

# Modal Logic for Rewriting Theories

Dirk Pattinson<sup>1</sup>

*Institut für Informatik  
Ludwig-Maximilians Universität  
Munich, Germany*

*Email: pattinso@informatik.uni-muenchen.de*

---

## Abstract

We view models of rewrite theories enriched with observations coalgebraically. This allows us on the one hand to use “off the shelf” logics for coalgebras to specify and, on the other hand, to verify properties of rewriting programs and to obtain results about the expressive power of such languages.

---

## 1 Introduction

Rewriting Logic has been proposed as a unifying framework for many different styles of programming language semantics (see [20,19]). The Maude language (see [4,3]) provides an implementation of (a sublanguage of) rewriting logic. This puts forward the question of a language, which can be used to *specify* properties of rewriting programs and, in turn, to verify these properties.

Differing from [17,5], the approach we propose is based on coalgebraic modal logic. By separating the *statical* properties (that is, properties of the algebraic term constructors) from the *dynamical* properties (given by the rewrite rules) of a rewrite theory, we are able to use appropriate logics for the specification of both.

Given a (one-sorted) rewrite theory  $\text{Th} = (\Sigma, E, L, \mathcal{R})$  and a  $(\Sigma, E)$  algebra  $A$ , the set  $\mathcal{R}$  of rewrite rules determines a transition relation  $R \subseteq A \times A$  on the carrier set of  $A$ . Turning this relation into a function  $\gamma : A \rightarrow \mathcal{P}(A)$  by defining  $\gamma(a) = \{a' \in A \mid aRa'\}$ , we obtain a coalgebra structure  $\gamma$  on the carrier of  $A$ .

This suggests that one can specify the *behaviour* of a rewrite theory by using a (modal) logic for coalgebras. Considering the transition relation  $R \subseteq A \times A$  as a Kripke frame, the induced logic is in general not strong enough

---

<sup>1</sup> Research partially supported by the DFG Graduiertenkolleg “Logik in der Informatik” and the DAAD programme “INIDA”

to formulate properties of interest. This is due to the fact that we only have the set  $\{\top, \perp\}$  of truth values as atomic propositions at hand, and formulas which we can build from these and the modal operators  $\Box$  and  $\Diamond$  only separate states (terms) with different termination properties, regardless of the result of the computation.

To overcome this lack of expressivity, we introduce additional *observers*, which allow us to deal with other properties of interest. Modally speaking, observers play the role of atomic propositions in Kripke models or of labels in in labelled transition systems, allowing us to reason about additional properties (different from possible termination at the next step). Eg. modelling bank accounts with rewriting logic, a possible observer for an account is the account balance. So the model of a rewrite theory with one observer takes the form

$$\gamma = \langle s, o \rangle : A \rightarrow \mathcal{P}(A) \times I$$

where  $I$  is the sort representing the integers,  $s$  computes successor states as above and  $o$  is the function which determines the observations.

Note that without the observer, the only property which we could observe (or express logically), is that an operation on bank accounts terminates. Having the observer at hand, we also have atomic propositions allowing us to observe (and, in turn, express logically) the account balance after the operation has terminated.

We thus extend the algebraic specification  $(\Sigma, E)$  with a set of observable sorts and observers protecting the original specification. This allows for equational specification of properties of the observation function  $o$  on the one hand, and for a modal specification of the behaviour of the transition relation on the other hand. In the framework presented, the rules of a rewrite theory have no impact on the structure of the formulas used to specify properties of the rewrite system. Given a set of term constructors, we view the rules as implementation of a system. In this light, it is the task of the specifier to lay down a signature and modal formulas constraining the behaviour of the rewrite rules, and, in turn the task of the implementor to produce rules which satisfy the specification.

The induced (modal) logic is shown to be strong enough to distinguish non-bisimilar elements while on the other hand it is weak enough not to distinguish bisimilar elements.

Apart from this expressivity results, we also show how satisfaction of modal formulas can be proved in a logical system by means of translating the modal formulas into first order logic. Two classes of models are considered: algebras for the underlying equational signature of a rewrite theory and reachable algebras, where every element of the carrier can be denoted by a ground term. In the latter case it turns out that one has to use infinitary logic in order to prove satisfaction of formulas, as it is the case when the language is extended with eventually and always operators.

## 2 Coalgebraic Modal Logic

### 2.1 Background on Coalgebras and Modal Logic

We just give the basic definitions which will be of concern to us in this exposition and refer the reader to [12,23] for a detailed account.

An algebra can be described categorically as a function  $\Sigma A \xrightarrow{\alpha} A$ , where  $\Sigma$  is an endofunctor on the category **Set** of sets. Dually, an  $\Omega$  coalgebra is a function  $C \xrightarrow{\gamma} \Omega C$ , where  $\Omega$  is an endofunctor on **Set**.

**Definition 2.1 (Coalgebras, Morphisms and Bisimulations)** *Suppose  $\Omega : \mathbf{Set} \rightarrow \mathbf{Set}$  is an endofunctor. An  $\Omega$ -coalgebra is a pair  $(C, \gamma)$  with  $C$  a set and  $\gamma : C \rightarrow \Omega C$  a function.*

*Suppose  $(C, \gamma)$  and  $(D, \delta)$  are  $\Omega$ -coalgebras. A coalgebra morphism  $f : (C, \gamma) \rightarrow (D, \delta)$  is a function  $f : C \rightarrow D$ , such that  $\delta \circ f = \Omega f \circ \gamma$ , that is, the diagram*

$$\begin{array}{ccc} C & \xrightarrow{f} & D \\ \downarrow \gamma & & \downarrow \delta \\ \Omega C & \xrightarrow{\Omega f} & \Omega D \end{array}$$

*commutes.*

*A relation  $R \subseteq C \times D$  is a bisimulation, if there exists a transition structure  $\rho : R \rightarrow \Omega R$  such that the projections  $\pi_1 : R \rightarrow C$  and  $\pi_2 : R \rightarrow D$  are coalgebra morphisms  $(R, \rho) \rightarrow (C, \gamma)$  and  $(R, \rho) \rightarrow (D, \delta)$ , respectively.*

*We call two elements  $c \in C$  and  $d \in D$  bisimilar (and denote this by  $c \simeq d$ ), if there exists a bisimulation relating  $c$  and  $d$ .*

In the sequel we will only be concerned with special types of functors, that is, functors  $\Omega$  of the form

$$\Omega Y = \mathcal{P}_f(Y) \times O_1 \cdots \times O_k \times Y^l,$$

where  $\mathcal{P}_f$  is the covariant finite powerset functor (mapping a set to the set of its finite subsets),  $O_1, \dots, O_k$  are (constant) sets and  $Y^l$  denotes the  $l$ -fold cartesian product of  $Y$  for some  $l \in \mathbb{N}$ .

To simplify notation, we write the structure maps  $\gamma : Y \rightarrow \Omega Y$  as  $\gamma = \langle s, (o_i), (p_j) \rangle$ .

Note that a function  $o_i$  maps a state  $y \in Y$  to an element of an ‘‘observable set’’  $O_i$  different from  $Y$ . We therefore understand the value  $o_i(y) \in O_i$  as an *observation* about the state  $y \in Y$  and the function  $o_i$  as an *observer function*. This is not the case with the  $p_j$ ’s, mapping a state  $y \in Y$  to a new state  $p_j(y) \in Y$ . So the function  $p_j$  corresponds to a *state change*. There are two essentially different ways to obtain a new state  $y'$  from a given state  $y \in Y$ : Either by applying the successor function  $s$ , obtaining a set of states (this will correspond to the rewriting process in later applications) or by applying one of the  $p_j$ ’s, obtaining a *single* new state. One can understand the  $p_j$ ’s as functions, which observe something of state type (which can then be turned

into an observation by applying an observer function  $o_i$ ). For this reason, the  $p_j$ 's will later be called *self-observers*.

The notion of bisimulation for functors of this particular shape can be characterised as follows:

**Proposition 2.2 (Characterisation of Bisimulation)** *Suppose  $\Omega Y = \mathcal{P}_f X \times O_1 \cdots \times O_k \times Y^l$  and  $\gamma = \langle s, o_1, \dots, o_k, p_1, \dots, p_l \rangle : Y \rightarrow \Omega Y$  is an  $\Omega$ -coalgebra. A relation  $R \subseteq Y \times Y$  is a bisimulation on  $(Y, \gamma)$  if and only if*

- (i)  $o_i(y_0) = o_i(y_1)$  for all  $(y_0, y_1) \in R$  and  $1 \leq i \leq k$ .
- (ii)  $(p_j(y_0), p_j(y_1)) \in R$  for all  $(y_0, y_1) \in R$  and  $1 \leq j \leq l$ .
- (iii) If  $(y_0, y_1) \in R$ , then for all  $z_0 \in s(y_0)$  there is a  $z_1 \in s(y_1)$  such that  $(z_0, z_1) \in R$ .
- (iv) If  $(y_0, y_1) \in R$ , then for all  $z_1 \in s(y_1)$  there is a  $z_0 \in s(y_0)$  such that  $(z_0, z_1) \in R$ .

The conditions (iii) and (iv) are the well known “zig-zag conditions” of bisimulation in modal logic (see eg. [2], Section 2.10 or [24], Section 5.3).

## 2.2 Multimodal Languages

We give the general definition of multimodal languages and their semantics following [10], Part 1, Section 5. This framework will be instantiated in the sequel to coalgebraic modal logic (in a way similar to [14]) and modal logic for rewriting theories with observations.

**Definition 2.3 (Multimodal Logic – Syntax)** *Let  $\text{AtProp}$ ,  $\text{ModOp}$  be sets of atomic propositions and modalities, respectively. The multimodal language  $\mathcal{L}_{(\text{AtProp}, \text{ModOp})}$  induced by  $\text{AtProp}$ ,  $\text{ModOp}$  is inductively generated by the following clauses:*

- $\perp$  and  $\varphi \in \text{AtProp}$  are formulas of  $\mathcal{L}_{(\text{AtProp}, \text{ModOp})}$ .
- If  $\varphi$  is a formula of  $\mathcal{L}_{(\text{AtProp}, \text{ModOp})}$ , then so is  $[o]\varphi$  for  $o \in \text{ModOp}$ .
- If  $\varphi$  and  $\psi$  are formulas of  $\mathcal{L}_{(\text{AtProp}, \text{ModOp})}$  then so is  $\varphi \rightarrow \psi$ .

We introduce the propositional connectives  $\wedge$ ,  $\vee$  as well as  $\perp$  and  $\neg$  in the usual way and let  $\langle o \rangle = \neg[o]\neg$  for  $o \in \text{ModOp}$ .

**Definition 2.4 (Multimodal logic – Semantics)** *Suppose  $\text{AtProp}$  and  $\text{ModOp}$  are sets of atomic propositions and modalities, respectively.*

A frame (or structure)  $\mathcal{M} = (A, (R_o)_{o \in \text{ModOp}})$  for  $\mathcal{L}_{(\text{AtProp}, \text{ModOp})}$  is given by a carrier set  $A$  and a family of relations  $R_o \subseteq A \times A$  for  $o \in \text{ModOp}$ . Given  $\mathcal{M}$ , a valuation of the propositional variables  $\text{AtProp}$  is a function  $V : A \rightarrow \mathcal{P}(\text{AtProp})$ .

Given a structure  $\mathcal{M} = (A, (R_o)_{o \in \text{ModOp}})$  and a valuation  $V : A \rightarrow \mathcal{P}(\text{AtProp})$ , satisfaction  $(\mathcal{M}, V, a) \models \varphi$  of a formula  $\varphi \in \mathcal{L}_{(\text{AtProp}, \text{ModOp})}$  at a state  $a \in A$  relative to a valuation  $V$  is inductively given by

- $(\mathcal{M}, V, a) \not\models \perp$ .
- $(\mathcal{M}, V, a) \models \varphi$  for  $\varphi \in \text{AtProp}$ , if  $\varphi \in V(a)$
- $(\mathcal{M}, V, a) \models \neg\varphi$  iff  $(\mathcal{M}, V, a) \not\models \varphi$ .
- $(\mathcal{M}, V, a) \models \varphi \rightarrow \psi$  if  $(\mathcal{M}, V, a) \models \psi$  or  $(\mathcal{M}, V, a) \not\models \varphi$ .
- $(\mathcal{M}, V, a) \models [o]\varphi$ , if, for all  $a' \in A$  with  $aR_o a'$ ,  $(\mathcal{M}, V, a') \models \varphi$ .

### 2.3 Coalgebraic Modal Logic

The material presented in this section summarises some results about coalgebras and modal logic and instantiates these results to the framework which will be the topic of discourse later.

Suppose  $\Omega$  is a functor on the category of sets given by  $\Omega Y = \mathcal{P}X \times O_1 \cdots \times O_k \times Y^l$ .

**Definition 2.5 (Induced Language  $\mathcal{L}_\Omega$ )** *The language  $\mathcal{L}_\Omega$  induced by  $\Omega$  is the multimodal language over the set  $\{i : o \mid 1 \leq i \leq k \wedge o \in O_i\}$  of atomic propositions and  $\{\Box, \Box_1, \dots, \Box_l\}$  of modalities.*

The intended meaning of the construct  $i : o$  is that the  $i$ -th observation function produces the result  $o$ .

**Definition 2.6 (Semantics of  $\mathcal{L}_\Omega$ )** *Suppose*

$$\gamma = \langle s, o_1, \dots, o_k, p_1, \dots, p_l \rangle : Y \rightarrow \Omega Y$$

*is an  $\Omega$ -coalgebra (structure).*

*The coalgebra  $(Y, \gamma)$  gives rise to a frame  $\mathcal{M} = (Y, R, (R_j)_{1 \leq j \leq l})$  for the language by defining*

- $yRy'$  iff  $y' \in s(y)$
- $yR_j y'$  iff  $y' = p_j(y)$

*and to a valuation  $V : Y \rightarrow \mathcal{P}(\text{AtProp})$  by  $V(y) = \{i : o \mid o_i(y) = o\}$ .*

*We say that a formula  $\varphi \in \mathcal{L}_\Omega$  is valid at point  $y \in Y$ , iff  $\varphi$  holds in the induced relational structure, that is*

$$(\gamma, y) \models \varphi \quad \text{iff} \quad (\mathcal{M}, V, y) \models \varphi.$$

*If  $(\gamma, y) \models \varphi$ , we also write  $y \models_\gamma \varphi$ .*

### 2.4 Expressive Power of Coalgebraic Modal Logic

**Proposition 2.7 (Invariance under Bisimulation)** *Suppose  $\gamma : Y \rightarrow \Omega Y$  is an  $\Omega$ -coalgebra and  $y_0, y_1 \in Y$  with  $y_0 \rightleftharpoons y_1$  and  $\varphi \in \mathcal{L}_\Omega$ . Then*

$$y_0 \models_\gamma \varphi \quad \text{iff} \quad y_1 \models_\gamma \varphi.$$

**Proof.** By induction on the structure of  $\varphi$  using Proposition 2.2. □

**Proposition 2.8 (Bisimilarity is logical equivalence)** *Suppose  $\gamma = \langle s, (o_i), (p_j) \rangle : Y \rightarrow \Omega Y$  is an  $\Omega$ -coalgebra such that  $s(y)$  is finite for every  $y \in Y$ .*

*If  $y_0, y_1 \in Y$  with  $y_0 \not\sim_\gamma y_1$ , then there exists a formula  $\varphi \in \mathcal{L}_\Omega$  such that  $y_0 \models_\gamma \varphi$  and  $y_1 \not\models_\gamma \varphi$ .*

**Proof.** The proof is essentially a simplification of [22], 4.7 and 4.8. □

### 3 Application to Rewriting Logic

We disregard the notion of labels in Rewrite Theory and focus on one-sorted theories with a single sort  $\sigma_0$ .

For multisorted signatures  $\Sigma$  with sorts  $(\sigma_i)_{i \in I}$  and a family  $X = (X_i)_{i \in I}$  of variables for each sort, we denote the freely generated term algebra by  $T_\Sigma X$  and the set of terms in  $T_\Sigma X$  of sort  $\sigma$  by  $(T_\Sigma X)_\sigma$ .

Given a  $\Sigma$ -algebra  $A$ , we denote the carrier set (or interpretation) of a sort  $\sigma$  of  $\Sigma$  by  $\llbracket \sigma \rrbracket$ . The same notation is used for the interpretation of terms: if  $t \in T_\Sigma X$ , we denote the interpretation of  $t$  in  $A$  wrt. a valuation  $\beta$  of the free variables by  $\llbracket t \rrbracket_A^\beta$ , or (if  $A$  and  $\beta$  are clear from the context) just by  $\llbracket t \rrbracket$ .

#### 3.1 Rewriting Theories under Consideration

We focus on “bare” rewriting logic, as described in [20,19] and on a variant which we call “computational” rewriting logic, which omits the reflexivity and the transitivity rule and can hence be considered as keeping track of the single steps which occur in the rewrite process. To be more exact, we consider rewriting theories  $\text{Th} = (\Sigma, E, \mathcal{R})$  where

- $\Sigma$  is an algebraic signature with a single sort  $\sigma_0$
- $E$  is a set of  $\Sigma$  equations, and
- $\mathcal{R}$  is a set of rewrite rules (see [20], section 2.1).

If we denote the  $E$ -equivalence class of a term  $t$  by  $[t]$ , we say that the set  $\mathcal{R}$  of rules *entails* a sequent  $[t] \rightarrow [t']$  in rewriting logic, if  $[t] \rightarrow [t']$  can be obtained by a finite number of applications of the structural rules of rewriting logic. This is denoted by  $\mathcal{R} \vdash_{\text{RWL}} [t] \rightarrow [t']$ .

Thinking of rewriting logic in terms of executable specifications, as implemented in the Maude language ([4,3]), it is reasonable to keep track of the intermediate states of a computation. We accomplish this by saying that  $\mathcal{R}$  entails a sequent  $[t] \rightarrow [t']$  in *computational rewriting logic*, and denote this by  $\mathcal{R} \vdash_{\text{CRWL}} [t] \rightarrow [t']$ , if the sequent  $[t] \rightarrow [t']$  can be derived by means of a finite number of applications of the rules

$$\text{(Cong)} \quad \frac{[t_{i_1}] \rightarrow [t'_{i_1}] \dots [t_{i_k}] \rightarrow [t'_{i_k}]}{[ft_1 \dots t_n] \rightarrow [ft'_1 \dots t'_n]}$$

for an  $n$ -ary function symbol  $f$  and  $1 \leq i_1 < \dots < i_k \leq n$ ,  $k \geq 1$  and  $t'_j = t_j$  for  $j \notin \{i_1, \dots, i_k\}$ , and

$$\text{(Repl)} \quad \frac{[s_{i_1}] \rightarrow [s'_{i_1}] \dots [s_{i_k}] \rightarrow [s'_{i_k}]}{[t(\bar{s}/\bar{x})] \rightarrow [t'(\bar{s}'/x)]}$$

for every rewrite rule  $[t(x_1, \dots, x_n)] \rightarrow [t'(x_1, \dots, x_n)] \in \mathcal{R}$  and  $\{i_1, \dots, i_k\} \subseteq \{1, \dots, n\}$ , where  $s'_j = s_j$  for  $j \notin \{i_1, \dots, i_k\}$ .

Note that the theory we develop is parametric in the actual notion of rewriting. One could also decree that  $\mathcal{R} \vdash [t] \rightarrow [t']$ , if  $[t] \rightarrow [t']$  is a one-step concurrent (or sequential) rewrite.

### 3.2 Rewriting with Observations and Induced Language

We introduce the concept of observation in the context of rewrite theories. The notion of “observational signature” introduced here is very similar to that considered in [7], who also introduce observations in order to obtain a logic to reason about and specify properties of rewrite theories. We will discuss the difference between the approach taken in loc. cit. and the present one after introducing the modal language  $\mathcal{L}_{(\text{Th}, \mathcal{E})}$ .

**Definition 3.1 (Rewrite Theories with Observations)** *Suppose  $\text{Th} = (\Sigma, E, \mathcal{R})$  is a rewrite theory. An observational extension of  $\text{Th}$  is a pair  $\mathcal{E} = (\Sigma^\circ, E^\circ)$ , where*

- $\Sigma^\circ$  extends  $\Sigma$  with new sorts and unary function symbols
- For all terms  $t_0, t_1 \in T_\Sigma X$ ,  $E^\circ \vdash t_0 = t_1$  iff  $E \vdash t_0 = t_1$ .

We call the pair  $(\text{Th}, \mathcal{E})$  a rewriting theory with observations. In this context, the set

$$\text{Obs}(\text{Th}, \mathcal{E}) = \{f : \sigma_0 \rightarrow \tau \in \Sigma^\circ \mid \tau \neq \sigma_0\}$$

is the set of observers of  $(\text{Th}, \mathcal{E})$  and

$$\text{SelfObs}(\text{Th}, \mathcal{E}) = \{f : \sigma_0 \rightarrow \sigma_0 \in \Sigma^\circ \mid f \notin \Sigma\}$$

the set of self-observers of  $(\text{Th}, \mathcal{E})$ .

While observers tell us directly about the outcome of an experiment, given a state  $s$ , self-observers correspond to experiments, whose result is a new state  $s'$ , which cannot be *directly* observed – one has to apply an observer function in order to obtain a “visible” result. Self-observers occur naturally in many places. Consider eg. a component in a state-based system, which implements an undo-operation. The undo-operation is (necessarily) of type  $S \rightarrow S$  (if  $S$  is the state space of the component). Using the self-observer induced by the undo-operation, one can formulate statements as “after we have performed some operation and (immediately afterwards) un-done it, we end up with a state indistinguishable from the state we set out with.”

**Definition 3.2 (Induced Language  $\mathcal{L}_{(\text{Th}, \mathcal{E})}$ )** Given a rewrite theory with observations  $(\text{Th}, \mathcal{E})$  and a family of variables  $X = (X_\sigma)_{\sigma \in \text{sorts}(\Sigma^o)}$ , the language  $\mathcal{L}_{(\text{Th}, \mathcal{E})}$  induced by  $(\text{Th}, \mathcal{E})$  is the multimodal language over the set

$$\text{AtProp} = \{o : v \mid o : \sigma_0 \rightarrow \sigma \in \text{Obs}(\text{Th}, \mathcal{E}) \wedge v \in (T_\Sigma X)_\sigma\}$$

and the set of modalities

$$\text{ModOp} = \{\Box\} \cup \{\Box_p \mid p \in \text{SelfObs}(\text{Th})\}.$$

**Remark 3.3 (Relationship to Coalgebraic Modal Logic)** Note that apart from a slightly different syntactic presentation (using the name of a function symbol instead of the position it appears in the coalgebraic signature functor), the language induced by a rewrite theory with observations is exactly the language induced by the coalgebraic signature functor

$$\Omega_{(\text{Th}, \mathcal{E})} Y = \mathcal{P}Y \times \prod_{o : \sigma_0 \rightarrow \sigma \in \text{Obs}(\text{Th}, \mathcal{E})} (T_\Sigma X)_\sigma \times \prod_{p \in \text{SelfObs}(\text{Th}, \mathcal{E})} Y.$$

in the sense of definition 2.5.

**Definition 3.4 (Semantics of the induced language)** Suppose  $(\text{Th}, \mathcal{E})$  is a rewriting theory with observations and  $X$  is a family of variables for each sort  $\sigma$  of  $\Sigma$ .

A model of  $(\text{Th}, \mathcal{E})$  is a  $(\Sigma^o, E^o)$ -algebra  $A$  together with a function  $s : \llbracket \sigma_0 \rrbracket \rightarrow \mathcal{P}(\llbracket \sigma_0 \rrbracket)$ , such that whenever  $\mathcal{R} \vdash_{\text{RWL}} [t_0] \rightarrow [t_1]$ , we have that  $\llbracket t_1 \rrbracket^\beta \in s(\llbracket t_0 \rrbracket^\beta)$  for all valuations  $\beta : X \rightarrow A$ .

Given a model  $M = (A, s)$  of  $(\text{Th}, \mathcal{E})$  and a valuation  $\beta : X \rightarrow A$ , we obtain a relational structure  $\mathcal{M} = (\llbracket \sigma_0 \rrbracket, R, (R_j)_{1 \leq j \leq l})$  by decreeing that

- $aRa'$  iff  $a' \in s(a)$
- $aR_p a'$  iff  $a' = \llbracket p \rrbracket(a)$  for  $p \in \text{SelfObs}(\text{Th}, \mathcal{E})$ .

and a valuation  $V : A \rightarrow \mathcal{P}(\text{AtProp})$  given by  $V(a) = \{o : v \mid \llbracket o \rrbracket(a) = \llbracket v \rrbracket^\beta\}$ . Now define validity  $(M, \beta, a) \models \varphi$  of a formula  $\varphi$  of  $\mathcal{L}_{(\text{Th}, \mathcal{E})}$  at the point  $a$  relative to the valuation  $\beta$  by

$$(M, \beta, a) \models \varphi \quad \text{iff} \quad (\mathcal{M}, V, a) \models \varphi.$$

We say that  $(M, \beta, t) \models \varphi$ , if  $(M, \beta, \llbracket t \rrbracket^\beta) \models \varphi$  and  $t \models \varphi$ , if, for all models  $M = (A, s)$  and all valuations  $\beta : X \rightarrow A$ , we have  $(M, \beta, \llbracket t \rrbracket^\beta) \models \varphi$ .

Note that we can use the same definition in order to define the semantics of the induced language with respect to computational rewriting logic (or any other form of rewriting deduction) by simply replacing deduction in rewriting logic  $\vdash_{\text{RWL}}$  by the appropriate rewrite relation.

The approach taken here differs from the one in [7] in the sense that [7] proposes a modal *action* logic, where each action is given by a (suitable combination of) rewrite rules. The approach taken here separates the task of

specification and implementation (that is, providing rewrite rules appropriate for the problem under consideration). Here rewrite rules (and their labels) do not enter into the modal language, thus also allowing different implementations to be compared.

### 3.3 Examples of Rewrite Theories and Modal Formulas

We present two examples of rewrite theories and modal formulas. In the first example, we present a (simple-minded) signature of natural numbers where we express the property that every ground term of that signature can always be rewritten to a normal form. The second example is bank account rewriting in the style of Maude where we assert that the total amount of money deposited in a bank remains constant (viewing a bank as a closed system).

#### *Termination in computational rewriting logic*

In computational rewriting logic without any extra observations, the only thing which can be expressed is termination of the rewrite process. Note that in absence of observations, the language is just built from  $\perp$ , propositional connectives and the  $\Box$ -operator. The property that (the equivalence class) of  $t$  cannot be rewritten any longer can be expressed by the formula  $\Box\perp$ , that is,  $t \models \Box\perp$  if (the equivalence class of)  $t$  has no successor state. The property that rewriting of  $t$  eventually terminates can be characterised by the formulas  $\neg\mathbf{A}\neg\Box\perp$ , using the always operator introduced in Section 5.1.

#### *Evaluation of Terms*

Suppose the algebraic signature  $\Sigma$  consists of a single sort  $N$  and the set of function symbols  $\{0 : \rightarrow N, s : N \rightarrow N, + : N, N \rightarrow N\}$  and the set  $E$  of equations is empty.

The extension  $\Sigma^o$  adds the sort  $B$  and the function symbols  $\{\mathbf{tt} : \rightarrow B, \mathbf{ff} : \rightarrow B, \wedge : B, B \rightarrow B, \mathbf{fe} : N \rightarrow B, \mathbf{cl} : N \rightarrow B\}$ . The intended meaning of the predicate  $\mathbf{fe}$  is to denote whether a term is fully evaluated, which is formalised by the equations  $E^o = \{\mathbf{fe}(0) = \mathbf{tt}, \mathbf{fe}(sx) = \mathbf{fe}(x), \mathbf{fe}(x + y) = \mathbf{ff}\}$ .

Now suppose  $(\text{Th}, \mathcal{E})$  is a rewrite theory with observations which realises the signature  $\Sigma^o$ , that is  $\text{Th} = (\Sigma, E, \mathcal{R})$  and  $\mathcal{E} = (\Sigma^o, E^o)$ , the rules of  $\text{Th}$  being the (recursive) equations for addition, viewed as rewrite rules.

We obtain an atomic formula  $\mathbf{fe} : \mathbf{tt}$  with the property that  $t \models \mathbf{fe} : \mathbf{tt}$ , if the term  $t$  can be denoted by a term in which the symbol “+” does not occur.

Similarly we can axiomatise a predicate  $\mathbf{cl}$  stating that a term is closed by the equations  $\mathbf{cl}(0) = \mathbf{tt}, \mathbf{cl}(sx) = \mathbf{cl}(x)$  and  $\mathbf{cl}(x + y) = \mathbf{cl}(x) \wedge \mathbf{cl}(y)$  and the obvious equations for  $\wedge$ . The property that every closed term has a successor which is fully evaluated can then be expressed by the formula  $\mathbf{cl} : \mathbf{tt} \rightarrow \Diamond\mathbf{fe} : \mathbf{tt}$ . Applying this to rewriting logic, we can only express that a term has at least one successor which is fully evaluated, that is, the rewrite rules  $\mathcal{R}$  are such that every closed term has a successor which is fully evaluated. However, in

computational rewriting logic, the properties that every term will be eventually fully evaluated and that this evaluation terminates can be only expressed by extending the language with an always-operator (see section 5.1).

*Bank account rewriting*

We consider a (simple-minded) specification of concurrent bank account rewriting. The sort we are observing in this example is the sort **Bank**, while the sorts **Account** and **Int** allow for atomic propositions enabling us to express statements about banks (and not just about termination of operations performed on banks). We use a Maude-like syntax and consider the rewrite theory given by

```
(mod ACCOUNT is

  sorts Account Bank Transfer .

  op <_:Account|amount:_) : Int Int -> Account .
  op transfer_from_to_ : Int Int Int -> Transfer .

  op [] : -> Bank .
  op [_] : Account -> Bank .
  op [_] : Transfer -> Bank .
  op _,_ : Bank Bank -> Bank [assoc comm id: [] ] .

  vars A1 A2 K K1 K2 : Int .

  rl [ transfer ] :
    [ < A1 : Account | amount : K1 > ],
    [ < A2 : Account | amount : K2 > ],
    [ transfer K from A1 to A2 ]
  =>
    [ < A1 : Account | amount : K1 - K > ],
    [ < A2 : Account | amount : K2 + K > ] .

  op total : Bank -> Int .

  var A : Int .
  var B : Bank .

  eq total([]) = 0 .
  eq total([ < A : Account | amount : K > ], B) = K + total(B) .
  eq total([ transfer K from A1 to A2 ], B) = total(B) .

endm)
```

Viewing a bank as a closed system, we can thus demand that  $t \models \mathbf{total} : x \rightarrow \Box \mathbf{total} : x$  for every term  $t$  of sort `bank`, expressing that the total amount of money deposited will remain constant.

Here we use the function `total` as an observer, which allows us to express a property of terms of sort `Bank`, revealing information about the term (in the example the total amount of money deposited). This would have not been possible without observers, where the information represented by a term would have been entirely hidden (and thus not usable in the logic).

It can be shown by means of first order translation that the rule given above validates this invariant.

### 3.4 From Models of Rewrite Theories to Coalgebras

It is worthwhile to note that, given a model  $M$  of a rewrite theory with observations, every valuation of the variables induces a coalgebra structure on the carrier set of  $\llbracket \sigma_0 \rrbracket$ . On the other hand, every valuation induces a translation between  $\mathcal{L}_{(\text{Th}, \mathcal{E})}$  and a (suitable) coalgebraic modal logic. This subsection makes this relationship precise.

Assume that  $(\text{Th}, \mathcal{E})$  is a rewrite theory with observations,  $M = (A, s)$  is a model of  $(\text{Th}, \mathcal{E})$  and  $\beta : X \rightarrow A$  is a valuation.

The coalgebraic *signature induced by  $M$*  is given by the functor

$$\Omega_M(Y) = \mathcal{P}(Y) \times \prod_{o:\sigma_0 \rightarrow \sigma \in \text{Obs}(\text{Th}, \mathcal{E})} \llbracket \sigma \rrbracket \times \prod_{p \in \text{SelfObs}(\text{Th}, \mathcal{E})} Y,$$

giving rise to a coalgebra structure

$$\gamma(a) = s(a) \times \prod_{o:\sigma_0 \rightarrow \sigma \in \text{Obs}(\text{Th}, \mathcal{E})} \llbracket o \rrbracket(a) \times \prod_{p \in \text{SelfObs}(\text{Th}, \mathcal{E})} \llbracket p \rrbracket(a).$$

The translation  $\mathcal{L}_{(\text{Th}, \mathcal{E})} \rightarrow \mathcal{L}_{\Omega_M}$  is given by  $(o : v)^{\text{tr}} = o : \llbracket v \rrbracket^\beta$ , and inductive extension to the whole of  $\mathcal{L}_{(\text{Th}, \mathcal{E})}$  following definition 2.3.

The following proposition can be proved by induction on the structure of formulas  $\varphi \in \mathcal{L}_{(\text{Th}, \mathcal{E})}$ :

**Proposition 3.5 (Validity, Coalgebraically)** *Let  $\varphi \in \mathcal{L}_{(\text{Th}, \mathcal{E})}$ . Then*

$$(M, \beta, a) \models \varphi \quad \text{iff} \quad (\gamma, a) \models (\varphi)^{\text{tr}}.$$

### 3.5 Expressive Power, revisited

Building on the work of Section 2.4, we can make the claim stated in the introduction precise and prove that  $\mathcal{L}_{(\text{Th}, \mathcal{E})}$  is strong enough to distinguish non-bisimilar points, while being weak enough in order not to distinguish bisimilar points.

**Theorem 3.6 (Bisimilarity, revisited)** *Suppose  $M = (A, s)$  is a model of a rewrite theory with observations  $(\text{Th}, \mathcal{E})$ .*

- (i) *Suppose  $a_0, a_1 \in \llbracket \sigma_0 \rrbracket$  and  $a_0 \simeq a_1$  with respect to the induced signature  $\Omega_M$ . Then*

$$(M, \beta, a_0) \models \varphi \quad \text{iff} \quad (M, \beta, a_1) \models \varphi$$

*for all  $\varphi \in \mathcal{L}_{(\text{Th}, \mathcal{E})}$ .*

- (ii) *Suppose  $s(a)$  is finite for every  $a \in \sigma_0$  and  $a_0, a_1 \in \llbracket \sigma_0 \rrbracket$  with  $a_0 \not\approx a_1$ . Then there exists a formula  $\varphi \in \mathcal{L}_{(\text{Th}, \mathcal{E})}$  and a valuation  $\beta : X \rightarrow A$  such that*

$$(M, \beta, a_0) \models \varphi \quad \text{and} \quad (M, \beta, a_1) \not\models \varphi.$$

**Proof.** Combine proposition 3.5, proposition 2.7 and proposition 2.8. □

## 4 Validity of Formulas via First Order Translation

So far we have only argued from a semantical point of view, that is, by defining a language and studying its semantical properties. This section demonstrates that we can also give a formal system which allows us to actually prove validity  $t \models \varphi$  of a formula at a particular term (viewed as a state of a system) within first order logic by translating  $\mathcal{L}_{(\text{Th}, \mathcal{E})}$  into first order logic following [1], Section 2.1, and using the completeness theorem.

We do this for two classes of models under consideration: algebras with a transition structure as introduced in 3.4 and *reachable* algebras, that is, algebras where every element of the carrier can be denoted by a ground term. Considering the second class of models, it turns out that we need infinitary logic to prove validity of formulas.

### 4.1 First Order Translation

**Definition 4.1 (First Order Translation)** *We consider the (multisorted) first order language  $\mathcal{L}_{(\widehat{\text{Th}}, \mathcal{E})}$  which consists of the sorts and function symbols of  $\Sigma^o$  and a binary relation  $R : \sigma_0, \sigma_0$ . Given a term  $t \in (T_{\Sigma}X)_{\sigma_0}$  the first order translation  $\varphi[t]$  of  $\varphi \in \mathcal{L}_{(\text{Th}, \mathcal{E})}$  is defined by induction on  $\mathcal{L}_{(\text{Th}, \mathcal{E})}$  as follows:*

- $(o : v)[t] \equiv ot = v$  for  $o \in \text{Obs}(\text{Th}, \mathcal{E})$ .
- $(\neg\varphi)[t] \equiv \neg(\varphi[t])$ ,  $(\varphi \rightarrow \psi)[t] \equiv \varphi[t] \rightarrow \psi[t]$ .
- $(\Box\varphi)[t] \equiv \forall y. tRy \rightarrow \varphi[y]$ , assuming that  $y$  is neither free in  $t$  nor in  $\varphi$ .
- $(\Box_p\varphi)[t] \equiv \varphi[pt]$  for  $p \in \text{SelfObs}(\text{Th})$ .

Depending on the notion of rewriting under consideration, one has to axiomatise the properties of the rewrite relation by different sets of first-order formulas. We do not make this axiomatisation explicit and take the set

$$\Phi_{\text{RWL}} = \{tRt' \mid \mathcal{R} \vdash [t] \rightarrow [t']\} \cup E^o$$

as given. Note that entailment “ $\vdash$ ” above can correspond to different notions of rewriting, take for example deduction in rewriting logic or deduction in computational rewriting logic. This enables us to prove

**Theorem 4.2** *Let  $t \in (T_\Sigma X)_{\sigma_0}$  and  $\varphi \in \mathcal{L}_{(\text{Th}, \mathcal{E})}$ . Then*

$$t \models \varphi \quad \text{iff} \quad \Phi_{\text{RWL}} \vdash \varphi[t]$$

where “ $\vdash$ ” is entailment in first order logic.

**Proof.** Remember that  $t \models \varphi$  means that  $(M, \beta, \llbracket t \rrbracket^\beta) \models \varphi$  for every model  $M$  of  $(\text{Th}, \mathcal{E})$ . By completeness of first order logic, we can reduce the statement to showing that  $t \models \varphi$  iff  $\Phi_{\text{RWL}} \models \varphi[t]$ . Note that every model  $M = (A, s)$  of  $(\text{Th}, \mathcal{E})$  induces a (first order) structure  $\hat{M}$  for  $\mathcal{L}_{(\hat{\text{Th}}, \mathcal{E})}$  by defining  $a \llbracket R \rrbracket a'$  iff  $a' \in s(a)$  (and keeping the interpretation of the sorts and function symbols).

The claim is proved by induction on the structure of  $\varphi$ , following the definition of validity in first order logic.  $\square$

#### 4.2 Validity in reachable models

We call a model  $M = (A, s)$  of a rewrite theory with observations *reachable*, if the algebra  $A$  is reachable, that is, every point in  $A$  is denotable by a ground term. Having models consisting only of denotable terms, we can pass to a simpler logic while retaining the same expressive power (cf theorem 3.6).

#### **Proposition 4.3 (Expressiveness with respect to reachable models)**

*Let  $M = (A, s)$  be a reachable model of a rewrite theory with observations  $(\text{Th}, \mathcal{E})$  and let  $\mathcal{L}_{(\text{Th}, \mathcal{E})}^g$  be the sublanguage of  $\mathcal{L}_{(\text{Th}, \mathcal{E})}$  which has the same modalities but where the atomic formulas are of the form  $o : v$  with  $v$  ground. Then bisimilar points of  $\llbracket \sigma_0 \rrbracket$  cannot be distinguished by formulas of  $\mathcal{L}_{(\text{Th}, \mathcal{E})}^g$  while non-bisimilar points can be distinguished in case  $s(a)$  is finite for every  $a \in \llbracket \sigma_0 \rrbracket$ .*

**Proof.** The first part is obvious. For the second part use the fact that different observations from a state  $a \in \llbracket \sigma_0 \rrbracket$  can be distinguished by observations  $o : v$  where  $v$  is ground.  $\square$

Since the class of reachable models is smaller than the class of models for a particular rewrite theory with observations, one expects that the set of formulas which is valid is larger, so the correspondence between provability in first order logic and validity can no longer be maintained.

If we extend the notion of entailment of first order logic by a rule saying that “universal formulas hold, if they hold for ground terms,” then the correspondence can be maintained. For details on the (infinitary) language  $\mathcal{L}_{\omega_1\omega}$  see [21,18].

To this end, we let

$$\Phi_{\text{RWL}}^\infty = \Phi_{\text{RWL}} \cup \bigwedge_{t \text{ ground}} \varphi(t/x) \rightarrow \forall x.\varphi(x),$$

implicitly assuming that only terms with the correct sorts are substituted, and define

$$t \models_{\text{Gen}} \varphi$$

iff  $(M, \beta, \llbracket t \rrbracket) \models \varphi$  for all reachable models  $M$  of  $\text{Th}$ .

**Theorem 4.4 (First order translation and reachable models)** *Let  $t \in (T_\Sigma X)_{\sigma_0}$  and  $\varphi \in \mathcal{L}_{(\text{Th}, \mathcal{E})}$ . Then*

$$t \models_{\text{Gen}} \varphi \quad \text{iff} \quad \Phi_{\text{RWL}}^\infty \vdash \varphi[t]$$

where “ $\vdash$ ” is entailment in first order infinitary logic  $\mathcal{L}_{\omega_1\omega}$ .

**Proof.** We use the omitting types theorem (see [21], theorem 3.5.1 or [13], theorem 6.15 for reference) and adapt the idea of the proof of [25], theorem 3.1.4. Showing that  $\Phi_{\text{RWL}}^\infty \vdash \varphi[t]$  implies that  $t \models_{\text{Gen}} \varphi$ , one just needs to show the validity of the infinitary axiom, which is easy. Now assume that  $t \models_{\text{Gen}} \varphi$  and assume for a contradiction that  $\Phi_{\text{RWL}}^\infty \not\vdash \varphi[t]$ . Then  $\Phi_{\text{RWL}}^\infty \cup \{\neg\varphi[t]\}$  is consistent and one can show that  $\Phi_{\text{RWL}}^\infty \cup \{\neg\varphi[t]\}$  locally omits the set

$$O = \bigcup_{\sigma \in \Sigma} \{x^\sigma \neq t \mid t \text{ ground term of sort } \sigma\}$$

Thus, by the omitting types theorem,  $\Phi_{\text{RWL}}^\infty \cup \{\neg\varphi[t]\}$  has a model which omits  $O$ , that is, a reachable model. From this model we can construct a model  $\mathcal{M} = (A, s)$  of  $(\text{Th}, \mathcal{E})$  by turning the interpretation of  $R$  into a function  $\llbracket \sigma_0 \rrbracket \rightarrow \mathcal{P}[\llbracket \sigma_0 \rrbracket]$ . By construction, we obtain a model which validates  $\neg\varphi$ , and we have reached a contradiction.  $\square$

### 4.3 Proof principles for bisimulation

Having established means to prove the validity of modal formulas at a particular state  $t \in (T_\Sigma X)_{\sigma_0}$ , we can also formulate a proof principle which allows us to establish that (the interpretation of) two terms  $t, t'$  are bisimilar in every model, namely when their first order translations  $\varphi[t]$  and  $\varphi[t']$  are equivalent in first order logic. This can be seen as a modal analogue of observational equality ([11], Definition 3.5) or of behavioural satisfaction ([8], Definition 5).

Let  $(\text{Th}, \mathcal{E})$  be a rewrite theory with observations. We call two terms  $t, t' \in (T_\Sigma X)_{\sigma_0}$  *bisimilar*, if, for every model  $\mathcal{M} = (A, s)$  of  $(\text{Th}, \mathcal{E})$  and every valuation  $\beta : X \rightarrow A$ , we have  $\llbracket t_0 \rrbracket^\beta \simeq \llbracket t_1 \rrbracket^\beta$ , where bisimilarity is wrt the induced coalgebraic signature (see section 3.4). Using this definition, we can prove

**Proposition 4.5 (Proof Principle for Bisimulation)** *Suppose  $t, t' \in (T_{\Sigma}X)_{\sigma_0}$ . Then  $t_0$  is bisimilar to  $t_1$  if and only if*

$$\forall \varphi \in \mathcal{L}_{(\text{Th}, \mathcal{E})}. \Phi_{\text{RWL}} \vdash \varphi[t_0] \iff \Phi_{\text{RWL}} \vdash \varphi[t_1].$$

**Proof.** Use theorem 3.6 and theorem 4.2. □

## 5 Extensions of the theory

This section sketches two extensions to the theory. Viewing rewriting logic as an executable specification language (as modelled by the rule system of computational rewriting logic, see section 3.1, the transition relation on states captures stepwise execution, that is, it is not reflexive. In order to express safety properties, we extend the language with an always-operator (which is not needed in rewriting logic deduction). In the second part, we show how the coalgebraic setting can be adapted to incorporate the case of multisorted rewrite theories.

### 5.1 Eventually and Always Operators

Let  $(\text{Th}, \mathcal{E})$  be a rewrite theory with observations.

**Definition 5.1 (Enrichment with always-operator)** *The language  $\mathcal{L}_{(\text{Th}, \mathcal{E})}^{\text{A}}$  is obtained from  $\mathcal{L}_{(\text{Th}, \mathcal{E})}$  by closing  $\mathcal{L}_{(\text{Th}, \mathcal{E})}$  under the rule*

- *If  $\varphi \in \mathcal{L}_{(\text{Th}, \mathcal{E})}^{\text{A}}$ , then so is  $\text{A}\varphi$ .*

*Given a model  $M$  of  $(\text{Th}, \mathcal{E})$  and  $a \in \llbracket \sigma_0 \rrbracket$ , we define  $a \models \text{A}\varphi$ , iff  $a \models \Box^n \varphi$  for all  $n \in \mathbb{N}$ ,  $n > 0$ .*

*The infinitary first order translation  $(\text{A}\varphi)[t]$  of  $\text{A}\varphi$ , given  $t \in (T_{\Sigma}X)_{\sigma_0}$  is the infinitary formula  $\bigwedge_{n \in \mathbb{N}} \Box^n(\varphi[t])$ .*

Since bisimulation can be characterised by means of logical equivalence (see theorem 3.6), we obtain immediately

**Proposition 5.2 (Expressive Power of  $\mathcal{L}_{(\text{Th}, \mathcal{E})}^{\text{A}}$ )** *Suppose  $M = (A, s)$  is a model of a rewrite theory  $(\text{Th}, \mathcal{E})$  with observations. If  $a_0, a_1 \in \llbracket \sigma_0 \rrbracket$  with  $a_0 \simeq a_1$ , then  $a_0$  and  $a_1$  satisfy the same formulas of  $\mathcal{L}_{(\text{Th}, \mathcal{E})}^{\text{A}}$ .*

Since the proof of the formula  $\text{A}\varphi$  involves reasoning about a possibly infinite number of successor states, this property cannot be expressed in first order logic any more. We can however extend definition 4.1 in such a way that we can prove  $\text{A}\varphi$  in  $\mathcal{L}_{\omega_1\omega}$ :

**Proposition 5.3 (Validity in the first order translation)** *Let  $t \in \mathcal{L}_{(\text{Th}, \mathcal{E})}^{\text{A}}$  and  $\varphi \in \mathcal{L}_{(\text{Th}, \mathcal{E})}^{\text{A}}$ . Then*

$$t \models \varphi \text{ iff } \Phi_{\text{RWL}} \vdash \varphi[t] \quad \text{and} \quad t \models_{\text{Gen}} \varphi \text{ iff } \Phi_{\text{RWL}}^{\infty} \vdash \varphi[t]$$

*where entailment “ $\vdash$ ” is entailment in the infinitary logic  $\mathcal{L}_{\omega_1\omega}$ .*

**Proof.** By 4.2, 4.4 and the completeness theorem for  $\mathcal{L}_{\omega_1\omega}$ , see [21], theorem 3.2.1.  $\square$

## 5.2 Multisorted Rewrite Theories

In order to apply the theory developed to multisorted rewrite theories, we first have to fix the notion of observation, which is done as in the one-sorted case. An *observational extension* of a multisorted rewrite theory is given by an extension of the underlying signature by new sorts, unary function symbols and a set of equations, which conservatively extend the original signature. However, one has to pay attention when considering the notions of observational formulas, in order to avoid that an observer  $o : \sigma \rightarrow \tau$  makes the algebraic structure of terms of sort  $\tau$  explicit.

### Definition 5.4 (Coalgebraic Modal Logic for multisorted Theories)

Suppose  $(\text{Th}, \mathcal{E})$  is an rewrite theory with observations. We define sets of formulas  $\mathcal{L}_{(\text{Th}, \mathcal{E})}^\sigma$  for each sort of  $\sigma$  by mutual induction:

- $\perp$  is a formula of  $\mathcal{L}_{(\text{Th}, \mathcal{E})}^\sigma$ .
- If  $o : \sigma \rightarrow \tau \in \text{Obs}(\text{Th}, \mathcal{E})$ , then  $\tau \notin \Sigma$  and  $v \in (T_\Sigma X)_\tau$ . then  $o : v$  is a formula of  $\mathcal{L}_{(\text{Th}, \mathcal{E})}^\sigma$ .
- if  $\varphi, \psi \in \mathcal{L}_{(\text{Th}, \mathcal{E})}^\sigma$ , then so are  $\Box_\sigma \varphi$  and  $\varphi \rightarrow \psi$ .
- If  $p : \sigma \rightarrow \tau \in \Sigma^o \setminus \Sigma$ ,  $\tau \in \Sigma$  and  $\varphi \in \mathcal{L}_{(\text{Th}, \mathcal{E})}^\tau$ , then  $\Box_p \varphi \in \mathcal{L}_{(\text{Th}, \mathcal{E})}^\sigma$ .

Every formula  $\varphi \in \mathcal{L}_{(\text{Th}, \mathcal{E})}^\sigma$  can, in the light of the first order translation, be viewed as a formula with one free variable of sort  $\sigma$ . Note the typing of the formulas which occur after the  $\Box_p$  for  $p : \sigma \rightarrow \tau \notin \Sigma$ .

Semantically, we consider models of the multisorted theory  $\mathcal{L}_{(\text{Th}, \mathcal{E})}$ , which satisfy the rewrite rules on a “sort-by-sort” basis, hence every formula of  $\mathcal{L}_{(\text{Th}, \mathcal{E})}^\sigma$  asserts properties of the rewrite process at sort  $\sigma$ . The notion of bisimulation we consider is then bisimulation “at sort  $\sigma$ ”, which can be defined as section of the global bisimulation relation  $\leftrightarrow \subseteq (\prod_{\sigma \in \Sigma} \llbracket \sigma \rrbracket)^2$  for any model of  $(\text{Th}, \mathcal{E})$ , by definition 2.1 (which can be uniformly lifted to the category  $\mathbf{Set}^n$ ). The expressivity results obtained in this framework then refer to this notion of bisimulation and can be shown essentially along the same lines as for one-sorted theories.

## 6 Conclusions and related work

The aim of this study was to present a logical framework which allows to reason about the actual process of deduction in rewriting logic. We have presented the concept of “rewrite theory with observations” which allows for what one might call behavioural specification of rewrite theories. Taking the often emphasised analogy *state*  $\leftrightarrow$  *term* seriously, we adopted a modal approach which allows for a language whose expressive power can be characterised to be at the right

level of abstraction (theorem 3.6) in the sense that observably equivalent states cannot be distinguished. Validity of observational formulas was characterised by means of a first order translation, which is parametric in the actual notion of rewriting employed, so the approach can be applied to “pure” rewriting deduction and to actual implementations of rewrite theories, where one clearly does not have the reflexivity and transitivity rule.

In contrast to [5], we do not focus on a particular setting (concurrent rewriting of object configurations), but instead tried to be as general as possible. We also emphasise the *logical* properties over the *computational* properties of a rewrite theory in the sense that we do not assume a specific model of concurrency.

The approach taken in [16,15] is closer to ours. There also coalgebraic techniques are employed to deal with state based systems. The main difference to the approach proposed here is that we mainly rely on models to give semantics to modal formulas and therefore achieve to prove expressivity results. Also, we do not restrict ourselves to object systems.

Similarities and dissimilarities to [7] have already been discussed in section 3.2. Summarising, one can say that the logic of [7] incorporates the rules of a rewrite theory into the logic (they give rise to the action terms), whereas the approach taken here disregards the rules as far as specification is concerned.

Finally we compare our approach to that of [6]. There, hidden algebra techniques and coinduction are used to formulate (and prove) properties of states. Although there is a close relationship between hidden algebra and coalgebra, we think that viewing models of rewrite theories enriched with observations as coalgebras has two slight advantages: First, we can use the by now well developed machinery of coalgebraic modal logic, and secondly, we can employ the notion of (coalgebraic) bisimulation to prove expressivity results for the logics under consideration. Note however, that the model theory used in [6] is more general than the one presented here, most notably we have not investigated whether languages (and models) induced satisfy the satisfaction condition for institutions ([9]). We leave this to further research.

## 7 Acknowledgements

The idea of this paper goes back to a visit to the department of computer science of Lisbon university made possible by a DAAD grant within the “INIDA” programme. The author would like to acknowledge the fruitful discussions with José Fiadeiro on this topic. Acknowledgements are also due to Alexander Knapp for discussions about rewriting logic and Alexander Kurz for many ideas about coalgebras and modal logic.

## References

- [1] H. Andreka, J. van Benthem, and I. Nemeti. Back and forth between modal and classical logic. *J. of the IGPL*, 3(5):685–720, 1995.
- [2] Alexander Chagrov and Michael Zakharyashev. *Modal Logic*, volume 35 of *Oxford Logic Guides*. Oxford Science Publications, 1997.
- [3] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and J. Quesada. A maude tutorial. Draft, March 2000. Available at <http://maude.csl.sri.com/papers/>.
- [4] M. Clavel, S. Eker, P. Lincoln, and J. Meseguer. Principles of maude. In *Proc. 1st Intl. Workshop on Rewriting Logic and its Applications*, Electronic Notes in Theoretical Computer Science, 1996.
- [5] Grit Denker. From rewrite theories to temporal logic theories. In *Proc. 2nd Intl. Workshop on Rewriting Logic and its Applications*, Electronic Notes in Theoretical Computer Science. Elsevier, 1998.
- [6] Razvăn Diaconescu. Foundations of behavioural specification in rewriting logic. *Electronic Notes in Theoretical Computer Science*, 4, 1996. Proc. First International Workshop on Rewriting Logics and its applications.
- [7] J. Fiadeiro, T. Maibaum, N. Martí-Oliet, J. Meseguer, and I. Pita. Towards a verification logic for rewriting logic. In *Recent Trends in Algebraic Development Techniques, WADT'99, France, Selected Papers*, volume 1827 of *Lecture Notes in Computer Science*. Springer, 1999.
- [8] J. Goguen and G. Malcolm. A hidden agenda. Technical Report CS97-538, UCSD, 1997.
- [9] Joseph Goguen and Rod Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39(1), 1992.
- [10] Robert Goldblatt. *Logics of Time and Computation*, volume 7 of *CSLI Lecture Notes*. Center for the Study of Language and Information, Stanford University, 1992. Second Edition.
- [11] R. Hennicker and M. Bidoit. Observational logic. In *Proc. AMAST '98, 7th International Conference on Algebraic Methodology and Software Technology*, number 1548 in *Lecture Notes in Computer Science*. Springer, 1999.
- [12] B. Jacobs and J. Rutten. A tutorial on (co)algebras and (co)induction. *EATCS Bulletin*, 62, 1997.
- [13] H. J. Keisler. Fundamentals of model theory. In J. Barwise and S. Feferman, editors, *Handbook of Mathematical Logic*. North Holland, 1977.
- [14] Alexander Kurz. Specifying coalgebras with modal logic. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *Coalgebraic Methods in Computer Science*

- (*CMCS'98*), volume 11 of *Electronic Notes in Theoretical Computer Science*, pages 57–72, 1998.
- [15] U. Lechner. Object-oriented specifications of distributed systems in the  $\mu$ -calculus and maude. *Electronic Notes in Theoretical Computer Science*, 4, 1996. Proc. First International Workshop on Rewriting Logics and its applications.
  - [16] U. Lechner. *Object-Oriented Specification of Distributed Systems*. PhD thesis, University of Passau, 1997.
  - [17] C. Lengauer and U. Lechner. Modal  $\mu$ -maude. In B. Freitag, C.B. Jones, C. Lengauer, and H.-J. Scheck, editors, *Object Orientation with Parallelism and Persistence*. Kluwer, 1996.
  - [18] M. Makkai. Admissible sets and infinitary logic. In J. Barwise and S. Feferman, editors, *Handbook of Mathematical Logic*. North Holland, 1977.
  - [19] Narciso Martí-Oliet and José Meseguer. Rewriting logic as a logical and semantic framework. In *Proc. 1st Intl. Workshop on Rewriting Logic and its Applications*, *Electronic Notes in Theoretical Computer Science*. Elsevier, 1996.
  - [20] José Meseguer. Research directions in rewriting logic. In U. Berger and H. Schwichtenberg, editors, *Computational Logic*, NATO Advanced Study Institute. Springer, 1998.
  - [21] M. Nadel.  $\mathcal{L}_{\omega_1\omega}$  and admissible fragments. In J. Barwise and S. Feferman, editors, *Model-Theoretic Logics*, *Perspectives in Mathematical Logic*. Springer, 1985.
  - [22] Martin Rößiger. Coalgebras and modal logic. In *Coalgebraic Methods in Computer Science (CMCS'00)*, volume 33 of *Electronic Notes in Theoretical Computer Science*, 2000.
  - [23] Jan Rutten. Universal coalgebra: A theory of systems. *Theoretical Computer Science*. To appear.
  - [24] Colin Stirling. Modal and temporal logics. In S. Abramsky, D. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2. Oxford Science Publications, 1992.
  - [25] M. Wirsing. Algebraic specification. In J. van Leuween, editor, *Handbook of Theoretical Computer Science*. Elsevier, 1990.