \wedge X_3 = 0)$
1.5	$s(X_1)$	$A = (x_1{\downarrow} = s(X_9) \wedge x_2{\downarrow} = 0 \wedge x_3{\downarrow} = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = s(X_9) \wedge X_2 = 0 \wedge X_3 = 0)$
Stop	(*on three branches*)	
1.1	$\emptyset$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = s(X_4) \wedge x_3{\downarrow} = s(X_5))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = 0 \wedge X_2 = s(X_4) \wedge X_3 = s(X_5))$
1.2	$\emptyset$	$A = (x_1{\downarrow} = s(X_6) \wedge x_2{\downarrow} = s(X_7) \wedge x_3{\downarrow} = X_3)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3, quot(X_6, X_7, X_3))$ $\sigma = (X_1 = s(X_6) \wedge X_2 = s(X_7))$
1.3	$s(quot(X_1, s(X_8), s(X_8)))$	$A = (x_1{\downarrow} = X_1 \wedge x_2{\downarrow} = 0 \wedge x_3{\downarrow} = s(X_8))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_2 = 0 \wedge X_3 = s(X_8))$
1.4	$quot(X_2, 0, 0)$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = X_2 \wedge x_3{\downarrow} = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = 0 \wedge X_3 = 0)$
1.5	$\emptyset$	$A = (x_1{\downarrow} = s(X_9) \wedge x_2{\downarrow} = 0 \wedge x_3{\downarrow} = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = s(X_9) \wedge X_2 = 0 \wedge X_3 = 0)$

Stop applies on states 1.1 and 1.5 because each term t of these states is not narrowable and contains only NF-variable (if any), which ensures that any ground instance of t is in normal form.

Stop also applies on state 1.2 if we choose for the induction ordering a simplification ordering based on the precedence $quot \succ_{\mathcal{F}} s, 0$. Indeed, $\mathcal{R}$ is well-covered, i.e. any normal form in $\mathcal{R}$ is composed only of constructor symbols. With the precedence chosen above, we then get $\theta(t) \succeq \theta(t{\downarrow})$, for any term t and ground instance θ with $Dom(\theta) \subseteq \mathcal{V}ar(t)$. Under these assumptions, for any θ satisfying $A = (x_1{\downarrow} = s(X_6) \wedge x_2{\downarrow} = s(X_7) \wedge x_3{\downarrow} = X_3)$, we have $\theta x_1 \succeq \theta s(X_6)$, $\theta x_2 \succeq \theta s(X_7)$ and $\theta x_3 \succeq \theta X_3$. By subterm property of the ordering, and because it is stable by instanciation, we get $\theta s(X_6) \succ \theta X_6$ and, by transitivity, $\theta x_1 \succ \theta X_6$. Likewise, we get $\theta x_2 \succ \theta X_7$. By $\mathcal{F}$-stability of the ordering, we finally have $\theta quot(x_1, x_2, x_3) \succ \theta quot(X_6, X_7, X_3)$, for any substitution θ satisfying A. Therefore the induction hypothesis applies on $quot(X_6, X_7, X_3)$.

Narrow

1.1	$\emptyset$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = s(X_4) \wedge x_3{\downarrow} = s(X_5))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = 0 \wedge X_2 = s(X_4) \wedge X_3 = s(X_5))$
1.2	$\emptyset$	$A = (x_1{\downarrow} = s(X_6) \wedge x_2{\downarrow} = s(X_7) \wedge x_3{\downarrow} = X_3)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3, quot(X_6, X_7, X_3))$ $\sigma = (X_1 = s(X_6) \wedge X_2 = s(X_7))$
1.3.1	$s(0)$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = 0 \wedge x_3{\downarrow} = s(X_8))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = 0)$
1.3.2	$s(quot(X_9, X_8, s(X_8)))$	$A = (x_1{\downarrow} = s(X_9) \wedge x_2{\downarrow} = 0 \wedge x_3{\downarrow} = s(X_8))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = s(X_9))$
1.4.1	$quot(0, 0, 0)$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = 0 \wedge x_3{\downarrow} = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_2 = 0)$
1.4.2	$s(X_{10})$	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = s(X_{10}) \wedge x_3{\downarrow} = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_2 = s(X_{10}))$
1.4.3	0	$A = (x_1{\downarrow} = 0 \wedge x_2{\downarrow} = 0 \wedge x_3{\downarrow} = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_2 = 0)$
1.5	$\emptyset$	$A = (x_1{\downarrow} = s(X_9) \wedge x_2{\downarrow} = 0 \wedge x_3{\downarrow} = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = s(X_9) \wedge X_2 = 0 \wedge X_3 = 0)$

The *abstract* step is empty, since we have no possible abstraction positions.
Indeed, on the term $t = s(quot(X_1, s(X_8), s(X_8)))$ of the state 1.3, the only position candidate for abstraction is 1 (since for any suffix position p we have ground instances $t|_p$ in normal form), and $t|_1 = quot(X_1, s(X_8), s(X_8))$ cannot be abstracted here.
As for the term $t = quot(X_2, 0, 0)$ of the state 1.4, each of its subterm $t|_i$ is such that its ground instances are in normal form, and then does not need to be abstracted.

According to the strategy S of application of inference rules, **Narrow** applies and yields the new states 1.3.1, 1.3.2, 1.4.1, 1.4.2 and 1.4.3.

Stop (*on four branches*)

1.1	$\emptyset$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = s(X_4) \wedge x_3\downarrow = s(X_5))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = 0 \wedge X_2 = s(X_4) \wedge X_3 = s(X_5))$
1.2	$\emptyset$	$A = (x_1\downarrow = s(X_6) \wedge x_2\downarrow = s(X_7) \wedge x_3\downarrow = X_3)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3, quot(X_6, X_7, X_3))$ $\sigma = (X_1 = s(X_6) \wedge X_2 = s(X_7))$
1.3.1	$\emptyset$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = 0 \wedge x_3\downarrow = s(X_8))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = 0)$
1.3.2	$\emptyset$	$A = (x_1\downarrow = s(X_9) \wedge x_2\downarrow = 0 \wedge x_3\downarrow = s(X_8))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3, s(quot(X_9, X_8, s(X_8))))$ $\sigma = (X_1 = s(X_9))$
1.4.1	$quot(0, 0, 0)$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = 0 \wedge x_3\downarrow = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_2 = 0)$
1.4.2	$\emptyset$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = s(X_{10}) \wedge x_3\downarrow = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_2 = s(X_{10}))$
1.4.3	$\emptyset$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = 0 \wedge x_3\downarrow = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_2 = 0)$
1.5	$\emptyset$	$A = (x_1\downarrow = s(X_9) \wedge x_2\downarrow = 0 \wedge x_3\downarrow = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = s(X_9) \wedge X_2 = 0 \wedge X_3 = 0)$

Stop applies on states 1.3.1 and 1.4.2 because each term t of these states is such that any of its ground instance is in normal form. **Stop** applies on the state 1.3.2 using the same reasoning as the one we got to apply **Stop** on the state 1.2 above.

Cut

1.1	$\emptyset$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = s(X_4) \wedge x_3\downarrow = s(X_5))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = 0 \wedge X_2 = s(X_4) \wedge X_3 = s(X_5))$
1.2	$\emptyset$	$A = (x_1\downarrow = s(X_6) \wedge x_2\downarrow = s(X_7) \wedge x_3\downarrow = X_3)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3, quot(X_6, X_7, X_3))$ $\sigma = (X_1 = s(X_6) \wedge X_2 = s(X_7))$
1.3.1	$\emptyset$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = 0 \wedge x_3\downarrow = s(X_8))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = 0)$
1.3.2	$\emptyset$	$A = (x_1\downarrow = s(X_9) \wedge x_2\downarrow = 0 \wedge x_3\downarrow = s(X_8))$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3, s(quot(X_9, X_8, s(X_8))))$ $\sigma = (X_1 = s(X_9))$
1.4.1	$\emptyset$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = 0 \wedge x_3\downarrow = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_2 = 0)$
1.4.2	$\emptyset$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = s(X_{10}) \wedge x_3\downarrow = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_2 = s(X_{10}))$
1.4.3	$\emptyset$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = 0 \wedge x_3\downarrow = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_2 = 0)$
1.5	$\emptyset$	$A = (x_1\downarrow = s(X_9) \wedge x_2\downarrow = 0 \wedge x_3\downarrow = 0)$ $C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$ $\sigma = (X_1 = s(X_9) \wedge X_2 = 0 \wedge X_3 = 0)$

Cut applies on state 1.4.1, because the LNS $\sigma = (X_2 = 0)$ of this state is the same as the LNS of the successful state 1.4.3. From now on, we have only successful states and **Shorten** applications are going to reduce the proof tree to a single successful state.

Shorten

1.1	$\emptyset$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = s(X_4) \wedge x_3\downarrow = s(X_5))$
		$C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$
		$\sigma = (X_1 = 0 \wedge X_2 = s(X_4) \wedge X_3 = s(X_5))$
1.2	$\emptyset$	$A = (x_1\downarrow = s(X_6) \wedge x_2\downarrow = s(X_7) \wedge x_3\downarrow = X_3)$
		$C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3, quot(X_6, X_7, X_3))$
		$\sigma = (X_1 = s(X_6) \wedge X_2 = s(X_7))$
1.3	$\emptyset$	$A = (x_1\downarrow = X_1 \wedge x_2\downarrow = 0 \wedge x_3\downarrow = s(X_8))$
		$C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$
		$\sigma = (X_2 = 0 \wedge X_3 = s(X_8))$
1.4	$\emptyset$	$A = (x_1\downarrow = 0 \wedge x_2\downarrow = X_2 \wedge x_3\downarrow = 0)$
		$C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$
		$\sigma = (X_1 = 0 \wedge X_3 = 0)$
1.5	$\emptyset$	$A = (x_1\downarrow = s(X_9) \wedge x_2\downarrow = 0 \wedge x_3\downarrow = 0)$
		$C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$
		$\sigma = (X_1 = s(X_9) \wedge X_2 = 0 \wedge X_3 = 0)$

Shorten

1	$\emptyset$	$A = (x_1\downarrow = X_1 \wedge x_2\downarrow = X_2 \wedge x_3\downarrow = X_3)$
		$C = (quot(x_1, x_2, x_3) > x_1, x_2, x_3)$
		$\sigma = \lambda$

Shorten

ϵ	$\emptyset$	$A = \top$
		$C = \top$
		$\sigma = \lambda$