

Structures for abstract rewriting

Marc Aiguier (aiguier@lami.univ-evry.fr)

¹Université d'Évry, LaMI CNRS UMR 8042,
523 pl. des Terrasses F-91000 Évry

Diane Bahrami (diane.bahrami@cea.fr)

CEA/LIST Saclay
F-91191 Gif sur Yvette Cedex

Abstract. When rewriting is used to generate convergent and complete rewrite systems in order to answer the validity problem for some theories, all the rewriting theories rely on a same set of notions, properties and methods. Rewriting techniques have mainly been used to answer the validity problem of equational theories, that is to compute congruences. However, recently, they have been extended in order to be applied to other algebraic structures such as pre-orders and orders. In this paper, we investigate an abstract form of rewriting, by following the paradigm of “*logical-system independency*”. To achieve this purpose, we provide a few simple conditions (or axioms) under which rewriting (and then the set of classical properties and methods) can be modeled, understood, studied, proven and generalized. This enables us to extend rewriting techniques to other algebraic structures than congruences and pre-orders such as congruences closed under monotonicity and modus-ponens. Finally, we introduce convergent rewrite systems that enable one to describe deduction procedures for their corresponding theory, and propose a Knuth-Bendix style completion procedure in this abstract framework.

Keywords: rewrite system, abstract rewriting, axiomatization, abstract deduction procedure, abstract completion procedure.

1. Introduction

1.1. MOTIVATION

One of the main purposes of rewriting is the generation of convergent and complete rewrite systems which can be used to automatically prove the validity of formulæ in some theories [16]. This technique has mainly been applied to answer the validity problem for equational theories, that is, to compute congruences (e.g. [6, 21] for surveys). This research activity started with the famous completion algorithm designed by Knuth and Bendix [36]. This algorithm provides for any equational theory, when it does not fail ¹, a rewrite system which is both terminating and confluent. Moreover, the equational theory and the rewrite system are proof theoretically equivalent. However, rewriting has recently been applied to other algebraic structures such as preorders and

¹ Unfailing completion procedures have also been defined [9, 32]. Unfailing completion procedures are programs that may terminate with a non-empty set of equalities, and only produce ground confluences.



orders [11, 39, 48–50, 53–55]. These researches are the consequence of a negative result. This negative result states that it is impossible to generate by the Knuth-Bendix algorithm a rewrite system equivalent to the equational presentation of lattice theory [27], although Withman’s algorithm has solved the word problem for this theory when represented under its ordered form. These works have then defined rewriting theories to solve the word problem of theories manipulating formulæ of the form $t \prec t'$ where \prec is any preorder: inclusion, subtyping, etc. J. Levy and J. Agustí opened this research field by applying rewriting to all pre-orders [39]. The difference with standard rewriting is that, when we have to orient a non-symmetric relation \sqsubseteq according to a reduction order \geq (i.e. a Noetherian order on terms), two rewrite relations are needed: both the intersection of \sqsubseteq and \geq , written $\rightarrow_{\sqsubseteq}$ and the intersection of \sqsubseteq and \leq , written \leftarrow_{\sqsubseteq} . This is due to the non-symmetry of \sqsubseteq : $\rightarrow_{\sqsubseteq} \neq (\leftarrow_{\sqsubseteq})^{-1}$. Such a pair of rewrite relations is called a *bi-rewrite system*. From these pioneer works, two generalizations have been proposed. G. Struth studied operational rewriting for any non-symmetric transitive relation [53, 55]. He generalized bi-rewrite systems to any pair of non-symmetric transitive relations A and B , and studied rewriting as a general theory of commutation from their composition. Then, he applied it to lattice theory [54]. M. Schorlemmer pushed on a bit further by defining a variant of the classical logic of first order predicates restricted to binary relations in which generalizations of Leibniz’s law (such as transitivity or typing) can be specified by using both the composition of binary relations and the set-theoretical inclusion. He studied rewriting on this logic in [50, 48, 49].

The question is: *can rewriting be extended to a larger class of algebraic structures than congruences, preorders or more generally the composition of binary relations?* To answer this question, we propose a general framework of rewriting by applying the paradigm of “*logical-system independency*”, that is, by providing a general framework and conditions (axioms), and by adapting and proving, within this general framework, classical definitions and results which underlie rewriting. The interest here is simple. From the study of all rewriting theories, we can observe that the same set of notions and results underlies rewriting. These main notions and results are the following:

- the way to define *good proofs* and *proofs to simplify*. In the equational rewriting setting, “good proofs” are valleys (i.e. elements of $\overset{*}{\rightarrow} \circ \overset{*}{\leftarrow}$) and “proofs to simplify” are peaks (i.e. elements of $\overset{*}{\leftarrow} \circ \overset{*}{\rightarrow}$).
- the result which states that any proof (i.e. defined by a combination of good proofs and proofs to simplify) can be identified with a good proof ²

² In the equational rewriting setting, this inclusion is called the *Church-Rosser property*, and is expressed by: $(\overset{*}{\leftarrow} \circ \overset{*}{\rightarrow}) \subseteq (\overset{*}{\rightarrow} \circ \overset{*}{\leftarrow})$

provided that proofs to simplify are identified with good proofs³, and conversely. We will name this result *Church-Rosser's result*⁴.

- the result which states that proofs to simplify can be eliminated and replaced by good proofs, step by step, by reducing basic proofs to simplify⁵ provided that rewrite systems are terminating, and that this process is terminating. This last result is well-known as *Newman's lemma*.
- the possibility to define for any Church-Rosser and terminating rewrite system, a decision procedure for its corresponding theory,
- and, the possibility to define a completion procedure which generates convergent rewrite systems for theories, when it does not fail.

Moreover, rewriting is the main technique used for prototyping algebraic specifications, and many new algebraic formalisms are (and will be) defined to answer some specific questions related to the activity of formal specification (observability, exception-handling, dynamic data-types, etc.). Hence, in order to be able to prototype (algebraic) specifications, one does not only need to define new formalisms, but one also has to adapt these classical notions, and to show that these fundamental results remain true for such formalisms. Up to now, this kind of approach - i.e. the study of some properties in the paradigm of “logical-system independency” - has been widely applied to semantic aspects of algebraic formalisms [25, 29, 47] and to theorem deduction [26, 44]. But as far as we know, operational aspects of algebraic formalisms (here represented by rewriting) have not received attention at this abstract level. Therefore, it is useful to provide an axiomatization of rewriting allowing one to generalize results which are well known for some specific formalisms. This is what we propose to do here. In this paper, we will then study rewriting in a generic way and propose a generalized form of usual results which underlie rewriting such as Church-Rosser's result and Newman's lemma. Moreover, given a convergent rewrite system (according to our new definition), we will also define a decision procedure for its corresponding theory. Finally, we will define a Knuth-Bendix style completion method in this generic framework with all expected results for it (mainly its correctness). As a result of the rewriting abstraction defined in the paper, all the results as well as the decision procedure and completion method established here are *de facto* generalizations of standard ones we find in different rewriting theories.

³ In the equational rewriting setting, this inclusion is called the *confluence property* and is expressed as follows: $\leftarrow \circ \rightarrow \subseteq \overset{*}{\rightarrow} \circ \overset{*}{\leftarrow}$

⁴ This name must not be confused with the so-called *Church-Rosser theorem*, which states the confluence of β -reduction in λ -calculus.

⁵ In the equational rewriting setting, this inclusion is called the *local confluence property* and is expressed as follows: $\leftarrow \circ \rightarrow \subseteq \overset{*}{\rightarrow} \circ \overset{*}{\leftarrow}$

Actually, all rewriting theories that we know satisfy the axioms given in this paper, and thus enter our framework (see the numerous examples developed in the paper).

1.2. RELATED WORK

In abstract rewriting, also called abstract reduction systems, a considerable amount of theory has been developed covering these basic topics (Church-Rosser's result, Newman's lemma, etc.). However, all results developed in abstract reduction systems concern congruences usually called Thue congruences. Here, we propose to apply rewriting techniques to compute over a class of algebraic structures larger than congruences and preorders such as congruences closed under monotonicity and modus-ponens (see Section 9 of this paper). Moreover, the idea of axiomatizing rewriting is not new and has been pursued with success, especially in the world of λ -calculus. Such axiomatizations deal with algebraic structures which are also congruences. The main works in this area are J.-J. Lévy's residual theory [40] and its extension by P.-A. Melliès [42]. In residual theory, the structure of the objects to be rewritten is abstracted through the redex notion (i.e. a place in rewritten objects which can be reduced by a rewrite rule). In this setting, many key properties of λ -calculus or of more general rewrite systems have been generalized, such as Church-Rosser's theorem [40, 42], the standardization theorem [30] or the stability theorem [43]. The main goal of these works was not to generate convergent and complete rewrite systems which answer the validity problem. However, we share the *axiomatic method* with them, that is we formulate through axioms a small number of simple properties which are shared by the different settings and which are needed to yield the fundamental results mentioned above. Finally, concerning the completion process, we can cite N. Dershowitz and C. Kirchner's work [22, 15] which generalizes the proof-ordering method to an abstract setting of arbitrary formal systems. This last work places itself downstream with respect to our work, and then completes it, in the sense that [22, 15] fix inference and the ordering on proofs whereas we give axioms to build such an ordering (see Proposition 6.11, Corollary 6.12, and Theorem 8.8). Actually, [15, 22] aim to give an abstract form to completion processes whereas we are interested in rewriting from every angle.

1.3. STRUCTURE OF THE PAPER

The paper is organized as follows: in Section 2, we recall standard notations about formal systems, theorem deduction and proof trees. In Section 3, we instantiate formal systems in order to deal with binary relations. Resulting formal systems will be called *rewriting formal systems*. In Sections 4 and 5,

we develop a generic framework for rewriting. In this framework, we adapt the standard definitions of rewrite system, rewriting step, derivation, termination, effluence (usually called *peak* in the equational rewriting setting) and proof by rewriting (usually called *valley*). Moreover, we give, for any abstract rewrite system, a decision procedure for its corresponding theory, and show its correctness and completeness with respect to the underlying theory. Section 6 gives four simple conditions (axioms) in order to obtain an abstract formulation of Church-Rosser's result and Newman's lemma. Therefore, our generic framework provides a basis for an abstract completion that is presented in Section 8. In Section 7, we extend our abstract framework in order to deal with rewriting modulo theories. Section 9 instantiates our abstract framework of rewriting for two logics: equational conditional logic and M. Schorlemmer's logic of special relations [49]. Conditional rewriting has been intensively studied in the literature [13, 24, 28, 34, 35] but only in order to answer the validity problem of unconditional equations (this example will be also instantiated in this paper to illustrate the definitions given in the paper). In the standard conditional rewriting setting, rewrite rules have conditions. Here, we will also include conditions in the rewriting process.⁶ Logic of special relations is an important example because it is presented as a logical framework in which many standard algebraic formalisms (equational, conditional, etc.) and non-standard ones (inclusion, preorder, equational typed logic, rewriting logic, etc.) can be encoded. We will show that the logic of special relations can be encoded as a rewriting formal system. This will enable us to obtain all the classical results such as presented by M. Schorlemmer [49], and to give a completion method devoted to this logic. As far as we know, completion had not been studied in this logic.

2. Preliminaries and notations

A *formal system* (a so-called *calculus*) $S = (F, R)$ over an alphabet A consists of a set F of strings over A (i.e. $F \subseteq A^*$), called *formulae*, and a finite set R of computable n -ary relations on F , called *inference rules*. Thus, a rule with arity n ($n \geq 1$) is a set of tuples $(\varphi_1, \dots, \varphi_n)$ of strings of F . Each sequence $(\varphi_1, \dots, \varphi_n)$ belonging to a rule r of R is called an *instance* of that rule with *premises* $\varphi_1, \dots, \varphi_{n-1}$ and *conclusion* φ_n . It is usually written $\frac{\varphi_1 \dots \varphi_{n-1}}{\varphi_n}$. A rule instance ι of R with conclusion φ is denoted by $\iota : \varphi$ and $\mathcal{L}(\iota)$ is the multiset of its premises. A *deduction* in S from a set of formulae Γ of F is a finite sequence (ψ_1, \dots, ψ_m) of formulae such that $m \geq 1$ and, for all i , $1 \leq i \leq m$,

⁶ Actually, this form of conditional rewriting is obviously included in works on superposition-based theorem proving for full first-order clauses with equality (e.g. see [10, 45]). An interesting work would be to compare these works with the conditional rewriting as developed in Section 9.

either ψ_i is an element of Γ or there is an instance $\frac{\varphi_1 \dots \varphi_n}{\psi_i}$ of a rule in \mathcal{S} where $\{\varphi_1, \dots, \varphi_n\} \subseteq \{\psi_1, \dots, \psi_{i-1}\}$. A *theorem* from a set of formulæ Γ in \mathcal{S} is a formula φ such that there exists a deduction in \mathcal{S} from Γ with φ as last element. This is usually denoted by $\Gamma \vdash \varphi$. Instances can also be composed to build *proof trees*. Thus, we obtain another way to denote deductions in formal systems. Formally, a *proof tree* π in a formal system \mathcal{S} is a finite tree whose nodes are labelled with formulæ of F in the following way: if a non-leaf node is labelled with φ_n and its predecessor nodes are labelled (from left to right) with $\varphi_1, \dots, \varphi_{n-1}$, then $\frac{\varphi_1 \dots \varphi_{n-1}}{\varphi_n}$ is an instance of a rule of \mathcal{S} . The previous notations on rule instances can be extended to proof trees: a proof tree π with root φ is denoted by $\pi : \varphi$, and $\mathcal{L}(\pi)$ is the multiset of its leaves. We denote by $\pi = (\pi_1, \dots, \pi_n, \varphi)_\iota$, with $n \in \mathbb{N}$, the proof tree whose the last inference rule is $\iota = \frac{\varphi_1 \dots \varphi_n}{\varphi}$ and such that, for every i , $1 \leq i \leq n$, π_i is the subtree of π leading to φ_i . Obviously, for any statement of the form $\Gamma \vdash \varphi$ in a formal system \mathcal{S} , there is an associated proof tree $\pi : \varphi$ whose leaves are axioms or formulæ from Γ . Two proof trees $\pi : \varphi$ and $\pi' : \varphi$ are *equivalent* with respect to a set of formulæ Γ if and only if both are associated to $\Gamma \vdash \varphi$.

Using a standard numbering of the tree nodes by natural number strings, we can refer to positions in a proof tree. Thus, given a proof tree π , a *position* of π is a string w on \mathbb{N} which represents the path from the root of π to the subtree at that position. This subtree is denoted by $\pi|_w$. Given a position $\omega \in \mathbb{N}^*$ in a proof tree π , $\pi[\pi']_\omega$ is the proof tree obtained from π by replacing the subtree $\pi|_\omega$ by π' . The trees $\pi|_\omega$ and π' necessarily have the same root. If π and $\pi' : \varphi$ are two proof trees and w is a leaf position of π such that $\pi|_w = \varphi$, then we use the expression $\pi \cdot_w \pi'$ rather than $\pi[\pi']_w$. This operation is called *composition* of π and π' on (leaf) position w . If R is a set of n -ary rules then R^\sharp is the set of proof trees inductively constructed from all rule instances in R , and closed under the composition operation.

3. Rewriting formal systems

3.1. DEFINITION

Rewriting is a method to reason with binary relations (equality [6, 21], inclusion [39] or other non-symmetric relations [11, 55], the ideal membership problem [17], etc.). These binary relations, contained in the set E in Definition 3.1 below, are defined on sets of elements that can be different from one rewriting theory to another (simple words, λ -terms, first order terms, graphs, etc.). Moreover, the behavior of these binary relations is specified by inference rules. For example, in the equational setting, the behavior of equality is specified by the reflexivity, transitivity and symmetry rules. If we extend to term equations, we add both context and substitution rules. We can then

notice that, in all rewriting theories, rewriting relations are specified thanks to a subset of these inference rules (e.g. substitution, context, reflexivity and transitivity), and then some of these inference rules are removed from the process (e.g. symmetry). Moreover, preserved inference rules can be split up into two disjoint sets, that we call RS and De , specifying rewriting steps and derivations, respectively. Removed inference rules will be put in the set Rmv . Rule instances of Rmv are removed because they generate basic loops in the rewriting process, and then lead to nonterminating rewrite relations. Thus, we propose the following general framework for rewriting, which applies to many settings.

DEFINITION 3.1 (Rewriting formal systems). *A rewriting formal system (rfs) is a 5-tuple $SP = (T, E, RS, De, Rmv)$ such that T is a set, E is a set of binary relations⁷ on T , and RS , De and Rmv are three disjoint sets of n -ary relations on the set F defined by: $F = \{p(u, v) \mid p \in E \wedge (u, v) \in p\}$.*

The set E in Definition 3.1 is the set of syntactically well-formed statements. Not in any ways, does this mean that these statements are true or false. For instance, in the rfs associated to the mono-sorted equational logic presented in Example 3.6 below, for the signature $\Sigma = \{0, succ^1\}$, E will contain equations of the form $succ^n(0) = succ^n(0)$, but also equations $succ^n(0) = succ^m(0)$ with $m \neq n \in \mathbb{N}$.⁸

REMARK 3.2. *The couple $(F, RS \cup De \cup Rmv)$ defines a formal system over the alphabet $A = E \cup T \cup \{(\cdot)\}$, according to the definition of Section 2.*

3.2. SEMI-COMMUTATION PROPERTY

We will see in the following that the division of the set of inference rules into the three sets RS , De and Rmv defines a search proof strategy which restricts the search proof space by selecting proof trees equipped with the following structure: RS 's rule instances are always above both De 's rule instances and Rmv 's rule instances. An important property to check is the completeness of the strategy, i.e., for every statement $\Gamma \vdash p(u, v)$ there exists a proof tree satisfying the above form. Classically, completeness is obtained by defining basic proof tree transformations (see the example of such a transformation just below). In [1–3, 12], we have studied how to define the three sets RS , De and Rmv from a generalization of these transformations that we have called *semi-commutation*. Let us recall briefly this work in this section.⁹ This notion

⁷ For any $p \in E$, we will use p both for the relation and for the symbol naming it.

⁸ $succ^n(0)$ is the ground term $\underbrace{succ(\dots(succ(0))\dots)}_{n \text{ times}}$.

⁹ For more explanations, we refer interested readers to our papers [2, 3].

of “semi-commutation” allows us to divide the set of inference rules into two sets Up and $Down$. Rules in Up and $Down$ are used to build rewriting steps and derivations, respectively. More precisely, Rmv will contains all the rules in Up and $Down$ that generate “basic looping schemata” in rewriting such as symmetry,¹⁰ and then $RS = Up \setminus Rmv$, and $De = Down \setminus Rmv$.

The notion of “semi-commutation” generalizes how some inference rules can “go over” other ones. For instance, in the equational logic (a complete presentation of this logic is given in Example 3.6 below) we have the following transformation:

$$\frac{\frac{t_1=t_2 \quad t_2=t_3}{t_1=t_3}}{C[t_1] = C[t_3]} \rightsquigarrow \frac{\frac{t_1=t_2}{C[t_1]=C[t_2]} \quad \frac{t_2=t_3}{C[t_2]=C[t_3]}}{C[t_1] = C[t_3]}$$

However, this does not hold for the tree

$$\frac{\frac{t_1=t_2}{C[t_1]=C[t_2]} \quad C[t_2] = t_3}{C[t_1] = t_3}$$

because it is not possible to transform such a tree in order to put transitivity above context. The reason is that we have no information on the structure of the term t_3 . The same transformations hold for context with symmetry and for substitution with symmetry and transitivity. Moreover, all inference rules are erased when they occur under reflexivity. Hence, it is sensible to put context and substitution in Up , and transitivity, symmetry and reflexivity in $Down$. Formally, the semi-commutation notion is defined as follows:

DEFINITION 3.3 (Structured proof tree). *Let T be a set. Let E be a set of binary relations on T . Let (F, R) be a formal system where $F = \{p(u, v) \mid p \in E \wedge (u, v) \in p\}$. Let $Up = (Up_p)_{p \in E}$ and $Down = (Down_p)_{p \in E}$ be two E -indexed non empty sets of rule instances of R such that for every $p \in E$, Up_p and $Down_p$ form a partition of the whole set of rule instances of the form $\frac{\Phi_1 \dots \Phi_n}{p(u, v)}$.*

Note $\mathcal{P}r_{Up > Down}$ the least set (according to the set-theoretical inclusion) inductively defined as follows:

- $Up^\# \cup Down \subseteq \mathcal{P}r_{Up > Down}$ ¹¹
- Let $\iota = \frac{\Phi_1 \dots \Phi_n}{p(u, v)} \in Down_p$ and let $(\pi : \Phi_i)_{1 \leq i \leq n}$ be a finite sequence of proof trees such that for every i , $1 \leq i \leq n$, $\pi_i \in \mathcal{P}r_{Up > Down}$. Then, $(\pi_1 : \Phi_1, \dots, \pi_n : \Phi_n, p(u, v))_\iota \in \mathcal{P}r_{Up > Down}$.

¹⁰ This notion of “basic looping schema” has been defined in D. Bahrami’s Phd manuscript [12] and in [2].

¹¹ See Section 2 for the definition of $_^\#$.

DEFINITION 3.4 (Semi-commutation). *With the notations of Definition 3.3, Up semi-commutes with $Down$ if and only if for every proof tree π of the form $(\iota_1 : \varphi_1, \dots, \iota_n : \varphi_n, \varphi)_\iota$ such that:*

- $\iota \in Up$
- $\forall 1 \leq i \leq n, \iota_i \in F \cup R$,
- $\exists 1 \leq i \leq n, \iota_i \in Down$

there exists an equivalent proof tree $\pi' : \varphi$ in $Pr_{Up > Down}$ with respect to $\mathcal{L}(\pi)$. We note $\pi \rightsquigarrow \pi'$ such a property between π and π' .

Hence, the semi-commutation property generates proof tree transformations defined as the closure of \rightsquigarrow under proof tree context and composition on leaf position (see the definition in Section 2). When these transformations are terminating then the completeness of the strategy holds. All the interest of the strategy defined by the division of inference rules into Up and $Down$ lies in this property. In [2, 3], we have provided a general framework under which the termination of \rightsquigarrow can be unified and generalized.

Definition 3.4 does not ensure unicity for the couple $(Up, Down)$ (when it exists). For instance, in the equational logic setting, many couples $(Up_=, Down_=)$ can be candidate. Let us suppose that $Up_=$ contains all rule instances except the instance of the transitivity rule $\frac{\frac{t_1=t_2 \quad t_2=t_3}{t_1=t_3} \quad t_3=t_4}{t_1=t_4}$ for given terms t_1, t_2 and t_3 (which is then the only instance of $Down_=$). We easily show that all the appropriate instances of symmetry, context and substitution rules semi-commute with this instance (see [3]). Moreover, we have:

$$\frac{\frac{\frac{t_1=t_2 \quad t_2=t_3}{t_1=t_3} \quad t_3=t_4}{t_1=t_4}}{t_1=t_4} \rightsquigarrow \frac{t_1=t_2 \quad \frac{t_2=t_3 \quad t_3=t_4}{t_2=t_4}}{t_1=t_4}.$$

Consequently, $Up_=$ semi-commutes with $Down_=$.

However, for many logics (anyway, all logics used in computer science and mathematics) the set of rule instances is represented by a finite set of rule schemata (up to meta-variable renaming), that is a single form with infinitely many instantiations. We take advantage of this to give a choice strategy for both sets: we put all the instances of a same rule schema in the same set (either Up or $Down$). This also makes the study of semi-commutation easier. Instead of checking the semi-commutation property on an infinite set of instances, we do it on the finite set of rule schemata. Going further in the implementation of this strategy would necessarily require two things:

1. formalizing the notion of rule schemata. But this turns out to be a difficult task. This is mainly due to heterogeneity of meta-variable meanings

manipulated within rule schemata. Indeed, some of them are interpreted like simple terms, others are formulæ or more complex mathematical objects such as mappings denoting substitutions. Moreover, these variables can be constrained under their form or other criteria (e.g. variables can only be substituted by defined terms, or conditions can be given on the scope of variables in the framework of first order logics). As far as we know, there are no works on a general formalization of rule schemata. Only some works have been defined for some classes of logics (propositional logic [51], first order logic [52]) and in the restricted framework of Hilbert-style calculus.

2. automatically checking the semi-commutation property between instances of *Up* and *Down*. As we have already noticed, the study of the semi-commutation property gives rise to basic proof tree transformation rules. For instance, in the equational logic setting, these basic transformation rules consist in “distributing” replacement, symmetry and substitution over transitivity. These basic transformation rules are usually tedious but easy to define. We can then reasonably hope to automatize their definition, especially because, in most formalisms equipped with a rewriting theory, applying the semi-commutation property consists in “distributing” rules in *Up* over rules in *Down*. It is easy to show that, in this case, transformations generated by the semi-commutation property are terminating by using proof terms for proofs with a recursive path ordering for the precedence ordering $>$ on $Up \cup Down$ defined by:

$$\iota > \iota' \iff \iota \in RS \wedge \iota' \in Down$$

3.3. EXAMPLES OF REWRITING FORMAL SYSTEMS

EXAMPLE 3.5 (Abstract reduction systems). *The rfs for abstract reduction systems is defined for any alphabet A by the tuple (T, E, RS, De, Rmv) where:*

- $T = A^*$,
- $E = \{\equiv\}$ s.t. $\equiv \stackrel{def}{=} T \times T$ (syntactic definition of word equations, that is all pairs of words are correct syntactic equations),
- RS is the set containing all instances of the following deduction rule:

$$Context \frac{\alpha \equiv \beta}{\delta_1.\alpha.\delta_2 \equiv \delta_1.\beta.\delta_2}$$

- De is the set containing all instances of the two following deduction rules:

$$\text{Reflexivity} \frac{}{\alpha \equiv \alpha} \quad \text{Transitivity} \frac{\alpha \equiv \beta \quad \beta \equiv \delta}{\alpha \equiv \delta}$$

- Rmv is the set of all instances of the following deduction rule:

$$\text{Symmetry} \frac{\alpha \equiv \beta}{\beta \equiv \alpha}$$

EXAMPLE 3.6 (Mono-sorted equational logic). Before defining the rfs for Mono-sorted equational logic, let us recall the basic definitions and notations of this logic. A signature Σ is a set of function names, each one equipped with an arity in \mathbb{N} . A function name f equipped with an arity $n \in \mathbb{N}$ will be noted f^n . Given a set of variables, let us denote by $T_\Sigma(V)$ the set of free terms with generators in V . Given a term $t \in T_\Sigma(V)$, $\text{Var}(t)$ denotes the set of variables occurring in t . A Σ -equation is any sentence of the form $t = t'$ where t and t' are terms in $T_\Sigma(V)$. A substitution is an application $\sigma : V \rightarrow T_\Sigma(V)$. It is naturally extended to terms and equations. The rfs for mono-sorted equational term rewriting is then defined for any signature Σ by the tuple (T, E, RS, De, Rmv) such that:

- $T = T_\Sigma(V)$,
- $E = \{=\}$ s.t. $\stackrel{\text{def}}{=} T \times T$ (i.e. all pairs of equations are syntactically correct),
- RS is the set of all instances of the two following deduction rules: $\sigma : V \rightarrow T_\Sigma(V)$ is a substitution and C is a context (i.e. a term with a unique occurrence of the constant \square), and $C[t]$ denotes the result of replacing in C the occurrence of \square by t

$$\text{Substitution} \frac{t = t'}{\sigma(t) = \sigma(t')} \quad \text{Context} \frac{t = t'}{C[t] = C[t']}$$

- De is the set of all instances of the two following deduction rules:

$$\text{Reflexivity} \frac{}{t = t} \quad \text{Transitivity} \frac{t = t' \quad t' = t''}{t = t''}$$

- Rmv is the set of all instances of the following deduction rule:

$$\text{Symmetry} \frac{t = t'}{t' = t}$$

EXAMPLE 3.7 (Conditional equational logic). *The rfs developed in this example is the logic that underlies the conditional rewriting to answer the validity problem of unconditional equations. Only rewrite rules will have conditions.*

Before defining this rfs let us recall some notions and notations of the conditional equational logic. Signatures, terms with variables, and substitution are defined as in Example 3.6.

Atoms are Σ -equations, and formulae are sentences of the form $\alpha_1 \wedge \dots \wedge \alpha_n \Rightarrow \alpha_{n+1}$ where for every i , $1 \leq i \leq n+1$, α_i is a Σ -equation.

In order to make conditional formulae enter the definition of rfs which only manipulates predicates, any formula of the form $c \Rightarrow t = t'$ where $c = \bigwedge_{1 \leq i \leq n} t_i = t'_i$

(i.e. c is a finite conjunction of equations), will be denoted by $t =_c t'$. Unconditioned equations $t = t'$ will be denoted by $t =_0 t'$. Hence, in the associated rfs, this gives rise to a family of predicates $=_c$ indexed by finite conjunctions, and inference rules will be n -ary relations on such formulae.

Therefore, given a signature Σ , we define the rfs (T, E, RS, De, Rmv) for the conditional equational logic as follows:

- $T = T_\Sigma(V)$,
- $E = \{=_c \mid c : \text{finite conjunction}\}$ s.t. for every c , $=_c \stackrel{\text{def}}{=} T \times T$ (syntactic definition),
- RS is the set of all instances of the following deduction rule: let $\sigma : V \rightarrow T_\Sigma(V)$ be a substitution and let C be a context

$$\text{Replacement/Congruence} \frac{t = \bigwedge_{1 \leq i \leq n} t_i = t'_i \quad \forall 1 \leq i \leq n, \sigma(t_i) =_0 \sigma(t'_i)}{C[\sigma(t)] =_0 C[\sigma(t')]}$$

- De is the set of all instances of the two following deduction rules:

$$\text{Reflexivity} \frac{}{t =_0 t} \quad \text{Transitivity} \frac{t =_0 t' \quad t' =_0 t''}{t =_0 t''}$$

- Rmv is the set of all instances of the following deduction rule:

$$\text{Symmetry} \frac{t =_0 t'}{t' =_0 t}$$

EXAMPLE 3.8 (Multi-sorted equational logic). *For this rfs, a signature Σ is a pair (S, F) where S is a set of sorts and F is a set of function names, each*

one equipped with an arity in $S^* \times S$. A set of variable $V = (V_s)_{s \in S}$ is an S -indexed family of sets. Given a set of variables, let us denote by $T_\Sigma(V) = (T_\Sigma(V)_s)_{s \in S}$ the S -indexed family of sets where for every $s \in S$, $T_\Sigma(V)_s$ is the set of free terms of sort s with generators in V . Σ -equations are any sentence of the form $t = t'$ where there is $s \in S$ such that t and t' are terms in $T_\Sigma(V)_s$. A substitution is an application $\sigma : V \rightarrow T_\Sigma(V)$ such that $\sigma(V_s) \subseteq T_\Sigma(V)_s$. It is naturally extended to terms and equations. The rfs for multi-sorted equational term rewriting is then defined for any signature Σ by the tuple (T, E, RS, De, Rmv) such that:

- $T = T_\Sigma(V)$,
- E is the S -set of binary relations $=_s$ on T such that $=_s \stackrel{def}{=} T_\Sigma(V)_s \times T_\Sigma(V)_s$ (all pairs of terms define equations if both terms are of the same sort),
- RS, De and Rmv are defined as in Example 3.6.

EXAMPLE 3.9 (Non-symmetric transitive logic). *The rfs for arbitrary (non-symmetric) transitive relations is defined as in Example 3.6 except that $E = \{\preceq\}$ such that $\preceq \stackrel{def}{=} T \times T$. Moreover, we assume that every function f^n is monotonic on all its positions with respect to \preceq , that is: $\forall 1 \leq i \leq n, t_i \preceq t'_i \Rightarrow f(t_1, \dots, t_i, \dots, t_n) \preceq f(t_1, \dots, t'_i, \dots, t_n)$.¹² Therefore, the deductive rules are:*

- RS is the set of all instances of the two following deduction rules:
 $\sigma : V \rightarrow T_\Sigma(V)$ is a substitution and C is a context

$$\text{Substitution} \frac{t \preceq t'}{\sigma(t) \preceq \sigma(t')} \quad \text{Monotonicity} \frac{t \preceq t'}{C[t] \preceq C[t']}$$

- De is the set of all instances of both following deduction rules:

$$\text{Reflexivity} \frac{}{t \preceq t} \quad \text{Transitivity} \frac{t \preceq t' \quad t' \preceq t''}{t \preceq t''}$$

- $Rmv = \emptyset$

¹² Otherwise, some restrictions need to be put on the context rule (called “Monotonicity” in this example and given below).

4. Rewrite systems

4.1. BASIC DEFINITION

Rewriting orients binary predicates. We briefly saw in the introduction of this paper that for any symmetric relation p , one rewriting relation suffices because $\rightarrow_p = (\leftarrow_p)^{-1}$, but for any non-symmetric transitive one, two are needed. This was first observed by J. Levy and J. Agustí and gave rise to bi-rewrite systems [39]. In our generalization, since binary predicates in E are not necessarily symmetric, we follow [39] and define rewrite systems as follows:

DEFINITION 4.1 (Rewrite systems). *Let $S\mathcal{P} = (T, E, RS, De, Rmv)$ be an rfs. An $S\mathcal{P}$ -rewrite systems \mathcal{R} is an E -sorted set of pairs of binary relations $(\rightarrow_p, \leftarrow_p)_{p \in E}$ on T such that: $\forall p \in E, \rightarrow_p \cup \leftarrow_p \subseteq p$ (compatibility with the syntactic definition of p given in $S\mathcal{P}$).*

EXAMPLE 4.2. *In the rfs for abstract reduction systems, we can consider the following set of rules from the alphabet $A = \{a, b, c, \dots, z\} \cup \{\epsilon, true, false\}$ which defines the lexicographic order: if ϵ denotes the empty word, then for every α and every β in A^**

$$\begin{aligned}
 \epsilon < \alpha &\rightarrow_{\equiv} true \\
 a < \tau &\rightarrow_{\equiv} true \text{ with } \tau \in \{b, c, d, \dots, z\} \\
 b < \tau &\rightarrow_{\equiv} true \text{ with } \tau \in \{c, d, \dots, z\} \\
 &\vdots \\
 y < z &\rightarrow_{\equiv} true \\
 \alpha < \epsilon &\rightarrow_{\equiv} false \\
 \tau < a &\rightarrow_{\equiv} false \text{ with } \tau \in \{b, c, d, \dots, z\} \\
 \tau < b &\rightarrow_{\equiv} false \text{ with } \tau \in \{c, d, \dots, z\} \\
 &\vdots \\
 z < y &\rightarrow_{\equiv} false \\
 \tau.\alpha < \tau.\beta &\rightarrow_{\equiv} \alpha < \beta \text{ with } \tau \in \{a, b, \dots, z\} \\
 \tau.\alpha < \tau'.\beta &\rightarrow_{\equiv} \tau < \tau' \text{ with } \tau \neq \tau' \in \{a, b, \dots, z\}
 \end{aligned}$$

Because \equiv is symmetric, \leftarrow_{\equiv} is not considered here.

EXAMPLE 4.3. *In the rfs for mono-sorted equational logic, we can consider the following set of rules from the signature $\Sigma = (\{0^0, s^1, +^2, \times^2\}, \{x, y\})$, which defines arithmetic:*

$$\begin{aligned}
 x + 0 &\rightarrow_{\equiv} x \\
 x + s(y) &\rightarrow_{\equiv} s(x + y) \\
 x \times 0 &\rightarrow_{\equiv} 0 \\
 x \times s(y) &\rightarrow_{\equiv} x \times y + x
 \end{aligned}$$

Because $=$ is symmetric, $\leftarrow =$ is not considered here.

EXAMPLE 4.4. In the rfs developed for the conditional equational logic, we can consider the following set of rules from the signature

$$\Sigma = (\text{true}^0, \text{false}^0, 0^0, \text{eq}^2, _ \text{mod} _ ^2, \text{gcd}^2)$$

which specifies the greatest common divisor: in order to improve rule readability, throughout the paper, we will write \rightarrow_c , $\rightarrow_{\mathcal{R}}^c$ and $\overset{*}{\rightarrow}_{\mathcal{R}}^c$ rather than $\rightarrow_{=c}$, $\rightarrow_{\mathcal{R}}^{\overline{=c}}$ and $\overset{*}{\rightarrow}_{\mathcal{R}}^{\overline{=c}}$

$$\begin{aligned} \text{gcd}(n, m) &\rightarrow_{\text{eq}?(n \text{ mod } m, 0)=\text{true}} m \\ \text{gcd}(n, m) &\rightarrow_{\text{eq}?(n \text{ mod } m, 0)=\text{false}} \text{gcd}(m, n \text{ mod } m) \end{aligned}$$

As in the two previous examples, because for every condition c , $=_c$ is symmetric, $\leftarrow =_c$ is not considered here.

EXAMPLE 4.5. In the rfs for non-symmetric transitive rewriting where \subseteq denotes the set-theoretical inclusion, we can consider the following set of rules from the signature $\Sigma = \{\cup^2\}$, which defines the inclusion theory of union [39]:

$$\begin{aligned} X \cup X &\rightarrow_{\subseteq} X \\ X &\leftarrow_{\subseteq} X \cup Y \\ Y &\leftarrow_{\subseteq} X \cup Y \end{aligned}$$

4.2. REWRITING STEPS AND REWRITINGS

We could be tempted to define rewriting steps and rewritings as the closures of binary relations \rightarrow_p and \leftarrow_p under *RS*'s and *De*'s rule instances, respectively, that is by orienting the conclusion of *RS*'s and *De*'s rule instances in the same direction as all their premises (this is how the standard rewriting relation is built in the unconditioned equational rewriting setting). But there are many deduction rules which do not satisfy such a condition. For instance, this is not observed by the rule *Replacement/Congruence* given in Example 3.7. Indeed, when dealing with conditional rewrite rules, we have (at least) three potentially interesting definitions of $\rightarrow_{\mathcal{R}}^{\emptyset}$:¹³ natural, join, and

¹³ Other rewriting relations of the form $\rightarrow_{\mathcal{R}}^c$ with $c \neq \emptyset$ are not considered because simply restricted to rewrite rules (i.e. $\rightarrow_{\mathcal{R}}^c = \rightarrow_c$). Indeed, the rfs defined in Example 3.6 is the logical setting which parameterizes classic conditional rewriting. But, the classic conditional rewriting was defined to answer the validity problem of unconditioned equations (the conclusions of all deduction rules are of the form $t =_{\emptyset} t'$). Only rewrite rules are with conditions.

normal rewriting. They were proposed in [24].¹⁴ Hence, $\rightarrow_{\mathcal{R}}^0$ can be defined as follows: given a rewrite system $\mathcal{R} = (\rightarrow_c)_{c:\text{conjunction}}$, then let us define $\Theta = \{t =_c t' \mid t \rightarrow_c t' \in \mathcal{R}\}$

1. *Natural conditional rewriting* $C[\sigma(t)] \rightarrow_{\mathcal{R}}^0 C[\sigma(t')]$ if $t \rightarrow \bigwedge_{1 \leq i \leq n} t'_i \in \mathcal{R}$ and for every i , $1 \leq i \leq n$, $\Theta \vdash \sigma(t_i) =_{\Theta} \sigma(t'_i)$,
2. *Join conditional rewriting* $C[\sigma(t)] \rightarrow_{\mathcal{R}}^0 C[\sigma(t')]$ if $t \rightarrow \bigwedge_{1 \leq i \leq n} t'_i \in \mathcal{R}$ and for every i , $1 \leq i \leq n$, $\sigma(t_i) \downarrow_{\Theta} \sigma(t'_i)$ where \downarrow_{Θ} means there is a term t'' such that $\sigma(t_i) \xrightarrow{\Theta}^0 t'' \xrightarrow{\Theta}^0 \sigma(t'_i)$, or
3. *Normal conditional rewriting* $C[\sigma(t)] \rightarrow_{\mathcal{R}}^0 C[\sigma(t')]$ if $t \rightarrow \bigwedge_{1 \leq i \leq n} t'_i \in \mathcal{R}$ and for every i , $1 \leq i \leq n$, $\sigma(t_i) \xrightarrow{\Theta}^0 \sigma(t'_i)$

We observe that in the three cases, the orientation of the conclusion $C[\sigma(t)] =_{\Theta} C[\sigma(t')]$ of the Replacement/Congruence rule only depends on the orientation of $t = \bigwedge_{1 \leq i \leq n} t'_i$ in \mathcal{R} .

In Section 9, we will extend the classic conditional rewriting by closing rewriting steps under monotonicity, transitivity and modus-ponens. This will give rise to the following deductive rule:

$$\text{Trans/Mod} \frac{t =_c t' \quad t' =_{c'} t'' \quad \forall i \in I, u_i =_{c_i} v_i}{t =_{\bar{c} \wedge \bigwedge_{i \in I} c_i} t''}$$

where $\bar{c} = c \wedge c' \setminus \{u_i = v_i \mid i \in I\}$.¹⁵

It is clear that the orientation of the conclusion $t =_{\bar{c} \wedge \bigwedge_{i \in I} c_i} t''$ will only depend on the orientation of $t =_c t'$ and $t' =_{c'} t''$. Each equation $u_i =_{c_i} v_i$ must only be satisfied according to one of the three possibilities among natural, normal or join rewriting which can be easily extended (see Section 9.1 for a complete presentation of this rewriting setting).

A similar phenomenon occurs in the logic of special relations [49]¹⁶. Here, two kinds of atoms are considered: $t \gamma t'$ and $\gamma \sqsubseteq \delta$, where t, t' are terms, and

¹⁴ We can also cite [37] for the natural rewriting, and [18, 19] for the join rewriting.

¹⁵ \bar{c} is then the conjunction of c and c' where each equation $u_i = v_i$ has been removed.

¹⁶ The general method developed in this paper has also been applied to M. Schorlemmer's logic with special relations. Actually, M. Schorlemmer does not treat sentences of the form $\gamma \sqsubseteq \delta$ as atoms of its logic of special relations; instead he assumes that the partial order on binary relations is "hardwired" into the signature. Subsequently, he does not consider the

γ, δ are binary relations on terms (the “special relations”). The composition rule is formulated as follows: $\frac{t\alpha' \quad t'\gamma'' \quad \alpha;\gamma\sqsubseteq\delta}{t\delta t''}$ (“;” is the relation composition and α, γ, δ and \sqsubseteq are binary relations which can be rewritten). Here, $t\delta t''$ is oriented in one direction if and only if both $t\alpha'$ and $t'\gamma''$ are oriented in that same direction. Therefore, the orientation of $t\delta t''$ only depends on the orientation of the first two premises. The third one must only be satisfied according to one of the three possibilities among natural, normal or join rewriting which can also be easily extended (see Section 9.2 for a complete presentation of this rewriting setting).

After seeing these three examples, it becomes obvious that some premises of rule instances have a special status. For any rule instance $\iota \in RS \cup De$, we gather its “special” premises in the multi-set $\mathcal{FL}(\iota) \subseteq \mathcal{L}(\iota)$ and call them *fixed leaves*. A sensible constraint is that $\mathcal{FL}(\iota)$ is non empty. The definition of these fixed leaves are *ad-hoc* for each rfs. Therefore, given a deduction rule in $RS \cup De$, the orientation of its conclusion will only be influenced by the orientation of its fixed leaves. In next definition, we will define in the abstract framework, both natural and normal rewritings. Join rewriting can also be abstractly defined but before that, we need to give an abstract meaning of the notion of valleys, which will be done in Section 5.

DEFINITION 4.6 (Rewriting step and rewriting relations). *Let \mathcal{R} be an SP -rewrite system. For every $p \in E$, \rightarrow_p^p and $\overset{*}{\rightarrow}_{\mathcal{R}}^p$ are two binary relations on T defined as the least binary relations (according to set-theoretical inclusion) inductively defined as follows:*

1. $\rightarrow_p \subseteq \rightarrow_{\mathcal{R}}^p$ and $\rightarrow_{\mathcal{R}}^p \subseteq \overset{*}{\rightarrow}_{\mathcal{R}}^p$, and
2. for every $\iota : p(t, t') \in RS$ (resp. $\iota : p(t, t') \in De$) such that:
 - for every leaf $p'(u, v) \in \mathcal{FL}(\iota)$, $u \rightarrow_{\mathcal{R}}^{p'} v$ (resp. $u \overset{*}{\rightarrow}_{\mathcal{R}}^{p'} v$), and
 - for every leaf $p'(u', v') \in \mathcal{L}(\iota) \setminus \mathcal{FL}(\iota)$,
 - Natural rewriting $\Theta \vdash p'(u', v')$ where $\Theta = \{p(t, t') \mid t \rightarrow_p t' \in \mathcal{R} \vee t \leftarrow_p t' \in \mathcal{R}\}$
 - Normal rewriting $u' \overset{*}{\rightarrow}_{\mathcal{R}}^{p'} v'$ or $u' \leftarrow_{\mathcal{R}}^{p'} v'$

we have $t \rightarrow_{\mathcal{R}}^p t'$ (resp. $t \overset{*}{\rightarrow}_{\mathcal{R}}^p t'$)

relation \sqsubseteq to be rewritten. Here, we will extend the rewriting theory for the logic of special relations by allowing to rewrite in addition the relation \sqsubseteq . This extension can be found in Section 9.

We denote by $\rightarrow_{\mathcal{R}} = \bigcup_{p \in E} \rightarrow_{\mathcal{R}}^p$ and $\overset{*}{\rightarrow}_{\mathcal{R}} = \bigcup_{p \in E} \overset{*}{\rightarrow}_{\mathcal{R}}^p$.

$\leftarrow_{\mathcal{R}} = \bigcup_{p \in E} \leftarrow_{\mathcal{R}}^p$ and $\overset{*}{\leftarrow}_{\mathcal{R}} = \bigcup_{p \in E} \overset{*}{\leftarrow}_{\mathcal{R}}^p$ are defined analogously.

REMARK 4.7. The two relations $\overset{*}{\rightarrow}_{\mathcal{R}}$ and $\overset{*}{\leftarrow}_{\mathcal{R}}$ should not be confused with the reflexive and transitive closures of $\rightarrow_{\mathcal{R}}$ and $\leftarrow_{\mathcal{R}}$. There are only the closures of $\rightarrow_{\mathcal{R}}$ and $\leftarrow_{\mathcal{R}}$ under rules in *De*.

EXAMPLE 4.8. In the abstract reduction system setting, a rewrite system \mathcal{R} is given by any binary relation \rightarrow_{\equiv} on A^* . Moreover, all the premises of each instance \mathfrak{v} of both deductive rules *Context* and *Transitivity* belong to $\mathcal{FL}(\mathfrak{v})$. Therefore, following Definition 4.6, $\rightarrow_{\mathcal{R}}^{\equiv}$ is the least binary relation on A^* satisfying the following clauses:

- $\rightarrow_{\equiv} \subseteq \rightarrow_{\mathcal{R}}^{\equiv}$;
- if $\alpha \rightarrow_{\mathcal{R}}^{\equiv} \beta$ then for every $(\delta_1, \delta_2) \in A^* \times A^*$, $\delta_1 \cdot \alpha \cdot \delta_2 \rightarrow_{\mathcal{R}}^{\equiv} \delta_1 \cdot \beta \cdot \delta_2$.

$\overset{*}{\rightarrow}_{\mathcal{R}}^{\equiv}$ is then the reflexive and transitive closure of $\rightarrow_{\mathcal{R}}^{\equiv}$.¹⁷

EXAMPLE 4.9. In the setting of mono-sorted equational rewriting, a rewrite system \mathcal{R} is given by any binary relation $\rightarrow_{=}$ on $T_{\Sigma}(V)$. Moreover, all the premises of each instance \mathfrak{v} of the three deductive rules *Substitution*, *Context* and *Transitivity* belong to $\mathcal{FL}(\mathfrak{v})$. Therefore, following Definition 4.6, $\rightarrow_{\mathcal{R}}^{\equiv}$ is the least binary relation on $T_{\Sigma}(V)$ satisfying the following clauses:

- $\rightarrow_{=} \subseteq \rightarrow_{\mathcal{R}}^{\equiv}$;
- if $t \rightarrow_{\mathcal{R}}^{\equiv} t'$ then $\sigma(t) \rightarrow_{\mathcal{R}}^{\equiv} \sigma(t')$ where $\sigma : V \rightarrow T_{\Sigma}(V)$ is any substitution;
- if $t \rightarrow_{\mathcal{R}}^{\equiv} t'$ then $C[t] \rightarrow_{\mathcal{R}}^{\equiv} C[t']$ where C is a context.

$\overset{*}{\rightarrow}_{\mathcal{R}}^{\equiv}$ is then the reflexive and transitive closure of $\rightarrow_{\mathcal{R}}^{\equiv}$.

EXAMPLE 4.10. In the classic conditional rewriting setting, a rewrite system \mathcal{R} is given by a family of binary relations \rightarrow_c (c being a finite conjunction of equations) on $T_{\Sigma}(V)$. Moreover, for every instance $\mathfrak{v} \frac{t = \bigwedge_{1 \leq i \leq n} t_i = t'_i \quad \forall 1 \leq i \leq n, \sigma(t_i) =_0 \sigma(t'_i)}{C[\sigma(t)] =_0 C[\sigma(t')]}$ of the deductive rule *Replacement/Congruence*, the set of its fixed leaves is $\mathcal{FL}(\mathfrak{v}) = \{t = \bigwedge_{1 \leq i \leq n} t_i = t'_i\}$. All the premises of every instance \mathfrak{v} of the deductive rule *Transitivity* belong to $\mathcal{FL}(\mathfrak{v})$. Therefore, following Definition 4.6, for

¹⁷ The symmetric closure of $\overset{*}{\rightarrow}_{\mathcal{R}}^{\equiv}$ is usually called Thue congruence.

every finite conjunction of equations $c \neq \emptyset$, $\rightarrow_{\mathcal{R}}^c = \rightarrow_c$, and $\rightarrow_{\mathcal{R}}^0$ is the least binary relation on $T_{\Sigma}(V)$ satisfying the following clause:

$$s \rightarrow_{\mathcal{R}}^0 t \Leftrightarrow \begin{cases} \exists l \rightarrow \bigwedge_{1 \leq i \leq n} r_i \in \mathcal{R}, \omega, \sigma, \\ s|_{\omega} = \sigma(l) \wedge t = s[\sigma(r)]_{\omega} \wedge \bigwedge_{1 \leq i \leq n} \sigma(l_i) \sim \sigma(r_i) \end{cases}$$

In the natural and normal rewritings, \sim is $\leftrightarrow_{\mathcal{R}}^*$ and $\rightarrow_{\mathcal{R}}^*$, respectively, where $\leftrightarrow_{\mathcal{R}}^*$ and $\rightarrow_{\mathcal{R}}^*$ are the reflexive, symmetric and transitive closure and the reflexive and transitive closure of $\rightarrow_{\mathcal{R}}^0$.

EXAMPLE 4.11. In the setting of multi-sorted equational rewriting, a rewrite system \mathcal{R} is given by an S -indexed family of binary relations $\rightarrow_{=,s}$ on $T_{\Sigma}(V)_s$, with $s \in S$. Moreover, all the premises of each instance ι of the three deductive rules Substitution, Context and Transitivity belong to $\mathcal{FL}(\iota)$. Therefore, following Definition 4.6, for every $s \in S$ $\rightarrow_{\mathcal{R}}^{\bar{s}}$ is the least binary relation on $T_{\Sigma}(V)_s$ satisfying the following clauses:

- $\rightarrow_{=,s} \subseteq \rightarrow_{\mathcal{R}}^{\bar{s}}$;
- if $t \rightarrow_{\mathcal{R}}^{\bar{s}} t'$ then $\sigma(t) \rightarrow_{\mathcal{R}}^{\bar{s}} \sigma(t')$ where $\sigma : V \rightarrow T_{\Sigma}(V)$ is any substitution;
- if $t \rightarrow_{\mathcal{R}}^{\bar{s}} t'$ then $C[t] \rightarrow_{\mathcal{R}}^{\bar{s}} C[t']$ where C is a context.

For every $s \in S$, $\rightarrow_{\mathcal{R}}^{\bar{s}}$ is then the reflexive and transitive closure of $\rightarrow_{\mathcal{R}}^{\bar{s}}$.

EXAMPLE 4.12. In the setting of non-symmetric transitive rewriting, a rewrite system \mathcal{R} is given by two binary relations on $T_{\Sigma}(V)$, \rightarrow_{\prec} and \leftarrow_{\prec} , respectively. Therefore, following Definition 4.6, $\rightarrow_{\mathcal{R}}^{\prec}$ and $\leftarrow_{\mathcal{R}}^{\prec}$ are both least binary relations on $T_{\Sigma}(V)$ satisfying the following clauses:

- $\rightarrow_{\prec} \subseteq \rightarrow_{\mathcal{R}}^{\prec}$ and $\leftarrow_{\prec} \subseteq \leftarrow_{\mathcal{R}}^{\prec}$;
- if $t \rightarrow_{\mathcal{R}}^{\prec} t'$ (resp. $t \leftarrow_{\mathcal{R}}^{\prec} t'$) then $\sigma(t) \rightarrow_{\mathcal{R}}^{\prec} \sigma(t')$ (resp. $\sigma(t) \leftarrow_{\mathcal{R}}^{\prec} \sigma(t')$) where $\sigma : V \rightarrow T_{\Sigma}(V)$ is any substitution;
- if $t \rightarrow_{\mathcal{R}}^{\prec} t'$ (resp. $t \leftarrow_{\mathcal{R}}^{\prec} t'$) then $C[t] \rightarrow_{\mathcal{R}}^{\prec} C[t']$ (resp. $C[t] \leftarrow_{\mathcal{R}}^{\prec} C[t']$) where C is a context.

$\rightarrow_{\mathcal{R}}^{\prec}$ and $\leftarrow_{\mathcal{R}}^{\prec}$ are then the reflexive and transitive closures of $\rightarrow_{\mathcal{R}}^{\prec}$ and $\leftarrow_{\mathcal{R}}^{\prec}$, respectively.

DEFINITION 4.13 (Convertibility relation). *With all the notations of Definition 4.6, we denote by $\leftrightarrow_{\mathcal{R}} = \bigcup_{p \in E} \leftrightarrow_{\mathcal{R}}^p$ where for every p , $\leftrightarrow_{\mathcal{R}}^p$ is the least binary relation satisfying: $t \leftrightarrow_{\mathcal{R}}^p t'$ iff either $t \rightarrow_p t'$ or $t \leftarrow_p t'$ or there exists $\iota : p(t, t') \in RS$ such that:*

- for every $p'(u, v) \in \mathcal{FL}(\mathfrak{t})$, $u \leftrightarrow_{\mathcal{R}}^{p'} v$
- for every $p'(u, v) \in \mathcal{L}(\mathfrak{t}) \setminus \mathcal{FL}(\mathfrak{t})$, $u \overset{*}{\leftrightarrow}_{\mathcal{R}}^{p'} v$

Finally, we denote by $\overset{*}{\leftrightarrow}_{\mathcal{R}} = \bigcup_{p \in E} \overset{*}{\leftrightarrow}_{\mathcal{R}}^p$ the closure of $\leftrightarrow_{\mathcal{R}}^p$ under trees of De and Rmv , that is $t \overset{*}{\leftrightarrow}_{\mathcal{R}}^p t'$ iff either $t \leftrightarrow_{\mathcal{R}}^p t'$ or there exists $\iota : p(t, t') \in De \cup Rmv$ such that: $\forall p'(u, v) \in \mathcal{L}(\mathfrak{t}), u \overset{*}{\leftrightarrow}_{\mathcal{R}}^{p'} v$. $\overset{*}{\leftrightarrow}_{\mathcal{R}}$ is called convertibility relation.

REMARK 4.14. *The relation $\overset{*}{\leftrightarrow}_{\mathcal{R}}$ should not be confused with the reflexive and transitive closure of $\leftrightarrow_{\mathcal{R}}$. It is only the closure of $\leftrightarrow_{\mathcal{R}}$ under rules in De and Rmv .*

From Definition 4.13, the convertibility relation $\overset{*}{\leftrightarrow}_{\mathcal{R}}$ defines proof strategies which restrict the proof search space by selecting proof trees equipped with the following structure: RS 's rule instances are always above both De 's rule instances and Rmv 's rule instances. Let us call such proof trees *rewrite trees*. We must check that derivability (i.e. syntactic consequences obtained from \vdash) coincides with convertibility in rewriting. This property, named *logicality*,¹⁸ is expressed by: $\Gamma \vdash p(t, t') \Leftrightarrow t \overset{*}{\leftrightarrow}_{\mathcal{R}}^p t'$ ($\Gamma = \{p(u, v) \mid u \rightarrow_p v \in \mathcal{R} \vee u \leftarrow_p v \in \mathcal{R}\}$).

Obviously, we have $t \overset{*}{\leftrightarrow}_{\mathcal{R}}^p t' \Rightarrow \Gamma \vdash p(t, t')$ because the convertibility relation defines rewrite proofs which are peculiar proof trees. The other direction is more difficult to prove because it requires that all statements of the form $\Gamma \vdash p(t, t')$ accept some rewrite proofs as proof trees. In all logics where the logicality result holds, this is checked thanks to basic proof tree transformations produced from the *semi-commutation* property of rule instances in RS with rule instances in $De \cup Rmv$. For instance, in the equational logic, substitution semi-commutes (denoted by \rightsquigarrow) with transitivity:

$$\text{Subst} \frac{\text{Trans} \frac{t=t' \quad t'=t''}{t=t''}}{\sigma(t)=\sigma(t'')} \rightsquigarrow \text{Trans} \frac{\text{Subst} \frac{t=t'}{\sigma(t)=\sigma(t')}}{\sigma(t)=\sigma(t'')} \quad \text{Subst} \frac{t'=t''}{\sigma(t')=\sigma(t'')}}{\sigma(t)=\sigma(t'')}$$

The difficulty is to show that the induced global proof tree transformation is normalizing. In [4], we have provided a general setting under which results

¹⁸ In the equational logic setting, this result is the so-called *Birkhoff's theorem*.

of normalization of proof trees such as, for instance, the logicality result in equational reasoning and the cut-elimination property in sequent or natural deduction calculi, can be unified and generalized. This has been achieved by giving simple conditions which are sufficient to ensure that such normalization results hold. These conditions are based on basic properties of elementary combinations of inference rules which assure that the induced “global” proof tree transformation processes do terminate. We refer the reader to terms of this generalized version of the logicality theorem as well as its proof in [3, 4]. Here, we suppose that convertibility coincides with deductions, that is: $\Gamma \vdash p(t, t') \Leftrightarrow t \xleftrightarrow{\mathcal{R}}^* t'$.

4.3. DERIVATIONS AND PROOFS

In the equational and non-symmetric transitive rewriting setting, derivations and proofs can be simply defined by a sequence of rewriting steps. The transitivity application order on this sequence does not matter: different orders always end in the same conclusion. This comes from the fact that transitivity instances can permute with each other:

$$\frac{\frac{t_1 r t_2 \quad t_2 r t_3}{t_1 r t_3} \quad t_3 r t_4}{t_1 r t_4} \rightsquigarrow \frac{t_1 r t_2 \quad \frac{t_2 r t_3 \quad t_3 r t_4}{t_2 r t_4}}{t_1 r t_4} \quad r \in \{\equiv, \preceq, =, =\emptyset\}$$

Notice that both trees can be represented by the same sequence $t_1 r t_2 r t_3 r t_4$.

In our generalization, we cannot define derivations and proofs as sequences of rewriting steps because the applications of rule instances in De do not permute with each other *a priori*. This leads us to denote them by trees. Hence, a derivation is a rewrite tree whose internal nodes and leaves are labelled by elements of $\xrightarrow{\mathcal{R}}^*$ (resp. $\xleftarrow{\mathcal{R}}^*$) and $\rightarrow_{\mathcal{R}}$ (resp. $\leftarrow_{\mathcal{R}}$), respectively. Formally, this is defined as follows:

NOTATION 4.15. *Let us note $t \xleftrightarrow{\mathcal{R}}^* t'$ and $t \xleftarrow{\mathcal{R}}^* t'$ to mean either $t \rightarrow_{\mathcal{R}}^* t'$ or $t \leftarrow_{\mathcal{R}}^* t'$, and $t \xrightarrow{\mathcal{R}}^* t'$ or $t \xleftarrow{\mathcal{R}}^* t'$, respectively.*

REMARK 4.16. *In any case, $t \xleftrightarrow{\mathcal{R}}^* t'$ should not be confused with $t \leftrightarrow_{\mathcal{R}}^* t'$. In the first case, rewriting direction does not matter but it exists. In the second case, we have closed $\rightarrow_{\mathcal{R}}^*$ under rule instances in Rmv .*

DEFINITION 4.17 (Derivations and proofs). *A derivation d is any tree*

$$\frac{d_1 : u_1 \xleftrightarrow{\mathcal{R}}^* v_1 \dots d_n : u_n \xleftrightarrow{\mathcal{R}}^* v_n}{u_{n+1} \xleftrightarrow{\mathcal{R}}^* v_{n+1}}$$

such that:

- each d_i is a derivation ($1 \leq i \leq n$),
- for all i, j , $1 \leq i, j \leq n+1$, $u_i \xrightarrow{*P_i} v_i \iff u_j \xrightarrow{*P_j} v_j$ (d_i and d_j are oriented in the same direction), and
- there is an instance $\iota : p_{n+1}(u_{n+1}, v_{n+1}) \in De$ such that:
 - $\mathcal{FL}(\iota) = \{p_1(u_1, v_1), \dots, p_n(u_n, v_n)\}$, and
 - for all $p'(u', v') \in \mathcal{L}(\iota) \setminus \mathcal{FL}(\iota)$, $p'(u', v')$ holds in \mathcal{R} (according to the normal or natural rewriting)

A proof π is any tree $\frac{\pi_1:u_1 \xrightarrow{*P_1} v_1, \dots, \pi_n:u_n \xrightarrow{*P_n} v_n}{u_{n+1} \xrightarrow{*P_{n+1}} v_{n+1}}$ such that:

- each π_i is a proof ($1 \leq i \leq n$), and
- there is an instance $\iota : p_{n+1}(u_{n+1}, v_{n+1}) \in De \cup Rmv$ such that:
 - if $\iota \in Rmv$ then $\mathcal{L}(\iota) = \{p_1(u_1, v_1), \dots, p_n(u_n, v_n)\}$,
 - else (i.e. $\iota \in De$) $\mathcal{FL}(\iota) = \{p_1(u_1, v_1), \dots, p_n(u_n, v_n)\}$, and for all $p'(u', v') \in \mathcal{L}(\iota) \setminus \mathcal{FL}(\iota)$, $u' \xrightarrow{*P'} v'$.

We will note $(d_1 : u_1 \xrightarrow{*P_1} v_1, \dots, d_n : u_n \xrightarrow{*P_n} v_n, u_{n+1} \xrightarrow{*P_{n+1}} v_{n+1})_\iota$ (resp. $(\pi_1 : u_1 \xrightarrow{*P_1} v_1, \dots, \pi_n : u_n \xrightarrow{*P_n} v_n, u_{n+1} \xrightarrow{*P_{n+1}} v_{n+1})_\iota$) to mean that the last rule applied in the derivation $\frac{d_1:u_1 \xrightarrow{*P_1} v_1 \dots d_n:u_n \xrightarrow{*P_n} v_n}{u_{n+1} \xrightarrow{*P_{n+1}} v_{n+1}}$ (resp. the proof $\frac{\pi_1:u_1 \xrightarrow{*P_1} v_1 \dots \pi_n:u_n \xrightarrow{*P_n} v_n}{u_{n+1} \xrightarrow{*P_{n+1}} v_{n+1}}$) is $\iota \in De$ (resp. $\iota \in De \cup Rmv$).

Derivations are then trees inductively constructed over instances in De with rewriting steps as generators such that all fixed leaves of instances in De are oriented in the same direction.

Proofs are also trees inductively constructed over instances in $De \cup Rmv$ with rewriting steps as generators, but satisfying that somewhere in the tree:

1. an instance of Rmv may occur, and/or
2. an instance of De may occur with (at least) two of its fixed leaves oriented in opposite directions.

EXAMPLE 4.18. *In the equational and non-symmetric transitive rewriting setting, for short the transitive rewriting setting, a sequence of rewriting steps then represents all trees composed of transitivity instances and whose leaves are the rewriting steps of the sequence, but in which the applications of the transitivity rule are not made in the same order. Hence, in our framework,*

from the sequence of rewritings $t_1 \xrightarrow{\overline{\mathcal{R}}} t_2 \xrightarrow{\overline{\mathcal{R}}} \dots \xrightarrow{\overline{\mathcal{R}}} t_n$, many derivations can be considered such as for instance the application of the transitivity rule from left to right:

$$\begin{array}{c}
 \frac{d_0 \overline{t_1 \rightarrow_{\mathcal{R}} t_2} \quad d'_0 \overline{t_2 \rightarrow_{\mathcal{R}} t_3}}{d_1 \overline{t_1 \xrightarrow{*} t_3}} \quad d'_1 \overline{t_3 \rightarrow_{\mathcal{R}} t_4}} \\
 \frac{\quad}{d_2 \overline{t_1 \xrightarrow{*} t_4}} \\
 \vdots \\
 \frac{d_{n-3} \overline{\dots} \quad d'_{n-3} \overline{t_{n-1} \rightarrow_{\mathcal{R}} t_n}}{d_{n-2} \overline{t_1 \xrightarrow{*} t_n}}
 \end{array}$$

Here, for each i , $1 \leq i \leq n-2$, each derivation d_i is denoted by $(d_{i-1} : t_1 \xrightarrow{*} t_{i+1}, d'_{i-1} : t_{i+1} \xrightarrow{\overline{\mathcal{R}}} t_{i+2}, t_1 \xrightarrow{\overline{\mathcal{R}}} t_{i+2})_{\substack{t_1=t_{i+1} \\ t_{i+1}=t_{i+2}}}$.

In the non-symmetric but transitive rewriting setting, because the transitivity rule is of the form $\frac{t \xrightarrow{\mathcal{R}'} t' \xrightarrow{\mathcal{R}'} t''}{t \xrightarrow{\mathcal{R}'} t''}$, sequences of rewriting are:

- either of the form $t_1 \xrightarrow{\mathcal{R}} t_2 \xrightarrow{\mathcal{R}} t_3 \dots \xrightarrow{\mathcal{R}} t_n$,
- or $t_n \xleftarrow{\mathcal{R}} t_{n-1} \xleftarrow{\mathcal{R}} t_{n-2} \dots \xleftarrow{\mathcal{R}} t_1$, that is $t_1 \xrightarrow{\mathcal{R}} t_2 \xrightarrow{\mathcal{R}} t_3 \dots \xrightarrow{\mathcal{R}} t_n$

Therefore, by applying the transitivity rule from left to right, for instance, we have both following derivations:

1.

$$\begin{array}{c}
 \frac{d_0 \overline{t_1 \xrightarrow{\mathcal{R}} t_2} \quad d'_0 \overline{t_2 \xrightarrow{\mathcal{R}} t_3}}{d_1 \overline{t_1 \xrightarrow{*} t_3}} \quad d'_1 \overline{t_3 \xrightarrow{\mathcal{R}} t_4}} \\
 \frac{\quad}{d_2 \overline{t_1 \xrightarrow{*} t_4}} \\
 \vdots \\
 \frac{d_{n-3} \overline{\dots} \quad d'_{n-3} \overline{t_{n-1} \xrightarrow{\mathcal{R}} t_n}}{d_{n-2} \overline{t_1 \xrightarrow{*} t_n}}
 \end{array}$$

$$\begin{array}{c}
\frac{d_0 \frac{t_1 \xrightarrow{\mathcal{R}} t_2}{t_1 \xrightarrow{\mathcal{R}} t_2} \quad d'_0 \frac{t_2 \xrightarrow{\mathcal{R}} t_3}{t_2 \xrightarrow{\mathcal{R}} t_3}}{d_1 \frac{t_1 \xrightarrow{\mathcal{R}} t_3}{t_1 \xrightarrow{\mathcal{R}} t_3}} \quad d'_1 \frac{t_3 \xrightarrow{\mathcal{R}} t_4}{t_3 \xrightarrow{\mathcal{R}} t_4}}{d_2 \frac{t_1 \xrightarrow{\mathcal{R}} t_4}{t_1 \xrightarrow{\mathcal{R}} t_4}} \\
\vdots \\
\frac{d_{n-3} \frac{\dots}{t_1 \xrightarrow{\mathcal{R}} t_{n-1}} \quad d'_{n-3} \frac{t_{n-1} \xrightarrow{\mathcal{R}} t_n}{t_{n-1} \xrightarrow{\mathcal{R}} t_n}}{d_{n-2} \frac{t_1 \xrightarrow{\mathcal{R}} t_n}{t_1 \xrightarrow{\mathcal{R}} t_n}}
\end{array}$$

As an example of proof, let us consider the following sequence of term rewriting:

$$t_1 \rightarrow_{\mathcal{R}} t_2 \leftarrow_{\mathcal{R}} t_3 \leftarrow_{\mathcal{R}} t_4 \rightarrow_{\mathcal{R}} t_5 \leftarrow_{\mathcal{R}} t_6 \rightarrow_{\mathcal{R}} t_7$$

The following trees are examples of proofs which are the results of different application orders of transitivity on the above sequence:

$$\begin{array}{c}
\frac{d_1^0 \frac{d_0^0 : t_1 \rightarrow_{\mathcal{R}} t_2 \quad d_0^1 : t_2 \leftarrow_{\mathcal{R}} t_3}{d_1^0 : t_1 \xrightarrow{\mathcal{R}} t_3} \quad d_1^1 \frac{d_0^2 : t_3 \leftarrow_{\mathcal{R}} t_4 \quad d_0^3 : t_4 \rightarrow_{\mathcal{R}} t_5}{d_1^1 : t_3 \xrightarrow{\mathcal{R}} t_5}}{d_2^0 \frac{t_1 \xrightarrow{\mathcal{R}} t_3 \quad t_3 \xrightarrow{\mathcal{R}} t_5}{t_1 \xrightarrow{\mathcal{R}} t_5}} \quad d_2^1 \frac{d_1^2 : t_5 \leftarrow_{\mathcal{R}} t_6 \quad d_1^3 : t_6 \rightarrow_{\mathcal{R}} t_7}{d_2^1 : t_5 \xrightarrow{\mathcal{R}} t_7}}{d_3 \frac{t_1 \xrightarrow{\mathcal{R}} t_5 \quad t_5 \xrightarrow{\mathcal{R}} t_7}{t_1 \xrightarrow{\mathcal{R}} t_7}}
\end{array}$$

$$\begin{array}{c}
\frac{d_1^0 \frac{d_0^0 : t_2 \leftarrow_{\mathcal{R}} t_3 \quad d_0^1 : t_3 \leftarrow_{\mathcal{R}} t_4}{d_1^0 : t_2 \xrightarrow{\mathcal{R}} t_4} \quad d_1^1 : t_4 \rightarrow_{\mathcal{R}} t_5}{d_2^0 \frac{t_2 \xrightarrow{\mathcal{R}} t_4 \quad d_1^1 : t_4 \rightarrow_{\mathcal{R}} t_5}{t_2 \xrightarrow{\mathcal{R}} t_5}} \quad d_2^1 \frac{d_1^2 : t_5 \leftarrow_{\mathcal{R}} t_6 \quad d_1^3 : t_6 \rightarrow_{\mathcal{R}} t_7}{d_2^1 : t_5 \xrightarrow{\mathcal{R}} t_7}}{d_3^0 \frac{d_1^0 : t_2 \xrightarrow{\mathcal{R}} t_5 \quad t_2 \xrightarrow{\mathcal{R}} t_5}{d_3^0 : t_1 \xrightarrow{\mathcal{R}} t_5} \quad d_3^1 \frac{d_2^2 : t_5 \leftarrow_{\mathcal{R}} t_6 \quad d_2^3 : t_6 \rightarrow_{\mathcal{R}} t_7}{d_3^1 : t_5 \xrightarrow{\mathcal{R}} t_7}}{d_4 \frac{t_1 \xrightarrow{\mathcal{R}} t_5 \quad t_5 \xrightarrow{\mathcal{R}} t_7}{t_1 \xrightarrow{\mathcal{R}} t_7}}
\end{array}$$

$$\begin{array}{c}
d_1^0 \frac{d_0^0 : t_1 \rightarrow_{\mathcal{R}} t_2 \quad d_0^1 : t_2 \leftarrow_{\mathcal{R}} t_3}{t_1 \xleftrightarrow{\mathcal{R}}^* t_3} \quad d_1^1 : t_3 \leftarrow_{\mathcal{R}} t_5 \\
d_2^0 \frac{t_1 \xleftrightarrow{\mathcal{R}}^* t_3 \quad d_1^1 : t_3 \leftarrow_{\mathcal{R}} t_5}{t_1 \xleftrightarrow{\mathcal{R}}^* t_4} \quad d_2^1 : t_4 \rightarrow_{\mathcal{R}} t_5 \\
\hline
d_3^1 \frac{d_2^2 : t_5 \leftarrow_{\mathcal{R}} t_6 \quad d_2^3 : t_6 \rightarrow_{\mathcal{R}} t_7}{t_5 \xleftrightarrow{\mathcal{R}}^* t_7} \\
d_4 \frac{t_1 \xleftrightarrow{\mathcal{R}}^* t_4 \quad d_3^1 \frac{d_2^2 : t_5 \leftarrow_{\mathcal{R}} t_6 \quad d_2^3 : t_6 \rightarrow_{\mathcal{R}} t_7}{t_5 \xleftrightarrow{\mathcal{R}}^* t_7}}{t_1 \xleftrightarrow{\mathcal{R}}^* t_7}
\end{array}$$

5. Properties of rewrite systems

In the transitive rewriting setting, rewrite systems are used to answer the validity problem of theories when they are both *confluent* and *terminating*, that is *convergent*. To define convergent rewrite systems in our framework, we must then give an abstract formulation of termination and confluence.

5.1. TERMINATION

A naive approach to define the termination of rewrite systems would be that derivations cannot be expanded indefinitely, expansion of derivations meaning that there are some other derivations which contain them as subtrees. Of course, this condition is sufficient but is too weak. Indeed, in the equational rewriting setting, the instantiation of such a definition would be: a rewrite system is terminating if every derivation $t_1 \rightarrow t_2 \rightarrow \dots \rightarrow t_n$ is:

- un-expandable to the right $t_1 \rightarrow t_2 \rightarrow \dots \rightarrow t_n \not\rightarrow t_{n+1}$
- un-expandable to the left $t_0 \not\rightarrow t_1 \rightarrow t_2 \rightarrow \dots \rightarrow t_n$

Condition 1. is the right one which is usually taken into account to define termination of rewrite systems. On the contrary, Condition 2. cannot be imposed, otherwise most classic rewrite systems would be considered as un-terminating what would be counter-intuitive.

In our generalization, expansion to the left or to the right has no meaning, because the algebraic structures under consideration are not necessarily transitive. We then have to denote derivations by trees (see the previous section). Therefore, we have to define a similar notion to the notion of expanding to the left and to the right but for trees. To achieve this purpose, we can notice that for every $\mathfrak{t} = \frac{p_1(u_1, v_1) \dots p_n(u_n, v_n)}{p(u, v)}$ in De , whether we are dealing with $\xrightarrow{\mathcal{R}}^P$ or $\xleftarrow{\mathcal{R}}^P$, we can associate a number i , $1 \leq i \leq n$, such that $p_i(u_i, v_i) \in \mathcal{FL}(\mathfrak{t})$, which denotes the position among the premises of \mathfrak{t} where derivations are to be expanded. We will denote this position by $\vec{\mathcal{E}x}(\mathfrak{t})$ (resp. $\overleftarrow{\mathcal{E}x}(\mathfrak{t})$). For

instance, in the equational rewriting setting, for every transitivity instance $\mathfrak{t} = \frac{u \equiv v \quad v \equiv w}{u \equiv w}$, $\vec{\mathcal{E}x}(\mathfrak{t}) = 1$ ($\overleftarrow{\mathcal{E}x}(\mathfrak{t})$ is not considered because equalities are symmetric). Hence, if we consider the derivation $d : u \xrightarrow{*}^{\equiv} \mathcal{R} v$, any expansion of d from \mathfrak{t} has necessarily the following form $\frac{d : u \xrightarrow{*}^{\equiv} \mathcal{R} v \quad d' : v \xrightarrow{*}^{\equiv} \mathcal{R} w}{u \xrightarrow{*}^{\equiv} \mathcal{R} w}$. From such an expanding position, we can define an expanding relation $\mathcal{E}_{\mathcal{R}}$ on derivations as follows:

$$d : u \xrightarrow{*}^{\equiv} \mathcal{R} v \quad \mathcal{E}_{\mathcal{R}} \quad \frac{d : u \xrightarrow{*}^{\equiv} \mathcal{R} v \quad d' : v \xrightarrow{*}^{\equiv} \mathcal{R} w}{u \xrightarrow{*}^{\equiv} \mathcal{R} w}$$

Let us also denote by $\mathcal{E}_{\mathcal{R}}$ the transitive closure of this expanding relation. Therefore, in the equational rewriting setting, we can re-define the termination of rewrite systems as follows: a rewrite system is terminating if and only if $\mathcal{E}_{\mathcal{R}}$ is Noetherian.¹⁹

Because $\mathcal{E}_{\mathcal{R}}$ has been defined on derivations denoted by trees, it can be generalized in our abstract framework as follows:

DEFINITION 5.1 (Expanding positions and expanding relation). *Let \mathcal{SP} be an rfs. Expanding positions are defined by two applications $\vec{\mathcal{E}x}, \overleftarrow{\mathcal{E}x} : De \rightarrow \mathbb{N}$ satisfying for every $\mathfrak{t} = \frac{p_1(u_1, v_1) \dots p_n(u_n, v_n)}{p(t, t')}$ in De :*

- $\vec{\mathcal{E}x}(\mathfrak{t}) \neq \overleftarrow{\mathcal{E}x}(\mathfrak{t})$,
- $1 \leq \vec{\mathcal{E}x}(\mathfrak{t}) \leq n$ and $1 \leq \overleftarrow{\mathcal{E}x}(\mathfrak{t}) \leq n$, and
- if $j = \vec{\mathcal{E}x}(\mathfrak{t})$ and $k = \overleftarrow{\mathcal{E}x}(\mathfrak{t})$ then $p_j(u_j, v_j)$ and $p_k(u_k, v_k)$ belong to $\mathcal{FL}(\mathfrak{t})$.

Let \mathcal{R} be an \mathcal{SP} -rewrite system. Let us denote by $\mathcal{E}_{\mathcal{R}}$ the binary relation on derivations defined as the transitive closure of:

$$d : u \xrightarrow{*}^{\mathcal{P}'} \mathcal{R} v \quad \mathcal{E}_{\mathcal{R}} \quad (d' = (d_1, \dots, d_m, t \xrightarrow{*}^{\mathcal{P}} t')_{\mathfrak{t}}) \iff d_{\vec{\mathcal{E}x}(\mathfrak{t})} = d$$

$$d : u \xrightarrow{*}^{\mathcal{P}'} \mathcal{R} v \quad \mathcal{E}_{\mathcal{R}} \quad (d' = (d_1, \dots, d_m, t \xrightarrow{*}^{\mathcal{P}} t')_{\mathfrak{t}}) \iff d_{\overleftarrow{\mathcal{E}x}(\mathfrak{t})} = d$$

¹⁹ A Noetherian relation is not necessarily irreflexive. A reflexive binary relation can be Noetherian provided that all infinite ordered sequences are stationary, that is there is a position $i \in \mathbb{N}$ from which the following elements in the sequence are equal.

EXAMPLE 5.2. *The symmetric and transitive rewriting setting has already been handled above. In the non-symmetric transitive rewriting setting, both expanding positions for every instance ι of the transitivity are respectively, $\overrightarrow{\mathcal{E}x}$ (ι) = 1 and $\overleftarrow{\mathcal{E}x}$ (ι) = 2. Therefore, the expanding relation for any \mathcal{SP} -rewrite system \mathcal{R} is defined as shown in the figure below.*

$$\begin{array}{c} \forall t, t', t'' \\ \begin{array}{c} \triangleleft d \\ t \xrightarrow{*} \xrightarrow{\prec} \mathcal{R} t' \end{array} \quad \mathcal{E}_{\mathcal{R}} \quad \frac{\begin{array}{c} \triangleleft d \quad \triangleleft d' \\ t \xrightarrow{*} \xrightarrow{\prec} \mathcal{R} t' \quad t' \xrightarrow{*} \xrightarrow{\prec} \mathcal{R} t'' \end{array}}{t \xrightarrow{*} \xrightarrow{\prec} \mathcal{R} t''} \\ \\ \forall t, t', t'' \\ \begin{array}{c} \triangleleft d \\ t' \xleftarrow{*} \xrightarrow{\prec} \mathcal{R} t \end{array} \quad \mathcal{E}_{\mathcal{R}} \quad \frac{\begin{array}{c} \triangleleft d' \quad \triangleleft d \\ t'' \xleftarrow{*} \xrightarrow{\prec} \mathcal{R} t' \quad t' \xleftarrow{*} \xrightarrow{\prec} \mathcal{R} t \end{array}}{t'' \xleftarrow{*} \xrightarrow{\prec} \mathcal{R} t} \end{array}$$

Let us observe that $\mathcal{E}_{\mathcal{R}}$ induces a partition of derivations, that is if $d \mathcal{E}_{\mathcal{R}} d'$ then d and d' have their conclusions which are oriented in the same direction.

Although defining termination from Noetherianess of $\mathcal{E}_{\mathcal{R}}$ is sufficient for both equational and conditional rewriting settings, this is too weak for the non-symmetric transitive one. Indeed, in such a rewriting setting, this would be equivalent to impose that both $\xrightarrow{*} \xrightarrow{\prec} \mathcal{R}$ and $\xleftarrow{*} \xrightarrow{\prec} \mathcal{R}$ are Noetherian orderings.²⁰ However, it has been shown in [39] that this is a weaker condition not sufficient to have Newman's lemma (i.e. the equivalence between the Church-Rosser and local confluence properties under termination of rewrite systems). To obtain such a result, we have to define the termination of rewrite systems by: $(\xrightarrow{*} \xrightarrow{\prec} \mathcal{R} \cup \xleftarrow{*} \xrightarrow{\prec} \mathcal{R})^*$ is a Noetherian ordering. This motivates the concept of reduction relation, that is a binary relation which is known to be terminating and embodies in its definition the closure properties under instances in RS and De . Moreover, proving termination of a rewrite system often requires to check an infinite set of expansions of each derivation in \mathcal{R} . A way to answer this problem is also to use the concept of reduction relation as is stemmed from Theorem 5.7 below. Here, as previously, reduction relations are not necessarily orders.

DEFINITION 5.3 (Reduction relation). *Let $\mathcal{SP} = (T, E, RS, De, Rmv)$ be an rfs equipped with both expanding position applications $\overrightarrow{\mathcal{E}x}$ and $\overleftarrow{\mathcal{E}x}$. A rewrit-*

²⁰ This is because two derivations d and d' satisfying $d \mathcal{E}_{\mathcal{R}} d'$ have their conclusions which are oriented in the same direction.

ing relation \succ is an E -indexed family of binary relations \succ_p on T satisfying for each $p \in E$:

- $\succ_p \subseteq p \cup p^{-1}$,
- \succ_p is closed under instances in RS and De (cf. Definition 4.6).

A reduction is any tree resulting from the inductive construction of \succ , that is any tree $r = \frac{r_1:u_1 \succ_{p_1} v_1 \dots r_n:u_n \succ_{p_n} v_n}{u_{n+1} \succ_{p_{n+1}} v_{n+1}}$ such that:

- each r_i is a reduction ($1 \leq i \leq n$), and
- there is an instance $\iota: p_{n+1}(u_{n+1}, v_{n+1}) \in De$ or $\iota: p_{n+1}(v_{n+1}, u_{n+1}) \in De$ such that:
 - for all i , $1 \leq i \leq n$, $p_i(u_i, v_i) \in \mathcal{FL}(\iota)$ or $p_i(v_i, u_i) \in \mathcal{FL}(\iota)$, and
 - for all $p'(u', v') \in \mathcal{L}(\iota) \setminus \mathcal{FL}(\iota)$, we have either $u' \succ_{p'} v'$ or $v' \succ_{p'} u'$.

As for derivations, we can define an expanding relation \mathcal{E}_\succ on reductions according to $\overset{\rightarrow}{\mathcal{E}x}$, that is:

$$r: u \succ v \mathcal{E}_\succ (r' = (r_1, \dots, r_m, t \succ t')_\iota) \iff r \overset{\rightarrow}{\mathcal{E}x(\iota)} = r'$$

Therefore, a reduction relation is any rewriting relation \succ such that its expanding relation \mathcal{E}_\succ is Nætherian.

EXAMPLE 5.4. In the abstract reduction system rewriting, a reduction relation is any well-founded order stable under context. In both mono-sorted and multi-sorted rewriting settings, a reduction relation is any well-founded order stable under context and substitution. In the conditional rewriting setting, a reduction relation is a family of well-founded binary relations \succ_c (c :finite conjunction of equations) such that \succ_\emptyset is an order stable under replacement/congruence, that is if $u \succ \bigwedge_{1 \leq i \leq n} t_i \succ v$ and for every i , $1 \leq i \leq n$, we have either $t_i \succ_\emptyset t'_i$ or $t'_i \succ_\emptyset t_i$, then, $u \succ_\emptyset v$.

In order to benefit from all classic methods that facilitate proof of termination such as reduction and simplification orders, we can establish that for every finite (possibly empty) conjunction c , $\succ_c \Rightarrow$ where \succ is a well-founded ordering on terms stable under substitution and context. This is compatible with the definition of reduction relations such as proposed in this paper. Indeed, $\succ = \bigcup_c \succ_c \Rightarrow$ is obviously stable under replacement/congruence.

Finally, in the setting of non-symmetric transitive rewriting, a reduction relation is any well-founded order monotonic and stable under substitution.

Termination of rewrite systems is then defined as follows:

DEFINITION 5.5 (Termination of rewrite systems). *Let \mathcal{R} be an \mathcal{SP} -rewrite system. Let us define, for every $p \in E$, $\succ_{\mathcal{R}}^p = (\rightarrow_p \cup (\leftarrow_p)^{-1})^* ((\dots)^*)^*$ meaning “closed under instances in RS and De ”. Obviously, $\succ_{\mathcal{R}} = (\succ_{\mathcal{R}}^p)_{p \in E}$ is a rewriting relation. Therefore, \mathcal{R} is terminating if and only if $\mathcal{E}_{\succ_{\mathcal{R}}}$ is a reduction relation.*

EXAMPLE 5.6. *In the equational rewriting setting, because $\rightarrow_{\overline{\mathcal{R}}} = (\leftarrow_{\overline{\mathcal{R}}})^{-1}$, we have $\succ_{\overline{\mathcal{R}}} = \overset{*}{\rightarrow}_{\overline{\mathcal{R}}}$ and then $\mathcal{E}_{\succ_{\overline{\mathcal{R}}}} = \mathcal{E}_{\overline{\mathcal{R}}}$. On the contrary, in the non-symmetric transitive rewriting setting, $\succ_{\overline{\mathcal{R}}} = (\rightarrow_{\overline{\mathcal{R}}} \cup (\leftarrow_{\overline{\mathcal{R}}})^{-1})^*$. Therefore, $\mathcal{E}_{\overline{\mathcal{R}}} \subseteq \mathcal{E}_{\succ_{\overline{\mathcal{R}}}}$.*

THEOREM 5.7. *An \mathcal{SP} -rewrite system \mathcal{R} is terminating if and only if there exists a reduction relation \succ such that: $\forall p \in E, \rightarrow_p \cup (\leftarrow_p)^{-1} \subseteq \succ$.*

Proof If \mathcal{R} is terminating, set $\succ = \succ_{\mathcal{R}}$. Conversely, we have $\mathcal{E}_{\succ_{\mathcal{R}}} \subseteq \mathcal{E}_{\succ}$. As \mathcal{E}_{\succ} is Noetherian, so is $\mathcal{E}_{\succ_{\mathcal{R}}}$. \square

In the term equational rewriting setting, this last theorem was established by Lankford [38] (see [6], p. 103, for the statement of this theorem). Here, Theorem 5.7 is a generalized form of this result.

5.2. GENERALIZED FORM OF CHURCH-ROSSER AND CONFLUENCE PROPERTIES

From Definition 4.17, proofs are composed of derivations connected together via rule instances in $De \cup Rmv$. Therefore, among proofs we have derivations as well as all proofs composed of derivations connected together via one instance of a rule in De with some orientation conflicts on fixed leaves. Let us call such proofs *basic proofs*. They are defined as follows:

DEFINITION 5.8 (Basic proofs). *A basic proof is either a derivation or a proof π of the form $(d_1, \dots, d_n, u \overset{*}{\leftarrow}_{\mathcal{R}}^p v)_\iota$ with $\iota \in De$ and for every $i, 1 \leq i \leq n$, d_i is a derivation. Let us denote by \mathcal{BP} the set of basic proofs.*

EXAMPLE 5.9. *In the equational rewriting setting, basic proofs are elements of $(\overset{*}{\rightarrow} \circ \overset{*}{\leftarrow}) \cup (\overset{*}{\leftarrow} \circ \overset{*}{\rightarrow})$, that is a basic proof is a proof which has one of the two following forms:*

1. $\frac{d_1: u \overset{*}{\leftarrow}_{\mathcal{R}} w \quad d_2: w \overset{*}{\rightarrow}_{\mathcal{R}} v}{u \overset{*}{\leftarrow}_{\mathcal{R}} v}, \text{ or}$
2. $\frac{d_1: u \overset{*}{\rightarrow}_{\mathcal{R}} w \quad d_2: w \overset{*}{\leftarrow}_{\mathcal{R}} v}{u \overset{*}{\leftarrow}_{\mathcal{R}} v}$

In the non-symmetric transitive rewriting setting, basic proofs are elements in $(\xrightarrow{*}_{\mathcal{R}} \circ \xleftarrow{*}_{\mathcal{R}}) \cup (\xleftarrow{*}_{\mathcal{R}} \circ \xrightarrow{*}_{\mathcal{R}})$, that is a basic proof is a proof which has one of the two following forms:

1. $\frac{d_1:u \xleftarrow{*}_{\mathcal{R}} w \quad d_2:w \xrightarrow{*}_{\mathcal{R}} v}{u \xleftrightarrow{*}_{\mathcal{R}} v}$, or
2. $\frac{d_1:u \xrightarrow{*}_{\mathcal{R}} w \quad d_2:w \xleftarrow{*}_{\mathcal{R}} v}{u \xleftrightarrow{*}_{\mathcal{R}} v}$

Obviously, proofs are then basic proofs connected together via a finite set of rule instances in $De \cup Rmv$. In order to reduce the proof search space when we want to establish that two elements are convertible, the adopted strategy is to replace some basic proof trees, called in this paper *effluences*, by other ones, called here *proofs by rewriting*. This is Church-Rosser's result, that we will generalize in next section. When all effluences can be replaced by proofs by rewriting for an \mathcal{SP} -rewrite system \mathcal{R} , \mathcal{R} is said *confluent*. To test automatically this property, effluences have to be eliminated step by step by replacing local effluences which are effluences whose derivations are restricted to rewriting steps (this is a general form of Newman's lemma that we will formally give and prove in the next section).

Therefore, the generalization of the confluence property requires us first to divide basic proofs into effluences and proofs by rewriting such that one must be able to generate proofs by rewriting from the decision procedure used to answer the validity problem of theories. Proofs by rewriting and effluences are then defined as follows:

DEFINITION 5.10 (Proof by rewriting and effluence). Proofs by rewriting and effluences form a partition \mathcal{PR} and $\mathcal{E}ff$ of \mathcal{BP} (the set of basic proofs) such that for any formula $p(u, v)$, the set $RS[p(u, v)]$ defined by:

$$\{u' \xleftrightarrow{*}_{\mathcal{R}} v' \mid \exists (d_1, \dots, d_n, u \xleftrightarrow{*}_{\mathcal{R}}^P v) \in \mathcal{PR}, \exists 1 \leq i \leq n, u' \xleftrightarrow{*}_{\mathcal{R}} v' \mathcal{E}_{\mathcal{R}} d_i\}$$

is finite and computable, that is we can construct the Turing-machine capable of listing all its members out of a given rewrite system.

A local effluence is an effluence of $\mathcal{E}ff$ of the form $(u_1 \xleftrightarrow{*}_{\mathcal{R}} v_1, \dots, u_n \xleftrightarrow{*}_{\mathcal{R}} v_n, u \xleftrightarrow{*}_{\mathcal{R}} v)$.

REMARK 5.11. Given a formula $p(u, v)$, the set $RS[p(u, v)]$ then contains all the rewriting steps that start the proof by rewriting with $u \xleftrightarrow{*}_{\mathcal{R}}^P v$ as conclusion.

The condition of Definition 5.10 will be useful to define a decision procedure used to answer the validity problem of theories (see the algorithm just below).

EXAMPLE 5.12. *In the equational rewriting setting, proofs by rewriting and effluences are valleys and peaks, respectively. Following our notations, peaks and valleys are defined as follows:*

- a peak is any basic proof tree of the form $\frac{d_1:u \xrightarrow{\ast} \overline{\mathcal{R}} w \quad d_2:w \xrightarrow{\ast} \overline{\mathcal{R}} v}{u \xrightarrow{\ast} \overline{\mathcal{R}} v}$
- a valley is either a derivation or any basic proof tree of the form $\frac{d_1:u \xrightarrow{\ast} \overline{\mathcal{R}} w \quad d_2:w \xrightarrow{\ast} \overline{\mathcal{R}} v}{u \xrightarrow{\ast} \overline{\mathcal{R}} v}$

This partition satisfies the condition of Definition 5.10. Indeed, suppose an equation $u = v$. Then, let us define the set S as follows:

$$S = \left\{ \begin{array}{c} \{u' \mid u \xrightarrow{\ast} \overline{\mathcal{R}} u'\} \\ \cup \\ \{v' \mid v \xrightarrow{\ast} \overline{\mathcal{R}} v'\} \end{array} \right\}$$

For any derivation $d : u \xrightarrow{\ast} \overline{\mathcal{R}} w$ ²¹ there is a term u' such that $d = \frac{d_1:u \xrightarrow{\ast} \overline{\mathcal{R}} u' \quad d_2:u' \xrightarrow{\ast} \overline{\mathcal{R}} w}{u \xrightarrow{\ast} \overline{\mathcal{R}} w}$.

This means, by the definition of the expanding position application $\mathcal{E}x$, that $d_1 : u \xrightarrow{\ast} \overline{\mathcal{R}} u' \mathcal{E}x d$. Consequently, we have:

$$RS[u = v] = \left\{ \begin{array}{c} \{u \xrightarrow{\ast} \overline{\mathcal{R}} u' \mid u' \in S\} \\ \cup \\ \{v \xrightarrow{\ast} \overline{\mathcal{R}} v' \mid v' \in S\} \end{array} \right\}$$

Moreover, for any rewrite system \mathcal{R} , the rewriting relation $\rightarrow_{\mathcal{R}}$ is finitely branching, that is each term has only finitely many direct successors which can be automatically listed. Therefore, $RS[u = v]$ is finite and computable.

On the contrary, if proofs by rewriting were peaks and effluences were valleys, then for some $u = v$, the set $RS[u = v]$ would not be computable anymore. Indeed, there could be an infinite set of terms t such that $u \xleftarrow{\ast} t \xrightarrow{\ast} v$.

Similarly, in the non-symmetric transitive rewriting setting, proofs by rewriting and effluences are elements in $(\xrightarrow{\ast} \overline{\mathcal{R}} \circ \xleftarrow{\ast} \overline{\mathcal{R}})$ and $\xleftarrow{\ast} \overline{\mathcal{R}} \circ \xrightarrow{\ast} \overline{\mathcal{R}}$, respectively. Then, suppose a formula $u \preceq v$ and define S as follows:

$$S = \left\{ \begin{array}{c} \{u' \mid u \xrightarrow{\ast} \overline{\mathcal{R}} u'\} \\ \cup \\ \{v' \mid v' \xleftarrow{\ast} \overline{\mathcal{R}} v\} \end{array} \right\}$$

²¹ Recall that $\xrightarrow{\ast} \overline{\mathcal{R}}$ is the transitive closure of $\rightarrow_{\mathcal{R}}$. Hence, $u \xrightarrow{\ast} \overline{\mathcal{R}} w$ contains at least a rewriting step.

For any derivation $d : u \xrightarrow{\mathcal{R}} w$ (resp. $d : u \xleftarrow{\mathcal{R}} w$) there is a term u' such that $d = \frac{d_1 : u \xrightarrow{\mathcal{R}} u' \quad d_2 : u' \xrightarrow{\mathcal{R}} w}{u \xrightarrow{\mathcal{R}} w}$ (resp. $d = \frac{d_1 : u \xleftarrow{\mathcal{R}} u' \quad d_2 : u' \xleftarrow{\mathcal{R}} w}{u \xleftarrow{\mathcal{R}} w}$). This means, by the definition of the expanding position application \overleftarrow{Ex} (resp. \overrightarrow{Ex}), that $d_1 : u \xrightarrow{\mathcal{R}} u' \overleftarrow{Ex} d$ (resp. $d_1 : u \xleftarrow{\mathcal{R}} u' \overrightarrow{Ex} d$). Consequently, we have:

$$RS[u = v] = \begin{cases} \{u \xrightarrow{\mathcal{R}} u' \mid u' \in \mathcal{S}\} \\ \cup \\ \{v' \xleftarrow{\mathcal{R}} v \mid v' \in \mathcal{S}\} \end{cases}$$

Moreover, for any rewrite system \mathcal{R} , both rewriting relations $\xrightarrow{\mathcal{R}}$ and $\xleftarrow{\mathcal{R}}$ are also finitely branching. Therefore, $RS[u = v]$ is finite and computable.

But, when dealing with conditional rewriting, the set $RS[u =_c v]$ is not necessarily computable. Indeed, performing rewriting steps requires to rewrite (join²² and normal rewriting) or to satisfy (natural rewriting) the premises of rewriting rules. In the literature, some conditions such as “decreasingness” [23], have been proposed so that joinability and rewriting are decidable.

Now, the join rewriting which had been left in pending in the previous section, can be defined. This gives rise to the following definition:

DEFINITION 5.13 (Join rewriting). *Let \mathcal{R} be an SP-rewrite system. For every $p \in E$, $\rightarrow_p^{\mathcal{R}}$ and $\xrightarrow{*}_p^{\mathcal{R}}$ are the least binary relations on T (according to the set-theoretical inclusion) inductively defined as follows:*

1. $\rightarrow_p \subseteq \rightarrow_p^{\mathcal{R}}$ and $\rightarrow_p^{\mathcal{R}} \subseteq \xrightarrow{*}_p^{\mathcal{R}}$, and
2. for every $\iota : p(t, t') \in RS$ (resp. $\iota : p(t, t') \in De$) such that:
 - for every leaf $p'(u, v) \in \mathcal{FL}(\iota)$, $u \xrightarrow{\mathcal{R}'} v$ (resp. $u \xrightarrow{*}_{\mathcal{R}'} v$), and
 - for every leaf $p'(u', v') \in \mathcal{L}(\iota) \setminus \mathcal{FL}(\iota)$, there is $\pi : u' \xrightarrow{*}_{\mathcal{R}'} v' \in \mathcal{PR}$,

we have $t \xrightarrow{\mathcal{R}'} t'$ (resp. $t \xrightarrow{*}_{\mathcal{R}'} t'$)

We define $\rightarrow_{\mathcal{R}} = \bigcup_{p \in E} \rightarrow_p^{\mathcal{R}}$ and $\xrightarrow{*}_{\mathcal{R}} = \bigcup_{p \in E} \xrightarrow{*}_p^{\mathcal{R}}$.

$\leftarrow_{\mathcal{R}} = \bigcup_{p \in E} \leftarrow_p^{\mathcal{R}}$ and $\xleftarrow{*}_{\mathcal{R}} = \bigcup_{p \in E} \xleftarrow{*}_p^{\mathcal{R}}$ are defined analogously.

²² A generalization of such a rewriting is given in Definition 5.13.

In the following of this section, we will only consider join rewriting.

In order to define a decision procedure from a given terminating rewrite system, the expanding relation $\mathcal{E}_{\mathcal{R}}$ of any \mathcal{SP} -rewrite system \mathcal{R} must satisfy two supplementary properties:

1. rewritings can be characterized by some derivations which can be generated step by step, and
2. when dealing with finite rewrite systems, the choice to extend derivations by rewriting steps is finite and computable, that is we can construct the Turing-machine capable of listing all its members out of a given rewrite system.

Formally, this is expressed as follows:

NOTATION 5.14. Let us define $\overleftrightarrow{\mathcal{E}x}$ to mean either $\overrightarrow{\mathcal{E}x}$ or $\overleftarrow{\mathcal{E}x}$.

DEFINITION 5.15 (Sensible rewrite system). An \mathcal{SP} -rewrite system \mathcal{R} is sensible if and only if its expanding relation $\mathcal{E}_{\mathcal{R}}$ satisfies both following properties:

1. **Generated step by step:** For any rewriting $u \xrightarrow{*}_{\mathcal{R}} v$, there is a derivation $d : u \xrightarrow{*}_{\mathcal{R}} v$ and a finite sequence of derivations $d_1 \mathcal{E}_{\mathcal{R}} \dots \mathcal{E}_{\mathcal{R}} d_n$ such that:
 - $d = d_n$,
 - $d_1 : u_1 \xrightarrow{\mathcal{R}} v_1$ is a rewriting step, and
 - for all i , $2 \leq i \leq n$, $d_i = (d'_1, \dots, d'_m, u_i \xrightarrow{*}_{\mathcal{R}} v_i)_1$ such that for every j , $1 \leq j \neq \overleftrightarrow{\mathcal{E}x}(\mathfrak{t}) \leq m$, d'_j is a rewriting step.
2. **Finite and computable derivation extension choice:** Let d be a derivation. Let us denote by $D[d]$ the set of derivations d' satisfying the following properties:
 - $d \mathcal{E}_{\mathcal{R}} d'$, and
 - d' is of the form $(d_1, \dots, d_n, \emptyset)_1$ such that for every j , $1 \leq j \neq \overleftrightarrow{\mathcal{E}x}(\mathfrak{t}) \leq n$, d_j is a rewriting step.

Then, for every derivation d , $D[d]$ is finite and computable.

$D[d]$ is the set of derivations which expand the derivation d by adding a rewriting step.

EXAMPLE 5.16. When dealing with transitive rewriting settings (i.e. both equational and non-symmetric transitive rewriting) for each rewriting $t \xrightarrow{*} t'$, since $\xrightarrow{*}$ is transitive, there exist n terms t_1, \dots, t_n such that:

$$\frac{\frac{t \rightarrow t_1 \quad t_1 \rightarrow t_2}{t \xrightarrow{*}_{\mathcal{R}} t_2} \quad t_2 \rightarrow t_3}{t \xrightarrow{*}_{\mathcal{R}} t_3} \quad \dots \quad \frac{t \xrightarrow{*}_{\mathcal{R}} t_n \quad t_n \rightarrow t'}{t \xrightarrow{*}_{\mathcal{R}} t'}$$

Finally, given a derivation $d : u \xrightarrow{*}_{\mathcal{R}} v$, we have seen in Example 5.12 that v can be rewritten in one step, into a finite set of terms v' , that is $d \mathcal{E}_{\mathcal{R}} \frac{d:u \xrightarrow{*}_{\mathcal{R}} v \quad d':v \rightarrow_{\mathcal{R}} v'}{d:u \xrightarrow{*}_{\mathcal{R}} v'}$.

Moreover, generating the set of terms v' such that $v \rightarrow_{\mathcal{R}} v'$ is a finite and computable task. Therefore, $D[d]$ defined as the set of derivations of the form $\frac{d:u \xrightarrow{*}_{\mathcal{R}} v \quad d':v \rightarrow_{\mathcal{R}} v'}{u \xrightarrow{*}_{\mathcal{R}} v'}$, is also finite and computable.

Based on the sensible \mathcal{SP} -rewrite system $\mathcal{R} = (\rightarrow_p, \leftarrow_p)_{p \in E}$, a decision procedure for its corresponding theory $Th(\Gamma) = \{p(u, v) \mid \Gamma \vdash p(u, v)\}$ where $\Gamma = \{p(u, v) \mid u \Leftarrow_p v \in \mathcal{R}\}$, can be defined. To prove $\Gamma \vdash p(u, v)$, we perform the following algorithm \mathcal{A} on $p(u, v)$, written $\mathcal{A}[p(u, v)]$:

Input a rewrite system \mathcal{R} and a formula $p(u, v)$.

Initialization $S := RS[p(u, v)]$, $Tmp := RS[p(u, v)]$, and $answer := false$;

Loop while $Tmp \neq \emptyset$ do:

- 1) choose d in Tmp and $Tmp := Tmp \setminus \{d\}$;
- 2) $Tmp := Tmp \cup D[d]$; (cf. Definition 5.15)
- 3) $S := S \cup Tmp$;
- 4) if there is $(d_1, \dots, d_n, u \xrightarrow{*}_{\mathcal{R}}^p v)_i \in \mathcal{PR}$ such that:
 - $d_i \in S$ ($1 \leq i \leq n$), and
 - $\forall p'(u', v') \in \mathcal{L}(i) \setminus \mathcal{FL}(i)$, $\mathcal{A}[p'(u', v')]$

then $Tmp := \emptyset$;

$answer := true$;

end of loop

Output $return(answer)$

The above procedure calls for some comments:

- As assured by Point 4) in the algorithm, proofs generated by the above decision procedure are proofs by rewriting.
- The kind of rewriting which is taken into account by the above procedure is the join one. Indeed, as this is expressed by the second bullet under Item 4, we apply the algorithm \mathcal{A} to each formula $p'(u', v') \in \mathcal{L}(\mathfrak{t}) \setminus \mathcal{F}\mathcal{L}(\mathfrak{t})$ with $p' \in E$. Hence, we try to find a proof by rewriting $\pi : u' \xrightarrow{*}^r_{\mathcal{R}} v'$.

EXAMPLE 5.17. *When dealing with transitive relations, the direct instantiation of the above procedure is as follows:*

Initialization $S := \{d \mid \exists w, d : u \rightarrow^r_{\mathcal{R}} w \vee d : w \leftarrow^r_{\mathcal{R}} v\}$ and $Tmp := \{d \mid \exists w, d : u \rightarrow^r_{\mathcal{R}} w \vee d : w \leftarrow^r_{\mathcal{R}} v\}$ with $r \in \{=, \preceq\}$, and $answer := false$.

Loop while $Tmp \neq \emptyset$ do:

- 1) choose $d : a \xrightarrow{*}^r_{\mathcal{R}} b$ in Tmp and $Tmp := Tmp \setminus \{d\}$; ²³
- 2) $Tmp := Tmp \cup \{d' : a \xrightarrow{*}^r_{\mathcal{R}} c \mid b \xrightarrow{*}^r_{\mathcal{R}} c \wedge (a \xrightarrow{*}^r_{\mathcal{R}} b \Leftrightarrow b \rightarrow^r_{\mathcal{R}} c)\}$; ²⁴
- 3) $S := S \cup Tmp$.
- 4) if there are $u \xrightarrow{*}^r_{\mathcal{R}} c$ and $c \xrightarrow{*}^r_{\mathcal{R}} v$ in S then $Tmp := \emptyset$;
 $answer := true$;

end of loop

Output $return(answer)$

The above procedure defines a breadth first search proof for the theory \mathcal{R} .

Note that the above algorithm is recursive, and hence can loop even when rewrite systems are terminating. Indeed, nothing prevents from applying again $\mathcal{A}[p(u, v)]$ in the execution of $\mathcal{A}[p'(u', v')]$. A sufficient condition to prevent such a situation is the following:

DEFINITION 5.18 (Adequacy). *Let \mathcal{SP} be an rfs. Let \bowtie be a binary relation on formulae defined by:*

$$p(u, v) \bowtie p'(u', v') \iff \exists \mathfrak{t} : p(u, v) \in De, p'(u', v') \in \mathcal{L}(\mathfrak{t}) \setminus \mathcal{F}\mathcal{L}(\mathfrak{t})$$

Therefore, \mathcal{SP} is adequate if and only if the transitive closure \bowtie^+ of \bowtie is Noetherian.

²³ a is necessarily either u or v .

²⁴ The last condition means that both rewritings are in the same direction.

The intuition under Definition 5.18 is that in any proof $\pi : u \xrightarrow{*}_{\mathcal{R}}^P v$, there is no subproof $(\pi_1, \dots, \pi_n, u' \xrightarrow{*}_{\mathcal{R}}^{P'} v')$ such that $p(u, v)$ belongs to unfixed leaves of \mathfrak{t} (i.e. $p(u, v) \in \mathcal{L}(\mathfrak{t}) \setminus \mathcal{FL}(\mathfrak{t})$). Would such a subproof exist, solving $p(u, v)$ by rewriting should require to search a proof by rewriting $\pi : u \xrightarrow{*}_{\mathcal{R}}^P v$ and would then loop.

This condition is obviously satisfied by all the examples developed up to here in this paper because the set of unfixed leaves of transitivity instances is empty. With conditional rewriting, undecidability problems of joinability have been relegated to the level of the execution of rewriting steps (see Example 5.12).

Given an rfs $\mathcal{SP} = (T, E, RS, De, Rmv)$, its adequacy can be obviously checked when the set E is finite and we are dealing with a finite set of inference rule schemas.

We are in a position to define properties of rewrite systems.

DEFINITION 5.19 (Properties of rewrite systems). *An \mathcal{SP} -rewrite system \mathcal{R} is:*

- confluent (*resp.* locally confluent) *if for any effluence (*resp.* local effluence) there is an equivalent proof by rewriting, and*
- Church-Rosser *if for any proof there is an equivalent proof by rewriting.*

For any adequate rfs \mathcal{SP} , Church-Rosser and terminating rewrite systems can be used to answer the validity problem as this is expressed by the following result:

THEOREM 5.20. *Let \mathcal{SP} be an adequate rfs. If the \mathcal{SP} -rewrite system $\mathcal{R} = (\rightarrow, \leftarrow)$ is Church-Rosser then $\Gamma \vdash p(u, v)$ if and only if the procedure defined above terminates and answers true. Moreover, if \mathcal{R} is Church-Rosser and terminates, then the validity problem is decidable.*

Proof Theorem 5.20 is composed of two properties. Let us prove the first one defined by the following equivalence:

$\Gamma \vdash p(u, v)$ ($p \in E$) *if and only if the procedure defined above terminates and answers true*

The *if* implication is easily proven from Point 4) of the procedure. By the generalized version of the logicality theorem established in [3], when $\Gamma \vdash p(u, v)$, we have necessarily $u \xrightarrow{*}_{\mathcal{R}}^P v$. As \mathcal{R} is Church-Rosser, then there is

a proof by rewriting $\pi : u \xrightarrow{\mathcal{R}}^p v$ which establishes the statement $\Gamma \vdash p(u, v)$. Therefore, let us prove the *only if* implication by induction with respect to the ordering \bowtie . Let $\pi = (d_1, \dots, d_n, u \xrightarrow{\mathcal{R}}^p v)_t$ be a proof by rewriting. From both constraints requiring that $RS[p(u, v)]$ and $D[d]$ are finite and computable for every derivation d , each derivation d_i ($1 \leq i \leq n$) will be generated in a finite time by the above decision procedure. Moreover, for every $p'(u', v') \in \mathcal{L}(t) \setminus \mathcal{F}\mathcal{L}(t)$, by induction hypothesis, the decision procedure terminates and answers true. Therefore, π will be generated in a finite time by the above decision procedure.

The second property is given by the implication:

if \mathcal{R} is Church-Rosser and terminates, then the validity problem is decidable

As \mathcal{R} is terminating and from all explanations given just before, given a formula $p(u, v)$, the set of proofs by rewriting $\pi : u \xrightarrow{\mathcal{R}}^p v$ can be generated in a finite time, when they exist. Therefore, for any $p(u, v)$, the algorithm answers in a finite time that the statement $\Gamma \vdash p(u, v)$ is true or not. \square

6. Generalization of Church-Rosser's result and Newman's lemma

Congruences (defined by a set of equations), which are a typical example of algebraic structure where rewriting has been intensively studied and applied, have a number of (implicit) properties which are useful for rewriting but which are not necessarily satisfied in all rfs (as they are not required in Definition 3.1). In this section, we will thus give, via a set of axioms, the conditions needed for a rfs to satisfy these useful properties. First of all, Church-Rosser's result establishes a correspondence between Church-Rosser systems and confluent systems. Obviously, Church-Rosser systems are confluent systems because effluences are peculiar proofs. The opposite implication is more difficult. In the transitive rewriting setting, this implication has a diagrammatic proof based on the two following observations:

1. proofs can be written as a series of maximal peaks (i.e. peaks which are not contained in another one) ;
2. replacement of any maximal peak by a valley decreases the number of maximal peaks.²⁵

In all rewriting theories, these two basic requirements are behind all proofs of Church-Rosser's result. Their abstraction requires first to give an abstract formulation of maximal peaks. This leads to the following definition:

²⁵ Actually, this number decreases of one unit.

DEFINITION 6.1 (Maximal effluences). Let $\pi : \wp$ and $\pi' : \wp$ be two proofs with $\mathcal{L}(\pi) = \mathcal{L}(\pi')$. Let $\omega \in \mathbb{N}^*$ be a position such that $\pi'_{|\omega}$ is an effluence. The pair (π', ω) is a maximal effluence of π if and only if there are no other pairs (π'', ω') satisfying the same conditions as (π', ω) and such that $\mathcal{L}(\pi'_{|\omega}) \subset \mathcal{L}(\pi''_{|\omega'})$.

Let us denote $NEM(\pi)$ the number of maximal effluences in π . The notation $\pi \leq_{em} \pi'$ means that $NEM(\pi) \leq NEM(\pi')$.

EXAMPLE 6.2. Let π be the tree obtained from the following rewriting sequence by applying transitivity instances from left to right:

$$t_1 \rightarrow t_2 \leftarrow t_3 \leftarrow t_4 \rightarrow t_5 \leftarrow t_6 \rightarrow t_7$$

The following pairs are examples of effluences of π :

$$\begin{array}{l}
 1. \left(\frac{\frac{\frac{t_1 \rightarrow t_2 \leftarrow t_3}{t_1 \overset{*}{\leftrightarrow} t_3} \quad \frac{t_3 \leftarrow t_4 \rightarrow t_5}{t_3 \overset{*}{\leftrightarrow} t_5}}{t_1 \overset{*}{\leftrightarrow} t_5} \quad \frac{t_5 \leftarrow t_6 \rightarrow t_7}{t_5 \overset{*}{\leftrightarrow} t_7}}{t_1 \overset{*}{\leftrightarrow} t_7}, \quad 0.1 \right) \\
 2. \left(\frac{\frac{\frac{t_2 \leftarrow t_3 \leftarrow t_4}{t_2 \overset{*}{\leftrightarrow} t_4} \quad t_4 \rightarrow t_5}{t_2 \overset{*}{\leftrightarrow} t_5} \quad \frac{t_5 \leftarrow t_6 \rightarrow t_7}{t_5 \overset{*}{\leftrightarrow} t_7}}{t_1 \overset{*}{\leftrightarrow} t_7}, \quad 0.1 \right) \\
 3. \left(\frac{\frac{\frac{t_1 \rightarrow t_2 \leftarrow t_3}{t_1 \overset{*}{\leftrightarrow} t_3} \quad t_3 \leftarrow t_4}{t_1 \overset{*}{\leftrightarrow} t_4} \quad \frac{t_4 \rightarrow t_5}{t_4 \rightarrow t_5} \quad \frac{t_5 \leftarrow t_6 \rightarrow t_7}{t_5 \overset{*}{\leftrightarrow} t_7}}{t_1 \overset{*}{\leftrightarrow} t_7}, \quad 1 \right)
 \end{array}$$

But only the second and the third ones are maximal effluences of π (not the first one because $\{t_3 \leftarrow t_4, t_4 \rightarrow t_5\}$ is included in $\{t_2 \leftarrow t_3, t_3 \rightarrow t_4, t_4 \rightarrow t_5\}$).

Given a finite set S of rewriting steps, there will be a finite set (possibly empty) of maximal effluences with premises in S if we can only build a finite set of proofs from S . This will be our first postulate:

AXIOM 1. *Given a finite set S of rewriting steps, there exists a finite set of proofs π with $\mathcal{L}(\pi) \subseteq S$.*

In our general setting, the first requirement above (i.e. writing proofs as a series of maximal peaks) obviously holds for any proof that does not contain instances of rules in Rmv . This is a consequence of the fact that effluences and proofs by rewriting form a partition of all basic proofs. Therefore, any proof without maximal effluences nor instances of rules in Rmv , is necessarily a proof by rewriting. But, when dealing with proofs containing some instances of rules in Rmv , we have to impose the following axiom:

AXIOM 2. *For any proof $(\pi_1, \dots, \pi_n, u \xrightarrow{*P} v)_\iota$ where each $\pi_i \in \mathcal{PR}$ ($1 \leq i \leq n$) and $\iota \in Rmv$, there is $\pi : u \xrightarrow{*P} v \in \mathcal{PR}$.*

EXAMPLE 6.3. *In the equational rewriting setting, Axiom 2 obviously holds. Indeed, for any $u \xrightarrow{*R} w \xleftarrow{*R} v$, we also have the valley $v \xrightarrow{*R} w \xleftarrow{*R} u$. In the non-symmetric transitive rewriting setting, Axiom 2 is meaningless because Rmv is empty.*

In our general setting, the second requirement (i.e. replacing a maximal proof in a proof by a rewriting proof decreases the number of maximal proofs) does not necessarily hold. Therefore, it has to be imposed:

AXIOM 3 (Cut in maximal effluences). *For any proof π , the replacement of one of its maximal effluences (π', w) by an equivalent proof by rewriting π'' (i.e. with the same conclusion as $\pi'_{|\omega}$) decreases the number of maximal effluences: $\pi'[\pi'']_{\omega} <_{em} \pi'$.*

EXAMPLE 6.4. *In the transitive rewriting setting, Axiom 3 holds. Indeed, we saw previously that proofs can be written as a series of maximal peaks. It is clear that the replacement of any maximal peak by a valley reduces by one unit the number of maximal peaks.*

Axioms 1-3 induce Church-Rosser's result:

THEOREM 6.5 (Generalized Church-Rosser's result). *For every rewrite system \mathcal{R} for which Axioms 1-3 hold, \mathcal{R} is Church-Rosser if and only if \mathcal{R} is confluent.*

Proof *The only if part.* Obvious.

The if part. Proved by induction on the number of maximal effluences in a proof.

- If π has no maximal effluences, then π is either a proof by rewriting or is of the form $(\pi_1, \dots, \pi_n, u \xrightarrow[\mathcal{R}]{*P} v)_\iota$ where each π_i is a proof by rewriting and $\iota \in Rmv$. The last case is then solved by Axiom 2.
- Let (π', ω) be a maximal effluence of π . Let π'' be a proof by rewriting with the same conclusion as π'_ω . By Axiom 3, $\pi'[\pi'']_\omega <_{em} \pi'$. Hence, because we have $\mathcal{L}(\pi) = \mathcal{L}(\pi')$, $\pi'[\pi'']_\omega <_{em} \pi$. By induction hypothesis, there exists a proof by rewriting with the same conclusion as $\pi'[\pi'']_\omega$, and thus with the same conclusion as π .

□

Newman's lemma makes sense provided that all effluences contain a local effluence. This leads to the third axiom:

AXIOM 4 (Existence of local effluence). *Any effluence $\pi : \wp$ contains a local effluence, that is there is a pair $(\pi' : \wp, \omega)$ such that $\mathcal{L}(\pi) = \mathcal{L}(\pi')$ and π'_ω is a local effluence.*

EXAMPLE 6.6. *In the transitive rewriting setting, every peak contains a unique local peak.*

In the transitive rewriting setting, Newman's lemma holds because termination of rewrite systems induces a Noetherian relation \gg on peaks defined as follows: $u \xleftarrow{*} t \xrightarrow{*} v \gg u' \xleftarrow{*} t' \xrightarrow{*} v'$ if and only if there exists a finite sequence of proofs $(\pi_1 : u \xrightarrow{*} v, \dots, \pi_n : u \xrightarrow{*} v)$ such that $\pi_1 = u \xleftarrow{*} t \xrightarrow{*} v$, for every i , $1 \leq i \leq n$, π_i has been obtained from π_{i-1} by replacing a local peak by an equivalent valley (i.e. with the same conclusion), and $u' \xleftarrow{*} t' \xrightarrow{*} v'$ is a maximal peak of π_n .

The abstract form of \gg is defined as follows:

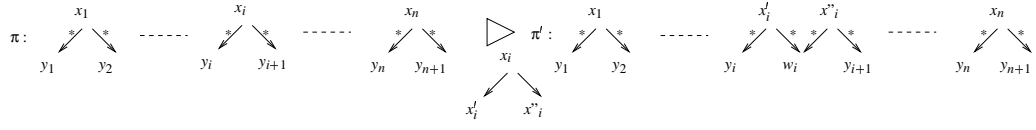
DEFINITION 6.7 (Relation on effluences). *Let \mathcal{R} be a rewrite system. Let \triangleright be the binary relation on proofs defined as follows: $\pi_1 \triangleright \pi_2$ if and only if there is a local effluence (π, ω) of π_1 , and a proof by rewriting π' with the same conclusion as π'_ω such that $\pi_2 = \pi[\pi']_\omega$.*

We denote by $\pi_1 \triangleright_{(\pi, \omega)} \pi_2$ the fact that π_2 has been obtained from π_1 by reducing the local effluence (π, ω) .

Therefore, let us denote \gg the binary relation on effluences defined as follows: $\pi_1 \gg \pi_2$ if and only if there is a proof π' and a maximal effluence (π', ω) of π' such that $\pi_1 \triangleright^ \pi'$,²⁶ and $\pi_2 = \pi'_\omega$.*

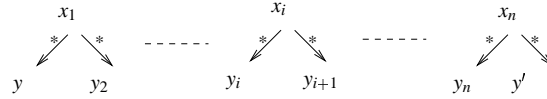
EXAMPLE 6.8. *In the transitive rewriting setting, \triangleright is defined as follows:*

²⁶ \triangleright^* is the reflexive and transitive closure of \triangleright .

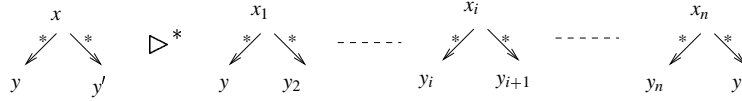


where the local peak $x_i' \leftarrow x_i \rightarrow x_i''$ in π has been replaced by the valley $x_i' \rightarrow w_i \leftarrow x_i''$.

Therefore, \gg is defined on peaks as follows: $y \leftarrow x \rightarrow y' \gg y_i \leftarrow x_i \rightarrow y_{i+1}$ if and only if there exists a proof



such that



PROPOSITION 6.9. \gg is transitive.

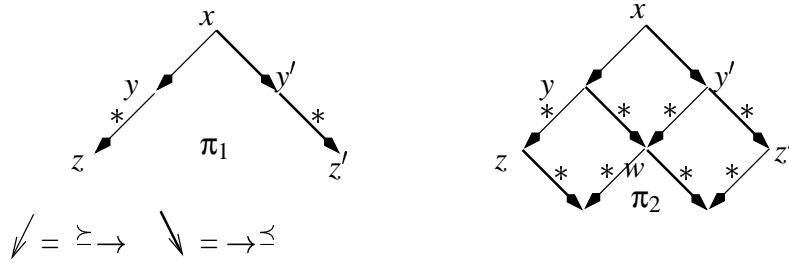
Proof Suppose $\pi_1 \gg \pi_2$ and $\pi_2 \gg \pi_3$. By Definition 6.7, $\pi_1 \gg \pi_2$ (resp. $\pi_2 \gg \pi_3$) means that there are a sequence $\pi_{11} \triangleright (\pi'_{11}, \omega_{11}) \pi_{12} \triangleright (\pi'_{12}, \omega_{12}) \cdots \triangleright (\pi'_{1n_1-1}, \omega_{1n_1-1}) \pi_{1n_1}$ (resp. $\pi_{21} \triangleright (\pi'_{21}, \omega'_{21}) \pi_{22} \triangleright (\pi'_{22}, \omega'_{22}) \cdots \triangleright (\pi'_{2n_2-1}, \omega'_{2n_2-1}) \pi_{2n_2}$) and a maximal effluence (π'_1, ω'_1) of π_{1n_1} (resp. (π'_2, ω'_2) of π_{2n_2}) such that $\pi_{11} = \pi_1$ and $\pi_2 = \pi'_{1|\omega'_1}$ (resp. $\pi_{21} = \pi_2$ and $\pi_3 = \pi'_{2|\omega'_2}$). $(\pi'_1[\pi'_{21}]_{\omega'_1}, \omega'_1 \cdot \omega_{21})$ is a local effluence of π_{1n_1} . More generally, we have for all i , $1 \leq i \leq n_2$, that $(\pi'_1[\pi'_{2i}]_{\omega'_1}, \omega'_1 \cdot \omega_{2i})$ is a local effluence of $\pi'_1[\pi_{2i-1}]_{\omega'_1}$. Consequently, we have

$$\pi'_{11}[\pi_{22}]_{\omega'_1} \triangleright (\pi'_1[\pi_{22}]_{\omega'_1}, \omega'_1 \cdot \omega_{22}) \cdots \triangleright (\pi'_1[\pi_{2n_2-1}]_{\omega'_1}, \omega'_1 \cdot \omega_{2n_2-1}) \pi'_{1n_1}[\pi_{2n_2}]_{\omega'_1}. \quad \square$$

In the equational rewriting setting, under the termination condition of rewrite systems, \gg is Noetherian because, for any peak $z \leftarrow y \leftarrow x \rightarrow y' \rightarrow z'$, when replacing the local peak $y \leftarrow x \rightarrow y'$ by any valley $y \rightarrow w \leftarrow y'$, we obtain two effluences $z \leftarrow y \rightarrow w$ and $w \leftarrow y' \rightarrow z'$, that is $z \leftarrow y \leftarrow x \rightarrow y' \rightarrow z' \gg z \leftarrow y \rightarrow w$ and $z \leftarrow y \leftarrow x \rightarrow y' \rightarrow z' \gg w \leftarrow y' \rightarrow z'$. When we do not have $y = z$ (resp. $y' = w$) nor $y = w$ (resp. $y' = z'$), then there exist intermediate points i_1 and i_2 (resp. i'_1 and i'_2) such that $w \leftarrow i_2 \leftarrow y \rightarrow i_1 \rightarrow z$ (resp. $w \leftarrow i'_1 \leftarrow y' \rightarrow i'_2 \rightarrow z'$). Obviously, we have the following derivations $x \rightarrow y \rightarrow i_1$ (resp. $x \rightarrow y' \rightarrow i'_1$) and $x \rightarrow y \rightarrow i_2$ (resp. $x \rightarrow y' \rightarrow i'_2$).

By repeatedly applying this kind of replacement, for any infinite sequence $\pi_1 \gg \dots \gg \pi_n \gg \dots$ we can generate an infinite derivation by connecting together the branches of transformed local peaks, and then termination implies that \gg is Noetherian. Therefore, \gg describes an algorithm that eliminates peaks, step by step. To define this algorithm, we need to use the local confluence of the rewrite system. Its termination results from the termination of the system.

However, in the non-symmetric transitive rewriting setting, $\pi_1 \gg \pi_2$ does not mean there exists a derivation built from rewriting steps of the local peaks which have been reduced to obtain π_2 . Indeed, we can have the following situation: $\pi_1 \gg \pi_2$ where



But, $x \xrightarrow{\lambda} y \xrightarrow{*} w$ is not a derivation. However, we have $x \succ_{\mathcal{R}} y \succ_{\mathcal{R}} w$ where $\succ_{\mathcal{R}} = (\rightarrow_{\mathcal{R}} \cup \rightarrow_{\mathcal{R}}^*)$. Consequently, the termination of rewrite systems also implies that \gg is Noetherian.

This will define our fourth axiom. Briefly, Axiom 5 will express that, for any $\pi \gg \pi'$, there exists an order of reduction of local effluences (i.e. a finite sequence $\pi \triangleright (\pi'_1, \omega_1) \dots \triangleright (\pi'_{m-1}, \omega_{m-1}) \pi''$ s.t. π'' is a maximal effluence of π') such that we can build a reduction by connecting together some rewriting steps (transformed into reductions) of each reduced local effluence (π'_i, ω_i) . This is expressed as follows:

AXIOM 5 (Connection). *Let \mathcal{R} be an \mathcal{SP} -rewrite system. For any $\pi_1 \gg \pi_2 = (d_1, \dots, d_n, \Phi)$ such that each d_i for $i = 1, \dots, n$ stands for a derivation, there exists a finite sequence of proofs*

$$\pi_1 \triangleright (\pi'_1, \omega_1) \dots \triangleright (\pi'_{m-1}, \omega_{m-1}) \pi_m$$

and a sequence of reductions $r_1 \mathcal{E}_{\succ_{\mathcal{R}}} \dots \mathcal{E}_{\succ_{\mathcal{R}}} r_m$ satisfying the following statement: if \bar{d} is the reduction obtained from d by replacing every node and every leaf $u \xrightarrow{*} v$ (resp. $u \xleftarrow{*} v$) by $u \succ v$ (resp. $u \prec v$), then

- π_2 is a maximal effluence of π_m ,

- there exists a rewriting step $rs \in \mathcal{L}(\pi'_{1|\omega_1})$ s.t. $r_1 = \overline{rs}$,²⁷
- for each $r_j = (r_1^j, \dots, r_{p_j}^j, \Phi_j)_{\mathfrak{v}}$ $2 \leq j \leq m$, and for every k , $1 \leq k \neq \vec{\mathcal{E}x}(\mathfrak{v}) \leq p_j$, there exists $rs^j \in \mathcal{L}(\pi'_{j|\omega_j})$ s.t. $r_k^j = \overline{rs^j}$, and
- for $r_m = (r_1^m, \dots, r_{p_m}^m, \Phi_m)_{\mathfrak{v}}$, and for any $1 \leq k \neq \vec{\mathcal{E}x}(\mathfrak{v}) \leq p_m$, $r_k^m \in \{\overline{d_1}, \dots, \overline{d_n}\}$.

EXAMPLE 6.10. In the transitive rewriting setting, given a rewrite system \mathcal{R} , for any $\pi_1 \gg \pi_2$ where $\pi_2 = y \xleftarrow{*} \mathcal{R} x \xrightarrow{*} \mathcal{R} y'$ there exists, by definition, a sequence S of proofs of the form:

$$\pi_1 \triangleright_{x_1'' \leftarrow \mathcal{R} x_1 \rightarrow \mathcal{R} x_1'} \pi_2 \triangleright_{x_2'' \leftarrow \mathcal{R} x_2 \rightarrow \mathcal{R} x_2'} \dots \triangleright_{x_{n-1}'' \leftarrow \mathcal{R} x_{n-1} \rightarrow \mathcal{R} x_{n-1}'} \pi_n'$$

such that π_2 is a maximal peak of π_n' . Therefore, let us define from S the following finite set of proof sequences as follows:

- $S_0 = \pi_{j_0}$ where $j_0 = \min\{2 \leq j \leq n \mid \pi_2 \text{ is a maximal peak of } \pi_j'\}$, and
- $S_{i+1} = \pi_{j_{i+1}} \triangleright_{x_{j_{i+1}}'' \leftarrow \mathcal{R} x_{j_{i+1}} \rightarrow \mathcal{R} x_{j_{i+1}}'} S_i$ such that $2 \leq j_{i+1} \leq j_i$ satisfies either $x_{j_i} = x_{j_{i+1}}'$ or $x_{j_i} = x_{j_{i+1}}''$

The process is stopped when at a step k there is no longer j_k , $1 \leq j_k \leq j_{k-1}$, satisfying either $x_{j_{k-1}} = x_{j_k}'$ or $x_{j_{k-1}} = x_{j_k}''$.

Hence, S_k is the minimal sequence ordered by \triangleright that can be associated to $\pi_1 \gg \pi_2$ where only reductions of local peaks which allow us to reach π_2 are preserved. Other reductions of local peaks are removed from S .

By definition, for any sequence S_i we have the reduction $x_{j_i} \rightarrow \mathcal{R} x_{j_{i-1}} \rightarrow \mathcal{R} \dots \rightarrow \mathcal{R} x_{j_1}$. Moreover, if k is the last step of the above process, then we have respectively that $\pi_{j_k} = \pi_1$ because π_1 is a peak and that either $x = x_{j_1}'$ or $x = x_{j_1}''$. Therefore, we can define $k-1$ reductions as follows:

- $r_1 = x_{j_k} \succ_{\mathcal{R}} x_{j_{k-1}}$, and
- $r_i = r_{i-1} \succ_{\mathcal{R}} x_{j_{k-i}}$

Let us define $r_k = r_{k-1} \succ_{\mathcal{R}} x \succ_{\mathcal{R}} z$ with $z \in \{y, y'\}$. Thus, we have $r_1 \mathcal{E}_{\mathcal{R}} \dots \mathcal{E}_{\mathcal{R}} r_k$. This proves that Axiom 5 holds in the transitive rewriting setting.

PROPOSITION 6.11. For any rewrite system \mathcal{R} for which Axiom 5 holds, if \mathcal{R} is terminating then \gg is Nætherian.

²⁷ Let us recall that rewriting steps are basic derivations.

Proof Let $\pi_1 \gg \pi_2 \gg \dots \gg \pi_n \gg \dots$ be a non-stationary infinite sequence of effluences. By Proposition 6.9, \gg is transitive. Therefore, by Axiom 5, we can build a non-stationary infinite sequence of reductions

$$r_1 \mathcal{E}_{\mathcal{R}} r_2 \mathcal{E}_{\mathcal{R}} \dots \mathcal{E}_{\mathcal{R}} r_n \mathcal{E}_{\mathcal{R}} \dots$$

□

COROLLARY 6.12. *If \mathcal{R} is terminating then \triangleright is Noetherian.*

This last result enables us to obtain our expected theorem:

THEOREM 6.13 (Generalized Newman's lemma). *For any rewrite system \mathcal{R} for which Axioms 1-5 hold, if \mathcal{R} is terminating then confluence follows from local confluence.*

Proof Suppose that \mathcal{R} is terminating and locally confluent. Then, by Proposition 6.11, \gg is Noetherian. Therefore, we will prove by well-founded induction on \gg that every effluence π has the property P defined by: $P(\pi)$ if and only if there exists a proof by rewriting with the same conclusion as π .

- Since \mathcal{R} is locally confluent, minimal elements with respect to \gg are local effluences. Obviously, $P(\pi)$ holds for each local effluence π .
- Let π be an effluence. By Axiom 4, π has a local effluence (π', ω) . Let us denote by $\pi_{loc} = \pi'_{|\omega}$. By hypothesis, π_{loc} has an equivalent proof by rewriting π_{rew} . By Definition 6.7, the replacement of π_{loc} by π_{rew} in π' yields a proof whose maximal effluences are smaller than π with respect to \gg . Thus, by induction hypothesis, each one of these maximal effluences can be replaced by an equivalent proof by rewriting. Moreover, by Axiom 3, such replacements reduce the number of maximal effluences of the initial proof.

We thus obtain a *proof by rewriting* equivalent to π in a finite number of step (new maximal effluences being smaller with respect to \gg and their number decreasing).

□

Theorem 6.13 is a basis for an abstract formulation of Knuth-Bendix completion. This will be given in Section 8.

7. Abstract rewriting modulo theories and more

This section is devoted to generalize, in our abstract framework, the notion of rewriting modulo theories. The way we will handle rewriting modulo theories in our abstract framework will also allow us to deal with rewriting of binary relations constrained by any other n -ary relations such as the definedness predicate in the framework of the partial function logic, or the membership predicate in the framework of label algebras [14].

Theories may contain formulæ which are inherently nonterminating (e.g. when dealing with commutative operators in the equational rewriting setting). As this is usual in the classical term rewriting theory, the obvious idea is not to use these non-orienting formulæ directly, but to take them into account when applying some other rewrite rules.

Here, we propose to apply this idea to our abstract rewriting. To achieve this purpose, we extend Definition 3.1 as follows:

DEFINITION 7.1 (Extended rewrite formal systems). *An extended rewrite formal system (erfs) is a 7-tuple $(T, E, P, RS, De, Rmv, Oth)$ ²⁸ such that T is a set, E is a set of binary relations on T , P is a set of n -ary relations on T disjoint of E , and RS, De, Rmv and Oth are four disjoint sets of n -ary relations on the set $F = \{p(u_1, \dots, u_n) \mid p \in E \cup P \wedge (u_1, \dots, u_n) \in p\}$ satisfying:*

- all instances in $RS \cup De \cup Rmv$ have conclusions of the form $p(u, v)$ with $p \in E$, and
- all instances in Oth have conclusions of the form $p(u_1, \dots, u_n)$ with $p \in P$.

EXAMPLE 7.2 (Rewriting modulo equational theories). *The erfs for term rewriting modulo equational theories is defined for any signature $\Sigma = (F, V)$ by the tuple $(T, E, P, RS, De, Rmv, Oth)$ such that:*

- T is the standard set of terms with generators in V ,
- $E = \{=\}$ such that $= \stackrel{def}{=} T \times T$,
- $P = \{\approx\}$ such that $\approx \stackrel{def}{=} T \times T$,
- RS is the set containing all instances of substitution and context rules, and the two following deductive rules:

$$\frac{s \approx s' \quad s' = t'}{s = t'} \quad \frac{s = t' \quad t' \approx t}{s = t}$$

²⁸ Oth for Other rules.

- *De* is the set of all instances of transitivity and reflexivity rules of $=$,
- *Rmv* is the set of all instances of symmetry rule of $=$, and
- *Oth* is the set of all instances of rules defined by the usual deductive rules of equational reasoning applying to equations of the form $t \approx t'$.

EXAMPLE 7.3 (Partial function logic). *Let us define the erfs for partial function logic with definedness, total valuation and existential equality [5]. First, we recall the basic notions and notations of this logic. Signatures are usual signatures, that is a set of partial function names equipped with an arity in \mathbb{N} . Given a set of variables V , $T_{\Sigma}(V)$ denotes the standard set of terms with generators in V . Given a term t in $T_{\Sigma}(V)$, $\text{Var}(t)$ denotes the set of variables occurring in t . Formulae are either equations $t = t'$ where t and t' are terms in $T_{\Sigma}(V)$, or formulae $D(t)$ where t is a term in $T_{\Sigma}(V)$. Semantically, $t = t'$ states that both sides of the equality are defined and denote the same value and $D(t)$ states that t is defined, i.e., t necessarily yields a value when it is evaluated. A substitution is a function $\sigma : V \rightarrow T_{\Sigma}(V)$. It is naturally extended to terms and formulae. Given a set of formulae Γ , a substitution σ is defined with respect to Γ if and only if for all $x \in V$, we have $\Gamma \vdash D(\sigma(x))$. Given a signature Σ , we define the erfs for partial function logic by the tuple $(T, E, P, RS, De, Rmv, Oth)$ such that:*

- T is $T_{\Sigma}(V)$,
- $E = \{=\}$ such that $= \stackrel{def}{=} T \times T$,
- $P = \{D\}$ such that $D \stackrel{def}{=} T$ (syntactic definition),²⁹
- RS is the set of all instances of the following deductive rules:

Replacement $\frac{t=t' \quad D(f(\dots, t, \dots))}{f(\dots, t, \dots) = f(\dots, t', \dots)}$

Substitution $\frac{t=t'}{\sigma(t) = \sigma(t')}$ where σ is a defined substitution

- *De* is the set of all instances of the transitivity rule and the following deductive rule:

$$\frac{D(t)}{t = t}$$

- *Rmv* is the set of all instances of the symmetry rule, and

²⁹ $D \stackrel{def}{=} T$ means that every axiom $D(t)$ is syntactically correct. Of course, this does not imply that all terms are defined.

– *Oth* is the set of all instances of rules defined by the following deductive rules:

$$\text{Strictness } \frac{D(f(t_1, \dots, t_n))}{D(t_i)} \quad \text{Existential equality } \frac{t=t'}{D(t)}$$

Rewrite systems are extended by adding a set Δ of formulæ of the form $p(u_1, \dots, u_n)$ with $p \in P$ in order to constrain rewriting. This leads to the following definition:

DEFINITION 7.4 (Extended rewrite systems). *Let $S\mathcal{P} = (T, E, P, RS, De, Rmv, Oth)$ be an erfs. An extended $S\mathcal{P}$ -rewrite system \mathcal{R} is defined as in Definition 4.1 except that we add a set Δ of formulæ of the form $p(u_1, \dots, u_n)$ with $p \in P$.*

EXAMPLE 7.5. *In the erfs for rewriting modulo an equational theory, we can consider the following rewrite system for the signature*

$$\Sigma = (\{0^0, 1^0, +^2, \times^2\}, \{x, y, z\})$$

which defines Boolean rings:

$$\Delta = \left\{ \begin{array}{l} x + y \approx y + x, \quad x \times y \approx y \times x, \\ (x + y) + z \approx x + (y + z), \quad (x \times y) \times z \approx x \times (y \times z) \end{array} \right\}$$

$$\rightarrow_{=} = \left\{ \begin{array}{l} x + x \rightarrow 0, \quad x \times x \rightarrow x, \\ 0 + x \rightarrow x, \quad 0 \times x \rightarrow 0, \quad \dots \\ x \times (y + z) \rightarrow (x \times y) + (x \times z), \quad 1 \times x \rightarrow x, \end{array} \right\}$$

(see [31] for the complete presentation of the rewrite system for Boolean rings).

Rewriting steps and their closure are then extended by adding the satisfaction of premises of the form $p(u_1, \dots, u_n)$ with $p \in P$ for any instance of rules in $RS \cup De \cup Rmv$. This leads to the following definition:

DEFINITION 7.6 (Rewriting and convertibility relations). *Extended Rewriting steps are defined as in Definition 4.6 except that we add to Point 2. the following step:*

– for every leaf $p'(u_1, \dots, u_n) \in \mathcal{L}(\pi)$ with $p' \in P$, we have $\Theta \vdash p'(u_1, \dots, u_n)$ with $\Theta = \Delta \cup \{p(u, v) \mid p \in E \wedge u \stackrel{*}{\leftrightarrow} v \in \mathcal{R}\}$

In the same way, the convertibility relation $\stackrel{}{\leftrightarrow}_{\mathcal{R}}$ is defined as in Definition 4.13 except we add the following condition:*

for every $p'(u_1, \dots, u_n) \in \mathcal{L}(t)$ with $p' \in P$, we have $\Theta \vdash p'(u_1, \dots, u_n)$.

Of course, termination of rewrite systems necessarily requires that statements of the form $\Theta \vdash p^!(u_1, \dots, u_n)$ are computable.

EXAMPLE 7.7 (Rewriting modulo equational theories). *In the framework of term rewriting modulo equational theories, a rewrite system is then given by a couple (Δ, \rightarrow) where Δ is a set of equations and \rightarrow is a binary relation on terms. Therefore, following Definition 7.6, $\rightarrow_{\mathcal{R}}$ is the least binary relation on terms satisfying the following clauses:*

- $\rightarrow \subseteq \rightarrow_{\mathcal{R}}$,
- if $t \rightarrow_{\mathcal{R}} t'$ then $\sigma(t) \rightarrow_{\mathcal{R}} \sigma(t')$ where $\sigma : V \rightarrow T_{\Sigma}(V)$ is any substitution,
- if $t \rightarrow_{\mathcal{R}} t'$ then $C[t] \rightarrow_{\mathcal{R}} C[t']$ where C is a context (i.e. a term with a unique occurrence of the constant \square , and $C[t]$ denotes the result of replacing in C the occurrence of \square by t),
- if $\Delta \vdash s \approx s'^{30}$ and $s' \rightarrow_{\mathcal{R}} t$ then $s \rightarrow_{\mathcal{R}} t$, and
- if $s \rightarrow_{\mathcal{R}} t'$ and $\Delta \vdash t' \approx t$ then $s \rightarrow_{\mathcal{R}} t$.

In the literature, $\rightarrow_{\mathcal{R}}$ is unusually denoted by $\rightarrow_{\mathcal{R}_{\Delta}}$. Therefore, $\rightarrow_{\mathcal{R}_{\Delta}}$ is a rewrite relation which is defined on term equivalence classes $[s]_{\Delta}$ where \equiv_{Δ} is the least congruence deduced from equations in Δ .

The main problem with $\rightarrow_{\mathcal{R}_{\Delta}}$ is that in order to reduce $[s]_{\equiv_{\Delta}}$ w.r.t. $\rightarrow_{\mathcal{R}_{\Delta}}$, we need to explore all $[s]_{\equiv_{\Delta}}$, that is enumerate all terms that are Δ -equivalent to s to find one that is reducible via $\rightarrow_{\mathcal{R}}$. This requires all Δ -equivalence classes to be finite. In order to improve efficiency of the method, $\rightarrow_{\mathcal{R}_{\Delta}}$ has to be refined. A way is to refine rewriting steps such that they involve matching modulo \equiv_{Δ} [33, 46]. This can be formalized in our abstract framework by removing, in the erfs defined in Example 7.2, all rules of RS and by replacing them by the following one:

$$\frac{t \approx \sigma(u) \quad u = t'}{f(\dots, t, \dots) = f(\dots, \sigma(t'), \dots)} \quad \sigma: \text{substitution}$$

Therefore, following Definition 7.6, $\rightarrow_{\mathcal{R}}$ is the least binary relation on terms satisfying the following clauses:

- $\rightarrow \subseteq \rightarrow_{\mathcal{R}}$,
- if $\Delta \vdash t = \sigma(u)$ and $u \rightarrow_{\mathcal{R}} t'$ then $C[t] \rightarrow_{\mathcal{R}} C[\sigma(t')]$

³⁰ Here, we have $\Delta \vdash s \approx s' \iff \Theta \vdash s \approx s'$ because all rules in Oth are defined from formulae of the form $t \approx t'$. They do not manipulate equations of the form $t = t'$.

A shorter definition of $\rightarrow_{\mathcal{R}}$ can be given as follows:

$s \rightarrow_{\mathcal{R}} t$ iff there is a rule $u \rightarrow v \in R$, a term w , a context C and a substitution σ such that $w \equiv_{\Delta} \sigma(u)$, $s = C[w]$ and $t = C[\sigma(v)]$.

In the literature, $\rightarrow_{\mathcal{R}}$ is denoted by $\rightarrow_{\mathcal{R}, \Delta}$. $\rightarrow_{\mathcal{R}, \Delta}$ is weaker than $\rightarrow_{\mathcal{R}, \Delta}$. Indeed, we have the following counter-example taken from [6]: if $\Delta = \{(x + y) + z = x + (y + z)\}$ and $R = \{0 + x \rightarrow x\}$, then $[(a + 0) + b] \equiv_{\Delta} \rightarrow_{\mathcal{R}, \Delta} [a + b] \equiv_{\Delta}$, but $(a + 0) + b$ is in normal form with respect to $\rightarrow_{\mathcal{R}, \Delta}$.

EXAMPLE 7.8 (Partial function logic). Given a rewrite system $\mathcal{R} = (\Delta, \rightarrow)$ in the erfs of the partial function logic, $\rightarrow_{\mathcal{R}}$ is the least binary relation on T satisfying the following clauses:

- $\rightarrow_{\subseteq} \rightarrow_{\mathcal{R}}$,
- if $t \rightarrow_{\mathcal{R}} t'$ and σ is a substitution defined with respect to Θ , then $\sigma(t) \rightarrow_{\mathcal{R}} \sigma(t')$,
- if $t \rightarrow_{\mathcal{R}} t'$ and C is a context such that $\Theta \vdash D(C[t])$, then $C[t] \rightarrow_{\mathcal{R}} C[t']$.

As previously, a shorter definition can be given as follows: $t \rightarrow_{\mathcal{R}} t'$ iff there are $u \rightarrow v \in \mathcal{R}$, a substitution σ and a context C such that:

- for every $x \in \text{Var}(u) \cup \text{Var}(v)$, $\Theta \vdash D(\sigma(x))$,
- $\Theta \vdash D(C[t])$, and
- $t = C[\sigma(u)]$ and $t' = C[\sigma(v)]$.

Some restrictions have to be made on Definition 4.17 to define derivations and proofs. These restrictions consist in removing from trees resulting from the inductive construction of both $\overset{*}{\rightarrow}_{\mathcal{R}}$ and $\overset{*}{\leftarrow}_{\mathcal{R}}$, the formulæ that do not concern relations to rewrite (i.e. formulæ of the form $p(u_1, \dots, u_n)$ with $p \in P$) in all rule instances of $De \cup Rmv$. This leads to the following definition:

DEFINITION 7.9 (Derivation and proof). With all the notations of Definition 7.6, derivations (resp. proofs) are inductively defined as follows:

- any element of $\overset{*}{\leftarrow}_{\mathcal{R}}$ (resp. $\overset{*}{\rightarrow}_{\mathcal{R}}$) is a derivation (resp. proof),
- if $d_1 : u_1 \overset{* P_1}{\leftarrow}_{\mathcal{R}} v_1, \dots, d_n : u_n \overset{* P_n}{\leftarrow}_{\mathcal{R}} v_n$ (resp. $\pi_1 : u_1 \overset{* P_1}{\rightarrow}_{\mathcal{R}} v_1, \dots, \pi_n : u_n \overset{* P_n}{\rightarrow}_{\mathcal{R}} v_n$) are derivations (resp. proofs) and $\iota : p(u, v) \in De$ (resp. $\iota : p(u, v) \in De \cup Rmv$) such that:

- $\mathcal{FL}(\mathfrak{v}) = \{p_1(u_1, v_1), \dots, p_n(u_n, v_n)\}$ (the set of fixed leaves for \mathfrak{v} is exactly the set $\{p_1(u_1, v_1), \dots, p_n(u_n, v_n)\}$),
- for all $p'(u', v') \in \mathcal{L}(\mathfrak{v}) \setminus \mathcal{FL}(\mathfrak{v})$ with $p' \in E$, we have $p'(u', v')$ holds in \mathcal{R} according to normal, join or natural rewriting,
- for all $p(w_1, \dots, w_m) \in \mathcal{L}(\mathfrak{v})$ with $p \in P$ we have $\Theta \vdash p(w_1, \dots, w_m)$, and
- (Only for derivations) for all $1 \leq i \neq j \leq n$, we have: $u_i \xrightarrow{*P_i}_{\mathcal{R}} v_i \iff u_j \xrightarrow{*P_j}_{\mathcal{R}} v_j$ (fixed leaves of \mathfrak{v} are oriented in the same way)

then $(d_1, \dots, d_n, u \xrightarrow{*P}_{\mathcal{R}} v)_{\mathfrak{v}}$ (resp. $(\pi_1, \dots, \pi_n, u \xrightarrow{*P}_{\mathcal{R}} v)_{\mathfrak{v}}$) is a derivation (resp. a proof).

$(u \xrightarrow{*P}_{\mathcal{R}} v)$ is oriented in the same way as all fixed leaves of \mathfrak{v})

All properties and results established in Section 5 and Section 6 are directly adaptable in the extended framework developed in this section.

EXAMPLE 7.10. *In both above examples, manipulated formulæ are equations. Therefore, the definition of the associated expanding relation is defined as in Example 5.2 except that for rewriting modulo equational theories, rewritten elements are term equivalence classes. Consequently, effluences and proofs by rewriting denote peaks and valleys, respectively. Confluence and local confluence are then the usual notions of confluence and local confluence, respectively. We can easily show that the four axioms hold. Therefore, Church-Rosser's result and Newman's lemma are satisfied.*

8. Completion of rewrite systems

In this section, we give a completion method which is adapted to our general framework. As it is usual since Bachmair, Dershowitz and Hsiang's works [7, 8], a completion method is described by a set of inference rules. Here, given an rfs $\mathcal{SP} = (T, E, RS, De, Rmv)$, inference rules work on pairs (Γ, \mathcal{R}) where Γ is a set of formulæ $p(u, v)$, and \mathcal{R} is an \mathcal{SP} -rewrite system.

For a pair (Γ, \mathcal{R}) , a *mixed proof* of $p(u, v)$ is defined as follows:

DEFINITION 8.1 (Mixed proof). *For every pair (Γ, \mathcal{R}) and every $p \in E$, let us denote by $\xrightarrow{*P}_{\Gamma, \mathcal{R}}$ the binary relation on T defined as the convertibility relation in Definition 4.13 for $\rightarrow_{\mathcal{R}}^P \cup \leftarrow_{\mathcal{R}}^P \cup \{(u, v) \mid p(u, v) \in \Gamma\}$. Let us call mixed proof every proof tree resulting from the inductive construction of $\xrightarrow{*P}_{\Gamma, \mathcal{R}} = \bigcup_{p \in E} \xrightarrow{*P}_{\Gamma, \mathcal{R}}$.*

By definition, given a pair (Γ, \mathcal{R}) , $\overset{*}{\leftrightarrow}_{\mathcal{R}} \subseteq \overset{*}{\leftrightarrow}_{\Gamma, \mathcal{R}}$. Hence, basic proofs are specific instances of mixed proofs. Mixed proofs are then basic proofs and proofs in $\overset{*}{\leftrightarrow}_{\Gamma}$ (the convertibility relation defined for $\{(u, v) \mid p(u, v) \in \Gamma\}$)³¹ connected together via a finite set of rules instances in $De \cup Rmv$. The notions of effluence, local effluence and maximal effluence are then defined as in Definition 5.10 and Definition 6.1 by replacing the word “proof” with “mixed proof”.

A completion procedure has in input:

- a set of formulæ Φ of \mathcal{SP} of the form $p(u, v)$ with $p \in E$, and
- a reduction relation \succ .

The first step of the procedure consists in defining the pair $(\Gamma_0, \mathcal{R}_0)$ such that $\Gamma_0 = \Phi$ and $\mathcal{R}_0 = \emptyset$.

The completion method is defined by the set C of inference rules given in Figure 1.

The inference rules of Figure 1 call for some comments:

- DEDUCE adds a formula which can be derived from local effluences of \mathcal{R} . In the equational rewriting setting, when rewrite systems are terminating, effluences (by applying Newman’s lemma) can be automatically tested by computing all overlaps between rules, so-called *critical pairs*. This result is known as the *critical pair lemma*. The process to perform critical pairs is terminating when the set of rules is finite. Here, critical pairs cannot be defined at this abstract level because they depend on both the main general unifier notion and the inductive structure of first order terms. Here, objects are simply elements of a set T ; no conditions are given on their structure.
- SIMPLIFY uses the rules of \mathcal{R} to simplify formulæ. It consists in looking for an instance τ in De whose conclusion belongs to Γ , removing it from Γ and adding in Γ all premises of τ that have not been rewritten. Condition 2. is necessary to establish that SIMPLIFY is sound. In the equational rewriting setting, SIMPLIFY is simply instantiated as follows:

$$\text{SIMPLIFY} \quad \frac{\Gamma \cup \{u=v\}, \mathcal{R}}{\Gamma \cup \{u'=v\}, \mathcal{R}} \quad \text{if } u \overset{*}{\rightarrow}_{\mathcal{R}} u'$$

This is obtained from the instance of the transitivity rule $\tau = \frac{u=u' \quad u'=v}{u=v}$.

Indeed, for such an instance, we have $\overset{\rightarrow}{Ex}(\tau) = 1$ (see Section 5.1), then

³¹ In the equational rewriting setting, elements of $\overset{*}{\leftrightarrow}_{\Gamma}$ are called “plateaux”.

DEDUCE	$\frac{\Gamma, \mathcal{R}}{\Gamma \cup \{p(u, v)\} \cup (\mathcal{L}(t) \setminus \mathcal{F}\mathcal{L}(t)), \mathcal{R}}$	if there is a local effluence $(u_1 \xleftrightarrow{\mathcal{R}} v_1, \dots, u_n \xleftrightarrow{\mathcal{R}} v_n, u \xleftrightarrow{\mathcal{R}}^P v)_t$
ORIENT 1	$\frac{\Gamma \cup \{p(u, v)\}, \mathcal{R}}{\Gamma, \mathcal{R} \cup \{u \rightarrow_p v\}}$	if $u \succ v$
ORIENT 2	$\frac{\Gamma \cup \{p(u, v)\}, \mathcal{R}}{\Gamma, \mathcal{R} \cup \{u \leftarrow_p v\}}$	if $u \prec v$
SIMPLIFY	$\frac{\Gamma \cup \{p(u, v)\}, \mathcal{R}}{\Gamma \cup \Gamma', \mathcal{R}}$	if there exists $\iota = \frac{p_1(u_1, v_1) \dots p_n(u_n, v_n)}{p(u, v)} \in De$ s.t. 1. either $u \xrightarrow{\mathcal{E}x(\iota)} \xrightarrow{*} \xrightarrow{p_{\mathcal{E}x(\iota)}} \mathcal{R} v \xrightarrow{\mathcal{E}x(\iota)}$ and in this case $\Gamma' = \{p_i(u_i, v_i) \mid 1 \leq i \neq \mathcal{E}x(\iota) \leq n\}$ or $u \xleftarrow{\mathcal{E}x(\iota)} \xleftarrow{*} \xleftarrow{p_{\mathcal{E}x(\iota)}} \mathcal{R} v \xleftarrow{\mathcal{E}x(\iota)}$ and in this case $\Gamma' = \{p_i(u_i, v_i) \mid 1 \leq i \neq \mathcal{E}x(\iota) \leq n\}$, 2. and for all $p_i(u_i, v_i) \in \Gamma'$, $u_i \xleftrightarrow{\Gamma \cup \{p(u, v)\}, \mathcal{R}}^* p_i v_i$

Figure 1. Inference rules for completion

$u \xrightarrow{\mathcal{R}}^* u'$, and then $\Gamma' = \{u' = v\}$.

Condition 2. (i.e. $u' \xleftrightarrow{\Gamma \cup \{u=v\}, \mathcal{R}}^* v$) is satisfied because we have $u' \xleftarrow{\mathcal{R}}^* u \xleftrightarrow{\Gamma \cup \{u=v\}} v$.

Both conditions of SIMPLIFY have to be computable. Obviously, Point 1. satisfies such a property because for any (Γ, \mathcal{R}) the rewrite system \mathcal{R} is terminating. However, Point 2. is unlikely to be computable. For instance, in the non-symmetric transitive rewriting setting, SIMPLIFY is instantiated as follows:

SIMPLIFY
$$\frac{\Gamma \cup \{u \preceq v\}, \mathcal{R}}{\Gamma \cup \Gamma', \mathcal{R}}$$
 either $u \xrightarrow{\mathcal{R}}^* u'$ and
in this case $\Gamma' = \{u' \preceq v\}$ and $u' \xleftrightarrow{\Gamma \cup \{u \preceq v\}}^* v$
or $u' \xleftarrow{\mathcal{R}}^* v$ and
in this case $\Gamma' = \{u \preceq u'\}$ and $u \xleftrightarrow{\Gamma \cup \{u \preceq v\}}^* u'$

But, as expressed above, SIMPLIFY is unlikely to be computable because of the proof obligation $u' \xleftrightarrow{\Gamma \cup \{u \preceq v\}}^* v$ (resp. $u \xleftrightarrow{\Gamma \cup \{u \preceq v\}}^* u'$). In Knuth-Bendix completion for non-symmetric transitive relations developed by G. Struth

in [56], SIMPLIFY is not considered but replaced by the three following inference rules:

SIMPLIFY

$$\frac{\Gamma \cup \{s \preceq t\}, \mathcal{R}}{\Gamma, \mathcal{R}} \quad \frac{\Gamma, \mathcal{R} \cup \{s \rightarrow_{\preceq} t\}}{\Gamma, \mathcal{R}} \quad \frac{\Gamma, \mathcal{R} \cup \{s \leftarrow_{\preceq} t\}}{\Gamma, \mathcal{R}}$$

if $s \preceq t$, $s \rightarrow_{\preceq} t$ or $s \leftarrow_{\preceq} t$ is redundant, that is it can be replaced by a smaller proof in $\Gamma \cup \mathcal{R}$ using the proof ordering and measure \succ .

A basic way to express redundancy in our abstract framework is as follows:

SIMPLIFY

$$\frac{\Gamma \cup \{p(u, v)\}, \mathcal{R}}{\Gamma, \mathcal{R}} \quad \frac{\Gamma, \mathcal{R} \cup \{u \rightarrow_p v\}}{\Gamma, \mathcal{R}} \quad \frac{\Gamma, \mathcal{R} \cup \{u \leftarrow_p v\}}{\Gamma, \mathcal{R}}$$

if there is a proof by rewriting $\pi : u \xrightarrow{*}_P^{\mathcal{R} \setminus \{u \preceq_p v\}} v$.

EXAMPLE 8.2. *In the equational rewriting setting, basic inference rules for completion are then the following:*

$$\begin{array}{lll} \text{DEDUCE} & \frac{\Gamma, \mathcal{R}}{\Gamma \cup \{u=v\}, \mathcal{R}} & \text{if } u \stackrel{=}{\mathcal{R}} \leftarrow u' \rightarrow \stackrel{=}{\mathcal{R}} v \\ \text{ORIENT} & \frac{\Gamma \cup \{u=v\}, \mathcal{R}}{\Gamma, \mathcal{R} \cup \{u \rightarrow v\}} & \text{if } u \succ v \\ \text{SIMPLIFY} & \frac{\Gamma \cup \{u=v\}, \mathcal{R}}{\Gamma \cup \{u'=v\}, \mathcal{R}} & \text{if } u \xrightarrow{*}_{\mathcal{R}} u' \end{array}$$

In the conditional rewriting setting, the direct instantiation of inference rules of Figure 1 is: we assume that for every finite conjunction c , $\succ_c = \succ$ where \succ is a well-founded ordering stable under substitution and context (cf. Example 5.4)

$$\begin{array}{lll} \text{DEDUCE} & \frac{\Gamma, \mathcal{R}}{\Gamma \cup \{u=0v\}, \mathcal{R}} & \text{if } u \stackrel{0}{\mathcal{R}} \leftarrow u' \rightarrow \stackrel{0}{\mathcal{R}} v \\ \text{ORIENT} & \frac{\Gamma \cup \{u=cv\}, \mathcal{R}}{\Gamma, \mathcal{R} \cup \{u \rightarrow cv\}} & \text{if } u \succ v \\ \text{SIMPLIFY} & \frac{\Gamma \cup \{u=0v\}, \mathcal{R}}{\Gamma \cup \{u'=0v\}, \mathcal{R}} & \text{if } u \xrightarrow{*}_0^{\mathcal{R}} u' \end{array}$$

With such rules, for the different rewrite systems \mathcal{R} the relation $\rightarrow_{\mathcal{R}}^0$ is not necessarily decidable. For this purpose, we saw that we have to manipulate decreasing rewrite systems. Therefore, \succ has to be a simplification order (i.e. contain in addition the proper subterm relation). Moreover, we have to add in the ORIENT rule the following condition: $\forall t = t' \in c, u \succ t \wedge u \succ t'$. This last

condition enables us to consider only decreasing rewrite systems for which joinability is decidable. From there, the two following inference rules which cannot be defined at our abstract level, can be added:

$$\begin{array}{l} \text{SIMPLIFY CONDITION 1} \quad \frac{\Gamma \cup \{u =_c \Delta t = \Delta' \Delta c' v\}, \mathcal{R}}{\Gamma \cup \{u =_{c \wedge w = w' \wedge c' v}\}, \mathcal{R}} \quad \text{if } t \xrightarrow{\ast 0}_{\mathcal{R}} w \text{ and } t' \xrightarrow{\ast 0}_{\mathcal{R}} w' \\ \text{SIMPLIFY CONDITION 2} \quad \frac{\Gamma, \mathcal{R} \cup \{u \rightarrow_{c \Delta t = \Delta' \Delta c' v}\}}{\Gamma, \mathcal{R} \cup \{u \rightarrow_{c \wedge w = w' \wedge c' v}\}} \quad \text{if } t \xrightarrow{\ast 0}_{\mathcal{R}} w \text{ and } t' \xrightarrow{\ast 0}_{\mathcal{R}} w' \end{array}$$

In the non-symmetric transitive rewriting setting, basic inference rules for completion are then the following:

$$\begin{array}{l} \text{DEDUCE} \quad \frac{\Gamma, \mathcal{R}}{\Gamma \cup \{u \preceq v\}, \mathcal{R}} \quad \text{if } u \xrightarrow{\preceq}_{\mathcal{R}} \leftarrow u' \rightarrow \xrightarrow{\preceq}_{\mathcal{R}} v \\ \text{ORIENT 1} \quad \frac{\Gamma \cup \{u \preceq v\}, \mathcal{R}}{\Gamma, \mathcal{R} \cup \{u \rightarrow \preceq v\}} \quad \text{if } u \succ v \\ \text{ORIENT 2} \quad \frac{\Gamma \cup \{u \preceq v\}, \mathcal{R}}{\Gamma, \mathcal{R} \cup \{u \leftarrow \preceq v\}} \quad \text{if } v \succ u \end{array}$$

SIMPLIFY

$$\frac{\Gamma \cup \{s \preceq t\}, \mathcal{R}}{\Gamma, \mathcal{R}} \quad \frac{\Gamma, \mathcal{R} \cup \{s \rightarrow \preceq t\}}{\Gamma, \mathcal{R}} \quad \frac{\Gamma, \mathcal{R} \cup \{s \leftarrow \preceq t\}}{\Gamma, \mathcal{R}} \quad \text{if } s \xrightarrow{\ast \preceq}_{\mathcal{R}} u \leftarrow \xrightarrow{\ast \preceq}_{\mathcal{R}} t$$

REMARK 8.3. Usually, tautologies are simply recognized on their syntactical structure. For instance, in equational logic, tautologies are all equations of the form $t = t$. Regarding conditional equational logic, tautologies are either of the form $t =_c t$ or $t =_c t'$ where $t = t' \in c$. In this case, tautologies are removed from Γ , and we add the following supplementary inference rule to the set C of Figure 1:

$$\text{DELETE} \quad \frac{\Gamma \cup \{p(u, v)\}, \mathcal{R}}{\Gamma, \mathcal{R}} \quad \text{if } \emptyset \vdash p(u, v)$$

8.1. CORRECTNESS

NOTATION 8.4. If $\frac{\Gamma, \mathcal{R}}{\Gamma', \mathcal{R}'}$ is an instance of any inference rule of Figure 1, we will denote this instance $(\Gamma, \mathcal{R}) \vdash (\Gamma', \mathcal{R}')$.

The inference rules of Figure 1 are sound, that is they do not change the underlying theory.

PROPOSITION 8.5. If $(\Gamma_1, \mathcal{R}_1) \vdash (\Gamma_2, \mathcal{R}_2)$ then $\xrightarrow{\ast}_{\Gamma_1, \mathcal{R}_1} = \xrightarrow{\ast}_{\Gamma_2, \mathcal{R}_2}$.

Proof This is obvious for the first three rules. For SIMPLIFY, let us denote by $\Gamma_1 = \Gamma \cup \{p(u, v)\}$ and $\Gamma_2 = \Gamma \cup \Gamma'$, $\mathcal{R}_1 = \mathcal{R}_2 = \mathcal{R}$. By hypothesis, for every $p_i(u_i, v_i) \in \Gamma'$ we have $u_i \xrightarrow{*}_{\Gamma_1, \mathcal{R}_1}^{p_i} v_i$. Therefore, we have $\xrightarrow{*}_{\Gamma_2, \mathcal{R}_2} \subseteq \xrightarrow{*}_{\Gamma_1, \mathcal{R}_1}$. Conversely, the three following hypothesis $\Gamma' \subseteq \Gamma_2$, $u \xrightarrow{\mathcal{E}_x(\iota)} v \xrightarrow{\mathcal{E}_x(\iota)}$ (resp. $u \xleftarrow{\mathcal{E}_x(\iota)} v \xleftarrow{\mathcal{E}_x(\iota)}$), and $\iota = \frac{p_1(u_1, v_1) \dots p_n(u_n, v_n)}{p(u, v)} \in De$ imply $u \xrightarrow{*}_{\Gamma_2, \mathcal{R}_2} v$. Consequently, $\xrightarrow{*}_{\Gamma_1, \mathcal{R}_1} \subseteq \xrightarrow{*}_{\Gamma_2, \mathcal{R}_2}$. \square

Moreover, any rewrite system generated by inference rules is terminating.

PROPOSITION 8.6. *If $\mathcal{E}_{\succ_{\mathcal{R}}} \subseteq \mathcal{E}_{\succ}$ and $(\Gamma, \mathcal{R}) \vdash (\Gamma', \mathcal{R}')$, then $\mathcal{E}_{\succ_{\mathcal{R}'}} \subseteq \mathcal{E}_{\succ}$.*

Proof Formulæ are oriented with the help of the reduction ordering \succ . \square

Finally, each step of the completion method defines mixed proof transformations. Here, reduced subproofs are twofold:

1. maximal effluences, and
2. proofs containing at least a leaf of the form $u \leftrightarrow_{\Gamma} v$.

If these transformations are terminating, and if at the end we have an empty set of formulæ to orient, then by Proposition 8.5, Proposition 8.6 and Theorem 5.20, we have obtained a decision procedure for the starting theory Γ . As usual, a sufficient condition has to be imposed to have such a result. This condition is natural when we are dealing with non-deterministic choices on formulæ of the form $p(u, v)$ to orient or to delete as well as rules to reduce. The underlying idea is that any finitely accessible choice is not indefinitely dismissed. We talk about *fairness*. Here, fairness is defined as follows:

DEFINITION 8.7 (Fairness). *Let $((\Gamma_i, \mathcal{R}_i))_{i \geq 0}$ be a chain such that $(\Gamma_0, \mathcal{R}_0) \vdash (\Gamma_1, \mathcal{R}_1) \vdash \dots$. We say that a formula is persistent if it occurs in any Γ_i beyond some rank. Let us denote by $\Gamma_{\infty} = \bigcup_{i \geq 0} \bigcap_{j \geq i} \Gamma_j$ the set of persistent formulæ, and*

by $R_{\infty} = \bigcup_{i \geq 0} \bigcap_{j \geq i} R_j$ the set of persistent rules.

The chain $((\Gamma_i, \mathcal{R}_i))_{i \geq 0}$ is fair if no formula of the form $p(u, v)$ is persistent and any local effluence is eventually transformed into a formula.

Since Bachmair's works, inference rules of completion can be used to define proof orders on mixed proofs. Here, the proof normalization consists essentially on replacing a local effluence into an equivalent proof by rewriting. Of course, this is not possible if Γ is not empty and if \mathcal{R} is not confluent. The idea due to Bachmair, Dershowitz and Hsiang is then to see

the completion process as a transformation of the axiom system allowing to write proofs more and more normal. To achieve this purpose, we have to show that inference rules of completion make decrease mixed proof complexity. In equation rewriting setting, this is obtained by using Bachmair's ordering. Hence, if we note $>$ the ordering on mixed proofs defined by:

- DEDUCE: $u \mathcal{R} \leftarrow t \rightarrow_{\mathcal{R}} v > u \leftrightarrow_{\Gamma} v$
- EFFLUENCE: $u \mathcal{R} \leftarrow t \rightarrow_{\mathcal{R}} v > u \xrightarrow{*}_{\mathcal{R}} w \mathcal{R} \xleftarrow{*} v$
- ORIENT:
 - $u \leftrightarrow_{\Gamma} v > u \rightarrow v$
 - $u \leftrightarrow_{\Gamma} v > v \rightarrow u$
- SIMPLIFY:
 - $u \leftrightarrow_{\Gamma} v > u \xrightarrow{*}_{\mathcal{R}} u' \leftrightarrow_{\Gamma} v$
 - $u \leftrightarrow_{\Gamma} v > u \leftrightarrow u' \mathcal{R} \xleftarrow{*} v$

then the closure \gg of $>$ under mixed proof tree contexts and transitivity is well-founded by using Bachmair's ordering defined as the extension of the reduction ordering \succ to multisets and then to multisets of these multisets when mixed complexity is defined by:

- *complexity of elementary proof*: $\|u \rightarrow_{\mathcal{R}} v\| = \{\{u\}\}$ and $\|u \leftrightarrow_{\Gamma} v\| = \{\{u, v\}\}$ ³²
- *complexity of mixed proofs*: the multiset of the complexity of its elementary proofs.

In our abstract framework, the ordering \gg is defined by: Let $\Gamma_{\omega} = \bigcup_{i \geq 0} \Gamma_i$ and $\mathcal{R}_{\omega} = \bigcup_{i \geq 0} \mathcal{R}_i$, and for all $p(u, v) \in \Gamma_{\omega}$ let us denote by $u \leftrightarrow_{\Gamma_{\omega}}^p v$ the mixed proof tree reduced to this leaf. Let $>$ be the binary relation on mixed proofs defined as follows:

- DEDUCE: $(u_1 \xleftrightarrow{\mathcal{R}_{\omega}} v_1, \dots, u_n \xleftrightarrow{\mathcal{R}_{\omega}} v_n, u \xleftrightarrow{*}_{\mathcal{R}_{\omega}}^p v) > u \leftrightarrow_{\Gamma_{\omega}}^p v$,
- ORIENT: $u \leftrightarrow_{\Gamma_{\omega}}^p v > u \xleftrightarrow{\mathcal{R}_{\omega}}^p v$,

³² $\{\{\dots\}\}$ denotes finite multisets.

- SIMPLIFY: $u \leftrightarrow_{\Gamma_\omega}^P v > (\dots, u \stackrel{*}{\underset{\mathcal{E}x(\iota)}{=}} \underset{\mathcal{R}_\omega}{\leftrightarrow} v \stackrel{*}{\underset{\mathcal{E}x(\iota)}{=}} \dots, u_i \leftrightarrow_{\Gamma_\omega} v_i, \dots, u \leftrightarrow_{\Gamma_\omega, \mathcal{R}_\omega}^* v)$, and
- EFFLUENCE: $\pi : u \leftrightarrow_{\mathcal{R}_\omega}^* v > \pi' : u \leftrightarrow_{\mathcal{R}_\omega}^* v$ if π is a local effluence and π' a proof by rewriting

Let us denote by \gg the closure of $>$ under mixed proof tree context and transitivity.

The equivalent of Bachmair's ordering cannot be abstractly defined because based on reduction orderings on terms whilst here reduction orderings are on derivations.

AXIOM 6 (Proof normalization). \gg is well-founded.

When fairness and Axiom 6 are ensured, proofs have normal forms with respect to \gg . The following normalization theorem is the basis of Church-Rosser property.

THEOREM 8.8. *If the chain $((\Gamma_i, \mathcal{R}_i))_{i \geq 0}$ is fair and Axiom 6 is satisfied, then for every mixed proof π obtained at step i there is an equivalent proof by rewriting in $\mathcal{R}_\infty = \bigcup_{i \geq 0} \bigcap_{j \geq i} \mathcal{R}_j$.*

Proof To prove Theorem 8.8, we reason by well-founded induction on \gg . Let $\pi : u \leftrightarrow_{\Gamma_i, \mathcal{R}_i}^* v$ be a mixed proof. By definition, if π is not a proof by rewriting, then it has a leaf of the form $u' \leftrightarrow_{\Gamma_i} v'$, or a local effluence (π', ω) . By fairness, $u' \leftrightarrow_{\Gamma_i} v'$ will have disappeared at a step $j_1 > i$, and the local effluence will be transformed into a formula at a step $j_2 > i$. In all cases, π is transformed into a mixed proof π' at a step j such that $j > i$ and $\pi \gg \pi'$. By induction hypothesis, there is a step $k > j$ and a proof by rewriting $\pi'' : u \leftrightarrow_{\Gamma_k, \mathcal{R}_k}^* v$. As rewrite rules are never removed from \mathcal{R}_ω , we have $\mathcal{R}_\infty = \mathcal{R}_\omega$. \square

8.2. REDUNDANT RULES PROCESSING

The inference rules of Figure 1 lead to nondeterministic and inefficient completion procedures. Indeed, rules which can be deduced from other rules are not necessarily removed. A way to answer this problem will be to add the inference of Figure 2.

Briefly, REDUCE uses rules of \mathcal{R} to simplify rules. It consists in looking for an instance ι in De whose conclusion belongs to $\leftrightarrow_{\Gamma, R}$, removing it from R and adding in Γ all premises of ι (those indexed by I) which cannot be rewritten. In the equational rewriting setting, REDUCE is instantiated into two inference rules as follows:

REDUCE $\frac{\Gamma, \mathcal{R} \cup \{u \stackrel{\leftarrow}{\leftarrow}_p v\}}{\Gamma \cup \Gamma', \mathcal{R}}$ if there exists $\iota = \frac{p_1(u_1, v_1) \dots p_n(u_n, v_n)}{p(u, v)} \in De$ s.t.

1. either $u \xrightarrow{\mathcal{E}x(\iota)} \xrightarrow{*} \mathcal{R} \xrightarrow{p_{\mathcal{E}x(\iota)}} v \xrightarrow{\mathcal{E}x(\iota)}$ and
in this case $\Gamma' = \{p_i(u_i, v_i) \mid 1 \leq i \neq \mathcal{E}x(\iota) \leq n\}$
or $u \xleftarrow{\mathcal{E}x(\iota)} \xleftarrow{*} \mathcal{R} \xleftarrow{p_{\mathcal{E}x(\iota)}} v \xleftarrow{\mathcal{E}x(\iota)}$ and
in this case $\Gamma' = \{p_i(u_i, v_i) \mid 1 \leq i \neq \mathcal{E}x(\iota) \leq n\}$,
2. and for all $p_i(u_i, v_i) \in \Gamma'$, $u_i \xleftrightarrow{*} \mathcal{R} \xrightarrow{p_i} v_i$

Figure 2. Inference rule for simplification

1. COMPOSE $\frac{\Gamma, \mathcal{R} \cup \{u \rightarrow v\}}{\Gamma \cup \{u = v'\}, \mathcal{R}}$ if $v \xleftrightarrow{*} \mathcal{R} v'$

This is obtained from the instance of the transitivity rule $\frac{u = v' \quad v' = v}{u = v}$. According to the rewriting orientation applied between v and v' , two cases can occur: either $v \xrightarrow{*} \mathcal{R} v'$ and then by a direct application of ORIENT 1, we have $u \rightarrow v'$, or $v \xleftarrow{*} \mathcal{R} v'$ and in this case the equation $u = v'$ can be removed from Γ at the next step by SIMPLIFY. For these two reasons, the inference rule usually given is the following:

COMPOSE $\frac{\Gamma, \mathcal{R} \cup \{u \rightarrow v\}}{\Gamma, \mathcal{R} \cup \{u \rightarrow v'\}, \mathcal{R}}$ if $v \xrightarrow{*} \mathcal{R} v'$

2. COLLAPSE $\frac{\Gamma, \mathcal{R} \cup \{u \rightarrow v\}}{\Gamma \cup \{u' = v\}, \mathcal{R}}$ if $u \xrightarrow{*} \mathcal{R} u'$

This is obtained from the instance of the transitivity rule $\frac{u = u' \quad u' = v}{u = v}$. Without a well-founded strict order on rewrite rules, it is known that COLLAPSE does not decrease complexity of proof trees through completion [8]. Moreover, it may even not preserve the starting equational theory. However, this well-founded strict order on rewrite rules is based on pattern-matching and subterm relation which cannot be defined at our abstract level (terms being simple elements of a set T without structuration).

Of course, we are again confronted with the problem that Point 2. of REDUCE condition is unlikely to be computable. However, we saw that in this case, a solution is to remove all redundant rules, that is rules $u \stackrel{\leftarrow}{\leftarrow}_p v \in \mathcal{R}$ for which there are proofs by rewriting $\pi : u \xleftrightarrow{*} \mathcal{R} \setminus \{u \stackrel{\leftarrow}{\leftarrow}_p v\} v$. In Section 4, we have supposed that the logicity property holds and then \mathcal{R} is Church-Rosser. Moreover, \mathcal{R} is terminating. By Theorem 5.20, we can decide if there

exists a proof by rewriting $u \xrightarrow{*} \mathcal{R} \setminus \{u \Leftarrow_p v\} v$. Therefore, we have a mechanical redundancy test.

9. Other examples

In this section, we will instantiate our abstract framework on two logics: extended conditional equational logic and M. Schorlemmer's logic of special relations [49]. This last logic is the most important of our applications because itself has been defined as a general framework for unifying a variety of rewriting theories. Here, we will show that we obtain the same results as M. Schorlemmer [49] and, furthermore, we will define a completion method which, as far as we know, has not been studied yet.

9.1. CONDITIONAL REWRITING

Conditional rewriting has been intensively studied (see [13, 24, 28, 34, 35]). Up to our knowledge, conditional rewriting has been studied to solve the word problem associated with unconditioned equations (this rewriting setting has been developed as a running example in both previous sections). Only rewrite rules are equipped with a condition, that is they are of the form $\bigwedge_{1 \leq i \leq n} t_i = t'_i \Rightarrow t \rightarrow t'$. Actually, when dealing with conditioned equations,

superposition-based theorem proving for full first-order clauses with equality (e.g. see [11, 45]) is usually used. Here, to exemplify our abstract framework, we propose to extend the classic conditional rewriting to formulæ of the form $\bigwedge_{1 \leq i \leq n} t_i = t'_i \Rightarrow t = t'$ by using exclusively rewriting techniques. To achieve this

purpose, rewritings and derivations will be closed under transitivity, monotonicity and modus-ponens, respectively. This will be obtained by inserting both monotonicity and modus-ponens in the deduction rule of transitivity. The resulting deduction rule will be called Trans/Mod (see below for its statement).

To define the associated rfs, we will use the notions and notations that have already been presented in Example 3.7. Let us recall that in order to enter conditional formulæ into the definition of rfs which only manipulates predicates, standard conditional formulæ $c \Rightarrow t = t'$ are represented by $t =_c t'$ in our framework (t and t' being terms and c being a finite conjunction of equations of the form $c = \bigwedge_{1 \leq i \leq n} t_i = t'_i$). In the following, we will use the following

supplementary notations:

- $u = v \in c$ to mean that the equation $u = v$ occurs in the finite conjunction c ,

- $c \subseteq c'$ to mean that all equations in c occur in c' , and
- $c \setminus c'$ to denote the finite conjunction of equations obtained by removing all equations of c that occur in c' .

9.1.1. Rewriting formal system

Given a signature Σ defined by a set of function names equipped with arities in \mathbb{N} , and a set of variables V , we define the rfs for conditional term rewriting $\mathcal{SP} = (T, E, RS, De, Rmv)$ as follows:

- $T = T_\Sigma(V)$,
- $E = \{=_c \mid c : \text{finite conjunction}\}$ such that for every c , $=_c \stackrel{def}{=} T_\Sigma(V) \times T_\Sigma(V)$ (syntactic definition),
- RS is the set of all instances of the following deduction rules: let $\sigma : V \rightarrow T_\Sigma(V)$ be a substitution and let C be a context

$$\text{Substitution} \frac{t =_c t'}{\sigma(t) =_{\sigma(c)} \sigma(t')} \quad \text{Context} \frac{t =_c t'}{C[t] =_c C[t']}$$

- De is the set of all instances of the two following deduction rules:

$$\text{Reflexivity} \frac{}{t =_\emptyset t}$$

$$\text{Trans/Mod} \frac{t =_c t' \quad t' =_{c'} t'' \quad \forall i \in I, u_i =_{c_i} v_i}{t =_{\bar{c} \wedge \bigwedge_{i \in I} c_i} t''}$$

where $\bar{c} = c \wedge c' \setminus \{u_i = v_i \mid i \in I\}$.

9.1.2. Conditional rewrite systems

Since all binary relations of E are symmetric, a conditional rewrite system for \mathcal{SP} is simply a family $\{\rightarrow_c\}_{c:\text{condition}}$.

Given an instance $\mathfrak{t} = \frac{t =_c t' \quad t' =_{c'} t'' \quad \forall i \in I, u_i =_{c_i} v_i}{t =_{\bar{c} \wedge \bigwedge_{i \in I} c_i} t''}$ of Trans/Mod, its set of fixed

leaves is $\mathcal{FL}(\mathfrak{t}) = \{t =_c t', t' =_{c'} t''\}$.

Following Definition 4.6, rewriting steps are defined as follows: $t \rightarrow_{\mathcal{R}}^c t'$ if and only if there exist a rule $u \rightarrow_{c'} v \in \mathcal{R}$, a substitution $\sigma : V \rightarrow T$, and a context C such that $t = C[\sigma(u)]$, $t' = C[\sigma(v)]$, and $c = \sigma(c')$. For every finite conjunction c , the rewriting relation $\rightarrow_{\mathcal{R}}^*{}^c$ is then defined as follows: if $t \xrightarrow{\mathcal{R}}^*{}^{c_1} t'$,

$t' \xrightarrow{\mathcal{R}}^* c_2 t''$, and $\bigwedge_{i \in I} u_i = v_i \subseteq c_1 \wedge c_2$ such that for every $i \in I$, there exists a finite conjunction c_i such that:

- Normal $u_i \xleftrightarrow{\mathcal{R}}^* c_i v_i$,
- Natural $u_i \xrightarrow{\mathcal{R}}^* c_i v_i$,
- Join $u_i \xrightarrow{\mathcal{R}}^* c' w \xrightarrow{\mathcal{R}}^* c'' v_i$ and $c_i = c' \wedge c''$

then $t \xrightarrow{\mathcal{R}}^* c t'$ where $c = \bar{c} \wedge \bigwedge_{i \in I} c_i$ and $\bar{c} = (c_1 \wedge c_2) \setminus (\bigwedge_{i \in I} u_i = v_i)$.

This definition of conditional rewriting subsumes the classical one such as defined in Example 4.10. Indeed, if $s \xrightarrow{\mathcal{R}}^0 t$ has been obtained from the rewrite rule $\bigwedge_{1 \leq i \leq n} l_i = r_i \Rightarrow l \rightarrow r$ in classic conditional rewriting, then this gives rise to the following rewritings:

$$\text{Subst.} \frac{l \rightarrow \bigwedge_{1 \leq i \leq n} l_i = r_i}{\bigwedge_{1 \leq i \leq n} \sigma(l_i) = \sigma(r_i)} \quad \text{Cont.} \frac{\sigma(l) \xrightarrow{\mathcal{R}}^* \bigwedge_{1 \leq i \leq n} \sigma(l_i) = \sigma(r_i) \quad \sigma(r)}{\bigwedge_{1 \leq i \leq n} \sigma(l_i) = \sigma(r_i)} \quad t =_0 t \quad \forall 1 \leq i \leq n, \sigma(l_i) =_0 \sigma(r_i)$$

$$\text{Trans/Mod} \frac{s = C[\sigma(t)] \xrightarrow{\mathcal{R}}^* \bigwedge_{1 \leq i \leq n} \sigma(l_i) = \sigma(r_i) \quad t = C[\sigma(r)]}{s =_0 t}$$

Proofs are trees composed of instances of Trans/Mod and Symmetry, while derivations are composed of instances of Trans/Mod only. Therefore, a derivation is any tree of the form $\frac{d_1: u \xrightarrow{\mathcal{R}}^* c v \quad d_2: v \xrightarrow{\mathcal{R}}^* c' w}{u \xrightarrow{\mathcal{R}}^* \bar{c} \wedge \bigwedge_{i \in I} c_i w}$ such that $\bar{c} = (c \wedge c') \setminus (\bigwedge_{i \in I} u_i = v_i)$ where for every $i \in I$, there is a finite conjunction c_i such that $u_i =_{c_i} v_i$ holds in \mathcal{R} .

9.1.3. Properties of conditional rewrite systems

We choose the expanding position application defined for any instance ι of the Trans/Mod rule by: $\text{Ex}(\iota) = 1$. Hence, $\mathcal{E}_{\mathcal{R}}$ is defined by:

$$d : u \xrightarrow{\mathcal{R}}^* c v \mathcal{E}_{\mathcal{R}} \frac{d_1 : u \xrightarrow{\mathcal{R}}^* c v \quad d_2 : v \xrightarrow{\mathcal{R}}^* c' w}{u \xrightarrow{\mathcal{R}}^* \bar{c} \wedge \bigwedge_{i \in I} c_i w}$$

As for every finite conjunction c , $=_c$ is symmetric, we then have, for every rewrite system \mathcal{R} , $\mathcal{E}_{\mathcal{R}} = \mathcal{E}_{\mathcal{R}}$. Therefore, termination of rewrite systems is equivalent to the fact that $\mathcal{E}_{\mathcal{R}}$ is Noetherian.

This leads to the following definition of effluences and proofs by rewriting:

- Effluences are proofs of the form

$$\frac{d_1 : u \xrightarrow{*c}_{\mathcal{R}} v \quad d_2 : v \xrightarrow{*c'}_{\mathcal{R}} w}{u \xrightarrow{*c'}_{\mathcal{R}} w} \quad \bar{c} \wedge \bigwedge_{i \in I} c_i$$

- Proof by rewriting are proofs of the form:

$$\frac{d_1 : u \xrightarrow{*c}_{\mathcal{R}} v \quad d_2 : v \xrightarrow{*c'}_{\mathcal{R}} w}{u \xrightarrow{*c'}_{\mathcal{R}} w} \quad \bar{c} \wedge \bigwedge_{i \in I} c_i$$

Let $u =_c v$ be a conditioned equation. Then, let us define the set \mathcal{S} as follows:

$$\mathcal{S} = \left[\begin{array}{c} \{u' \mid u \rightarrow_{\mathcal{R}}^{c'} u'\} \\ \cup \\ \{v' \mid v \rightarrow_{\mathcal{R}}^{c'} v'\} \end{array} \right]$$

Then, let us show that $RS[u =_c v] = \left[\begin{array}{c} \{u \rightarrow_{\mathcal{R}}^{c'} u' \mid \exists c', u' \in \mathcal{S}\} \\ \cup \\ \{v \rightarrow_{\mathcal{R}}^{c'} v' \mid \exists c', v' \in \mathcal{S}\} \end{array} \right]$. Indeed,

for every finite conjunction c , $\rightarrow_{\mathcal{R}}^c$ is finitely branching, that is each term has only finitely many direct successors. Moreover, $\rightarrow_{\mathcal{R}}^c$ is decidable.³³ Hence, \mathcal{S} is finite and thus computable.

First of all, observe that every proof tree of the form:

$$\frac{t_1 =_{c_1} t_2 \quad \frac{t_2 =_{c_2} t_3 \quad t_3 =_{c_3} t_4 \quad \forall i \in I, u_i =_{c_i} v_i}{t_2 =_{\bar{c}_1 \wedge \bigwedge_{i \in I} c_i} t_4} \quad \forall j \in J, u_j =_{c_j} v_j}{t_1 =_{\bar{c}_2 \wedge \bigwedge_{j \in J} c_j} t_4}$$

where $\bar{c}_1 = c_2 \wedge c_3 \setminus \{u_i =_{c_i} v_i \mid i \in I\}$ and $\bar{c}_2 = c_1 \wedge \bar{c}_1 \setminus \{u_j =_{c_j} v_j \mid j \in J\}$, can be transformed into the following equivalent proof tree (i.e. with the same conclusion):

³³ In the classic conditional rewriting, rewriting steps are not necessarily computable because they depend on decidability of joinability. Here, this does not occur because the satisfaction of premises of conditional rules has been relegated to the level of rewriting step composition (i.e. in Trans/Mod).

$$\frac{\frac{t_1 =_{c_1} t_2 \quad t_2 =_{c_2} t_3 \quad \forall i \in I', u_i =_{c_i} v_i}{t_1 =_{\bar{c}_1 \wedge \bigwedge_{i \in I'} c_i} t_3} \quad t_3 =_{c_3} t_4 \quad \forall j \in J', u_j =_{c_j} v_j}{t_1 =_{\bar{c}_2 \wedge \bigwedge_{j \in J} c_j} t_4}$$

with

$$\{u_i = v_i \mid i \in I'\} = \begin{cases} \{u_i = v_i \mid i \in I\} \cap c_2 \\ \cup \\ \{u_j = v_j \mid j \in J\} \cap c_1 \end{cases}$$

$$\{u_j = v_j \mid j \in J'\} = \begin{cases} \{u_i = v_i \mid i \in I\} \cap c_3 \\ \cup \\ \{u_j = v_j \mid j \in J\} \cap (\bar{c}_1 \wedge \bigwedge_{i \in I'} c_i) \end{cases}$$

We have then: $\bar{c}_2 \wedge \bigwedge_{j \in J} c_j = \bar{c}_1 \wedge \bigwedge_{j \in J} c_j$.

Therefore, for any derivation $d : u \xrightarrow{*}_{\mathcal{R}}^c v$, there exists an equivalent derivation d' of the form:

$$\frac{d_1 : u \xrightarrow{c_1}_{\mathcal{R}} u' \quad d_2 : u' \xrightarrow{*}_{\mathcal{R}}^{c_2} v}{u \xrightarrow{*}_{\mathcal{R}}^c v}$$

This means, by the definition of the expanding position application, that $d_1 :$

$$u \xrightarrow{c_1}_{\mathcal{R}} u' \in \mathcal{E}_{\mathcal{R}} d'. \text{ Therefore, } RS[u =_c v] = \begin{cases} \{u \xrightarrow{c'}_{\mathcal{R}} u' \mid u' \in S\} \\ \cup \\ \{v \xrightarrow{c'}_{\mathcal{R}} v' \mid v' \in S\} \end{cases}. \text{ Consequently,}$$

$RS[u =_c v]$ is finite and thus computable.

Moreover, we have:

- for each rewriting $t \xrightarrow{*}_{\mathcal{R}}^c t'$, there exists a derivation $d : t \xrightarrow{*}_{\mathcal{R}}^c t'$. Therefore, by the above transformation between instances of Trans/Mod, there exists an equivalent derivation d' of the form:

$$\frac{\frac{t \xrightarrow{c_1}_{\mathcal{R}} t_1 \quad t_1 \xrightarrow{c_2}_{\mathcal{R}} t_2}{t \xrightarrow{*}_{\mathcal{R}}^{\bar{c}_1} t_2} \quad t_2 \xrightarrow{c_3}_{\mathcal{R}} t_3}{t \xrightarrow{*}_{\mathcal{R}}^{\bar{c}_2} t_3} \quad \dots \quad \frac{t \xrightarrow{*}_{\mathcal{R}}^{\bar{c}_{n-1}} t_n \quad t_n \xrightarrow{c_{n+1}}_{\mathcal{R}} t'}{t \xrightarrow{*}_{\mathcal{R}}^{\bar{c}_n = c} t'}}$$

- given a derivation $d : u \xrightarrow{*}_{\mathcal{R}}^c v$, we have seen just above that v can be rewritten in one step, into a finite set of terms v' , that is $d \mathcal{E}_{\mathcal{R}} \frac{d:u \xrightarrow{*}_{\mathcal{R}}^c v \quad d':v \xrightarrow{c'}_{\mathcal{R}} v'}{u \xrightarrow{*}_{\mathcal{R}}^{c \wedge c'} v'}$.

Moreover, generating the set of terms v' such that $v \rightarrow_{\mathcal{R}}^{c'} v'$ is a finite and computable task. Therefore, $D[d]$ defined as the set of derivations of the form $\frac{d:u \xrightarrow{*}_{\mathcal{R}}^c v \quad d':v \xrightarrow{c'}_{\mathcal{R}} v'}{\overline{c} \wedge \bigwedge_{i \in I} c_i} \quad u \xrightarrow{*}_{\mathcal{R}}^{i \in I} v'}$, is finite. However, without supplementary

conditions, $D[d]$ is not necessarily computable. Indeed, it is required for every $i \in I$ that $u_i =_{c_i} v_i$ holds in \mathcal{R} . Therefore, this process can loop. This had been already observed in the classic conditional rewriting when we were generating the set $RS[u =_c v]$. To solve this problem, the condition which is imposed on rewrite systems, is decreasingness [23]. Here, this condition is expressed as follows:

DEFINITION 9.1 (Decreasing). *A conditional rewrite system is decreasing if there is a well-founded ordering $>$ on terms which satisfies the following properties:*

1. $\rightarrow_{\mathcal{R}} \subseteq >$,
2. $>$ contains the proper subterm relation, and
3. for each rule $l \rightarrow_c r$ with $c = \bigwedge_{1 \leq i \leq n} u_i = v_i$, $\sigma(l) > \sigma(u_i)$ and $\sigma(l) > \sigma(v_i)$ for all substitutions σ and all indices $1 \leq i \leq n$.

Decreasingness is essentially the same condition as imposed on conditional equations by theorem-proving procedures in [18, 19]. Indeed, in [18, 19], a conditional equation $c \Rightarrow u = v$ is decreasing if for every substitution σ , $\sigma(u) > \sigma(v)$ (similar to Point 1. in Definition 9.1) and the proofs of the conditions only involve terms smaller than $\sigma(u)$ (similar to Point 3. in Definition 9.1). The point of decreasingness as defined in [18, 19] is to enable one to use equations by using only the decreasing instances of the equations, much like in unfailing Knuth-Bendix completion or Aka ordered completion where one uses the ordered instances of equations to perform simplification. Here, we do not present the approach developed in [18, 19] which defines theorem-proving procedures for conditional equations rather than completion procedures.

THEOREM 9.2. *If \mathcal{R} is decreasing then $\xrightarrow{*}_{\mathcal{R}}$ is decidable.*

Proof Let us prove this property by induction on the ordering that makes \mathcal{R} decreasing. Let s be a term which is not a normal form. By definition,

there is a rewrite rule $l \rightarrow_c r$ with $c = \bigwedge_{1 \leq i \leq n} u_i = v_i$, a context C and a substitution σ such that $s = C[\sigma(l)] \rightarrow_{\mathcal{R}}^{\sigma(c)} t = C[\sigma(r)]$. By definition, $s > C[\sigma(r)]$. Therefore, by the induction hypothesis, there is a normal form w such that $C[\sigma(r)] \xrightarrow{*}_{\mathcal{R}}^{c'} w$ with $c' = \bigwedge_{1 \leq j \leq k} u'_j = v'_j$. By definition, $C[\sigma(l)] > \sigma(l) > \sigma(u_i), \sigma(v_i)$ for $i, 1 \leq i \leq n$, and $C[\sigma(r)] > \sigma(r) > \sigma(u'_j), \sigma(v'_j)$ for $j, 1 \leq j \leq k$. Therefore, by the induction hypothesis, we can decide for all $i, 1 \leq i \leq n$ and all $j, 1 \leq j \leq k$, if for some c_i and c'_j , $u_i = v_i$ and $u'_j = v'_j$ hold in \mathcal{R} . Consequently, for every $c'' \subseteq \sigma(c) \wedge c'$ we can decide if $s \xrightarrow{*}_{\mathcal{R}}^{c'' \wedge cd} w$ where $cd = \bigwedge_{i \in I} c_i \wedge \bigwedge_{j \in J} c'_j$ such that $cd \subseteq (\sigma(c) \wedge c') \setminus c''$ and for every $i \in I$ and $j \in J$, $u_i =_{c_i} v_i$ and $u'_j =_{c'_j} v'_j$ hold in \mathcal{R} . \square

As a corollary of Theorem 9.2, for any decreasing rewrite system \mathcal{R} , the set $D[d]$ is computable.

Hence, from the two points above, we can conclude that decreasing conditional rewrite systems are sensible.

By the first condition of the notion of sensibleness, derivations can be written as sequences of rewriting steps together with a finite conjunction c . Hence, any derivation $d : t \xrightarrow{*}_{\mathcal{R}}^c t'$ is written $(t \rightarrow_{\mathcal{R}}^{c_1} t_1 \rightarrow_{\mathcal{R}}^{c_2} t_2 \dots \rightarrow_{\mathcal{R}}^{c_{n+1}} t', c)$ where all $t_i \rightarrow_{\mathcal{R}}^{c_i} t_{i+1}$ are the leaves (i.e. rewriting steps) occurring in d . Moreover, among all derivations associated with a sequence $(t \rightarrow_{\mathcal{R}}^{c_1} t_1 \rightarrow_{\mathcal{R}}^{c_2} t_2 \dots \rightarrow_{\mathcal{R}}^{c_{n+1}} t', c)$ we have the following one:

$$\frac{\frac{t \rightarrow_{\mathcal{R}}^{c_1} t_1 \quad t_1 \rightarrow_{\mathcal{R}}^{c_2} t_2}{t \xrightarrow{*}_{\mathcal{R}}^{c_1 \wedge c_2} t_2} \quad t_2 \rightarrow_{\mathcal{R}}^{c_3} t_3}{t \xrightarrow{*}_{\mathcal{R}}^{c_1 \wedge c_2 \wedge c_3} t_3} \quad \dots \quad \frac{t \xrightarrow{*}_{\mathcal{R}}^{c_1 \wedge \dots \wedge c_{n-1}} t_n \quad t_n \rightarrow_{\mathcal{R}}^{c_{n+1}} t'}{t \xrightarrow{*}_{\mathcal{R}}^c t'}}$$

where only in the last application of the Trans/Mod rule, we check that for every $u_i = v_i \in (\bigwedge_{1 \leq j \leq n+1} c_j) \setminus c$ there exists a finite conjunction $c_i \subseteq c$ such that $u_i =_{c_i} v_i$ holds in \mathcal{R} .

In the same way, a proof can be written as a sequence of rewriting steps (with some orientation conflicts) together with a finite conjunction c .

The abstract decision procedure given in Section 5 is then instantiated as follows:

Input a decreasing rewrite system \mathcal{R} and a conditional equation $u =_c v$.

Initialization $S, Tmp := RS[u =_c v]$, $Tmp := RS[u =_c v]$, and $answer := false$;

Loop while $Tmp \neq \emptyset$ do:

- 1) choose $d : t_1 \rightarrow_{\mathcal{R}}^{c_1} t_2 \dots \rightarrow_{\mathcal{R}}^{c_n} t_n$ in Tmp and $Tmp := Tmp \setminus \{d\}$;³⁴
- 2) $Tmp := Tmp \cup \{t_1 \rightarrow_{\mathcal{R}}^{c_1} t_2 \dots \rightarrow_{\mathcal{R}}^{c_n} t_n \rightarrow_{\mathcal{R}}^{c_n} t_{n+1} \mid t_n \rightarrow_{\mathcal{R}}^{c_n} t_{n+1}\}$;
- 3) $S := S \cup Tmp$;
- 4) if there exist $u \rightarrow_{\mathcal{R}}^{c_{11}} t_{11} \dots \rightarrow_{\mathcal{R}}^{c_{1n}} w$ and $v \rightarrow_{\mathcal{R}}^{c_{21}} t_{21} \dots \rightarrow_{\mathcal{R}}^{c_{2m}} w$ in S , such that:

$$\forall u_i = v_i \in \left(\bigwedge_{1 \leq i \leq n} c_{1i} \wedge \bigwedge_{1 \leq j \leq m} c_{2j} \right) \setminus c, \exists c_i \subseteq c, \mathcal{A}[u_i =_{c_i} v_i]$$

then $Tmp := \emptyset$;
 $answer := true$;

end of loop

Output $return(answer)$

Here, the rfs associated with the conditional logic is not adequate. But, with the condition of decreasingness, the above process cannot loop.

A conditional rewrite system \mathcal{R} is:

- *confluent* if and only if for every $\frac{d_1:u \xrightarrow{\mathcal{R}}^{c_1} t \quad d_2:t \xrightarrow{\mathcal{R}}^{c_2} v}{u \xleftrightarrow{\mathcal{R}}^c v}$ there exists $\frac{d_3:u \xrightarrow{\mathcal{R}}^{c_3} t \quad d_4:t \xrightarrow{\mathcal{R}}^{c_4} v}{u \xleftrightarrow{\mathcal{R}}^c v}$
- *Church-Rosser* if and only if for every proof $\pi : u \xleftrightarrow{\mathcal{R}}^c v$ there exists $\frac{d_1:u \xrightarrow{\mathcal{R}}^{c_1} t \quad d_2:t \xrightarrow{\mathcal{R}}^{c_2} v}{u \xleftrightarrow{\mathcal{R}}^c v}$.

Axiom 2 holds because, for any $\frac{d_1:u \xrightarrow{\mathcal{R}}^{c_1} t \quad d_2:t \xrightarrow{\mathcal{R}}^{c_2} v}{u \xleftrightarrow{\mathcal{R}}^c v}$, we also have $\frac{d_2:v \xrightarrow{\mathcal{R}}^{c_2} t \quad d_1:t \xrightarrow{\mathcal{R}}^{c_1} u}{v \xleftrightarrow{\mathcal{R}}^c u}$.

Axiom 3 holds because, as in the standard equational rewriting, proofs can be written as a series of maximal peaks, and then replacing any maximal peak

³⁴ Observe that t_1 is necessarily either u or v .

by a valley reduces by one unit the number of maximal peaks. Therefore, we have as corollary of Theorem 6.5 that Church-Rosser is equivalent to confluence.

Every peak obviously contains a local peak, and so, Axiom 4 is obvious. Finally, we can easily adapt the proof developed in Example 6.10 to show that Axiom 5 holds.

9.1.4. Completion

Inference rules for completion are the following: let \succ be a reduction order which contains the proper subterm relation

DELETE1	$\frac{\Gamma \cup \{u=0u\}, \mathcal{R}}{\Gamma, \mathcal{R}}$	
DELETE2	$\frac{\Gamma \cup \{u=cv\}, \mathcal{R}}{\Gamma, \mathcal{R}}$	if $u = v \in c$
DEDUCE	$\frac{\Gamma, \mathcal{R}}{\Gamma \cup \{u=cv\}, \mathcal{R}}$	if $(u \xrightarrow{c_1} u' \rightarrow_{\mathcal{R}}^{c_2} v, c)$
ORIENT	$\frac{\Gamma \cup \{u=cv\}, \mathcal{R}}{\Gamma, \mathcal{R} \cup \{u \rightarrow cv\}}$	if $u \succ v$ and for all $l = r \in c$, $u \succ l$ and $u \succ r$
SIMPLIFY	$\frac{\Gamma \cup \{u=cv\}, \mathcal{R}}{\Gamma \cup \{u'=c_2v\} \cup \{t=c't' \mid \mathcal{P}(t, t', c')\}, \mathcal{R}}$	if 1. $u \xrightarrow{c_1} u'$ 2. $\exists c_2, u' \xrightarrow{*}_{\Gamma \cup \{u=cv\}, \mathcal{R}}^{c_2} v$ 3. $\forall t = t' \in (c_1 \wedge c_2) \setminus c$, $\exists c' \subseteq c, t \xrightarrow{*}_{\Gamma \cup \{u=cv\}, \mathcal{R}}^{c'} t'$
	$\text{where } \mathcal{P}(t, t', c') \Leftrightarrow \left[\begin{array}{l} (t = t' \in (c_1 \wedge c_2) \setminus c) \wedge \\ (c' \subseteq c) \wedge \\ (t \xrightarrow{*}_{\Gamma \cup \{u=cv\}, \mathcal{R}}^{c'} t') \end{array} \right.$	

The supplementary condition in the inference rule ORIENT enables us to obtain decreasing rewrite systems \mathcal{R} in pairs (Γ, \mathcal{R}) manipulated by the above completion.

Mixed proofs can also be written as a series of proof steps that are either rewriting steps or equations. Hence, by using Bachmair's ordering, we can easily show that Axiom 6 is satisfied. However, the two last conditions of the inference rule SIMPLIFY are unlikely to be computable because of both proof obligations $u' \xrightarrow{*}_{\Gamma \cup \{u=cv\}, \mathcal{R}}^{c_2} v$ and $t \xrightarrow{*}_{\Gamma \cup \{u=cv\}, \mathcal{R}}^{c'} t'$. To solve these problems,

a solution is to break down the deductive rule Trans/Mod into two deductive rules:

$$\text{Transitivity} \frac{t =_c t' \quad t' =_{c'} t''}{t =_{c \wedge c'} t''}$$

$$\text{ModusPonens} \frac{t =_{c \wedge \bigwedge_{i \in I} u_i = v_i} t' \quad u_i =_c v_i}{t =_c t'}$$

In this new rfs, all above notions are easily adaptable and left to the reader. Of course, both new rules are more restrictive than Trans/Mod because in Modus-Ponens, each equation $u_i = v_i$ is satisfied with respect to the finite conjunction c .

Both new rules give rise to the following inference rules:

$$\text{SIMPLIFY1} \quad \frac{\Gamma \cup \{u =_c v\}, \mathcal{R}}{\Gamma \cup \{u' =_{c_2} v\}, \mathcal{R}} \quad \text{if } 1. u \xrightarrow{c_1} \mathcal{R} u' \text{ with } c_1 \subseteq c \text{ and } c = c_1 \wedge c_2$$

$$\text{SIMPLIFY2} \quad \frac{\Gamma \cup \{u =_c v\}, \mathcal{R}}{\Gamma \cup \{s =_{c'} t \mid s = t \in c'\}, \mathcal{R}} \quad \text{if } 1. u \xrightarrow{c \wedge c'} \mathcal{R} v$$

9.2. REWRITING OF SPECIAL RELATIONS

9.2.1. Rewriting formal system

The logic of special relations is both a restriction and an extension of first order logic. It is a restriction because it only considers binary predicates. It is an extension because it allows to specify composition laws such as transitivity or subtyping, and then to define new binary relations. This is obtained by adding two supplementary syntactic operators “ \smile ” and “ $;$ ” which denote respectively, the opposite and the composition of binary relations. These new operators allow to build, in addition to terms with variables of the form $f(t_1, \dots, t_n)$, *relational terms* of the form $\widetilde{\alpha}$ and $\alpha; \beta$ where α and β are binary relations. Formulæ are also extended by expressing that a binary relation is included into another one. More formally, the rfs for logic of special relations is defined as follows: a signature $\Sigma = (F, R)$ contains a set F of function names with arities in \mathbb{N} , and a set R of binary relation names. The set of terms with variables is the standard set $T_\Sigma(V)$ of free terms with generators in V . The set of *relational terms* TR_Σ is the least set inductively defined as follows:

- $\mathbb{I} \in TR_\Sigma$ (\mathbb{I} is the identity relation neutral for composition)
- $R \subseteq TR_\Sigma$,
- if $\alpha, \beta \in TR_\Sigma$ then $\alpha; \beta \in TR_\Sigma$, and

- if $\alpha \in TR_\Sigma$ then $\widetilde{\alpha} \in TR_\Sigma$.

“;” denotes the composition of binary relations, and is associative. “ $\widetilde{}$ ” denotes the opposite of binary relations, and is an anti-involution, that is, $\widetilde{\widetilde{\alpha}} = \alpha$, $\widetilde{\mathbb{I}} = \mathbb{I}$ and $\alpha; \beta = \beta; \widetilde{\alpha}$.

Monotonicity and antimonotonicity are inherent features of function names of signatures, in the same sense as their arities. Hence, a function f^n of F which is said *positive in position i with respect to (α, β)* satisfies the following property:

$$t_i \alpha'_i \implies f(t_1, \dots, t_i, \dots, t_n) \beta f(t_1, \dots, t'_i, \dots, t_n)$$

This can be easily extended to subterm position p within a term t as follows:

- a term $f(t_1, \dots, t_n) \in T_\Sigma(V)$ is positive in position $1 \leq i \leq n$ with respect to (α, β) if and only if f is positive in i with respect to (α, β) , and
- if $p.q$ is a subterm position in a term $t \in T_\Sigma(V)$ such that $t_{|_p}$ is positive in q with respect to (α, γ) and t is positive in p with respect to (γ, β) , then t is positive in $p.q$ with respect to (α, β) .

Atoms are either $u\alpha v$ where u, v are terms in $T_\Sigma(V)$ and $\alpha \in TR_\Sigma$, or $\alpha \sqsubseteq \beta$ where $\alpha, \beta \in TR_\Sigma$. Semantically, \sqsubseteq means the set-theoretical inclusion.

EXAMPLE 9.3. *As explained just above, the logic of special relations has been defined as an universal algebraic formalism in which other algebraic formalisms can be encoded. Here, we are going to illustrate such an encoding for the standard and typed equational logic.*

1. Standard equational logic. Given a set F of function names with arities in \mathbb{N} , we define the signature $\Sigma = (F, R)$ in the logic of special relations as follows:

- $R = \{=\}$, and
- for each function symbol $f^n \in F$, f is positive in every position $1 \leq i \leq n$ for $(=, =)$.

Moreover, for every theory Γ we add the three following axioms:

- a) $\mathbb{I} \sqsubseteq =$ (reflexivity)
- b) $= \sqsubseteq \widetilde{=}$ (symmetry)
- c) $=; = \sqsubseteq =$ (transitivity)

2. Typed equational logic. *Typed equational logic* [41] extends standard equational logic by treating sorts semantically (i.e. by axioms). Its formulae are Horn clauses involving equations $t = t'$ as well as type assignments $t : t'$ where t, t' are terms of $T_\Sigma(V)$. Therefore, given a set F of function names signature, the signature $\Sigma = (F, R)$ in the logic of special relations is defined as for the standard equational logic except that $R = \{=, :\}$. Moreover, in addition to the three above axioms specifying equality, for every theory Γ , we have the two following axioms to specify type assignments:

a) $=; \sqsubseteq$: (typing of equals)

b) $::; =\sqsubseteq$: (type equality)

Given a signature $\Sigma = (F, R)$ and a set of variables V , we define the rfs (T, E, RS, De, Rmv) as follows:

- $T = T_\Sigma(V) \cup TR_\Sigma$,
- $E = TR_\Sigma \cup \{\sqsubseteq\}$ such that for every $\alpha \in TR_\Sigma$, $\alpha \stackrel{def}{=} T_\Sigma(V) \times T_\Sigma(V)$ and $\sqsubseteq \stackrel{def}{=} TR_\Sigma \times TR_\Sigma$,
- If we index each set of deductive rules instances RS , De , and Rmv by the set E , we have:
 - De_{\sqsubseteq} contains all the instances of the following deductive rules:

$$\text{Reflexivity} \frac{}{t \sqsubseteq t}$$

- for every $\alpha \in TR_\Sigma$:
 - * RS_α contains all instances of the two following deductive rules: if $\sigma : V \rightarrow T_\Sigma(V)$ is a substitution and C is a context positive in subterm position p of the symbol \square in C with respect to (β, α) , then

$$\text{Substitution} \frac{t\alpha t'}{\sigma(t)\alpha\sigma(t')} \quad \text{Context} \frac{t\beta t'}{C[t]\alpha C[t']}$$

- * De_α contains all instances of the following deductive rule:

$$\text{Composition} \frac{u\gamma v \quad v\beta w \quad \gamma; \beta \sqsubseteq \alpha}{u\alpha w}$$

- * Rmv_α contains all instances of the following deductive rule:

$$\text{Converse} \frac{u\beta v}{v\beta u}$$

and then $\alpha = \widetilde{\beta}$

- and:

- * RS_{\sqsubseteq} is the set containing all instances of the two following deductive rules:

$$\text{Duality} \frac{\alpha \sqsubseteq \beta}{\widetilde{\alpha} \sqsubseteq \widetilde{\beta}} \quad \text{Context}_{\sqsubseteq} \frac{\alpha \sqsubseteq \beta}{\gamma; \alpha; \delta \sqsubseteq \gamma; \beta; \delta}$$

- * De_{\sqsubseteq} is the set containing all instances of the following deductive rule:

$$\text{Reflexivity} \frac{}{\alpha \sqsubseteq \alpha} \quad \text{Transitivity} \frac{\alpha \sqsubseteq \beta \quad \beta \sqsubseteq \delta}{\alpha \sqsubseteq \delta}$$

- * Rmv_{\sqsubseteq} is empty.

9.2.2. Rewrite systems

Rewriting developed here will be defined to solve the word problem of two kinds of atoms:

1. relational atoms of the form $u\alpha v$, and
2. inclusion atoms of the form $\alpha \sqsubseteq \beta$.

Rewrite systems for the logic of special relations is then defined for any binary relation α in TR_Σ (resp. for \sqsubseteq) by a pair of binary relations $(\rightarrow_\alpha, \leftarrow_\alpha)$ (resp. $(\rightarrow_{\sqsubseteq}, \leftarrow_{\sqsubseteq})$). Given an instance $\mathfrak{t} = \frac{u\alpha v \quad v\beta w \quad \alpha; \beta \sqsubseteq \gamma}{u\gamma w}$ (resp. $\mathfrak{t} = \frac{\alpha \sqsubseteq \gamma \quad \gamma \sqsubseteq \delta}{\alpha \sqsubseteq \delta}$), its set of fixed leaves is $\mathcal{FL}(\mathfrak{t}) = \{u\alpha v, v\beta w\}$ (resp. $\mathcal{FL}(\mathfrak{t}) = \{\alpha \sqsubseteq \gamma, \gamma \sqsubseteq \delta\}$), and then $\mathcal{L}(\mathfrak{t}) \setminus \mathcal{FL}(\mathfrak{t}) = \{\alpha; \beta \sqsubseteq \gamma\}$ (resp. $\mathcal{L}(\mathfrak{t}) \setminus \mathcal{FL}(\mathfrak{t}) = \emptyset$).

Following Definition 4.6, a rewriting step is then defined as follows:

- for every $\alpha \in TR_\Sigma$, we have: $t \rightarrow_{\mathcal{R}}^\alpha t'$ (resp. $t \leftarrow_{\mathcal{R}}^\alpha t'$) if and only if there are a rewrite rule $u \rightarrow^\beta v$ (resp. $u \leftarrow^\beta v$), a substitution $\sigma : V \rightarrow T_\Sigma(V)$ and a context C positive in position p of the constant \square with respect to (β, α) such that $t = C[\sigma(u)]_p$ and $t' = C[\sigma(v)]_p$.
- $\rightarrow_{\mathcal{R}}^{\sqsubseteq}$ is the least set satisfying the following clauses:
 - $\rightarrow_{\sqsubseteq} \subseteq \rightarrow_{\mathcal{R}}^{\sqsubseteq}$,

- if $\alpha \rightarrow_{\mathcal{R}}^{\sqsubseteq} \beta$ then $\widetilde{\alpha} \rightarrow_{\mathcal{R}}^{\sqsubseteq} \widetilde{\beta}$,
- if $\alpha \rightarrow_{\mathcal{R}}^{\sqsubseteq} \beta$ then $\gamma; \alpha; \delta \rightarrow_{\mathcal{R}}^{\sqsubseteq} \gamma; \beta; \delta$.

$\leftarrow_{\mathcal{R}}^{\sqsubseteq}$ is defined as above by replacing $\rightarrow_{\sqsubseteq}$ and $\rightarrow_{\mathcal{R}}^{\sqsubseteq}$ by \leftarrow_{\sqsubseteq} and $\leftarrow_{\mathcal{R}}^{\sqsubseteq}$, respectively.

Closure of rewriting steps is then defined as follows:

- for every $\gamma \in TR_{\Sigma}$, $\xrightarrow{\gamma}_{\mathcal{R}}^*$ is the least set satisfying the following clauses:
 - $t \xrightarrow{\gamma}_{\mathcal{R}}^* t$ if $\gamma = \mathbb{I}$,
 - $\rightarrow_{\mathcal{R}}^{\gamma} \subseteq \xrightarrow{\gamma}_{\mathcal{R}}^*$, and
 - if $t \xrightarrow{\alpha}_{\mathcal{R}}^* t'$, $t' \xrightarrow{\beta}_{\mathcal{R}}^* t''$ and:
 - * *Normal* $\alpha; \beta \xrightarrow{\sqsubseteq}_{\mathcal{R}}^* \gamma$,
 - * *Natural* $\alpha; \beta \xrightarrow{\sqsubseteq}_{\mathcal{R}}^* \gamma$
 - * *Join* $\alpha; \beta \xrightarrow{\sqsubseteq}_{\mathcal{R}}^* \delta \xrightarrow{\sqsubseteq}_{\mathcal{R}}^* \gamma$
 then $t \xrightarrow{\gamma}_{\mathcal{R}}^* t''$.
- $\xrightarrow{\sqsubseteq}_{\mathcal{R}}^*$ is the least set (according to the set-theoretical inclusion) containing $\rightarrow_{\mathcal{R}}^{\sqsubseteq}$ and closed under transitivity.

From the definitions above, derivations can be defined by sequences of form $(t_1 \xrightarrow{\alpha_1}_{\mathcal{R}}^* t_2 \xrightarrow{\alpha_2}_{\mathcal{R}}^* \dots \xrightarrow{\alpha_n}_{\mathcal{R}}^* t_{n+1}, \beta)$ or $\alpha_1 \xrightarrow{\sqsubseteq}_{\mathcal{R}}^* \alpha_2 \xrightarrow{\sqsubseteq}_{\mathcal{R}}^* \dots \xrightarrow{\sqsubseteq}_{\mathcal{R}}^* \alpha_{n+1}$. A derivation $(t_1 \xrightarrow{\alpha_1}_{\mathcal{R}}^* t_2 \xrightarrow{\alpha_2}_{\mathcal{R}}^* \dots \xrightarrow{\alpha_n}_{\mathcal{R}}^* t_{n+1}, \beta)$ leads to $t_1 \xrightarrow{\beta}_{\mathcal{R}}^* t_{n+1}$ and means that $\alpha_1; \dots; \alpha_n \sqsubseteq \beta$ holds in \mathcal{R} . Proofs are defined similarly.

9.2.3. Properties of rewrite systems

For every instance \mathfrak{t} of both deductive rules composition and transitivity, we have $\overrightarrow{Ex}(\mathfrak{t}) = 1$ and $\overleftarrow{Ex}(\mathfrak{t}) = 2$. Hence, termination of rewrite systems is expressed as the fact that $\mathcal{E}_{\succ_{\mathcal{R}}}$ is Noetherian where $\mathcal{E}_{\succ_{\mathcal{R}}} = (\mathcal{E}_{\succ_{\mathcal{R}}^{\alpha}})_{\alpha \in TR_{\Sigma}} \cup \mathcal{E}_{\succ_{\mathcal{R}}^{\sqsubseteq}}$ such that:

- for every $\alpha \in TR_{\Sigma}$, $\succ_{\mathcal{R}}^{\alpha}$ is the least binary relation that contains $\rightarrow_{\mathcal{R}}^{\alpha} \cup (\leftarrow_{\mathcal{R}}^{\alpha})^{-1}$, and is stable under substitution, positive context and composition, that is
 - if $t \succ_{\mathcal{R}}^{\alpha} t'$ then $\sigma(t) \succ_{\mathcal{R}}^{\alpha} \sigma(t')$,

- if $t \succ_{\mathcal{R}}^{\beta} t'$ and C is a positive context with respect to (β, α) then $C[t] \succ_{\mathcal{R}}^{\alpha} C[t']$, and
 - if $u \succ^{\gamma} v$, $v \succ^{\beta} w$ and $\gamma; \beta \succ_{\mathcal{R}}^{\square} \alpha$, then $u \succ^{\alpha} w$.
- $\succ_{\mathcal{R}}^{\square}$ is the least binary relation that contains $\rightarrow_{\mathcal{R}}^{\square} \cup (\leftarrow_{\mathcal{R}}^{\square})^{-1}$, and is stable under duality, context $_{\square}$ and transitivity, that is
- if $\alpha \succ_{\mathcal{R}}^{\square} \beta$ then $\widetilde{\alpha} \succ_{\mathcal{R}}^{\square} \widetilde{\beta}$,
 - if $\alpha \succ_{\mathcal{R}}^{\square} \beta$ then $\gamma; \alpha; \delta \succ_{\mathcal{R}}^{\square} \gamma; \beta; \delta$, and
 - if $\alpha \succ_{\mathcal{R}}^{\square} \beta$ and $\beta \succ_{\mathcal{R}}^{\square} \delta$ then $\alpha \succ_{\mathcal{R}}^{\square} \delta$.

Therefore, a reduction relation $\succ = (\bigcup_{\alpha \in TR_{\Sigma}} \succ_{\alpha}) \cup \succ_{\square}$ is a binary relation satisfying:

- for every $\alpha \in TR_{\Sigma}$, \succ_{α} is a well-founded relation that contains $\rightarrow_{\mathcal{R}}^{\alpha} \cup (\leftarrow_{\mathcal{R}}^{\alpha})^{-1}$, and is stable under substitution, positive context and composition, and
- \succ_{\square} is a well-founded order that contains $\rightarrow_{\mathcal{R}}^{\square} \cup (\leftarrow_{\mathcal{R}}^{\square})^{-1}$, and is stable under duality and context $_{\square}$.

In order to benefit from classic methods to show termination of rewrite systems, we can set down, for every $\alpha \in TR_{\Sigma}$, that $\succ_{\alpha} = >$ where $>$ is a well-founded order stable under substitution and positive context. The resulting binary relation $\succ = > \cup \succ_{\square}$ is obviously a reduction relation because transitivity closure is less restrictive than composition closure.

This leads to the following definition of effluences and proofs by rewriting:

- effluences are proofs of the form $(s \xleftarrow{\alpha}^*_{\mathcal{R}} t \xrightarrow{\beta}^*_{\mathcal{R}} u, \gamma)$, and
- proofs by rewriting are proofs of the form $(s \xrightarrow{\alpha}^*_{\mathcal{R}} t \xleftarrow{\beta}^*_{\mathcal{R}} u, \gamma)$.

such that $\alpha; \beta \square \gamma$ holds in \mathcal{R} .

Regarding the binary relation \square , effluences and proofs by rewriting are respectively, peaks $(\xleftarrow{\square}^*_{\mathcal{R}} \circ \xrightarrow{\square}^*_{\mathcal{R}})$ and valleys $(\xrightarrow{\square}^*_{\mathcal{R}} \circ \xleftarrow{\square}^*_{\mathcal{R}})$.

Let $u\alpha\beta$ be a formula. Therefore, let us define the set \mathcal{S} as follows:

$$\mathcal{S} = \left[\begin{array}{c} \{(u', \gamma) \mid u \xrightarrow{\gamma}_{\mathcal{R}} u'\} \\ \cup \\ \{(v', \gamma) \mid v' \xleftarrow{\gamma}_{\mathcal{R}} v\} \end{array} \right.$$

Then, let us show that $RS[u =_{\alpha} v] = \left[\begin{array}{c} \{u \rightarrow_{\mathcal{R}}^{\gamma} u' \mid (u', \gamma) \in \mathcal{S}\} \\ \cup \\ \{v' \leftarrow_{\mathcal{R}}^{\gamma} v \mid (v', \gamma) \in \mathcal{S}\} \end{array} \right]$. Indeed, for

every binary relation $\gamma \in TR_{\Sigma}$, both $\rightarrow_{\mathcal{R}}^{\gamma}$ and $\leftarrow_{\mathcal{R}}^{\gamma}$ are finitely branching, that is each term has only finitely many direct successors. Moreover, $\rightarrow_{\mathcal{R}}^{\gamma}$ and $\leftarrow_{\mathcal{R}}^{\gamma}$ are decidable. Therefore, \mathcal{S} is finite and computable.

Observe that for every derivation $u \xrightarrow{\pm}_{\mathcal{R}}^{\alpha} w$ (resp. $w \xleftarrow{\pm}_{\mathcal{R}}^{\alpha} v$) there is a term u' (resp. v') and $\delta, \gamma \in TR_{\Sigma}$ such that $u \rightarrow_{\mathcal{R}}^{\delta} u' \xrightarrow{\pm}_{\mathcal{R}}^{\gamma} w$ (resp. $w \xleftarrow{\pm}_{\mathcal{R}}^{\gamma} v' \leftarrow_{\mathcal{R}}^{\delta} v$) and $\delta; \gamma \sqsubseteq \alpha$ holds in \mathcal{R} . This then means that $u \rightarrow_{\mathcal{R}}^{\delta} u' \mathcal{E}_{\mathcal{R}} u \xrightarrow{\pm}_{\mathcal{R}}^{\alpha} w$ (resp.

$v' \leftarrow_{\mathcal{R}}^{\delta} v \mathcal{E}_{\mathcal{R}} w \xleftarrow{\pm}_{\mathcal{R}}^{\alpha} v$). Therefore, $RS[u\alpha v] = \left[\begin{array}{c} \{u \rightarrow_{\mathcal{R}}^{\gamma} u' \mid (u', \gamma) \in \mathcal{S}\} \\ \cup \\ \{v' \leftarrow_{\mathcal{R}}^{\gamma} v \mid (v', \gamma) \in \mathcal{S}\} \end{array} \right]$. Con-

sequently, $RS[u\alpha v]$ is finite and computable.

Moreover, we have:

- for each rewriting $u \xrightarrow{*}_{\mathcal{R}}^{\alpha} v$ (resp. $u \xleftarrow{*}_{\mathcal{R}}^{\alpha} v$), there exists a sequence $u \rightarrow_{\mathcal{R}}^{\alpha_1} u_1 \rightarrow_{\mathcal{R}}^{\alpha_2} \dots \rightarrow_{\mathcal{R}}^{\alpha_n} v$ (resp. $u \leftarrow_{\mathcal{R}}^{\alpha_1} u_1 \leftarrow_{\mathcal{R}}^{\alpha_2} \dots \leftarrow_{\mathcal{R}}^{\alpha_n} v$) such that $\alpha_1; \dots; \alpha_n \sqsubseteq \alpha$ holds in \mathcal{R} .
- given a derivation $u \xrightarrow{*}_{\mathcal{R}}^{\alpha} v$ (resp. $u \xleftarrow{*}_{\mathcal{R}}^{\alpha} v$), v (resp. u) can be rewritten in one step, into a finite set of terms v' (resp. u') such that there exists $\gamma \in TR_{\Sigma}$ satisfying $v \rightarrow_{\mathcal{R}}^{\gamma} v'$ (resp. $u' \leftarrow_{\mathcal{R}}^{\gamma} u$). Moreover, generating the set of all v' (resp. u') such that there exists $\gamma \in TR_{\Sigma}$ satisfying $v \rightarrow_{\mathcal{R}}^{\gamma} v'$ (resp. $u' \leftarrow_{\mathcal{R}}^{\gamma} u$), is a finite and computable task. However, $D[u \xrightarrow{*}_{\mathcal{R}}^{\alpha} v]$ (resp. $D[u \xleftarrow{*}_{\mathcal{R}}^{\alpha} v]$) defined as the set of derivations $(u \xrightarrow{*}_{\mathcal{R}}^{\alpha} v \rightarrow_{\mathcal{R}}^{\gamma} v', \beta)$ (resp. $(u' \leftarrow_{\mathcal{R}}^{\gamma} u \xleftarrow{*}_{\mathcal{R}}^{\alpha} v, \beta)$), is not necessarily finite and computable because of the arbitrary choice of β and the proof obligation that $\alpha; \gamma \sqsubseteq \beta$ (resp. $\gamma; \alpha \sqsubseteq \beta$) holds in \mathcal{R} . To have both finiteness and computableness of $D[u \xrightarrow{*}_{\mathcal{R}}^{\alpha} v]$ (resp. $D[u \xleftarrow{*}_{\mathcal{R}}^{\alpha} v]$), we need that $\rightarrow_{\mathcal{R}}^{\sqsubseteq} \cup (\rightarrow_{\mathcal{R}}^{\sqsubseteq})^{-1}$ is terminating. Let us observe that right and left-hand sides of rewrite rules of the form $\alpha \rightarrow_{\mathcal{R}}^{\sqsubseteq} \beta$ are ground (relational terms have no variables). Therefore, when the set of this kind of rewrite rules is finite then it is known that its termination is decidable [6]. Moreover, in practice, relational atoms have usually the following form: $\alpha_1; \alpha_2; \dots; \alpha_n \sqsubseteq \beta$ with for every i , $1 \leq i \leq n$, $\alpha_i \in R$ and $\beta \in R$. Such atoms are sufficient to express usual composition laws such as transitivity of a relation α ($\alpha; \alpha \sqsubseteq \alpha$), typing ($:: = \sqsubseteq$: and $:: : \sqsubseteq$:) or Leibniz law ($\cong; \alpha \sqsubseteq \alpha$).³⁵ Such atoms are obviously oriented

³⁵ Leibniz law is expressed by $\frac{x=y \quad P(x)}{P(y)}$ where P is a unary predicate. Here, $P(_) = _ \alpha z$.

as $\alpha_1; \dots; \alpha_n \rightarrow_{\sqsubseteq} \beta$. The resulting rewrite system is then called *right-reduced*, that is all rewrite rules are right-ground and their right-hand sides are irreducible. In this case, the rewrite system is terminating.

For the binary relation \sqsubseteq , the three sets $RS[\alpha \sqsubseteq]$, $D[\alpha \xrightarrow{*}_{\mathcal{R}} \sqsubseteq \beta]$ and $D[\alpha \xleftarrow{*}_{\mathcal{R}} \sqsubseteq \beta]$ are defined similarly. Hence, we conclude that when dealing with a rewrite-system \mathcal{R} equipped with a terminating rewriting relation for \sqsubseteq then \mathcal{R} is sensible. This gives rise to the following decision procedure:

Input a rewrite system \mathcal{R} and a formula ϕ of the form $u\alpha v$ or $\alpha \sqsubseteq \beta$.

Initialization $S := RS[\phi]$, $Tmp := RS[\phi]$ and $answer := false$;

Loop while $Tmp \neq \emptyset$ do:

1) choose d in Tmp and $Tmp := Tmp \setminus \{d\}$;

2) $Tmp := Tmp \cup D[d]$;

3) $S := S \cup Tmp$;

4) if ϕ is of the form $u\alpha v$ then:

if there exist $(u \xrightarrow{\alpha_{11}}_{\mathcal{R}} t_{11} \dots \xrightarrow{\alpha_{1n}}_{\mathcal{R}} w, \beta_1)$ and
 $(w \xleftarrow{\alpha_{21}}_{\mathcal{R}} t_{21} \dots \xleftarrow{\alpha_{2m}}_{\mathcal{R}} v, \beta_2)$ in S , such that:

$$\mathcal{A}[\beta_1 \sqsubseteq \alpha] \wedge \mathcal{A}[\beta_2 \sqsubseteq \alpha]$$

then $Tmp := \emptyset$;

$answer := true$;

else (i.e. ϕ is of the form $\alpha \sqsubseteq \beta$):

if there exist $\alpha \xrightarrow{\sqsubseteq}_{\mathcal{R}} \dots \xrightarrow{\sqsubseteq}_{\mathcal{R}} \gamma$ and $\gamma \xleftarrow{\sqsubseteq}_{\mathcal{R}} \dots \xrightarrow{\sqsubseteq}_{\mathcal{R}} \beta$ in S

then $Tmp := \emptyset$;

$answer := true$;

end of loop

Output $return(answer)$

Here, the logic for special relations is adequate, and then Theorem 5.20 holds. A rewrite system \mathcal{R} is then:

- *Confluent* if and only if for every $(u \xleftarrow{*}_{\mathcal{R}} \alpha t \xrightarrow{*}_{\mathcal{R}} \beta v, \tau)$, there exists $(u \xrightarrow{*}_{\mathcal{R}} \delta w \xleftarrow{*}_{\mathcal{R}} \gamma v, \tau)$,³⁶ and

³⁶ In [49], τ is $\alpha; \beta$.

- Church-Rosser if and only if for every $u \xrightarrow{*}_{\mathcal{R}}^{\alpha} v$, there exists $(u \xrightarrow{*}_{\mathcal{R}}^{\delta} w \xrightarrow{*}_{\mathcal{R}}^{\gamma} v, \alpha)$.

Arguments to show that the four axioms of Section 6 hold are similar to those that can be found in the different examples developed in this paper, and then are left to the reader.

9.2.4. Completion

Inference rules for completion are then the following: let \succ be a reduction relation

- DEDUCE 1 $\frac{\Gamma, \mathcal{R}}{\Gamma \cup \{u\gamma v, \alpha; \beta \sqsubseteq \gamma\}, \mathcal{R}}$ if $(u \xleftarrow{*}_{\mathcal{R}}^{\alpha} t \xrightarrow{*}_{\mathcal{R}}^{\beta} v, \gamma)$
- DEDUCE 2 $\frac{\Gamma, \mathcal{R}}{\Gamma \cup \{\alpha \sqsubseteq \beta\}, \mathcal{R}}$ if $\alpha \xleftarrow{*}_{\mathcal{R}}^{\sqsubseteq} \delta \xrightarrow{*}_{\mathcal{R}}^{\sqsubseteq} \beta$
- ORIENT 1 $\frac{\Gamma \cup \{u\alpha v\}, \mathcal{R}}{\Gamma, \mathcal{R} \cup \{u \rightarrow_{\alpha} v\}}$ if $u \succ_{\alpha} v$
- ORIENT 2 $\frac{\Gamma \cup \{u\alpha v\}, \mathcal{R}}{\Gamma, \mathcal{R} \cup \{u \leftarrow_{\alpha} v\}}$ if $v \succ_{\alpha} u$
- ORIENT 3 $\frac{\Gamma \cup \{\alpha \sqsubseteq \beta\}, \mathcal{R}}{\Gamma, \mathcal{R} \cup \{\alpha \rightarrow_{\sqsubseteq} v\}} \beta$ if $\alpha \succ_{\sqsubseteq} \beta$
- ORIENT 4 $\frac{\Gamma \cup \{\alpha \sqsubseteq \beta\}, \mathcal{R}}{\Gamma, \mathcal{R} \cup \{\alpha \leftarrow_{\sqsubseteq} v\}} \beta$ if $\beta \succ_{\sqsubseteq} \alpha$
- SIMPLIFY 1 $\frac{\Gamma \cup \{u\alpha v\}, \mathcal{R}}{\Gamma \cup \{u'\delta v\} \cup \{\gamma; \delta \sqsubseteq \alpha\}, \mathcal{R}}$ if 1. $u \xrightarrow{+}_{\mathcal{R}}^{\gamma} u'$
 2. $\exists \delta \in TR_{\Sigma}, \left\{ \begin{array}{l} u' \xleftarrow{*}_{\Gamma \cup \{u\alpha v\}, \mathcal{R}}^{\delta} v \\ \wedge \\ \gamma; \delta \xleftarrow{*}_{\Gamma \cup \{u\alpha v\}, \mathcal{R}}^{\sqsubseteq} \alpha \end{array} \right.$
- SIMPLIFY 2 $\frac{\Gamma \cup \{u\alpha v\}, \mathcal{R}}{\Gamma \cup \{u\delta v'\} \cup \{\delta; \gamma \sqsubseteq \alpha\}, \mathcal{R}}$ if 1. $v' \xleftarrow{+}_{\mathcal{R}}^{\gamma} v$
 2. $\exists \delta \in TR_{\Sigma}, \left\{ \begin{array}{l} u \xleftarrow{*}_{\Gamma \cup \{u\alpha v\}, \mathcal{R}}^{\delta} v' \\ \wedge \\ \delta; \gamma \xleftarrow{*}_{\Gamma \cup \{u\alpha v\}, \mathcal{R}}^{\sqsubseteq} \alpha \end{array} \right.$
- SIMPLIFY 3 $\frac{\Gamma \cup \{\alpha \sqsubseteq \beta\}, \mathcal{R}}{\Gamma \cup \{\delta \sqsubseteq \beta\}, \mathcal{R}}$ if $\alpha \xrightarrow{+}_{\mathcal{R}}^{\sqsubseteq} \delta$ and $\delta \xleftarrow{*}_{\Gamma \cup \{\alpha \sqsubseteq \beta\}}^{\sqsubseteq} \beta$
- SIMPLIFY 4 $\frac{\Gamma \cup \{\alpha \sqsubseteq \beta\}, \mathcal{R}}{\Gamma \cup \{\alpha \sqsubseteq \delta\}, \mathcal{R}}$ if $\delta \xleftarrow{+}_{\mathcal{R}}^{\sqsubseteq} \beta$ and $\alpha \xleftarrow{*}_{\Gamma \cup \{\alpha \sqsubseteq \delta\}}^{\sqsubseteq}$

The arguments to show that Axiom 6 holds, are similar to other examples developed in this paper. However, the two last conditions of the inference

rule SIMPLIFY 1 and SIMPLIFY 2 are unlikely to be computable because of the arbitrary choice of δ and both proof obligations $u' \xleftrightarrow{\Gamma \cup \{u\alpha v\}, \mathcal{R}}^* \delta v$ (resp. $u \xleftrightarrow{\Gamma \cup \{u\alpha v\}, \mathcal{R}}^* \delta v'$) and $\gamma; \delta \xleftrightarrow{\Gamma \cup \{u\alpha v\}, \mathcal{R}}^* \alpha$ (resp. $\delta; \gamma \xleftrightarrow{\Gamma \cup \{u\alpha v\}, \mathcal{R}}^* \alpha$). We saw previously that when dealing with rewrite rules for \sqsubseteq which are right-reduced (which is often the case in practice), $\xrightarrow{\mathcal{R}}^* \sqsubseteq$ and $\xleftarrow{\mathcal{R}}^* \sqsubseteq$ are decidable, and then so is $\xleftrightarrow{\mathcal{R}}^* \sqsubseteq$. But, the problem remains with $u' \xleftrightarrow{\Gamma \cup \{u\alpha v\}, \mathcal{R}}^* \delta v$ and $u \xleftrightarrow{\Gamma \cup \{u\alpha v\}, \mathcal{R}}^* \delta v'$. Following [56], we then replace SIMPLIFY 1 and SIMPLIFY 2 by the three following inference rules:

SIMPLIFY

$$\frac{\Gamma \cup \{s\alpha t\}, \mathcal{R}}{\Gamma, \mathcal{R}} \quad \frac{\Gamma, \mathcal{R} \cup \{s \rightarrow_{\alpha} t\}}{\Gamma, \mathcal{R}} \quad \frac{\Gamma, \mathcal{R} \cup \{s \leftarrow_{\alpha} t\}}{\Gamma, \mathcal{R}}$$

if $s\alpha t$, $s \rightarrow_{\alpha} t$ or $u \leftarrow_{\alpha} v$ is redundant, that is it can be replaced by a smaller proof in $\Gamma \cup \mathcal{R}$ using the proof ordering and measure \succ .

10. Conclusion and perspectives

In this article, we have presented an axiomatic version of abstract rewriting used to generate convergent and complete rewrite systems. Hence, in a generic (i.e., logical-system independent) way, we have given meaning to the usual notions and results which underlie rewriting, such as rewrite systems, derivations, effluences, proofs by rewriting, termination, Church-Rosser' result or Newman's lemma. Moreover, for any Church-Rosser and confluent rewrite system \mathcal{R} , we have defined a decision procedure that is correct and complete with respect to the underlying theory of \mathcal{R} . Finally, a completion method has been presented in this abstract framework. This axiomatization follows a first paper in which we generalized the logicality theorem which states the correspondence between derivability (syntactic consequences obtained from \vdash) and convertibility in rewriting ($\xleftrightarrow{*}$) [3].

Several research issues can be continued. First, in standard term rewriting, when rewrite systems are terminating, Newman's lemma can be automatically tested. This is the critical pair Lemma. This result states that confluence of terminating systems can be effectively tested by checking joinability of a finite set of equations called *critical pairs*, formed by overlapping left-hand sides. Authors in [15, 20, 22] define an abstract notion of critical pairs as a proof which is not in normal form, but all its subproofs are. In our abstract framework, this exactly corresponds to any proof whose direct subproofs are derivations or proof by rewritings. Hence, such a notion of critical pairs in [15, 20, 22] does not correspond to the expected one, that is proofs which reflect all local effluences such that, given a rewrite system \mathcal{R} , the whole

set of critical pairs can be automatically produced. In the standard rewriting theories, to compute critical pairs, we use the inductive structure of terms. In our abstract framework, rewritten objects are simply elements of a set T ; no conditions are given on their structure. Therefore, we plan to restrict the definition of rfs to a generalization of first order terms (i.e. elements equipped with an inductive structure from basic elements playing the rôle of variables). To achieve this, we think that Lévy and Mellies's nice residual theory could provide pieces of solutions.

Another important research issue would be to find a way to adapt usual results of term rewriting such as modularity in our abstract framework.³⁷

³⁷ A property is said to be *modular* if it is preserved through disjoint union of rewrite systems.

Acknowledgements

Special thanks are due to Claude Kirchner and Michael Rusinowitch for many fruitful and instructive discussions and numerous useful comments. We also thank the three reviewers for their careful reading of this text and helpful comments.

References

1. M. Aiguier. Contribution à la définition et à l'abstraction de paradigme et de technique de conception et de vérification dans le domaine des méthodes formelles dédiées au génie-logiciel. University of Evry val-d'Essonne, Habilitation à Diriger les Recherches (HDR), 2003.
2. M. Aiguier and D. Bahrani. Une approche générique de la réécriture. *Techniques et Science Informatiques (TSI)*, 22(4), 2003.
3. M. Aiguier, D. Bahrani, and C. Dubois. On a generalised logicality theorem. In *Artificial Intelligence, Automated Reasoning, and Symbolic Computation (AISC)*, volume 2385 of *Lecture Notes in Artificial Intelligence*, pages 51–64. Springer, 2002.
4. M. Aiguier, C. Boin, and D. Longuet. On generalized theorems for normalization of proof trees. Technical report, LaMI, University of Evry val-d'Essonne, 2005.
5. E. Astesiano and M. Cerioli. Free objects and equational deduction for partial conditional specifications. *Theoretical Computer Science*, 152:91–138, 1995.
6. F. Baader and T. Nipkow. *Term Rewriting and All That*. C.U. Press, 1998.
7. L. Bachmair. *Canonical Equational Proofs*. Progress in Theoretical Computer Science. Birkhäuser Springer, 1991.
8. L. Bachmair and N. Dershowitz. Equational inference, canonical proofs, and proof orderings. *Journal of ACM*, 41(2):236–276, 1994.
9. L. Bachmair, N. Dershowitz, and D.-A. Plaisted. *Resolution of Equations in Algebraic Structures*, volume II of *Rewriting Techniques*, chapter Completion without failure, pages 1–30. Academic Press, 1989.
10. L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4:217–247, 1994.
11. L. Bachmair and H. Ganzinger. Rewrite techniques for transitive relations. In *Proceedings of the annual Symposium on Logic in Computer Science*, pages 384–393. IEEE Computer Society, 1994.
12. D. Bahrani. *Une axiomatisation de la réécriture abstraite*. PhD thesis, LaMI, University of Évry, 2003.
13. J. Bergstra and J. Klop. Conditional rewrite rules: Confluence and termination. *Journal of Computer and System Sciences*, 32:323–362, 1986.
14. G. Bernot, P. Le Gall, and M. Aiguier. Label algebras and exception handling. *Science of Computer Programming*, 23:227–286, 1994.
15. M.-P. Bonacina and N. Dershowitz. Abstract canonical inference. *ACM Transactions on Computational Logic*, 2005. to appear.
16. M.-P. Bonacina and J. Hsiang. Towards a foundation of completion procedures as semidecision procedures. *Theoretical Computer Science*, 146:199–242, 1995.
17. B. Buchberger. Gröbner bases: an algorithmic method in polynomial ideal theory. In N.-K. Bose, editor, *Multidimensional Systems Theory*, pages 184–232, 1985.

18. N. Dershowitz. Canonical sets of Horn clauses. In A. Leach, B. Monien, and M. Rodriguez Artalejo, editors, *Proceedings of the 18th International Colloquium on Automata, Languages and Programming*, volume 510 of *Lecture Notes in Computer Science*, pages 267–278. Springer, 1991.
19. N. Dershowitz. Ordering-based strategies for Horn clauses. In J. Mylopoulos and R. Reiter, editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 118–124, 1991.
20. N. Dershowitz. Canonicity. In I. Dahn and L. Vigneron, editors, *Proc. of Fourth International Workshop on First-order Theorem Proving*, volume 86 of *Electronic Notes in Theoretical Computer Science*. Elsevier Science Publishers, 2003.
21. N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In *Handbook of TCS*, volume B: Formal Models and Semantics, pages 243–320. Elsevier, Amsterdam, 1990.
22. N. Dershowitz and C. Kirchner. Abstract canonical presentations. *Theoretical Computer Science*, 2005. to appear.
23. N. Dershowitz and M. Okada. A rationale for conditional equational programming. *Theoretical Computer Science*, 75:111–138, 1990.
24. N. Dershowitz, M. Okada, and G. Sivakumar. Canonical conditional rewrite systems. In E. Lusk and R. Overbeek, editors, *CADE*, volume 310 of *Lecture Notes in Computer Science*, pages 538–549. Springer, 1988.
25. H. Ehrig, M. Grosse-Rhode, and U. Wolter. On the role of category theory in the area of algebraic specification. In *Recent Trends in Algebraic Development Techniques*, volume 1130 of *Lecture Notes in Computer Science*, pages 17–48. Springer, 1995.
26. J. Fiadeiro and A. Sernadas. Structuring theories on consequence. In *Recent Trends in Algebraic Development Techniques*, volume 332 of *Lecture Notes in Computer Science*, pages 44–72. Springer, 1988.
27. R. Freese, J. Jecek, and J. Nation. Term rewriting system for lattice theory. *Journal of Symbolic Computation*, 16(3):279–288, 1993.
28. H. Ganzinger. A completion procedure for conditional equations. *Journal of Symbolic Computation*, 11:51–81, 1991.
29. J. Goguen and R. Burstall. Institutions: abstract model theory for specification and programming. *Journal of the ACM*, 39(1):95–146, 1992.
30. G. Gonthier, J.-J. Lévy, and P.-A. Melliès. An abstract standardisation theorem. In *Annual Symposium on Logic in Computer Science*, pages 72–81. IEEE Computer Society, 1992.
31. J. Hsiang. Refutational theorem proving using term rewriting systems. *Artificial Intelligence*, 25:255–300, 1985.
32. J. Hsiang and M. Rusinowitch. On word problems in equational theories. In Th.Ottman, editor, *Proceedings of the 14th International Conference on Automata, Languages and Programming*, volume 267 of *Lecture Notes in Computer Science*, pages 54–71. Springer, 1987.
33. J.-P. Jouannaud and H. Kirchner. Completion of a set of rules modulo a set of equations. *SIAM Journal of Computing*, 15:1155–1196, 1986.
34. S. Kaplan. Conditional rewrite rules. *Theoretical Computer Science*, 33:175–193, 1984.
35. S. Kaplan and J.-L. Remy. *Resolution of Equations in Algebraic Structures*, volume II of *Rewriting Techniques*, chapter Completion algorithms for conditional rewriting systems, pages 1–30. Academic Press, 1989.
36. D. Knuth and P. Bendix. Simple word problems in universal algebras. In *Computational Problems in Abstract Algebras*, pages 263–297. Pergamon Press, 1970.
37. E. Kounalis and M. Rusinowitch. On word problems in Horn theories. *Journal of Symbolic Computation*, 11(1-2):113–128, 1991.

38. D. Lankford. Some approaches to equality for computational logic: A survey and assessment. Memo ATP-36, Automatic Theorem Proving Project, University of Texas, Austin, 1977.
39. J. Levy and J. Agustí. Bi-rewrite systems. *Journal of Symbolic Computation*, 22:279–314, 1996.
40. J.-J. Lévy.
41. V. Manca, A. Salibra, and G. Scollo. Equational type logic. *Theoretical Computer Science*, 77:131–159, 1990.
42. P.-A. Melliès. *Description abstraite des systèmes de réécriture*. PhD thesis, Université Paris 7, 1996.
43. P.-A. Melliès. A stability theorem in rewriting theory. In *Annual Symposium on Logic in Computer Science*, pages 287–299. IEEE Computer Society, 1998.
44. J. Meseguer. General logics. In *Logic Colloq.'87*, pages 275–329. Holland, 1989.
45. R. Nieuwenhuis and A. Rubio. Theorem proving with ordering and equality constrained clauses. *Journal of Symbolic Computation*, 19(4):321–351, 1995.
46. G.-E. Peterson and M.-E. Stickel. Complete sets of reductions for some equational theories. *Journal of ACM*, 28:223–264, 1981.
47. A. Salibra and G. Scollo. A soft stairway to institutions. In M. Bidoit and C. Choppy, editors, *Recent Trends in Algebraic Development Techniques*, volume 655 of *Lecture Notes in Computer Science*, pages 310–329. Springer, 1993.
48. M. Schorlemmer. Term rewriting in a logic of special relations. In *Proceedings of the third International Workshop on Rewriting Logic and its Applications*, volume 15 of *Electronic Notes in Theoretical Computer Science*. North Holland, 1998.
49. M. Schorlemmer. On specifying and reasoning with special relations. In *Monografies de l'IIIA*, volume 10 of *IIIA-CSIC monograph*. Institut d'Investigació en Intel·ligència Artificial (CSIC), 1999.
50. M. Schorlemmer. Term rewriting in a logic of special relations. In *Proceedings of Algebraic Methodology and Software Technology*, volume 1546 of *Lecture Notes in Computer Science*, pages 178–195. Springer, 1999.
51. A. Sernadas, C. Sernadas, and C. Caleiro. Fibring of logics as a categorial construction. *Journal of Logic and Computation*, 9(2):149–179, 1999.
52. A. Sernadas, C. Sernadas, C. Caleiro, and T. Mossakowski. Categorical fibring of logics with terms and binding operators. In D. Gabbay and M. de Rijke, editors, *Frontiers of Combining Systems 2*, Research Studies Press, pages 295–316, 2000.
53. G. Struth. Non-symmetric rewriting. Technical report, MPI für Informatik, 1996.
54. G. Struth. On the word problem for free lattices. In H. Comon, editor, *Proceedings of Rewriting Techniques and Applications*, volume 1232 of *Lecture Notes in Computer Science*, pages 128–141. Springer, 1997.
55. G. Struth. *Canonical Transformations in Algebra, Universal Algebra and Logic*. PhD thesis, Institut Für Informatik, University of Saarlandes, 1998.
56. G. Struth. Knuth-Bendix completion for non-symmetric transitive relations. In M. van den Brand and R. Verma, editors, *Proceedings of the Second International Workshop on Rule-Based Programming (RULE2001)*, volume 59 of *Electronic Notes in Theoretical Computer Science*. Elsevier Science Publishers, 2001.