

# An abstract way to define rewriting logic

Marc Aiguier<sup>a,1</sup>, Diane Bahrami<sup>b</sup>, Delphine Longuet<sup>a</sup>

<sup>a</sup> *Université d'Évry, LaMI CNRS UMR 8042,  
Tour Evry II, 523 pl. des Terrasses de l'Agora F-91000 Évry*

<sup>b</sup> *CEA-LIST Saclay  
F-91191 Gif sur Yvette Cedex*

---

## Abstract

Since rewriting logic has been introduced, it has shown its adequateness both as a semantic and a logical framework. But the numerous applications of the rewriting logic in the above two areas has shown the importance of increasing its expressive power. Therefore, in order to facilitate this work, we will study in this paper how to generalize the transformation that from the equational logic has resulted in the rewriting logic. To achieve this purpose, we will show that there exists a valid and useful notion of rewriting logic associated to any rewriting theory fitting an abstract framework developed by two of the authors in previous papers.

*Key words:* Rewriting formal system, abstract rewrite system, abstract rewriting logic, reachability and provability models, soundness and completeness results

---

## 1 Introduction

Since rewriting logic has been introduced [21], it has shown its adequateness both as a semantic framework, particularly for concurrent and distributed computation, and as a logical framework, that is, a meta-logic in which other logics can be represented. Indeed, the basic axioms of this logic, which are rewrite rules of the form  $t \rightarrow t'$  where  $t$  and  $t'$  are terms over a given signature, can be read into two ways: either as the local transition of a concurrent system or the inference rule of some logic. For the former, rewriting logic then extends (equational) algebraic specifications to deal with dynamic and concurrent systems. Indeed, algebraic specifications have proven to be well-suited for describing complex data structures and the functional aspects of a software system. However, they are insufficient when applied to dynamic and

---

<sup>1</sup> e-mail: [aiguier@lami.univ-evry.fr](mailto:aiguier@lami.univ-evry.fr)

distributed systems. For the latter, rewriting logic is then a “universal” logic within which other formalisms can be translated.

The numerous applications of rewriting logic in the above two areas has shown the importance of increasing its expressive power. The expressive power of the standard rewriting logic can be increased in two ways, by extending either the computational capabilities such as introducing some probabilistic laws to basic transitions  $t \rightarrow t'$  [8,7] or real-time aspects [18], or the logical capabilities by considering another logic than the conditional equational logic to parameterize rewriting logic such as the membership equational logic [22] with frozen operators [9].

When we observe all these extensions, at each time, they lead to the three questions:

- (i) What are the rules of deduction for this extended rewriting logic?
- (ii) What are the models of a rewrite theory? Are there initial and free models?
- (iii) Is rewriting logic complete with respect to its model theory?

In the future, other applications will certainly lead to extend the standard rewriting logic to other peculiar aspects. These new extensions naturally lead to answer the three above questions. However, as this has been observed in [7,9], these extensions are usually nontrivial generalizations of the original inference rules, model theory, initial and free models, and completeness theorem for rewriting logic over equational logic as developed in [21]. Therefore, in order to facilitate this work, it can be useful to study how to define rewriting logic and how to answer the three above questions at a more abstract level. This is what we propose to do in this paper. This requires first to give an abstract form of logics which parameterize rewriting logic, and then to study rewriting in this abstract framework of logics. In previous papers [3,2,1], we proposed such a general framework of rewriting by applying the paradigm “*logical-system independent*”, that is providing a general framework and conditions (axioms), and adapting and proving the classical definitions and results which underlie rewriting. Such an abstraction allowed us to unify and generalize many different rewriting theories. Another interest of such an abstraction is 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 also has to adapt these classical notions, and show that these fundamental results remain true for such formalisms. Up to now, this kind of approach to study some properties in the paradigm “*logical-system independent*”, has been widely applied to semantic aspects of algebraic formalisms [12,15,24] and to theorem deduction [13,23]. 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.

The present paper is then devoted to the next step: showing that there exists a valid and useful notion of rewriting logic in this abstract framework. Hence, the present work continues the development of the abstract framework of rewriting developed in [2,1].

In the abstract rewriting theory developed in [3,1], abstraction is twofold:

- (i) Rewritten objects are just elements of a set without any peculiar structure as to be inductively defined from a set of function names and variables.
- (ii) rewriting relations are specified by inference rules just defined as  $n$ -ary relations of formal systems. Hence, no property is supposed on them such as for instance transitivity.

The consequence of both above points is that the work presented here does not aim at generalizing the approach developed by Meseguer and many others, that is providing a logical support to a very powerful version of transition systems. Indeed, in this case, rewriting logic is based on a notion of relation which is not symmetric (because change is not in general reversible) but transitive. The present paper goes beyond by only generalizing the transformation that from the equational logic has resulted in the rewriting logic. Besides, we will show in Section 7 that the rewriting logic over membership equational logic [9] in an instance of our framework.

This paper is organized as follows: In Section 2, we recall standard notations about formal systems, theorem deduction and proof trees. In order to be as self-contained as possible, Section 3 and Section 4 summarize relevant definitions of [2,1]. In Section 5 is introduced the notion of rewriting logic at this abstract level. Section 6 proposes a model theoretic semantics for abstract rewriting logic. The theorems proving the soundness and completeness of the abstract rewriting logic with respect to this semantics are presented. Finally, Section 7 exemplifies the abstract framework.

To instantiate our definitions, concepts and results, we will present the conditional rewriting logic [21] as a running example and the rewriting logic parameterized by the conditional membership equational logic [9] in Section 7. Other examples such as constrained and timed rewriting logics [18,17] can be found in [4].

## 2 Preliminaries

A *formal system* (a so-called *calculus*)  $\mathcal{S} = (F, R)$  over an alphabet  $A$  consists of a set  $F$  of strings over  $A$ , called *formulae*, and a set  $R$  of  $n$ -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*  $\varphi_n$ . It is usually written  $\frac{\varphi_1 \dots \varphi_{n-1}}{\varphi_n}$ . If  $n = 1$ , the instance is called an *axiom* and is written  $\frac{}{\varphi_1}$ . A *deduction* in  $\mathcal{S}$  from a set of formulae  $\Gamma$  of  $F$  is a finite sequence  $(\psi_1, \dots, \psi_m)$  of formulae such that  $m \geq 1$  and, for all  $i = 1, \dots, m$ , either  $\psi_i$  is an element of  $\Gamma$  or there is an instance  $\frac{\varphi_1 \dots \varphi_{n-1}}{\varphi_n}$  of a rule in  $\mathcal{S}$  where  $\varphi_n = \psi_i$  and  $\{\varphi_1, \dots, \varphi_{n-1}\} \subseteq \{\psi_1, \dots, \psi_{i-1}\}$ . A *theorem* from a set of formulae  $\Gamma$  in  $\mathcal{S}$  is a formula  $\varphi$  such that there exists a deduction in  $\mathcal{S}$  from  $\Gamma$  with last element  $\varphi$ . The existence of such a deduction is usually denoted by the meta-statement  $\Gamma \vdash \varphi$ . Instances of rules can also be composed to build *proof trees*. Formally, a *proof tree*  $\pi$  in a formal system  $\mathcal{S}$  is a finite tree whose nodes are labelled with formulae of  $F$  in the following way: if a non-leaf node is labelled with  $\varphi_n$  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  $\mathcal{S}$ . Moreover, the leaves in  $\pi$  are either axioms or else rules with no premise and conclusion of which is an element of a given set of hypotheses  $\Gamma$ . We write  $\pi = (\pi_1, \dots, \pi_n, \varphi)_\iota$ , with  $n \in \mathbb{N}$ , the proof tree whose last inference rule is  $\iota = \frac{\varphi_1, \dots, \varphi_n}{\varphi}$  and such that, for every  $i \in \{1, \dots, n\}$ ,  $\pi_i$  is the subtree of  $\pi$  leading to  $\varphi_i$ .

### 3 Rewriting formal system

Here, we define an abstract framework of logics for which there exists a notion of rewrite system with an associated notion of rewriting logic (see the two next sections).

Rewriting is a method to reason with binary relations (equality [5,11], inclusion [19] or other non-symmetric relations [6,25], the ideal membership problem [10], etc.). These binary relations (the set  $E$  in Definition 3.1) are defined on sets of elements that are homogeneous but that can be different from one rewriting theory to another (simple words,  $\lambda$ -terms, first order terms, graphs, etc.). Moreover, the behavior of these binary relations is specified by inference rules. For example, in the equational rewriting 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) while others are removed of the process (e.g. symmetry). Moreover, preserved inference rules can be split up into two disjoint sets, called  $RS$  and  $De$ , specifying rewriting steps and derivations, respectively. Removed inference rules will be put in the set  $Rmv$ . Typically, rule instances of  $Rmv$  are removed because they generate basic loops in rewriting process, and then lead to obvious nonterminating rewrite relations. Finally, these binary predicates can be constrained by other n-ary predicates (the set

$P$  in Definition 3.1) such as for instance the definability predicate  $D$  in partial algebras or the membership predicate “:” in the membership equational logic. The inference rules defining the behavior of these extra predicates will be put in the set  $Oth$ . This leads to extend formal systems as follows:

**Definition 3.1** [Rewriting formal systems] A *rewriting formal system* (rfs) is a 7-tuple  $\mathcal{SP} = (T, E, P, RS, De, Rmv, Oth)$  such that  $T$  is a set,  $E$  and  $P$  are disjoint sets of binary and  $n$ -ary relations on  $T$ , 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:

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

**Remark 3.2** The couple  $\mathcal{S} = (F, RS \cup De \cup Rmv \cup Oth)$  is a formal system over the alphabet  $E \cup P \cup T \cup \{(;, ;)\}$ .

**Example 3.3** [Conditional equational logic] In this example, we define the logic which parameterizes the conditional rewriting logic associated to the conditional term rewriting modulo a set of equations. Before defining the rfs for this logic, let us recall some definitions and notations useful to this purpose. A signature  $\Sigma$  is a set of function names, each ones equipped with an arity in  $\mathbb{N}$ . Given a set of variables  $V$ , let us note  $T_\Sigma(V)$  the set of terms, free with generators in  $V$ . Given a term  $t \in T_\Sigma(V)$ ,  $Var(t)$  denotes the set of variables occurring in  $t$ .

Atoms are  $\Sigma$ -equations of the form  $t = t'$  where  $t$  and  $t'$  are terms in  $T_\Sigma(V)$ . Formulae are then sentences of the form  $\alpha_1 \wedge \dots \wedge \alpha_n \Rightarrow \alpha_{n+1}$  where for every  $1 \leq i \leq n + 1$ ,  $\alpha_i$  is a  $\Sigma$ -equation, and theories are any set of formulae. A substitution is a mapping  $\sigma : V \rightarrow T_\Sigma(V)$ . It is naturally extended to terms equations and conditional formulae.

In order to fit conditional formulae into the definition of rfs which only manipulates predicates, any formula of the form  $c \Rightarrow t = t'$  where  $c$  is a finite conjunction of equations, will be noted  $t =_c t'$ . Unconditioned equations  $t = t'$  will be noted  $t =_\emptyset t'$ . Hence, in the associated rfs, this gives rise to a family of predicates  $=_c$  indexed by finite conjunctions of equations, and inference rules will be  $n$ -ary relations on such formulae.

Therefore, given a signature  $\Sigma$  and a set of  $\Sigma$ -equations  $Eq$ , we define the rfs  $\mathcal{SP} = (T, E, P, RS, De, Rmv, Oth)$  for the conditional equational logic as follows: Let  $\Gamma$  be a set of formulae  $t =_c t'$

- $T = T_\Sigma(V)$ ,
- $E = \{=_c \mid c : \text{finite conjunction}\}$  is a set of equalities with for every  $c : \text{conjunction}$ ,  $=_c \stackrel{def}{=} T_\Sigma(V) \times T_\Sigma(V)$  (syntactical definition of equations <sup>2</sup>),

<sup>2</sup> Any couple of terms  $(t, t')$  is a well-formed equation. In any way, this does not mean that

- $P = \{\approx\}$  with  $\approx \stackrel{def}{=} T_\Sigma(V) \times T_\Sigma(V)$ ,
- $RS$  is the set defined by the following deduction rules:
  - (i) **Reflexivity** for each  $t \in T_\Sigma(V)$ ,

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

- (ii) **Replacement** for each  $t =_c t' \in \Gamma$  with  $c = \bigwedge_{1 \leq i \leq n} t_i = t'_i$  and every  $\sigma, \sigma' : V \rightarrow T_\Sigma(V)$ ,

$$\frac{\forall x \in \text{Var}(t) \cup \text{Var}(t'), \sigma(x) =_\emptyset \sigma'(x) \quad \forall 1 \leq i \leq n, \sigma(t_i) =_\emptyset \sigma'(t'_i)}{\sigma(t) =_\emptyset \sigma'(t')}$$

- (iii) **Congruence** for each  $t(x_1, \dots, x_n)$ ,

$$\frac{\forall 1 \leq i \leq n, t_i =_\emptyset t'_i}{t(t_1/x_1, \dots, t_n/x_n) =_\emptyset t(t'_1/x_1, \dots, t'_n/x_n)}$$

- (iv) **Equality**

$$\frac{t \approx u \quad u =_\emptyset v \quad v \approx t'}{t =_\emptyset t'}$$

- $De$  is the set defined by the following deduction rule:
  - (i) **Transitivity**

$$\frac{t =_\emptyset t' \quad t' =_\emptyset t''}{t =_\emptyset t''}$$

- $Rmv$  is the set defined by the following deduction rule:

$$\text{Symmetry} \quad \frac{t =_\emptyset t'}{t' =_\emptyset t}$$

- $Oth$  is the set defined by all the standard rules of equational reasoning applied on equations of the form  $t \approx t'$  at which we add the following deduction rule:

$$\text{Axiom} \quad \frac{t = t' \in Eq}{t \approx t'}$$

## 4 Abstract rewriting

In this section, we recapitulate how to define the notion of rewrite systems and derivations in rfs from [2,1]. In [2,1], we also gave a meaning, in the abstract framework of rfs, to the usual notions of effluences and proofs by rewriting

---

it is true.

(abstractions of peaks and valleys, respectively, usual in term rewriting), termination, Church-Rosser property, etc. From these notions, we then gave sufficient conditions to ensure the fundamental results which underlie rewriting used to generate canonical rewrite systems, such as Newman's lemma. Then, this has allowed us to define a generic completion method *à la Knuth-Bendix*. We refer the interested reader to our papers [2,1] for the complete presentation of these notions, results and extensions.

**Definition 4.1** [Rewrite systems] Let  $\mathcal{SP} = (T, E, P, RS, De, Rmv, Oth)$  be a rfs. A  $\mathcal{SP}$ -rewrite systems  $\mathcal{R}$  is an  $E$ -sorted set of binary relations  $(\rightarrow_p)_{p \in E}$  on  $T$  such that:  $\forall p \in E, \rightarrow_p \subseteq p$  (compatibility with the syntactic definition of  $p$  given in  $\mathcal{SP}$ ).

**Example 4.2** In the rfs developed in Example 3.3, we can consider the following set of rules from the signature  $\Sigma = (true^0, false^0, 0^0, eq^{?2}, - mod \_{}^2, gcd^2)$ , which specifies the greatest common divisor:

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

As another example, dealing with rewriting modulo a set of equations, we can consider the following rewrite system from the signature  $\Sigma = (\{0^0, 1^0, +^2, \times^2\}, \{x, y, z\})$  which defines Boolean rings:

$$\begin{aligned} Eq &= \left\{ \begin{array}{ll} x + y \approx y + x, & x \times y \approx y \times x, \\ (x + y) + z \approx x + (y + z), & (x \times y) \times z \approx x \times (y \times z) \end{array} \right\}, \\ \rightarrow_{=_{\emptyset}} &= \left\{ \begin{array}{ll} x + x \rightarrow 0, & x \times x \rightarrow x, \\ 0 + x \rightarrow x, & 0 \times x \rightarrow 0, \dots \\ x \times (y + z) \rightarrow (x \times y) + (x \times z), & 1 \times x \rightarrow x, \end{array} \right\}. \end{aligned}$$

(see [16] for the complete presentation of this rewrite system)

We could be tempted to define rewriting steps and derivations as the closure of each binary relation  $\rightarrow_p$  under  $RS$ 's and  $De$ 's rule instances, respectively, that is 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** of the logic that parameterizes conditional rewriting and given by: for each  $t =_c t'$ <sup>3</sup> with  $c = \bigwedge_{1 \leq i \leq n} t_i = t'_i$  and

every  $\sigma, \sigma' : V \rightarrow T_{\Sigma}(V)$ ,

<sup>3</sup>  $t =_c t'$  and  $t =_{\emptyset} t'$  denote, respectively, the conditional formula  $c \Rightarrow t = t'$  where  $c$  is a finite conjunction of equations, and the equation  $t = t'$ . This transformation is useful in order to fit conditional formulae into the definition of rfs.

$$\frac{\forall x \in \text{Var}(t) \cup \text{Var}(t'), \sigma(x) =_{\emptyset} \sigma'(x) \quad \forall 1 \leq i \leq n, \sigma(t_i) =_{\emptyset} \sigma'(t'_i)}{\sigma(t) =_{\emptyset} \sigma'(t')}$$

Indeed, when dealing with conditional rewriting rules, we have (at least) three potentially interesting definitions of  $\rightarrow_{\mathcal{R}}^{\emptyset}$ : given a rewrite system  $\mathcal{R} = (\rightarrow_{=c})_{c:\text{equationconjunction}}$ , then let us define  $\Theta = \{t =_c t' \mid t \rightarrow_{=c} t' \in \mathcal{R}\}$

- (i) *Natural conditional rewriting*  $\sigma(t) \rightarrow_{\mathcal{R}}^{\emptyset} \sigma'(t')$  if for every  $x \in \text{Var}(t) \cup \text{Var}(t')$ ,  $\sigma(x) \rightarrow_{\mathcal{R}}^{\emptyset} \sigma'(x)$  and for every  $1 \leq i \leq n$ ,  $\Theta \vdash \sigma(t_i) =_{\emptyset} \sigma'(t'_i)$ ,
- (ii) *Join conditional rewriting*  $\sigma(t) \rightarrow_{\mathcal{R}}^{\emptyset} \sigma'(t')$  if for every  $x \in \text{Var}(t) \cup \text{Var}(t')$ ,  $\sigma(x) \rightarrow_{\mathcal{R}}^{\emptyset} \sigma'(x)$  and for every  $1 \leq i \leq n$ ,  $\sigma(t_i) \downarrow_{=_{\emptyset}} \sigma(t'_i)$  where  $\downarrow_{=_{\emptyset}}$  means there is a term  $t''$  such that  $\sigma(t_i) \xrightarrow{*}_{=_{\emptyset}} t'' =_{\emptyset} \xleftarrow{*} \sigma(t'_i)$ , or
- (iii) *Normal conditional rewriting*  $\sigma(t) \rightarrow_{\mathcal{R}}^{\emptyset} \sigma'(t')$  if for every  $x \in \text{Var}(t) \cup \text{Var}(t')$ ,  $\sigma(x) \rightarrow_{\mathcal{R}}^{\emptyset} \sigma'(x)$  and for every  $1 \leq i \leq n$ ,  $\sigma(t_i) \xrightarrow{*}_{=_{\emptyset}} \sigma(t'_i)$

After seeing this example, it becomes obvious that some premises of rule instances in  $RS \cup De$  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*. 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 the next definition, we will only define in the abstract framework, normal rewriting. Both natural and join rewriting can also be abstractly defined. In order to simplify the presentation, we do not present here the abstract form of these notions which, however, can be found in our paper [1].

**Definition 4.3** [Rewriting step and rewriting relations] Let  $\mathcal{R}$  be a  $\mathcal{SP}$ -rewrite system. For every  $p \in E$ ,  $\rightarrow_{\mathcal{R}}^p$  and  $\xrightarrow{*}_{\mathcal{R}}^p$  are two binary relations on  $T$  defined as the least binary relations (according to the set-theoretical inclusion) inductively defined as follows:

- (i)  $\rightarrow_p \subseteq \rightarrow_{\mathcal{R}}^p$  and  $\rightarrow_{\mathcal{R}}^p \subseteq \xrightarrow{*}_{\mathcal{R}}^p$ , and
  - (ii) 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.  $u \xrightarrow{*}_{\mathcal{R}}^{p'} v$ ), and
    - *Normal rewriting*:
      - for every leaf  $p'(u', v') \in \mathcal{L}(\iota) \setminus \mathcal{FL}(\iota)$  with  $p' \in E$ ,  $u' \xrightarrow{*}_{\mathcal{R}}^{p'} v'$ , and
      - for every leaf  $p(t_1, \dots, t_n) \in \mathcal{L}(\iota) \setminus \mathcal{FL}(\iota)$  with  $p \in P$ ,  $\Theta \vdash p(t_1, \dots, t_n)$
- we have  $t \rightarrow_{\mathcal{R}}^p t'$  (resp.  $t \xrightarrow{*}_{\mathcal{R}}^p t'$ )

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

**Example 4.4** From Example 3.3, for any rule instance  $\iota \in RS \cup De$ ,  $\mathcal{FL}(\iota)$  contains all its premises of the form  $t =_{\emptyset} t'$  except if  $\iota$  is an instance of the rule **Replacement**. In this last case, we have  $\mathcal{FL}(\iota) = \{\sigma(x) =_{\emptyset} \sigma'(x) \mid x \in$



$Var(t) \cup Var(t')$ . Therefore, rewriting steps are then defined as follows:

- for every  $t \in T_\Sigma(V)$ ,  $t \xrightarrow{\overline{\mathcal{R}}^0} t$ ,
- $\xrightarrow{=0} \subseteq \xrightarrow{\overline{\mathcal{R}}^0}$ ,
- for every  $t \xrightarrow{\bigwedge_{i \leq n} u_i = v_i} t' \in \mathcal{R}$ , and every  $\sigma, \sigma' : V \rightarrow T_\Sigma(V)$ , if for every  $x \in Var(t) \cup Var(t')$ ,  $\sigma(x) \xrightarrow{\overline{\mathcal{R}}^0} \sigma'(x)$  and for every  $1 \leq i \leq n$ ,  $\sigma(t_i) \xrightarrow{* \overline{\mathcal{R}}^0} \sigma(t'_i)$  then  $\sigma(t) \xrightarrow{\overline{\mathcal{R}}^0} \sigma(t')$ ,
- for every  $t(x_1, \dots, x_n) \in T_\Sigma(V)$ , if for every  $1 \leq i \leq n$ ,  $t_i \xrightarrow{\overline{\mathcal{R}}^0} t'_i$  then  $t(t_1/x_1, \dots, t_n/x_n) \xrightarrow{\overline{\mathcal{R}}^0} t(t'_1/x_1, \dots, t'_n/x_n)$ , and
- if  $u \xrightarrow{\overline{\mathcal{R}}^0} v$  and there exists  $s, t \in T_\Sigma(V)$  such that  $Eq \vdash s \approx u$ , and  $Eq \vdash v \approx t$ , then  $s \xrightarrow{\overline{\mathcal{R}}^0} t$ .

Note both congruence and replacement rules have many premises. This allows to apply rules in parallel to all arguments of an operator (congruence), or in correspondence of all variables of a rule (replacement). Hence, a single rewrite step can apply various rules in parallel.

For every finite conjunction of equations  $c \neq \emptyset$ , the rewriting relation  $\xrightarrow{* \overline{\mathcal{R}}^c} = \xrightarrow{\overline{\mathcal{R}}^c} = \xrightarrow{=c}$ . Finally,  $\xrightarrow{* \overline{\mathcal{R}}^0}$  is the transitive closure of  $\xrightarrow{\overline{\mathcal{R}}^0}$ .

## 5 Generic form of rewriting logic

In rewriting logic, basic axioms are rewrite rules of the form  $t \rightarrow t'$  considered as sequents and inference rules are n-ary relations on these basic axioms. Inference rules are simply obtained by replacing in every deductive rule of  $De \cup RS$  formulae of the form  $p(t, t')$  by  $t \rightarrow_p t'$ , and by erasing rules in  $Rmv$ . Formally, we have

**Definition 5.1** Let  $\mathcal{SP} = (T, E, P, RS, De, Rmv, Oth)$  be a rfs. For any  $r \in RS \cup De$ , let us note  $\vec{r} = \{\vec{\iota} \mid \iota : p(t, t') \in r \wedge p \in E\}$  where for any  $\iota = \frac{\varphi_1 \dots \varphi_n}{p(t, t')} \in r$ ,  $\vec{\iota}$  is the instance  $\frac{\vec{\varphi}_1 \dots \vec{\varphi}_n}{t \rightarrow_p t'}$  where for all  $1 \leq i \leq n$ :

- $\vec{\varphi}_i = t_i \rightarrow_{p_i} t'_i$  if  $\varphi_i = p_i(t_i, t'_i)$  and  $p_i \in E$ , or
- $\vec{\varphi}_i = \varphi_i$ , otherwise.

**Definition 5.2** [Abstract rewriting logic (ARL)] Let  $\mathcal{SP} = (T, E, P, RS, De, Rmv, Oth)$  be a rfs. Let  $\mathcal{R} = (\rightarrow_p)_{p \in E}$  be a  $\mathcal{SP}$ -rewrite system. We say that  $\mathcal{R}$  entails a sequent  $t \rightarrow_p t'$  and write  $\mathcal{R} \vdash_{Ded} t \rightarrow_p t'$  if and only if  $t \rightarrow_p t'$  can be obtained by the following set  $Ded$  of deduction rules:

$$Ded = \{\vec{r} \mid r \in RS \cup De\} \cup Oth$$

$\mathcal{SP}$ -rewrite systems are then theories for the underlying abstract rewriting logic.

**Example 5.3** [The conditional rewriting logic] The conditional rewriting logic which formalizes the conditional term rewriting modulo a set of equations is defined as follows:

- sentences are sequents of the form  $t \rightarrow_{=c} t'$  where  $c$  is a finite (possibly empty) conjunction of equations,
- a rewriting theory  $\mathcal{R}$  is a set of sequents, and
- a rewriting theory  $\mathcal{R}$  entails the sequent  $t \rightarrow_{=c} t'$  if it is obtained by the finite application of the following deduction rules:
  - (i) **Replacement** for each  $t \rightarrow_c t' \in \mathcal{R}$  with  $c = \bigwedge_{1 \leq i \leq n} t_i = t_i$  and every  $\sigma, \sigma' : V \rightarrow T_\Sigma(V)$ ,

$$\frac{\forall x \in \text{Var}(t) \cup \text{Var}(t'), \sigma(x) \rightarrow_{=0} \sigma'(x) \quad \forall 1 \leq i \leq n, \sigma(t_i) \rightarrow_{=0} \sigma'(t_i)}{\sigma(t) \rightarrow_{=0} \sigma'(t')}$$

- (ii) **Congruence** for each  $t(x_1, \dots, x_n)$ ,

$$\frac{\forall 1 \leq i \leq n, t_i \rightarrow_{=0} t'_i}{t(t_1/x_1, \dots, t_n/x_n) \rightarrow_{=0} f(t'_1/x_1, \dots, t'_n/x_n)}$$

- (iii) **Equality**

$$\frac{t \approx u \quad u \rightarrow_{=0} v \quad v \approx t'}{t \rightarrow_{=0} t'}$$

- (iv) **Reflexivity** for each  $t \in T_\Sigma(V)$ ,

$$\overline{t \rightarrow_{=0} t}$$

- (v) **Transitivity**

$$\frac{t \rightarrow_{=0} t' \quad t' \rightarrow_{=0} t''}{t \rightarrow_{=0} t''}$$

- (vi) all rule instances in *Oth* given in Example 3.3.

## 6 Semantics

In this section, we will answer the following question: *what are the models of abstract rewriting logic?*

To achieve this purpose, we follow the approach initiated in [9] to define a model-theoretical presentation of rewrite theories in terms of the models of a suitable theory of the first-order logic. As this was observed in [9], two kinds of models can be defined:

- (i) Reachability models which focus just on what elements of  $T$  can be reached from a certain element  $t$  via sequences of rewriting, ignoring how the rewrites can lead to them.
- (ii) Provability models which focus, unlike reachability models, both on what elements of  $T$  can be reached from a certain element  $t$  via sequences of rewriting and how the rewrites can lead to them. In [9], such models are called concurrent models because, as in [21], they are defined from both congruence and replacement rules which have many premises. Therefore, this allows to apply rewrite rules in parallel to all arguments of an operator (congruence) or in correspondence of variables of a rewrite rule (replacement). Congruence and replacement rules are strongly dependent on inductive structure of terms. In rfs, elements of  $T$  are simple objects without any inductive structure. Consequently, at this abstract level, concurrent models do not make sense anymore.

### 6.1 Reachability models

**Definition 6.1** [Reachability relation] Let  $\mathcal{R}$  be a  $\mathcal{SP}$ -rewrite system. Let us define  $\rightarrow_{\mathcal{R}}$  the  $E$ -sorted set of binary relations on  $T$  as follows:

$$\forall p \in E, t \rightarrow_{\mathcal{R}}^p t' \iff \mathcal{R} \vdash t \rightarrow_p t'$$

**Remark 6.2** Although the notation is the same, the reachability relation has not to be confused with the rewrite relation  $\rightarrow_{\mathcal{R}}$  which has been defined in Definition 4.3. However, if rewriting coincides with derivability in ARL [4], both above binary relations denote the same subset of  $T \times T$ .

The mono-sorted first-order predicate logic is sufficient to define a model-theoretical presentation of the reachability relation associated to a  $\mathcal{SP}$ -rewrite system  $\mathcal{R}$ .

**Definition 6.3** [The theory  $Reach(\mathcal{R})$ ] Let  $\mathcal{SP} = (T, E, P, RS, De, Rmv, Oth)$  be a rfs. Let  $\mathcal{R}$  be a  $\mathcal{SP}$ -rewrite system. The first order theory  $Reach(\mathcal{R})$  contains the signature  $\Sigma_{\mathcal{R}} = (\mathcal{F}, \mathcal{C}, \mathcal{P})$ <sup>4</sup> and the set  $Ax$  of sentences defined respectively, as follows:

- Signature:
  - $\mathcal{F} = \emptyset$ ,  $\mathcal{C} = T$ , and
  - $\mathcal{P} = \{\rightarrow_p^2 \mid p \in E\} \cup P$
- Sentences:
  - $\forall p \in E, \forall t, t' \in T, t \rightarrow_p t' \in \mathcal{R} \implies t \rightarrow_p t' \in Ax$ , and
  - $\forall \iota = \frac{\varphi_1 \dots \varphi_n}{\varphi} \in \mathcal{Ded}, \bigwedge_{1 \leq i \leq n} \varphi_i \implies \varphi \in Ax$ .

Definition 6.3 call for some comments:

<sup>4</sup>  $\mathcal{F}$ ,  $\mathcal{C}$  and  $\mathcal{P}$  are respectively the set of function, constant and predicate names.

- The above theory  $Reach(\mathcal{R})$  contains many (usually an infinite number of) sentences in  $Ax$ . The reason is that we associate a sentence to each rule instance. Consequently, all sentences in  $Ax$  are ground, that is, all terms which occur in sentences are elements of  $T$ . But, as  $T$  is not equipped with any inductive structure from a set of operators, elements in  $T$  are simple constants, and then the set of operators with arity greater than 1 is empty.
- For logics which parameterize existing rewriting logics, a shorter description can be given. Indeed, as this is usual in most logics (anyway all logics used in computing science and mathematics) the underlying inference relation  $\vdash$  is generated from a finite set of deductive rules, that is a single form with infinitely many instantiations. This allows to denote all the instances by a set of generic forms (up to meta-variable renaming). In this case, generic terms which occur in such deductive rules can be replaced by variables in the sentences of  $Ax$ .

**Example 6.4** As explained in the above comments, we are going to benefit from the fact that inference rules given in Example 5.3 are deductive rules and terms are inductively defined from a set of operator names, to give a shorter description of the theory  $Reach(\mathcal{R})$  than the one given in Definition 6.3. Therefore, this gives rise to the following description: let  $\mathcal{R}$  be a rewriting theory over a signature  $\Sigma$ ,

- The signature  $\Sigma_{\mathcal{R}} = (\mathcal{F}, \mathcal{C}, \mathcal{P})$  is defined by:
  - $\mathcal{F} = \{f^n \mid f \in \Sigma, n \geq 1\}$ ,  $\mathcal{C} = \{f^0 \mid f \in \Sigma\}$ , and
  - $\mathcal{P} = \{\rightarrow_{=c}^2, \leftarrow_{=c}^2 \mid c \text{ finite conjunction}\} \cup \{\approx^2\}$
- sentences in  $Ax$  are: to indicate that a term  $t$  has its variables among  $\{x_1, \dots, x_n\}$ , we write  $t(x_1, \dots, x_n)$ , and then  $t(t_1, \dots, t_n)$  is the term obtained from  $t$  by replacing all variable occurrences  $x_i$  by  $t_i$ 
  - sentences in  $\mathcal{R}$  and  $Eq$ ,
  - for every  $t \rightarrow_{=} t' \in \mathcal{R}$ ,
 
$$\bigwedge_{i \leq n} t_i = t'_i$$

$$\bigwedge_{j \leq m} y_j \rightarrow_{=0} y'_j \wedge \bigwedge_{i \leq n} t_i(y_1, \dots, y_m) \rightarrow_{=0} t'_i(y_1, \dots, y_m)$$

$$\Rightarrow t(y_1, \dots, y_m) \rightarrow_{=0} t'(y'_1, \dots, y'_m)$$
- for every  $f^n \in \Sigma$ ,
 
$$\bigwedge_{i \leq n} x_i \rightarrow_{=0} x'_i \Rightarrow f(x_1, \dots, x_n) \rightarrow_{=0} f(x'_1, \dots, x'_n)$$
  - $x \approx y \wedge y \rightarrow_{=0} z \wedge z \approx w \Rightarrow x \rightarrow_{=0} w$
  - $x \rightarrow_{=0} x$
  - $x \rightarrow_{=0} y \wedge y \rightarrow_{=0} z \Rightarrow x \rightarrow_{=0} z$
  - Usual equality axioms for the predicate  $\approx$

**Definition 6.5** [Reachability models] Let  $\mathcal{R}$  be a  $\mathcal{SP}$ -rewrite system. A *reachability model* of  $\mathcal{R}$  is any first-order structure of  $Reach(\mathcal{R})$ .

**Example 6.6** From the theory  $Reach(\mathcal{R})$  developed in Example 6.4, a model  $\mathcal{M}$  is a set  $U$  together for any finite conjunction  $c$  with a binary relation  $\rightarrow_{=c}^{\mathcal{M}}$ . For the empty conjunction,  $\rightarrow_{=\emptyset}^{\mathcal{M}}$  is reflexive and transitive. Therefore, the carrier  $U$  of  $\mathcal{M}$  can be naturally regarded as a category. This is how the semantics of the standard rewriting logic has been defined in [21]. In this case, all syntactical notions can be interpreted in the language of category theory. Indeed, from the congruence rule, it is obvious to show that the semantics of operator names and then terms with variables, are functors. Therefore, rewritings become natural transformations between functors. Actually, the semantics of rewrite rules in the language of category theory is more complicated because of conditions. Indeed, it is obvious to show from the replacement rule, that the semantics of unconditional rewrite rules of the form  $t \rightarrow_{=\emptyset} t'$  is a natural transformation  $\gamma : t^{\mathcal{M}} \Rightarrow t'^{\mathcal{M}}$ .

When conditions occur, rewrite rules define natural transformations between functors resulting of the composition of each functor associated to each term occurring in the conclusion and the subequalizer functor used to solve conditions <sup>5</sup>.

**Definition 6.7** [Herbrand's model] Let  $\mathcal{R}$  be a  $\mathcal{SP}$ -rewrite system. Let  $\mathcal{I}$  be the first order structure over  $\Sigma_{\mathcal{R}}$  defined as follows:

- the carrier  $I$  is  $T$ ,
- for every  $t \in \mathcal{C}$ ,  $t^{\mathcal{I}} = t$ ,
- for every  $p \in E$ ,  $\rightarrow_p^{\mathcal{I}} = \rightarrow_{\mathcal{R}}$ , and
- for every  $p \in P$ ,  $(t_1, \dots, t_n) \in p^{\mathcal{I}} \Leftrightarrow \mathcal{R} \vdash p(t_1, \dots, t_n)$ .

**Theorem 6.8 (Completeness)** For  $\mathcal{R}$  a  $\mathcal{SP}$ -rewrite theory,

$$Reach(\mathcal{R}) \models t \rightarrow_p t' \iff \mathcal{R} \vdash t \rightarrow_p t'$$

**Proof (Sketch)**  $Reach(\mathcal{R})$  is a universal Horn theory. Therefore,  $\mathcal{I}$  is initial in the category of first-order structures that satisfy sentences in  $Reach(\mathcal{R})$  [20]. By Definition 6.7, we obviously have:

$$\mathcal{I} \models \varphi \iff \mathcal{R} \vdash \varphi$$

Consequently, we can write:

$$\begin{aligned} Reach(\mathcal{R}) \models t \rightarrow_p t' &\iff \mathcal{I} \models t \rightarrow_p t' \quad \mathcal{I} \text{ is initial} \\ &\iff \mathcal{R} \vdash t \rightarrow_p t' \end{aligned}$$

□

<sup>5</sup> It is well-known that solutions of substitutions are equalizer between both morphisms associated to terms of equations [14]. Here, terms are semantically denoted by functors. Therefore, subequalizer is the generalization of the notion of equalizer of two functors. We refer the reader to [21] for the complete exposition of this notion.

Actually, we have a more general completeness result:

**Theorem 6.9** *For  $\mathcal{R}$  a  $\mathcal{SP}$ -rewrite theory,*

$$\text{Reach}(\mathcal{R}) \models \varphi \iff \mathcal{R} \vdash \varphi$$

$\varphi$  is over  $\text{Reach}(\mathcal{R})$ , that is, is either of the form  $t \rightarrow_p t'$  or of the form  $p(t_1, \dots, t_n)$  with  $p \in P$ .

**Proof** The proof is similar to the proof of Theorem 6.8.  $\square$

## 6.2 Provability models

As usual, the idea is to attach a proof term to each sequent, so-called decorated sequents. In the standard rewriting logic (this is also true for its extension developed in [9]) proof terms are built from variables, operators in signatures (congruence), labels of rewrite rules in  $\mathcal{R}$  (replacement), and “;” to compose rewritings (transitivity). Here, inference rules (proofs) cannot be implicitly taken into account (built) from operators of signatures, variables and other primitive symbols such as “;”. The reason is no information is given on both the structure of elements in  $T$  and the form of inference rules. Therefore, a symbol operator  $f_\iota : s_{\varphi_1} \times \dots \times s_{\varphi_n} \rightarrow s_{\varphi_{n+1}}$  has to be associated to any inference rule  $\iota = \frac{\varphi_1 \dots \varphi_n}{\varphi_{n+1}} \in \text{Ded}$  where  $s_{\varphi_i}$   $1 \leq i \leq n+1$  is a sort name which semantically contains every proof tree  $\pi : \varphi$ . For rewriting rules in a  $\mathcal{SP}$ -rewrite system  $\mathcal{R}$ , we will index rewriting rules by labels. Therefore, this leads to extend  $\mathcal{SP}$ -rewrite systems as follows:

**Definition 6.10** [Labelled rewrite system] Let  $L$  be a set. A labelled  $\mathcal{SP}$ -rewrite system  $\mathcal{R}$  is an  $E$ -sorted set of ternary relations  $(\rightarrow_p)_{p \in E}$  on  $L \times T \times T$ . For every  $(l, t, t')$  in  $\rightarrow_p$ , we will use the notation  $l : t \rightarrow_p t'$ .

This naturally leads to specify provability models in the many-sorted first order predicate logic:

**Definition 6.11** [The theory  $\text{Proof}(\mathcal{R})$ ] Let  $\mathcal{SP} = (T, E, P, RS, De, Rmv, Oth)$  be a rfs. Let us note  $\mathcal{S}$  the underlying formal system associated to  $\mathcal{SP}$  (see Remark 3.2). Let  $\mathcal{R}$  be a labelled  $\mathcal{SP}$ -rewrite system. The first order theory  $\text{Proof}(\mathcal{R})$  contains the signature  $\Sigma_{\mathcal{R}} = (S, \mathcal{F}, \mathcal{P})$  and the set  $Ax$  of sentences defined respectively, as follows:

- Signature:
  - $S = \{s_\varphi \mid \varphi \in \mathcal{S}\},$
  - $\mathcal{F} = \left[ \begin{array}{c} \{f_\iota : s_{\varphi_1} \times \dots \times s_{\varphi_n} \rightarrow s_{\varphi_{n+1}} \mid \iota = \frac{\varphi_1 \dots \varphi_n}{\varphi_{n+1}} \in \text{Ded}\} \\ \cup \\ \{l : \rightarrow s_{t \rightarrow_p t'} \mid l : t \rightarrow_p t' \in \mathcal{R}\} \end{array} \right],$
  - $\mathcal{P} = \{Pr_\varphi : s_\varphi \mid \varphi \in \mathcal{S}\}$
- Sentences:

- $\forall l : \rightarrow s_{t \rightarrow_p t'} \in \mathcal{F}$ ,  $Pr_{t \rightarrow_p t'}(l)$ , and
  - $\forall \iota = \frac{\varphi_1 \dots \varphi_n}{\varphi} \in \mathcal{Ded}$ ,  $\bigwedge_{1 \leq i \leq n} Pr_{\varphi_i}(x_{\varphi_i}) \implies Pr_{\varphi}(f_{\iota}(x_{\varphi_1}, \dots, x_{\varphi_n})) \in Ax$ .
- where  $x_{\varphi_i}$  is a variable of sort  $s_{\varphi_i}$ .

$Proof(\mathcal{R})$  is complete with respect to inference rules of ARL as expressed by the following result:

**Theorem 6.12 (Completeness I)** *For any rewrite theory  $\mathcal{R}$ , we have:*

$$\mathcal{R} \vdash t \rightarrow_p t' \iff \exists \pi \in T_{\Sigma_{\mathcal{R}}}(X)_{s_{t \rightarrow_p t'}}, Proof(\mathcal{R}) \models Pr_{t \rightarrow_p t'}(\pi)$$

$X$  is any set of variables which contains the subset  $\{x_{\varphi} \mid \varphi \in \mathcal{S}\}$ .

**Proof** The ‘‘Only if’’ part is obvious. The ‘‘If’’ part is proven by mathematical induction on the structure of proof trees.

- *basic case* Both cases have to be considered:
  - (i)  $l : t \rightarrow_p t' \in \mathcal{R}$ . In this case,  $Pr_{t \rightarrow_p t'}(l) \in Proof(\mathcal{R})$ .
  - (ii) there is a rule  $\iota \xrightarrow{t \rightarrow_p t'} \in \mathcal{Ded}$ . In this case,  $Pr_{t \rightarrow_p t'}(f_{\iota}) \in Proof(\mathcal{R})$ .
- *general case* there is a proof tree  $\pi = (\pi_1 : \varphi_1, \dots, \pi_n : \varphi_n, t \rightarrow_p t')_{\iota}$ . By induction hypothesis, for every  $1 \leq i \leq n$ , there exists  $\pi'_i \in T_{\Sigma_{\mathcal{R}}}(X)_{s_{\varphi_i}}$  such that  $Proof(\mathcal{R}) \vdash Pr_{\varphi_i}(\pi'_i)$ . Therefore, by assuming that we use the Hilbert calculus for the first-order logic, by instantiation and modus-ponens, we have  $Proof(\mathcal{R}) \vdash Pr_{t \rightarrow_p t'}(f_{\iota}(\pi'_1, \dots, \pi'_n))$ .

As the Hilbert calculus for the first-order logic is complete, we have then  $Proof(\mathcal{R}) \models Pr_{t \rightarrow_p t'}(f_{\iota}(\pi'_1, \dots, \pi'_n))$ .  $\square$

From the definition of  $Proof(\mathcal{R})$  the above completeness result holds for any formula  $\varphi$  of the underlying formal system  $\mathcal{S}$ , that is:

**Theorem 6.13 (Completeness)** *For any rewrite theory  $\mathcal{R}$ , we have:*

$$\mathcal{R} \vdash \varphi \iff \exists \pi \in T_{\Sigma_{\mathcal{R}}}(X)_{s_{\varphi}}, Proof(\mathcal{R}) \models Pr_{\varphi}(\pi)$$

$X$  is any set of variables which contains the subset  $\{x_{\varphi} \mid \varphi \in \mathcal{S}\}$ .

**Proof** The proof is similar to the one given to prove Theorem 6.12.  $\square$

## 7 An instance of our general approach

The rewriting logic defined in this section is parameterized by a generalization of the conditional Membership Equational Logic (MEL), called *MEL with frozen operators* [9].

The conditional membership equational logic (MEL) belongs to the family of algebraic specification formalisms that have been defined to extend basic algebraic specifications in order to support subsorts and partially of function

symbols. Before presenting the rfs for conditional membership equational logic with frozen operators, let us recall the basic notions and notations of this logic.

A MEL signature with frozen operators (called generalized MEL signature in [9]) is a triple  $(K, \Sigma, S)$  (just  $\Sigma$  in the following) where:

- $K$  is a set of *kinds*,
- $\Sigma = (K, F)$  is a standard many-kinded signatures where each function name  $f : k_1 \times \dots \times k_n \rightarrow k$  is together with a set  $\Phi(f) \subseteq \{1, \dots, n\}$  of *frozen arguments positions*, and
- $S$  is a  $K$ -indexed family of sets  $S_k$  (so called  $K$ -set).

Given a  $K$ -set  $V$  of variables, for every  $k \in K$ ,  $T_\Sigma(V)_k$  is the standard set of terms of kind  $k$ , free with generating in  $V$ , and  $T_\Sigma(V)$  is the  $K$ -indexed family  $(T_\Sigma(V)_k)_{k \in K}$ . Let us define  $\Phi$  and  $\nu$  the two binary relations on  $T_\Sigma(V)$  as follows:

$$\Phi(t, t') \iff \exists p \in \mathbb{N}, \exists 1 \leq i \leq p, \exists \alpha = \alpha_1.i.\alpha_2 \in Pos(t), \left\{ \begin{array}{l} t' = t_{|\alpha} \\ \wedge \\ t_{|\alpha_1} = f(t_1, \dots, t_p) \\ \wedge \\ i \in \Phi(f) \end{array} \right.$$

$$\nu(t, t') \iff \exists \alpha \in Pos(t), t' = t_{|\alpha} \wedge \neg \Phi(t, t')$$

Let us define  $\Phi(t) = \{x \mid \Phi(t, x)\}$  and  $\nu(t) = \{x \mid \nu(t, x)\}$ .

Atoms are either equations  $t = t'$  where  $t$  and  $t'$  are terms of the same kind, or membership formula  $t : s$  where  $t$  is a term of kind  $k$  and  $s \in S_k$ . In [9], conditions of rewrite rules are increased to allow equations, memberships and rewritings. This leads naturally to consider in the underlying rfs, three kinds of  $K$ -indexed family of equality predicates:

- $\approx_k$  to make rewritings modulo a set of equations  $Eq$ ,
- $\equiv_k$  to increase conditions in order to allow equations, and
- $=_k$  to denote equations which will be transformed into rewritings.

Conditional formulae are then any sentence  $\alpha_1 \wedge \dots \wedge \alpha_n \Rightarrow \alpha$  where each  $\alpha_i$  ( $1 \leq i \leq n$ ) is either of the form  $t_i =_k t'_i$ , or  $t_i \equiv_k t'_i$  or  $t_i :_k s_i$ , and  $\alpha$  is of the form  $t =_k t'$ . A substitution is a  $K$ -indexed family of application  $\sigma_k : V_k \rightarrow T_\Sigma(V)_k$ . It is naturally extended to terms and formulae.

Given a MEL signature  $\Sigma$  and a set of equations  $Eq$ , we define the rfs  $\mathcal{SP}$  by the tuple  $(T, E, RS, De, Rmv, Oth)$  such that: Let  $\Gamma$  be a theory in MEL with frozen operators



- $T = T_\Sigma(V) \cup (\bigcup_{k \in K} S_k)$ ,
- $E = \{=_{k,c} \mid k \in K, c : \text{finite conjunction}\}$  s.t.  $=_{k,c} \stackrel{def}{=} T_\Sigma(V)_k \times T_\Sigma(V)_k$  (syntactic definition of equations),
- $P = \{:_k, \equiv_k, \approx_k \mid k \in K\}$  s.t.  $:_k \stackrel{def}{=} T_\Sigma(V)_k \times S_k$  (syntactic definition of memberships), and  $\equiv_k, \approx_k \stackrel{def}{=} T_\Sigma(V)_k \times T_\Sigma(V)_k$ ,
- $RS$  is the set defined by the following deduction rules:
  - (i) **Reflexivity** for each  $k \in K$  and each  $t \in T_\Sigma(V)_k$ ,

$$\overline{t =_{k,\emptyset} t}$$

- (ii) **Replacement** for each  $t =_{k,c} t'$  with  $c = \bigwedge_{i \in I} t_i \equiv_{k_i} t'_i \wedge \bigwedge_{j \in J} :_{k_j} s_j \wedge \bigwedge_{l \in L} t_l =_{k_l, \emptyset} t'_l$  and all substitutions  $\sigma, \sigma'$ ,

$$\frac{\begin{array}{l} \forall i \in I, \sigma(t_i) \equiv_{k_i} \sigma(t'_i) \quad \forall j \in J, \sigma(t_j) :_{k_j} s_j \quad \forall l \in L, \sigma(t_l) =_{k_l, \emptyset} \sigma(t'_l) \\ \forall x \in \Phi(t) \cup \Phi(t'), \sigma(x) \equiv \sigma'(x) \quad \forall x \in \nu(t) \cap \nu(t'), \sigma(x) =_{\emptyset} \sigma'(x) \end{array}}{\sigma(t) =_{k,\emptyset} \sigma'(t')}$$

- (iii) **Congruence** for each  $t(x_1, \dots, x_n)$  with  $x_i \in V_{k_i}$ , if we note  $I \subseteq \{1, \dots, n\}$  and  $J = \{1, \dots, n\} \setminus I$  such that  $\Phi(t) = \{x_i \mid i \in I\}$  and  $\nu(t) = \{x_j \mid j \in J\}$ , then

$$\frac{\forall i \in I, t_i \equiv_{k_i} t'_i \quad \forall j \in J, t_j =_{k_j, \emptyset} t'_j}{t(t_1/x_1, \dots, t_n/x_n) =_{k,\emptyset} t(t'_1/x_1, \dots, t'_n/x_n)}$$

- (iv) **Equality1**

$$\frac{t \approx_k u \quad u =_{k,\emptyset} v \quad v \approx_k t'}{t =_{k,\emptyset} t'}$$

- $De$  is the set defined by the following deduction rule:

$$\text{Transitivity} \frac{t =_{k,\emptyset} t' \quad t' =_{k,\emptyset} t''}{t =_{k,\emptyset} t''}$$

- $Rmv$  is the set defined by the following deduction rule:

$$\text{Symmetry} \frac{t =_{k,\emptyset} t'}{t' =_{k,\emptyset} t}$$

- $Oth$  is the set defined by all the standard rules of equational reasoning for each of the predicates  $\equiv_k$  and  $\approx_k$  at which we add the two following deduction rules:

(i)

$$\mathbf{Axiom} \frac{t = t' \in Eq, t, t' \in T_\Sigma(V)_k}{t \approx_k t'}$$

(ii)

$$\mathbf{Equality2} \frac{t \approx_k u \quad u \equiv_k v \quad v \approx_k t'}{t \equiv_k t'}$$

For any rule instance  $\iota \in RS \cup De$ ,  $\mathcal{FL}(\iota)$  contains all its premises of the form  $t =_{k,\emptyset} t'$  except if  $\iota$  is an instance of the rule **Replacement**. In this last case,  $\mathcal{FL}(\iota) = \{\sigma(x) =_{\emptyset} \sigma'(x) \mid x \in \nu(t) \cap \nu(t')\}$ . Therefore, if we note  $Cnj$  the set of all finite conjunctions of atoms, then a rewrite system  $\mathcal{R}$  is a  $K \times Cnj$ -indexed set of binary relations  $\rightarrow_{=_{k,c}} \subseteq T_\Sigma(V)_k \times T_\Sigma(V)_k$ <sup>6</sup>. Rewriting steps are then defined as follows:

- for every  $t \in T_\Sigma(V)_k$ ,  $(t, t) \in \rightarrow_{\mathcal{R}}^{\overline{=}_{k,\emptyset}}$ ,
- $\rightarrow_{=_{k,\emptyset}} \subseteq \rightarrow_{\mathcal{R}}^{\overline{=}_{k,\emptyset}}$ ,
- for every  $t \rightarrow_{=_{k,c}} t' \in \mathcal{R}$  with  $c = \bigwedge_{i \in I} t_i \equiv_{k_i} t'_i \wedge \bigwedge_{j \in J} t_j :_{k_j} s_j \wedge \bigwedge_{l \in L} t_l =_{k_l, \emptyset} t'_l$  and all substitutions  $\sigma, \sigma'$ , if for every  $x \in \nu(t) \cap \nu(t')$ ,  $\sigma(x) \rightarrow_{\mathcal{R}}^{\overline{=}_{\emptyset}} \sigma'(x)$  and:
  - $\forall i \in I, \Theta \vdash \sigma(t_i) \equiv_{k_i} \sigma'(t'_i)$ ,
  - $\forall j \in J, \Theta \vdash \sigma(t_j) :_{k_j} s_j$ ,
  - $\forall x \in \Phi(t) \cup \Phi(t'), \Theta \vdash \sigma(x) \equiv \sigma'(x)$ , and
  - *Normal rewriting*  $\forall l \in L, \sigma(t_l) \xrightarrow{*}_{\mathcal{R}}^{\overline{=}_{k_l, \emptyset}} \sigma'(t'_l)$
 then  $\sigma(t) \rightarrow_{\mathcal{R}}^{\overline{=}_{k,\emptyset}} \sigma(t')$ ,
- for every  $f : k_1 \times \dots \times k_n \rightarrow k \in \Sigma$ , if for every  $1 \leq i \leq n$ ,  $t_i \rightarrow_{\mathcal{R}}^{\overline{=}_{k_i, \emptyset}} t'_i$  then  $f(t_1, \dots, t_n) \rightarrow_{\mathcal{R}}^{\overline{=}_{k, \emptyset}} f(t'_1, \dots, t'_n)$ , and
- if  $u \rightarrow_{\mathcal{R}}^{\overline{=}_{k,\emptyset}} v$  and there exists  $s, t \in T_\Sigma(V)_k$  such that  $Eq \vdash s \approx_k u$ , and  $Eq \vdash v \approx_k t$ , then  $s \rightarrow_{\mathcal{R}}^{\overline{=}_{k,\emptyset}} t$ .

The associated rewriting logic is then defined by the following inference rules: let  $\mathcal{R}$  be a set of sequents of the form  $t \rightarrow_{=_{k,c}} t'$

(i) **Reflexivity** for each  $k \in K$  and each  $t \in T_\Sigma(V)_k$ ,

$$\overline{t \rightarrow_{=_{k,\emptyset}} t}$$

(ii) **Replacement** for each  $t \rightarrow_{=_{k,c}} t' \in \mathcal{R}$  with  $c = \bigwedge_{i \in I} t_i \equiv_{k_i} t'_i \wedge \bigwedge_{j \in J} t_j :_{k_j} s_j \wedge$ 

$$\bigwedge_{l \in L} t_l =_{k_l, \emptyset} t'_l \text{ and all substitutions } \sigma, \sigma',$$

<sup>6</sup> Here,  $\leftarrow_{=_{k,c}}$  is not needed because  $=_{k,c}$  is symmetric.

$$\frac{\begin{array}{l} \forall i \in I, \sigma(t_i) \equiv_{k_i} \sigma(t'_i) \quad \forall j \in J, \sigma(t_j) :_{k_j} s_j \quad \forall l \in L, \sigma(t_l) \rightarrow_{=_{k_l,0}} \sigma(t'_l) \\ \forall x \in \Phi(t) \cup \Phi(t'), \sigma(x) \equiv \sigma'(x) \quad \forall x \in \nu(t) \cap \nu(t'), \sigma(x) \rightarrow_{=0} \sigma'(x) \end{array}}{\sigma(t) \rightarrow_{=_{k,0}} \sigma'(t')}$$

- (iii) **Congruence** for each  $t(x_1, \dots, x_n)$  with  $x_i \in V_{k_i}$ , if we note  $I \subseteq \{1, \dots, n\}$  and  $J = \{1, \dots, n\} \setminus I$  such that  $\Phi(t) = \{x_i \mid i \in I\}$  and  $\nu(t) = \{x_j \mid j \in J\}$ , then

$$\frac{\forall i \in I, t_i \equiv_{k_i} t'_i \quad \forall j \in J, t_j \rightarrow_{=_{k_j,0}} t'_j}{t(t_1/x_1, \dots, t_n/x_n) \rightarrow_{=_{k,0}} t(t'_1/x_1, \dots, t'_n/x_n)}$$

- (iv) **Equality1**

$$\frac{t \approx_k u \quad u \rightarrow_{=_{k,0}} v \quad v \approx_k t'}{t \rightarrow_{=_{k,0}} t'}$$

- (v) **Transitivity**

$$\frac{t \rightarrow_{=_{k,0}} t' \quad t' \rightarrow_{=_{k,0}} t''}{t \rightarrow_{=_{k,0}} t''}$$

In [9], the membership equational logic has been used to specify both theories  $Reach(\mathcal{R})$  and  $Proof(\mathcal{R})$ . As the membership equational logic does not deal with predicates except equality and membership, to specify  $Reach(\mathcal{R})$  in [9], it has been added for any kind  $k \in K$  of the MEL signature  $\Sigma$  which underlies the rfs, a new kind  $Pair_k$  with three sorts  $Ar_k^0$ ,  $Ar_k^1$ , and  $Ar_k$ , and two operators  $\rightarrow: k \times k \rightarrow Pair_k$  and  $;;: Pair_k \times Pair_k \rightarrow Pair_k$ . The kind  $Pair_k$  contains all rewritings, and  $Ar_k^0$ ,  $Ar_k^1$  and  $Ar_k$  denote respectively, idle rewrites, one-step rewrites and rewrites of arbitrary length. Finally,  $\rightarrow$  and  $;;$  denote respectively, rewritings and composition of rewritings. For lack of space, we cannot present both theories but we refer the reader to [9] for the complete presentation of both the reachability and provability theories. If we note  $MELReach(\mathcal{R})$  and  $Reach(\mathcal{R})$  the MEL reach theory as defined in [9] and the first order theory as defined in this paper, respectively, we can show that:

$$MELReach(\mathcal{R}) \models t \rightarrow t' : Ar_k \iff Reach(\mathcal{R}) \models t \rightarrow_{=_{k,0}} t'$$

Similar results are obtained with the theory  $Proof(\mathcal{R})$  as specified in this paper and the one developed in [9].

## 8 Conclusion

In this paper, we have shown the existence of a notion of rewriting logic for any rewriting theory satisfying the conditions of a general framework of rewriting.

This has given rise to an abstract form of rewriting logic for which we have studied the model theoretical semantics, and given an initiality theorem and two theorems proving respectively the soundness and completeness of the abstract rewriting logic with respect to this semantics.

In order to validate our approach, we are continuing to check “by hand” that we can indeed cover other already known extensions of the rewriting logic such as rewriting logic with probabilities [7].

## References

- [1] Aiguier, M. and D. Bahrami, *Structures for abstract rewriting*, Technical report, La.M.I, Université d’Évry val-d’Essonne (2003), available at [www.lami.univ-evry.fr/~aiguier](http://www.lami.univ-evry.fr/~aiguier).
- [2] Aiguier, M. and D. Bahrami, *Une approche générique de la réécriture*, *Technique et Science Informatiques (TSI)* **22** (2003).
- [3] Aiguier, M., D. Bahrami and C. Dubois, *On a generalised logicality theorem*, in: *AISC’2002*, L.N.A.I. **2385** (2002), pp. 51–64.
- [4] Aiguier, M., D. Bahrami and D. Longuet, *Abstract rewriting logic*, Technical report, University of Evry (2005), available at [www.lami.univ-evry.fr/~aiguier/](http://www.lami.univ-evry.fr/~aiguier/).
- [5] Baader, F. and T. Nipkow, “Term Rewriting and All That,” C.U. Press, 1998.
- [6] Bachmair, L. and H. Ganzinger, *Rewrite techniques for transitive relations*, in: *9th IEEE Symposium on Logic in Computer Science*, 1994, pp. 384–393.
- [7] Bournez, O. and M. Hoyrup, *Rewriting logic and probabilities*, in: S. Tison, editor, *Rewriting Techniques and Applications (RTA)*, L.N.C.S. **2706** (2003), pp. 61–75.
- [8] Bournez, O. and C. Kirchner, *Probabilistic rewrite strategies: Applications to elan*, in: S. Tison, editor, *Rewriting Techniques and Applications (RTA)*, L.N.C.S. **2378** (2002), pp. 252–266.
- [9] Bruni, R. and J. Meseguer, *Generalized rewrite theories*, in: J.-C.-M. Baeten, J.-K. Lenstra, J. Parrow and G.-J. Woeginger, editors, *Proceedings of ICALP 2003, 30th International Colloquium on Automata, Languages and Programming*, L.N.C.S. **2719** (2003), pp. 252–266.
- [10] Buchberger, B., *Gröbner bases: an algorithmic method in polynomial ideal theory*, in: N.-K. Bose, editor, *Multidimensional Systems Theory*, 1985, pp. 184–232.
- [11] Dershowitz, N. and J.-P. Jouannaud, *Rewrite systems*, in: *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics (B)*, Elsevier, 1990 pp. 243–320.
- [12] Ehrig, H., M. Grosse-Rhode and U. Wolter, *On the role of category theory in the area of algebraic specification*, in: *WADT*, L.N.C.S. **1130** (1995), pp. 17–48.

- [13] Fiadeiro, J. and A. Sernadas, *Structuring theories on consequence*, in: *Recent Trends in DT Spec.*, L.N.C.S. **332** (1988), pp. 44–72.
- [14] Goguen, J., “Resolution of Equations in Algebraic Structures,” Academic Press Inc., 1989 pp. 217–263.
- [15] Goguen, J. and R. Burstall, *Institutions: abstract model theory for specification and programming*, Journal of the ACM **39** (1992), pp. 95–146.
- [16] Hsiang, J., *Refutational theorem proving using term rewriting systems*, Artificial Intelligence **25** (1985), pp. 255–300.
- [17] Kirchner, C., *Réécriture, résolution d’équations et preuves en logique égalitaire*, EJC, GDR programmation du CNRS, Nantes (1998).
- [18] Kosiuczenko, P. and M. Wirsing., *Timed rewriting logic with an application to object-based specification*, Science of Computer Programming **28** (1997), pp. 225–246.
- [19] Levy, J. and J. Agusti, *Bi-rewrite systems*, Journal of Symbolic Computation **22** (1996), pp. 279–314.
- [20] Mahr, B. and J.-A. Mokowsky, *Characterizing specification language which admit initial semantics*, in: *Proc. of 8th CAAP*, number 159 in L.N.C.S. (1983), pp. 300–316.
- [21] Meseguer, J., *Conditional rewriting logic as a unified model of concurrency*, Theo. Comp. Sci. **96** (1992), pp. 73–155.
- [22] Meseguer, J., *Membership algebra as a logical framework for equational specification*, in: *WADT’97*, number 1376 in L.N.C.S. (1998), pp. 18–61.
- [23] Meseguer, J., *General logics*, in: *Logic Colloq. ’87* (1989), pp. 275–329.
- [24] Salibra, A. and G. Scollo, *A soft stairway to institutions*, in: *Recent Trends in Data Type Specification*, L.N.C.S. **655** (1993), pp. 310–329.
- [25] Struth, G., “Canonical Transformations in Algebra, Universal Algebra and Logic,” Ph.D. thesis, Institut Für Informatik, University of Saarlandes (1998).