# Generalized Rewrite Theories*

Roberto Bruni[1,2] and José Meseguer[2]

[1] Dipartimento di Informatica, Università di Pisa, Italia.
[2] CS Department, University of Illinois at Urbana-Champaign, USA.
bruni@di.unipi.it,meseguer@cs.uiuc.edu

**Abstract.** Since its introduction, more than a decade ago, rewriting logic has attracted the interest of both theorists and practitioners, who have contributed in showing its generality as a semantic and logical framework and also as a programming paradigm. The experimentation conducted in these years has suggested that some significant extensions to the original definition of the logic would be very useful in practice. In particular, the Maude system now supports subsorting and conditions in the equational logic for data, and also frozen arguments to block undesired nested rewritings; moreover, it allows equality and membership assertions in rule conditions. In this paper, we give a detailed presentation of the inference rules, model theory, and completeness of such generalized rewrite theories.

## Introduction

This paper develops new semantic foundations for a generalized version of rewriting logic. Since its original formulation [10], a substantial body of research (see the more than 300 references listed in the special TCS issue [6], and the four WRLA Proceedings in the ENTCS series, Vols. 4, 15, 36, and 71) has shown that rewriting logic (RL) has good properties as a *semantic framework*, particularly for concurrent and distributed computation, and also as a *logical framework*, a meta-logic in which other logics can be naturally represented. Indeed, the computational and logical meanings of a rewrite $t \to t'$ are like two sides of the same coin. Computationally, $t \to t'$ means that the state component $t$ can *evolve* to the component $t'$. Logically, $t \to t'$ means that from the formula $t$ one can *deduce* the formula $t'$. RL has also been shown to have good properties as a *declarative programming paradigm*, as demonstrated by the mature implementations of the ELAN [12], CafeOBJ [3], and Maude [2] languages.

The close contact with many applications in all the above areas has served as a good stimulus for a *substantial increase in expressive power* of the rewriting logic formalism by generalization along several dimensions:

1. Since a rewrite theory is essentially a triple $\mathcal{R} = (\Sigma, E, R)$, with $(\Sigma, E)$ an equational theory, and $R$ a set of labeled rewrite rules that are applied *modulo* the equations $E$, it follows that rewriting logic is *parameterized by the choice of an underlying equational logic*; therefore, generalizations towards more expressive equational logics yield more expressive versions of rewriting logic.
2. Another dimension along which expressiveness can be increased is by allowing *more general conditions* in conditional rewrite rules.
3. Yet another dimension has to do with *forbidding rewriting under certain operators or operator positions* (frozen operators and arguments). Although this could be regarded as a purely *operational aspect*, the need for it in many applications suggests supporting it directly at the semantic level of rewrite theories.

* Research supported by the MIUR Project COFIN 2001013518 CoMeta, by the FET-GC Project IST-2001-32747 Agile, and by ONR Grant N00014-02-1-0715. The first author is also supported by a CNR fellowship for research on Information Sciences and Technologies.

In this paper we generalize rewrite theories along these three dimensions. Along dimension 1, we select *membership equational logic* (MEL) [11] as the underlying equational logic. This is a very expressive many-kinded Horn logic whose atomic formulas are equations $t = t'$ and memberships $t : s$. It contains as special cases the order-sorted, many-sorted, and unsorted versions of equational logic. Along dimension 2, assuming an underlying MEL theory $(\Sigma, E)$, we allow for conditional rewrite rules of the form,

$$(\forall X)\ r{:}t \to t' \text{ if } \bigwedge_{i\in I} p_i = q_i \ \wedge\ \bigwedge_{j\in J} w_j : s_j \ \wedge\ \bigwedge_{l\in L} t_l \to t'_l$$

where $r$ is the rule label, all terms are $\Sigma$-terms, and the rule can be made conditional to other equations, memberships, and rewrites being satisfied. Finally, along dimension 3, we allow declaring certain operator arguments as *frozen*, thus blocking rewriting under them. This leads us to define a *generalized rewrite theory* (GRT) as a four tuple, $\mathcal{R} = (\Sigma, E, \phi, R)$, where $(\Sigma, E)$ is a membership equational theory, $R$ is a set of labeled conditional rewrite rules of the general form above, and $\phi$ is a function assigning to each operator $f : k_1 \dots k_n \to k$ in $\Sigma$ the subset $\phi(f) \subseteq \{1, \dots, n\}$ of its frozen arguments.

As already mentioned, such a notion of generalized rewrite theory has been arrived at through a long and extensive contact with many applications. In fact, practice has gone somewhat ahead of theory: all the above generalizations have already been implemented in the latest alpha versions of Maude 2.0. The importance of generalizing rewrite theories along dimension 1 has to do with the greater expressiveness allowed by having sorts, subsorts, subsort overloaded operators, and partial functions; all this is further explained in Section 1.2. We can illustrate the importance of generalizing along dimensions 2 and 3 with an example showing that, in essence, this brings RL and *structural operational semantics* (whose strong relationship had already been emphasized in [5,7,8]) closer than ever before. Consider for example a reactive process calculus with a nondeterministic choice operator $+$ specified by SOS rules of the form,

$$\frac{P \to P'}{P + Q \to P'}\ \textbf{left choice} \qquad \frac{Q \to Q'}{P + Q \to Q'}\ \textbf{right choice}$$

The corresponding rewrite theory $\mathcal{R}$ will then have two conditional rules, like

$$\textbf{left choice} : P + Q \to P' \ \text{ if } \ P \to P' \qquad \textbf{right choice} : P + Q \to Q' \ \text{ if } \ Q \to Q'$$

Furthermore, both arguments of $+$ should be *frozen*, i.e., $\phi(+) = \{1, 2\}$. If we add to this process calculus a sequential composition $P; Q$, the fact that $Q$ should not be able to evolve until $P$ has finished its task can be straightforwardly modeled by declaring the second argument of $\_;\_$ as frozen, plus the rule $\checkmark; Q \to Q$ (where $\checkmark$ is the "correct termination" process), which throws away the operator $\_;\_$, unfreezing its second argument. Hence, (un)frozen arguments can naturally model *reactive contexts*, i.e., the distinguished set of environments where reactions can take place. Note that frozen arguments are for rewrite theories the analogous of the *strategy annotations* used for equational theories in OBJ, CafeOBJ, and Maude to improve efficiency and/or to guarantee the termination of computations, replacing unrestricted equational rewriting by so-called *context-sensitive rewriting* [4]. Thus, in Maude, rewriting with both equations $E$ and rules $R$ can be made context-sensitive. The usefulness of having frozen attributes in rewrite theories has emerged gradually. Stehr, Meseguer, and Ölveczky first proposed *frozen kinds* [13]. The generalization of this to a subset $\Omega \subseteq \Sigma$ of *frozen operators* emerged in a series of email exchanges between Stefani and the second author. The subsequent generalization of freezing operator arguments selectively brings us to the just mentioned two levels (for equations and for rules) of context-sensitive rewriting.

Given the above notion of GRT, the paper addresses the following questions:

- What are rewriting logic's *rules of deduction* for generalized rewrite theories?
- What are the *models* of a rewrite theory? Are there initial and free models?
- Is rewriting logic *complete* with respect to its model theory, so that a rewrite is provable from a rewrite theory $\mathcal{R}$ if and only if it is satisfied by all models of $\mathcal{R}$?

The answers given (all in the affirmative) are in fact nontrivial *generalizations* of the original inference rules, model theory, initial and free models, and completeness theorem for rewriting logic over unsorted equational logic, as developed in [10]. In summary, therefore, this paper develops new *semantic foundations* for a generalized version of rewriting logic, along several dimensions that have been found to substantially increase its expressiveness in concrete applications. At the programming language level, this paper does also provide the needed mathematical semantics for Maude 2.0.

**Synopsis.** In § 1.1 we recap from [10] the original presentation of RL, and in § 1.2 we overview membership equational logic. § 2 and § 3 present the original contributions of the paper, introducing generalized rewrite theories, their proof theory, their model theory, and the completeness results. Note that the algebras of reachability and decorated sequents are expressed as membership equational theories themselves (a framework not available when [10] was published). Conclusions are drawn in the last section.

# 1 Background

## 1.1 Conditional rewriting logic

Though in the rewriting community it is folklore that rewrite theories are parametric w.r.t. the underlying equational logic of data specification, the details have been fully spelled out only for unsorted equational logic, and rules of the form (1) below.

Since only unsorted theories were treated in [10], here, but not in the rest of the paper where ordered sorts are used, an (equational) *signature* is a family of sets of *function symbols* (also *operators*) $\Sigma = \{\Sigma_n\}_{n\in\mathbb{N}}$ indexed by arities $n$, and a *theory* is a pair $(\Sigma, E)$ where $E = \{(\forall X_i)\ t_i = t_i'\}_{1\leq i\leq m}$ is a set of (universally quantified) $\Sigma$-equations, with $t_i, t_i' \in \mathbb{T}_\Sigma(X_i)$ two $\Sigma$-terms with variables in $X_i$. We let $t =_E t'$ denote the congruence modulo $E$ of two terms $t, t'$ and let $[t]_E$ or just $[t]$ denote the $E$-equivalence class of $t$ modulo $E$. We shall denote by $t[u_1/x_1, \ldots, u_n/x_n]$ (abbreviated $t[\vec{u}/\vec{x}]$) the term obtained from $t$ by simultaneously replacing the occurrences of $x_i$ by $u_i$ for $1 \leq i \leq n$.

**Definition 1.1 (Conditional rewrite theory).** *A (labeled) conditional rewrite theory $\mathcal{R}$ is a tuple $\mathcal{R} = (\Sigma, E, R)$, where $(\Sigma, E)$ is an unsorted equational theory and $R$ is a set of (labeled) conditional rewrite rules having the form below, with $t, t', t_i, t_i' \in \mathbb{T}_\Sigma(X)$.*

$$(\forall X)\ r{:}t \to t' \quad \textit{if} \quad t_1 \to t_1' \ \wedge \ \cdots \ \wedge \ t_\ell \to t_\ell'. \tag{1}$$

The theory $(\Sigma, E)$ defines the static data structure for the states of the system (e.g., a free monoid for strings, or a free commutative monoid for multisets), while $R$ defines the dynamics (e.g., productions in phrase-structure grammars or transitions in Petri nets).

Given a rewrite theory $\mathcal{R}$, its *rewriting logic* is a sequent calculus whose sentences have the form $(\forall X)\ t \to t'$ (with the dual, logico-computational meaning explained in the Introduction). We say that $\mathcal{R}$ *entails* a sequent $(\forall X)\ t \to t'$, and write $\mathcal{R} \vdash (\forall X)\ t \to t'$, if $(\forall X)\ t \to t'$ can be obtained by means of the inference rules in Figure 1. Roughly, **(Reflexivity)** introduces idle computations, **(Transitivity)** expresses the sequential composition of rewrites, **(Equality)** means that rewrites are applied modulo

$$\frac{t \in \mathbb{T}_\Sigma(X)}{(\forall X)\; t \to t}\ \textbf{Reflexivity} \qquad \frac{(\forall X)\; t_1 \to t_2, \qquad (\forall X)\; t_2 \to t_3}{(\forall X)\; t_1 \to t_3}\ \textbf{Transitivity}$$

$$\frac{E \vdash (\forall X)\; t = u, \qquad (\forall X)\; u \to u', \qquad E \vdash (\forall X)\; u' = t'}{(\forall X)\; t \to t'}\ \textbf{Equality}$$

$$\frac{f \in \Sigma_n, \qquad (\forall X)\; t_i \to t_i' \ \text{ for } \ i \in [1,n]}{(\forall X)\; f(t_1,\ldots,t_n) \to f(t_1',\ldots,t_n')}\ \textbf{Congruence}$$

$$\frac{\begin{array}{c}(\forall X)\; r{:}t \to t' \text{ if } \bigwedge_{1\leq i\leq \ell} t_i \to t_i' \in R, \qquad \theta,\theta'{:}X \to \mathbb{T}_\Sigma(Y)\\ (\forall Y)\; \theta(t_i) \to \theta(t_i') \ \text{ for } \ 1 \leq i \leq \ell, \qquad (\forall Y)\; \theta(x) \to \theta'(x) \ \text{ for } \ x \in X\end{array}}{(\forall Y)\; \theta(t) \to \theta'(t')}\ \begin{array}{l}\textbf{Nested}\\ \textbf{Replacement}\end{array}$$

**Fig. 1.** Deduction rules for conditional rewrite theories.

the equational theory $E$, **(Congruence)** says that rewrites can be nested inside larger contexts. The most complex rule is **(Nested Replacement)**, stating that given a rewrite rule $r \in R$ and two substitutions $\theta,\theta'$ for its variables such that for each $x \in X$ we have $\theta(x) \to \theta'(x)$, then $r$ can be concurrently applied to the rewrites of its arguments, once that the conditions of $r$ can be satisfied in the initial state defined by $\theta$. Since rewrites are applied modulo $E$, the sequents can be equivalently written $(\forall X)\ [t] \to [t\,']$.

From the model-theoretic viewpoint, the sequents can be decorated with *proof terms* in a suitable algebra that exactly captures concurrent computations. We remark that each rewrite theory $\mathcal{R}$ has initial and free models and that a completeness theorem reconciles the proof theory and the model theory, stating that a sequent is provable from $\mathcal{R}$ if and only if it is satisfied in all models of $\mathcal{R}$ (called *$\mathcal{R}$-systems*).

Roughly, the algebra of sequents contains the terms $[t]$ in $\mathbb{T}_{\Sigma,E}$ for idle rewrites, with the operators and equations in $(\Sigma,E)$ lifted to the level of sequents (e.g., if $\alpha_i{:}[t_i] \to [t_i']$ for $i \in [1,n]$, then $f(\alpha_1,\ldots,\alpha_n){:}[f(t_1,\ldots,t_n)] \to [f(t_1',\ldots,t_n')]$), plus the concatenation operator $\_;\_$ for composing $\alpha_1{:}[t_1] \to [t_2]$ and $\alpha_2{:}[t_2] \to [t_3]$ to $\alpha_1;\alpha_2{:}[t_1] \to [t_3]$ via **(Transitivity)**, and finally an additional operator $r$ with arity $|X|+\ell$ for each rule $r \in R$ of the form (1). For example, if $\{\beta_i{:}[\theta(t_i)] \to [\theta(t_i')]\}_{1\leq i\leq \ell}$ and $\{\alpha_x{:}[\theta(x)] \to [\theta'(x)]\}_{x\in X}$ are used as premises in **(Nested Replacement)**, then the conclusion is decorated by $r(\vec{\alpha},\vec{\beta})$. The axioms express: (i) that sequents form the arrows of a category with $\_;\_$ as composition and idle rewrites $[t]$ as identities; (ii) the functoriality of the $(\Sigma,E)$-structure, and (iii) the so-called *decomposition* and *exchange* laws, saying that the application of $r$ to $[\theta(t)]$ is concurrent w.r.t. the rewrites of the arguments of $t$.

### 1.2 Membership equational logic

In many applications, unsorted signatures are not expressive enough to reflect in a natural way the features of the system to be modeled. The expressiveness can be increased by supporting sorts (e.g., `Bool`, `Nat`, `Int`) via *many-sorted* signatures and relating them via *order-sorted* signatures (e.g., `NzNat` $<$ `Nat` $<$ `Int`). Equations in $E$ can be made more expressive by allowing *conditions* for their applications. Such conditions can be other equalities, or membership assertions. Conditional membership assertions are also useful. *Membership equational logic* (MEL) [11] possesses all the above features (generalizing order-sorted equational logic) and is supported by Maude [2].

A MEL *signature* is a triple $(K, \Sigma, S)$ (just $\Sigma$ in the following), with $K$ a set of *kinds*, $\Sigma = \{\Sigma_{\delta,k}\}_{(\delta,k) \in K^* \times K}$ a many-kinded signature and $S = \{S_k\}_{k \in K}$ a $K$-kinded family of disjoint sets of sorts. The kind of a sort $s$ is denoted by $[s]$. A MEL $\Sigma$-algebra $A$ contains a set $A_k$ for each kind $k \in K$, a function $A_f : A_{k_1} \times \cdots \times A_{k_n} \to A_k$ for each operator $f \in \Sigma_{k_1 \cdots k_n, k}$ and a subset $A_s \subseteq A_k$ for each sort $s \in S_k$, with the meaning that the elements in sorts are well-defined, while elements without a sort are *errors*. We write $\mathbb{T}_{\Sigma,k}$ and $\mathbb{T}_{\Sigma}(X)_k$ to denote respectively the set of ground $\Sigma$-terms with kind $k$ and of $\Sigma$-terms with kind $k$ over variables in $X$, where $X = \{x_1 : k_1, \ldots, x_n : k_n\}$ is a set of kinded variables.

Given a MEL signature $\Sigma$, *atomic formulae* have either the form $t = t'$ ($\Sigma$-equation) or $t : s$ ($\Sigma$-membership) with $t, t' \in \mathbb{T}_{\Sigma}(X)_k$ and $s \in S_k$; and $\Sigma$-*sentences* are conditional formulae of the form $(\forall X)\ \varphi$ if $\bigwedge_i p_i = q_i \wedge \bigwedge_j w_j : s_j$, where $\varphi$ is either a $\Sigma$-equation or a $\Sigma$-membership and all the variables in $\varphi$, $p_i$, $q_i$, and $w_j$ are in $X$. A MEL theory is a pair $(\Sigma, E)$ with $\Sigma$ a MEL signature and $E$ a set of $\Sigma$-sentences. We refer to [11] for the detailed presentation of $(\Sigma, E)$-algebras, sound and complete deduction rules, initial and free algebras, and theory morphisms.

Order-sorted notation $s_1 < s_2$ can be used to abbreviate the conditional membership $(\forall x : k)\ x : s_2$ if $x : s_1$. Similarly, an operator declaration $f : s_1 \times \cdots \times s_n \to s$ corresponds to declaring $f$ at the kind level and giving the membership axiom $(\forall x_1 : k_1, \ldots, x_n : k_n)\ f(x_1, \ldots, x_n) : s$ if $\bigwedge_{1 \le i \le n} x_i : s_i$. We write $(\forall x_1 : s_1, \ldots, x_n : s_n)\ t = t'$ in place of $(\forall x_1 : k_1, \ldots, x_n : k_n)\ t = t'$ if $\bigwedge_{1 \le i \le n} x_i : s_i$. Moreover, for a list of variables of the same sort $s$, we write $(\forall x_1, \ldots, x_n : s)$, and let the sentence $(\forall X)\ t : k$ mean $t \in \mathbb{T}_{(\Sigma,E)}(X)_k$.

## 2 Generalized rewrite theories and deduction

In this section we present the foundations of rewrite theories over MEL theories and where operators can have frozen arguments.

A *generalized operator* is a function symbol $f : k_1 \cdots k_n \to k$ together with a set $\phi(f) \subseteq \{1, \ldots, n\}$ of frozen argument positions. We denote by $\nu(f)$ the set $\{1, \ldots, n\} \setminus \phi(f)$ of *unfrozen* arguments, and say that $f$ is *unfrozen* if $\phi(f) = \varnothing$.

**Definition 2.1 (Generalized signatures).** *A* generalized MEL signature $(\Sigma, \phi)$ *is a* MEL *signature* $\Sigma$ *whose function symbols are generalized operators. The function* $\phi : \Sigma \to \wp_f(\mathbb{N})$ *assigns to each* $f \in \Sigma$ *its set of frozen arguments* ($\wp_f(\mathbb{N})$ *denotes the set of finite sets of natural numbers and for any* $f : k_1 \cdots k_n \to k$ *in* $\Sigma$ *we assume* $\phi(f) \subseteq \{1, \ldots, n\}$*).*

If the $i$th position of $f$ is frozen, then in $f(t_1, ..., t_n)$ any subterm of $t_i$ is frozen. This can be made formal by considering the usual tree-like representation of terms (the same subterm can occur in many distinct positions that are not necessarily all frozen). Positions in a term are denoted by strings of natural numbers, indicating the sequences of branches we must follow from the root to reach that position. For example, the term $t = f(g(a,b,c), f(h(a,b), f(b,c)))$ has two occurrences of the constant $c$ at positions 1.3 and 2.2.2, respectively. We let $t_\pi$ and $t(\pi)$ denote, respectively, the subterm of $t$ occurring at position $\pi$, and its topmost operator. For $\lambda$ the empty position, we let $t_\lambda$ denote the whole term $t$. In the example above, we have $t_{2.1} = h(a,b)$ and $t(2.1) = h$.

**Definition 2.2 (Frozen occurrences).** *The occurrence* $t_\pi$ *of the subterm of* $t$ *at position* $\pi$ *is* frozen *if there exist two positions* $\pi_1$, $\pi_2$ *and a natural number* $n$ *such that* $\pi = \pi_1.n.\pi_2$ *and* $n \in \phi(t(\pi_1))$. *The occurrence* $t_\pi$ *is called* unfrozen *if it is not frozen.*

In the example above, for $\phi(f) = \phi(g) = \varnothing$ and $\phi(h) = \{1\}$, we have that $t_{2.1.1} = a$ is frozen (because $t(2.1) = h$), while $t_{1.1} = a$ is unfrozen (because $t(\lambda) = f$ and $t(1) = g$).

$$\frac{t \in \mathbb{T}_\Sigma(X)_k}{(\forall X)\, t \to t}\ \textbf{Reflexivity} \qquad \frac{(\forall X)\, t_1 \to t_2, \qquad (\forall X)\, t_2 \to t_3}{(\forall X)\, t_1 \to t_3}\ \textbf{Transitivity}$$

$$\frac{E \vdash (\forall X)\, t = u, \qquad (\forall X)\, u \to u', \qquad E \vdash (\forall X)\, u' = t'}{(\forall X)\, t \to t'}\ \textbf{Equality}$$

$$\frac{\begin{array}{c} f \in \Sigma_{k_1 \cdots k_n, k}, \qquad t_i, t'_i \in \mathbb{T}_\Sigma(X)_{k_i} \text{ for } i \in [1,n] \\ t'_i = t_i \text{ for } i \in \phi(f), \qquad (\forall X)\, t_j \to t'_j \text{ for } j \in \nu(f) \end{array}}{(\forall X)\, f(t_1,\ldots,t_n) \to f(t'_1,\ldots,t'_n)}\ \textbf{Congruence}$$

$$\frac{\begin{array}{c} (\forall X)\, r{:}t \to t' \text{ if } \bigwedge_{i\in I} p_i = q_i \wedge \bigwedge_{j \in J} w_j : s_j \wedge \bigwedge_{l \in L} t_l \to t'_l \in R \\ \theta, \theta' {:} X \to \mathbb{T}_\Sigma(Y), \qquad \theta(x) = \theta'(x) \text{ for } x \in \phi(t,t') \\ E \vdash (\forall Y)\, \theta(p_i) = \theta(q_i) \text{ for } i \in I, \qquad E \vdash (\forall Y)\, \theta(w_j) : s_j \text{ for } j \in J \\ (\forall Y)\, \theta(t_l) \to \theta(t'_l) \text{ for } l \in L, \qquad (\forall Y)\, \theta(x) \to \theta'(x) \text{ for } x \in \nu(t,t') \end{array}}{(\forall Y)\, \theta(t) \to \theta'(t')}\ \textbf{Nested Replacement}$$

**Fig. 2.** Deduction rules for generalized rewrite theories.

**Definition 2.3 (Frozen variables).** *Given $t \in \mathbb{T}_\Sigma(X)$ we say that the variable $x \in X$ is* frozen *in $t$ if there exists a frozen occurrence of $x$ in $t$, otherwise it is called* unfrozen.

We let $\phi(t)$ and $\nu(t)$ denote, respectively, the set of frozen and unfrozen variables of $t$. Analogously, $\phi(t_1,\ldots,t_n)$ (resp. $\nu(t_1,\ldots,t_n)$) denotes the set of variables for which a frozen occurrence appears in at least one $t_i$ (resp. that are unfrozen in all $t_i$).

By combining conditional rewrite theories with MEL specifications and frozen arguments, we obtain a rather general notion of rewrite theory.

**Definition 2.4 (Generalized rewrite theory).** *A* generalized rewrite theory *(GRT) is a tuple $\mathcal{R} = (\Sigma, E, \phi, R)$ consisting of: (i) a generalized MEL signature $(\Sigma, \phi)$ with say kinds $k \in K$, sorts $s \in S$, and $K^* \times K$-indexed set of generalized operators $f \in \Sigma$ with frozen arguments according to $\phi$; (ii) a MEL theory $(\Sigma, E)$; (iii) a set $R$ of (universally quantified) labeled conditional rewrite rules $r$ having the general form*

$$(\forall X)\, r{:}t \to t' \text{ if } \bigwedge_{i\in I} p_i = q_i \wedge \bigwedge_{j \in J} w_j : s_j \wedge \bigwedge_{l \in L} t_l \to t'_l \qquad (2)$$

*where, for appropriate kinds $k$ and $k_l$ in $K$, $t, t' \in \mathbb{T}_\Sigma(X)_k$ and $t_l, t'_l \in \mathbb{T}_\Sigma(X)_{k_l}$ for $l \in L$.*

### 2.1 Inference in generalized rewriting logic

Given a GRT $\mathcal{R} = (\Sigma, E, \phi, R)$, a *sequent* of $\mathcal{R}$ is a pair of (universally quantified) terms of the same kind $t, t'$, denoted $(\forall X) t \to t'$ with $X = \{x_1 : k_1, \ldots, x_n : k_n\}$ a set of kinded variables and $t, t' \in \mathbb{T}_\Sigma(X)_k$ for some $k$. We say that $\mathcal{R}$ *entails* the sequent $(\forall X)\, t \to t'$, and write $\mathcal{R} \vdash (\forall X)\, t \to t'$, if the sequent $(\forall X)\, t \to t'$ can be obtained by means of the inference rules in Figure 2, which are briefly described below.

**(Reflexivity)**, **(Transitivity)**, and **(Equality)** are the usual rules for idle rewrites, concatenation of rewrites, and rewriting modulo the MEL theory $E$. **(Congruence)** allows rewriting the arguments of a generalized operator, but we add the condition that frozen arguments must stay idle (note that $t'_i = t_i$ is syntactic equality). Any unfrozen argument can still be rewritten, as expressed by the premise $(\forall X)\, t_j \to t'_j$ for $j \in \nu(f)$.

**(Nested Replacement)** takes into account the application of a rewrite rule in its most general form (2). It specifies that for any rewrite rule $r \in R$ and for any (kind-preserving) substitution $\theta$ such that the condition of $r$ is satisfied when $\theta$ is applied to all terms $p_i, q_i, w_j, t_l, t'_l$ involved, then it is possible to apply the rewrite $r$ to $\theta(t)$. Moreover, if $\theta'$ is a second (kind-preserving) substitution for the variables in $X$ such that $\theta$ and $\theta'$ coincide on all frozen variables $x \in \phi(t,t')$ (second line of premises), while the rewrites $(\forall Y)\ \theta(x) \to \theta'(x)$ are provable for the unfrozen variables $x \in \nu(t,t')$ (last premise), then such nested rewrites can be applied concurrently with $r$.

Of course, any unsorted rewrite theory can be regarded as a GRT where: (i) $\Sigma$ has a unique kind and no sorts; (ii) all the operators are total and unfrozen (i.e., $\phi(f) = \varnothing$ for any $f \in \Sigma$); (iii) conditions in rewrite rules contain neither equalities nor membership predicates. In this case, deduction via rules for conditional rewrite theories (Figure 1) coincides with deduction via rules for generalized rewrite theories (Figure 2).

**Theorem 2.1.** *Let $\mathcal{R}$ be a conditional rewrite theory, and let $\hat{\mathcal{R}}$ denote its corresponding* GRT. *Then:* $\mathcal{R} \vdash (\forall X)\ t \to t' \quad \Leftrightarrow \quad \hat{\mathcal{R}} \vdash (\forall X)\ t \to t'$.

## 3 Models of generalized rewrite theories

In this section, exploiting MEL, we define the reachability and concurrent model theories of GRTs and state completeness results.

### 3.1 Reachability models

Reachability models focus just on *what* terms/states can be reached from a certain state $t$ via sequences of rewrites, ignoring *how* the rewrites can lead to them.

**Definition 3.1 (Reachability relation).** *Given a* GRT $\mathcal{R} = (\Sigma, E, \phi, R)$, *its* reachability relation $\to_{\mathcal{R}}$, *is defined proof-theoretically, for each kind $k$ in $\Sigma$ and each $[t],[t'] \in \mathbb{T}_{\Sigma,E}(X)_k$, by the equivalence:* $[t] \to_{\mathcal{R}} [t'] \quad \Leftrightarrow \quad \mathcal{R} \vdash (\forall X)\ t \longrightarrow t'$.

The above definition is sound because we have the following easy lemma.

**Lemma 3.1.** *Let $\mathcal{R} = (\Sigma, E, \phi, R)$ be a* GRT, *and $t \in \mathbb{T}_\Sigma(X)_k$. If $\mathcal{R} \vdash (\forall X)\ t \longrightarrow t'$, then $t' \in \mathbb{T}_\Sigma(X)_k$. Moreover, for any $t, u, u', t' \in \mathbb{T}_\Sigma(X)_k$ such that $u \in [t]_E$, $u' \in [t']_E$ and $\mathcal{R} \vdash (\forall X)\ u \longrightarrow u'$, then $\mathcal{R} \vdash (\forall X)\ t \longrightarrow t'$.*

The reachability relation admits a model-theoretic presentation in terms of the free models of a suitable MEL theory. We give the details below as a "warm up" for the model-theoretic concurrent semantics given in the next section. The idea is that $\to_{\mathcal{R}}$ can be defined as the family of relations, indexed by the kinds $k$, given by interpreting the sorts $Ar_k$ in the free model of the following MEL theory $Reach(\mathcal{R})$.

**Definition 3.2 (The theory $Reach(\mathcal{R})$).** *The membership equational theory $Reach(\mathcal{R})$ contains the signature and sentences in $(\Sigma, E)$ together with the following extensions:*

1. *For each kind $k$ in $\Sigma$ we add:*
   (a) *a new kind $[Pair_k]$ (for $k$-indexed binary relations on terms of kind $k$) with four sorts $Ar^0_k$, $Ar^1_k$, $Ar_k$ and $Pair_k$ and subsort inclusions: $Ar^0_k\ Ar^1_k < Ar_k < Pair_k$;*
   (b) *the operators $(\_ \to \_) : k\ k \longrightarrow Pair_k$ (pair constructor),* $\mathsf{s}, \mathsf{t} : Pair_k \longrightarrow k$ *(source and target projections), and $(\_;\_) : [Pair_k]\ [Pair_k] \longrightarrow [Pair_k]$ (concatenation);*

*(c) the (conditional) equations and memberships*

$$(\forall x,y:k)\ \mathsf{s}(x\to y)=x$$
$$(\forall x,y:k)\ \mathsf{t}(x\to y)=y$$
$$(\forall z:Pair_k)\ (\mathsf{s}(z)\to \mathsf{t}(z))=z$$
$$(\forall x:k)\ (x\to x):Ar^0_k$$
$$(\forall x,y,z:k)\ (x\to z):Ar_k \quad if \quad (x\to y):Ar_k\wedge(y\to z):Ar_k$$
$$(\forall x,y,z:k)\ (x\to y);(y\to z)=(x\to z).$$

2. *Each $f:k_1\dots k_n\longrightarrow k$ in $\Sigma$ with $\nu(f)\neq\varnothing$ is lifted to $f:[Pair_{k_1}]\cdots[Pair_{k_n}]\longrightarrow [Pair_k]$, and for each $i\in\nu(f)$ we declare $f:Ar^0_{k_1}\cdots Ar^1_{k_i}\cdots Ar^0_{k_n}\longrightarrow Ar^1_k$; we then give, for each $i\in\nu(f)$, the equation below, where $X_i=\{x_1:k_1,\dots,x_n:k_n,y_i:k_i\}$*

$$(\forall X_i)\ f((x_1\to x_1),...,(x_i\to y_i),...,(x_n\to x_n))=f(x_1,...,x_n)\to f(x_1,...,y_i,...,x_n).$$

3. *For each rule $(\forall X)\ r:t\to t'$ if $\bigwedge_{i\in I}p_i=q_i\wedge\bigwedge_{j\in J}w_j:s_j\wedge\bigwedge_{l\in L}t_l\to t'_l$ in R, with, say $t,t'$ of kind k, and $t_l,t'_l$ of kind $k_l$, we give the conditional membership,*

$$(\forall X)\ (t\to t'):Ar^1_k\ \ if\ \bigwedge_{i\in I}p_i=q_i\wedge\bigwedge_{j\in J}w_j:s_j\wedge\bigwedge_{l\in L}t_l\to t'_l:Ar_{k_l}.$$

The sorts $Ar^0_k$ and $Ar^1_k$ contain respectively idle rewrites and one-step rewrites of $k$-kinded terms, while the sort $Ar_k$ contains $k$-rewrites of arbitrary length. The **(Congruence)** rule is modeled so that exactly one unfrozen argument can be rewritten in one-step (see item 2 in Definition 3.2), and **(Nested Replacement)** is restricted so that no nested rewrites can take place concurrently (item 3). Nevertheless, these two restrictions on how the inference rules are modeled do not alter the reachability relation $Ar_k$, because one-step rewrites can be composed in any admissible interleaved fashion (see the fifth axiom at point 1.(c)). Note that the concatenation operator $\_;\_$ is not really necessary, but its introduction facilitates the proof of Theorem 3.2.

The theory $Reach(\mathcal{R})$ provides an algebraic model for the reachability relation. For ground terms, such a model is given by the interpretation of the sorts $Ar_k$ in the initial model $\mathbb{T}_{Reach(\mathcal{R})}$. For terms with variables in $X$, the reachability model is the free algebra $\mathbb{T}_{Reach(\mathcal{R})}(X)$. This can be summarized by the following theorem:

**Theorem 3.1.** *For $\mathcal{R}=(\Sigma,E,\phi,R)$ a GRT and $t,t'\in T_\Sigma(X)_k$ we have the equivalences:*

$$\begin{array}{lcl}\mathcal{R}\vdash(\forall X)\ t\to t' & \Leftrightarrow & Reach(\mathcal{R})\vdash(\forall X)\ (t\to t'):Ar_k\\ & \Leftrightarrow & Reach(\mathcal{R})\models(\forall X)\ (t\to t'):Ar_k\\ & \Leftrightarrow & [(t\to t')]\in\mathbb{T}_{Reach(\mathcal{R})}(X)_{Ar_k}.\end{array}$$

### 3.2 Concurrent models

In general, many proofs concluding that $\mathcal{R}\vdash(\forall X)t\to t'$ are possible. However: (1) some of the proofs can be computationally equivalent, because they represent different interleaved sequences for the same concurrent computation, but (2) not all those proofs are necessarily equivalent, as they may, e.g., differ in the underlying set of applied rewrite rules, or in the different causal connections between the applications of the same rules. In this section, we show how to extend the notion of decorated sequents to GRTs, so as to define an algebraic model of *true concurrency* for $\mathcal{R}$.

As usual, decorated sequents are first defined by attaching a *proof term* (i.e., an expression built from variables, operators in $\Sigma$, and labels in $R$) to each sequent, and then by quotienting out proof terms modulo suitable functoriality, decomposition, and exchange laws. We can present $\mathcal{R}$'s algebra of sequents as the initial (or free) algebra of a suitable MEL theory $Proof(\mathcal{R})$. With respect to the classical presentation via decorated deduction rules, the MEL specification allows a standard algebraic definition of initial and loose semantics. Moreover, here we can naturally support many-sorted, order-sorted, and MEL data theories instead of just unsorted equational theories as in [10].

The construction of $Proof(\mathcal{R})$ is analogous to that of $Reach(\mathcal{R})$. The kind $[Pair_k]$ of $Reach(\mathcal{R})$ is replaced here by a kind $[Rw_k]$, whose elements include the proofs of concurrent computations. The initial and final states are still defined by means of the source ($\mathsf{s}$) and target ($\mathsf{t}$) operators. Moreover, since the proof of an idle rewrite $[t] \to [t]$ is $[t]$ itself, we can exploit subsorting to make $k$ a sort of kind $[Rw_k]$. The sorts $Rw_k^1$ and $Rw_k$ are the analogous of $Ar_k^1$ and $Ar_k$. The sort $Ar_k^1$ was introduced in $Reach(\mathcal{R})$ to deal with the "restricted" form of **(Congruence)** and **(Nested Replacement)**. Having decorations at hand, we can restore the full expressiveness of the two inference rules, but the sort $Rw_k^1$ is still useful in axiomatizing proof-decorated sequents: we define the **(Equality)** rule on $Rw_k^1$, lifting the equational theory $E$ to one-step rewrites, and then exploit functoriality and transitivity to extend $E$ to rewrites of arbitrary length in $Rw_k$.

**Definition 3.3 (The theory $Proof(\mathcal{R})$).** *The membership equational theory $Proof(\mathcal{R})$ contains the signature and sentences of $(\Sigma, E)$ together with the following extensions:*

1. *Each kind $k$ in $\Sigma$ becomes a sort $k$ in $Proof(\mathcal{R})$, with $s < k$ for any $s \in S_k$ in $\Sigma$.*
2. *For each kind $k$ in $\Sigma$ we add:*
   (a) *a new kind $[Rw_k]$ (for $k$-indexed decorated rewrites on $\Sigma$-terms of kind $k$) with sorts all sorts in $k$ and the (new) sorts $k$, $Rw_k^1$ and $Rw_k$, with: $k\ \ Rw_k^1 < Rw_k$;*
   (b) *the (overloaded) operators $(\_;\_) : [Rw_k]\,[Rw_k] \longrightarrow [Rw_k]$ and $\mathsf{s}, \mathsf{t} : Rw_k \longrightarrow k$;*
   (c) *the (conditional) equations and memberships*

$$
\begin{aligned}
(\forall x : k)\ \mathsf{s}(x) &= x \\
(\forall x : k)\ \mathsf{t}(x) &= x \\
(\forall x, y : Rw_k)\ x;y : Rw_k &\quad \textit{if} \quad \mathsf{t}(x) = \mathsf{s}(y) \\
(\forall x, y : Rw_k)\ \mathsf{s}(x;y) = \mathsf{s}(x) &\quad \textit{if} \quad \mathsf{t}(x) = \mathsf{s}(y) \\
(\forall x : Rw_k, y : Rw_k)\ \mathsf{t}(x;y) = \mathsf{t}(y) &\quad \textit{if} \quad \mathsf{t}(x) = \mathsf{s}(y) \\
(\forall x : k, y : Rw_k)\ x;y = y &\quad \textit{if} \quad x = \mathsf{s}(y) \\
(\forall x : Rw_k, y : k)\ x;y = x &\quad \textit{if} \quad \mathsf{t}(x) = y \\
(\forall x, y, z : Rw_k)\ x;(y;z) = (x;y);z &\quad \textit{if} \quad \mathsf{t}(x) = \mathsf{s}(y)\ \wedge\ \mathsf{t}(y) = \mathsf{s}(z).
\end{aligned}
$$

3. *We lift each operator $f : k_1 \dots k_n \longrightarrow k$ in $\Sigma$ to $f : [Rw_{k_1}] \cdots [Rw_{k_n}] \longrightarrow [Rw_k]$, and for $\nu(f) = \{i_1, ..., i_m\}$ we overload $f$ by $f : k_1 \cdots Rw_{k_{i_1}} \cdots Rw_{k_{i_m}} \cdots k_n \longrightarrow Rw_k$ and $f : k_1 \cdots Rw_{k_{i_j}}^1 \cdots k_n \longrightarrow Rw_k^1$ for $j = 1, \dots, m$, with equations*

$$
\begin{aligned}
(\forall X)\ \mathsf{s}(f(x_1, \dots, x_n)) &= f(\mathsf{s}(x_1), \dots, \mathsf{s}(x_n)) \\
(\forall X)\ \mathsf{t}(f(x_1, \dots, x_n)) &= f(\mathsf{t}(x_1), \dots, \mathsf{t}(x_n)),
\end{aligned}
$$

*where $X = \{x_1 : k_1, \dots, x_{i_1} : Rw_{k_{i_1}}, \dots, x_{i_m} : Rw_{k_{i_m}}, \dots, x_n : k_n\}$ and the equation*

$$
\begin{aligned}
&(\forall Y)\ f(x_1, \dots, (x_{i_1}; y_{i_1}), \dots, (x_{i_m}; y_{i_m}), \dots, x_n)) = \\
&\qquad f(x_1, \dots, x_n); f(x_1, \dots, y_{i_1}, \dots, y_{i_m}, \dots, x_n) \quad \textit{if} \quad \textstyle\bigwedge_{1 \le j \le m} \mathsf{t}(x_{i_j}) = \mathsf{s}(y_{i_j}),
\end{aligned}
$$

*where $Y = \{x_1 : k_1, \dots, x_{i_1}, y_{i_1} : Rw_{k_{i_1}}, \dots, x_{i_m}, y_{i_m} : Rw_{k_{i_m}}, \dots, x_n : k_n\}$.*

4. *For each equation* $(\forall x_1 : k_1, \ldots, x_n : k_n)\ t = t'$ *if* $\bigwedge_{i \in I} p_i = q_i \wedge \bigwedge_{j \in J} w_j : s_j$ *in E, we let* $X = \{x_1 : Rw_{k_1}, \ldots, x_n : Rw_{k_n}\}$ *and add the conditional equation*

$$(\forall X)\ t = t' \ \ if \bigwedge_{i \in I} p_i = q_i \wedge \bigwedge_{j \in J} \mathsf{s}(w_j) : s_j \wedge \bigwedge_{j \in J} \mathsf{t}(w_j) : s_j \wedge \bigwedge_{x_h \in \phi(t,t')} x_h : k_h \wedge \bigwedge_{x_h \in \nu(t,t')} x_h : Rw^1_{k_h}.$$

5. *For each rule* $(\forall X)\ r : t \to t'$ *if* $\bigwedge_{i \in I} p_i = q_i \wedge \bigwedge_{j \in J} w_j : s_j \wedge \bigwedge_{l \in L} t_l \to t'_l$ *in R, with, say,* $X = \{x_1 : k_1, \ldots, x_n : k_n\}$, $t, t'$ *of kind k, and* $t_l, t'_l$ *of kind* $k'_l$ *with* $L = \{1, \ldots, \ell\}$, *we add the operator* $r$: $[Rw_{k_1}] \cdots [Rw_{k_n}][Rw_{k'_1}] \cdots [Rw_{k'_\ell}] \to [Rw_k]$ *with*
   (a) *the conditional membership for characterizing basic one-step rewrites:*

$$(\forall x_1 : k_1, \ldots, x_n : k_n, y_1 : Rw_{k'_1}, \ldots, y_\ell : Rw_{k'_\ell})\ r(\vec{x}, \vec{y}) : Rw^1_k \quad if \quad \Delta$$

   *where* $\Delta = (\bigwedge_{i \in I} p_i = q_i \wedge \bigwedge_{j \in J} w_j : s_j \wedge \bigwedge_{l \in L} \mathsf{s}(y_l) = t_l \wedge \bigwedge_{l \in L} \mathsf{t}(y_l) = t'_l)$ *checks that the conditions for the application of the rule r are satisfied;*
   (b) *the conditional equations and memberships*

$$(\forall Y)\ r(\vec{z}, \vec{y}) : Rw_k \quad if \quad \Delta \wedge \Psi$$
$$(\forall Y)\ \mathsf{s}(r(\vec{z}, \vec{y})) = t \quad if \quad \Delta \wedge \Psi$$
$$(\forall Y)\ \mathsf{t}(r(\vec{z}, \vec{y})) = t'[\mathsf{t}(\vec{z})/\vec{x}] \quad if \quad \Delta \wedge \Psi$$

   *where* $Y = \{x_1 : k_1, \ldots, x_n : k_n, z_1 : Rw_{k_1}, \ldots, z_n : Rw_{k_n}, y_1 : Rw_{k'_1}, \ldots, y_\ell : Rw_{k'_\ell}\}$, $\Delta$ *is as before, and* $\Psi = (\bigwedge_{x_h \in \phi(t,t')} z_h = x_h \wedge \bigwedge_{x_h \in \nu(t,t')} \mathsf{s}(z_h) = x_h)$;
   (c) *the decomposition law*

$$(\forall Z)\ r(\vec{z}, \vec{y}) = r(\vec{x}, \vec{y}); t'[\vec{z}/\vec{x}] \quad if \quad \Delta \wedge \Psi$$

   *where* $Z = \{x_1 : k_1, \ldots, x_n : k_n, z_1 : Rw_{k_1}, \ldots, z_n : Rw_{k_n}, y_1 : Rw_{k'_1}, \ldots, y_\ell : Rw_{k'_\ell}\}$, *while* $\Delta$ *and* $\Psi$ *are as before;*
   (d) *the exchange law*

$$(\forall W)\ r(\vec{x}, \vec{y}); t'[\vec{z}/\vec{x}] = t[\vec{z}/\vec{x}]; r(\mathsf{t}(\vec{z}), \vec{y'}) \quad if \quad \Delta \wedge \Psi \wedge \Delta' \wedge \Phi$$

   *where* $W = \{x_1 : k_1, \ldots, x_n : k_n, z_1 : Rw^1_{k_1}, \ldots, z_n : Rw^1_{k_n}, y_1 : Rw_{k'_1}, \ldots, y_\ell : Rw_{k'_\ell}, y'_1 : Rw_{k'_1}, \ldots, y'_\ell : Rw_{k'_\ell}\}$, $\Delta$ *and* $\Psi$ *are as before,* $\Delta' = (\bigwedge_{i \in I} p_i[\mathsf{t}(\vec{z})/\vec{x}] = q_i[\mathsf{t}(\vec{z})/\vec{x}] \wedge \bigwedge_{j \in J} w_j[\mathsf{t}(\vec{z})/\vec{x}] : s_j \wedge \bigwedge_{l \in L} \mathsf{s}(y'_l) = t_l[\mathsf{t}(\vec{z})/\vec{x}] \wedge \bigwedge_{l \in L} \mathsf{t}(y'_l) = t'_l[\mathsf{t}(\vec{z})/\vec{x}])$ *checks that the conditions for the application of the rule r are satisfied after applying the rewrites* $\vec{z}$ *to the arguments of t, and* $\Phi = (\bigwedge_{l \in L} y_l; t'_l[\mathsf{t}(\vec{z})/\vec{x}] = t_l[\mathsf{t}(\vec{z})/\vec{x}]; y'_l)$ *states the correspondence between the "side" rewrites* $\vec{y}$ *and* $\vec{y'}$ *(via* $\vec{z}$*).*

We briefly comment on the definition of $Proof(\mathcal{R})$. The operators defined at point 2.(b) are the obvious source/target projections and sequential composition of rewrites, with the axioms stating that, for each $k$, the rewrites in $Rw_k$ are the arrows of a category with objects in $k$. The operators $f$ in $\Sigma$ are lifted to functors over rewrites in 3, while the equations in $E$ are extended to rewrites in 4. It is worth noting that: (i) when $f \in \Sigma$ is lifted, only unfrozen positions can have rewrites as arguments, and therefore the functoriality is stated w.r.t. unfrozen positions only; (ii) the axioms in $E$ are extended to one-step rewrites only (in unfrozen positions), hence, they hold for sequences of rewrites if and only if they can be proved to hold for each rewrite step. Point 5.(a) defines the basic one-step rewrites, i.e., where no rewrite occurs in the arguments $\vec{x}$. Point 5.(b) accounts for nested rewrites $\vec{z}$ below $r$, provided that the side-conditions of $r$ are satisfied by the initial state; in particular note that the expression $r(\vec{z},\vec{y})$ is always equivalent to $r(\vec{x},\vec{y});t'[\vec{z}/\vec{x}]$ (see decomposition law), where first $r$ is applied at the top of the term and then the arguments are rewritten according to $\vec{z}$ under $t'$. Finally, the exchange law states that, under suitable hypotheses, the arguments $\vec{x}$ can be equivalently rewritten first, and the rewrite rule $r$ applied later. Note that, as in the equations extending $E$, the exchange law is stated for one-step nested rewrites only. Nevertheless, it can be used in conjunction with the decomposition law to prove the exchange law for arbitrary long sequences of rewrites (provided that it can be applied step-by-step).

An important property for $Proof(\mathcal{R})$ is the preservation of the underlying state theory $(\Sigma,E)$. Otherwise, the additional axioms in $Proof(\mathcal{R})$ might collapse terms that are different in $(\Sigma,E)$. In this regard, the fact of adding the sorts $Rw^1_k$ and $Rw_k$ on top of $k$ is a potential source of term collapses. However, we can prove that, for any GRT $\mathcal{R}$, the theory $Proof(\mathcal{R})$ is a conservative extension of the underlying theory $(\Sigma,E)$.

**Proposition 3.1.** *Let $\mathcal{R} = (\Sigma,E,\phi,R)$ be a* GRT*, and let $t,t' \in T_\Sigma(X)_k$, and $s \in S_k$ for some kind $k$. Then, for any formula $\varphi$ of the form $t : k$ or $t : s$ or $t = t'$ we have that: $E \vdash (\forall X)\ \varphi \quad \Leftrightarrow \quad Proof(\mathcal{R}) \vdash (\forall X)\ \varphi$.*

The main result is that $Proof(\mathcal{R})$ is complete w.r.t. the inference rules in Figure 2.

**Theorem 3.2 (Completeness I).** *For any* GRT *$\mathcal{R} = (\Sigma,E,\phi,R)$ and any $t,t' \in T_\Sigma(X)_k$, we have: $\mathcal{R} \vdash (\forall X)\ t \to t' \quad \Leftrightarrow \quad \exists\alpha.\ Proof(\mathcal{R}) \vdash (\forall X)\ \alpha : Rw_k \wedge \mathsf{s}(\alpha) = t \wedge \mathsf{t}(\alpha) = t'$.*

The relevance of the MEL theory $Proof(\mathcal{R})$ is far beyond the essence of reachability, as it precisely characterizes the class of computational models of $\mathcal{R}$.

**Definition 3.4 (Concurrent models of $\mathcal{R}$).** *Let $\mathcal{R}$ be a* GRT*. A* concurrent model *of $\mathcal{R}$ is a $Proof(\mathcal{R})$-algebra.*

Since $Proof(\mathcal{R})$ is an ordinary MEL theory, it admits initial and free models [11]. Hence, the completeness result can be consolidated by stating the equivalence between formulae provable in $Proof(\mathcal{R})$ using MEL deduction rules, formulae holding for all concurrent models of $\mathcal{R}$ and formulae valid in the initial and free concurrent models.

**Theorem 3.3 (Completeness II).** *For $\mathcal{R}$ a* GRT *and for any* MEL *sentence $\varphi$ over $Proof(\mathcal{R})$ (and thus, for $\varphi$ any of the formulae $\alpha : Rw_k$, $\mathsf{s}(\alpha) = t$, $\mathsf{t}(\alpha) = t'$), we have: $Proof(\mathcal{R}) \vdash (\forall X)\ \varphi \quad \Leftrightarrow \quad Proof(\mathcal{R}) \models (\forall X)\ \varphi \quad \Leftrightarrow \quad \mathbb{T}_{Proof(\mathcal{R})}(X) \models (\forall X)\ \varphi$.*

Theorems 3.1, 3.2 and 3.3 can be combined together to state a stronger completeness result for $Proof(\mathcal{R})$, showing the equivalence between deduction at the level of GRTs, their (initial and free) reachability models, and their (initial and free) concurrent models.

By Theorem 2.1 we have that the specialized versions of all our results for GRT over unsorted equational theories without frozen arguments and without equality / membership conditions in rewrite rules coincide with the classical ones. In particular, if $\mathcal{R}$ is an ordinary rewrite theory, any $\mathcal{R}$-system is a concurrent model of the corresponding GRT $\hat{\mathcal{R}}$, because there is a forgetful functor $\mathbf{M}_{\mathcal{R}}$ from the category of $Proof(\hat{\mathcal{R}})$-algebras to the category of $\mathcal{R}$-systems. Indeed, the functor $\mathbf{M}_{\mathcal{R}}$ preserves initial and free models.

## Conclusion

We have defined *generalized rewrite theories* to substantially extend the expressiveness of rewriting logic in many applications. We have given rules of deduction for these theories, defined their models as MEL algebras, and shown that initial and free models exist (for both reachability and true concurrency models). We have also shown that this generalized rewriting logic is complete with respect to its model theory, and that our results generalize the original results for unsorted rewrite theories in [10]. Future work will make more explicit the 2-categorical nature of our model theory, and will develop the semantics of *generalized rewrite theory morphisms*, extending the ideas in [9].

When evaluating the trade-offs between the complexity of the presentation and the expressiveness of the proposed rewrite theories, we have preferred to give the precise foundational semantics for the most general form of rewrite theories used in practice. Although the result suggests that MEL is expressive enough to embed GRTs just as MEL theories plus some syntactic sugar, we argue that the intrinsic separation of concerns in GRTs (i.e., equational vs operational reasoning) is fundamental in most applications.

The theory $\mathit{Proof}(\mathcal{R})$ has an obvious reading as the GRT counterpart of the classic Curry-Howard isomorphism. Along this line of research there is a flourishing literature that focuses on the full integration of type theory with rewriting logic. We just mention the joint work of Stehr with the second author on the formalization of Pure Type Systems in RL [14], and the work of Cirstea, Kirchner and Liquori on the ρ-calculus [1].

*Acknowledgment.* We thank Mark-Oliver Stehr, Peter Ölveczky, and Jean-Bernard Stefani for helping us along the way to frozen operators, and all the members of the Maude team for invaluable insights towards more general notions of rewrite theory. We warmfully thank Miguel Palomino, Narciso Martí-Oliet and the anonymous reviewers for many helpful comments.

## References

1. H. Cirstea, C. Kirchner, and L. Liquori. The Rho Cube. *Proc. FoSSaCS'01*, *LNCS* 2030, pp. 168–183. Springer, 2001.
2. M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and J. Quesada. Maude: Specification and programming in rewriting logic. *TCS* 285:187–243, 2002.
3. R. Diaconescu and K. Futatsugi. *CafeOBJ Report: The language, proof techniques, and methodologies for object-oriented algebraic specification*, *AMAST Series in Computing* volume 6. World Scientific, 1998.
4. S. Lucas. Termination of rewriting with strategy annotations. *Proc. LPAR'01*, *Lect. Notes in Artificial Intelligence* 2250, pp. 669–684. Springer, 2001.
5. N. Martí-Oliet and J. Meseguer. Rewriting logic as a logical and semantic framework. *Handbook of Philosophical Logic* volume 9, pp. 1–87. Kluwer, second edition, 2002.
6. N. Martí-Oliet and J. Meseguer. Rewriting logic: roadmap and bibliography. *Theoret. Comput. Sci.* 285(2):121–154, 2002.
7. N. Martí-Oliet, K. Sen, and P. Thati. An executable specification of asynchronous pi-calculus semantics and may testing in Maude 2.0. *Proc. WRLA'02*, *ENTCS* 71. Elsevier, 2002.
8. N. Martí-Oliet and A. Verdejo. Implementing CCS in Maude 2. *Proc. WRLA'02*, *ENTCS* 71. Elsevier, 2002.
9. J. Meseguer. Rewriting as a unified model of concurrency. Technical Report SRI-CSL-90-02R, SRI International, Computer Science Laboratory, 1990.
10. J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoret. Comput. Sci.*, 96:73–155, 1992.
11. J. Meseguer. Membership algebra as a logical framework for equational specification. *Proc. WADT'97*, *LNCS* 1376, pp. 18–61. Springer, 1998.
12. Protheo Team. The ELAN home page, 2001. www page `http://elan.loria.fr`.
13. M.-O. Stehr, J. Meseguer, and P. Ölveczky. Rewriting logic as a unifying framework for Petri nets. *Unifying Petri Nets*, *LNCS* 2128, pp. 250–303. Springer, 2001.
14. M.-O. Stehr and J. Meseguer. Pure Type Systems in Rewriting Logic. *Proc. LFM'99*. 1999.