Functorial Semantics of Rewrite Theories*

José Meseguer

University of Illinois at Urbana-Champaign, USA
meseguer@cs.uiuc.edu

Abstract. This paper develops a close analogy between Lawvere's functorial semantics of equational theories [21], and a similar 2-functorial semantics for rewrite theories, which specify concurrent systems and whose models are "true concurrency" models of such systems. This has the advantage of unifying within a single 2-functorial framework both models and rewrite theory morphisms. Such morphisms are used in Maude to "put rewrite theories together" in different constructions, including parameterized rewrite theory specifications.

1 Introduction

We owe to Lawvere [21], Bénabou [2], and to Eilenberg and Moore [14] the important insight that algebraic semantics can be developed not just on the category **Set** of sets and functions, but on general categories satisfying minimal requirements. This insight has been exploited in computer science since the 1970s, leading to many fruitful extensions of algebraic semantics beyond the traditional universal algebra of algebraic data types. Since Hartmut Ehrig has made fundamental contributions to both the original theory of algebraic data types and to extending algebraic semantics in new directions, including, for example, graph rewriting, and Petri-net based concurrent system specifications, it seems appropriate for this occasion to discuss the way in which the semantics of rewriting logic [25] is a form of universal algebra on the 2-category **Cat**, the same way that the semantics of traditional algebraic specifications is furnished by universal algebra on the category **Set**.

Of course, in traditional algebraic semantics we are not only interested in the categories of models (the algebras) but on "putting theories together" in a categorical way as originally proposed by Burstall and Goguen in [6] and further developed in the work of Hartmut Ehrig and other researchers (see his two-volume work with Mahr [12, 13] and references there). Theories are "put together" by means of categorical constructions in the category **EqtlTh** of equational theories and equational theory morphisms. Although the most common description of **EqtlTh** is in terms of theory *presentations*, the most elegant and presentation-independent way of describing **EqtlTh** is as the category of Lawvere-Bénabou theories [21, 2]. This has the conceptual advantage of unifying the semantics of the category of theories **EqtlTh** and that of the difference

* Research supported by ONR Grant N00014-02-1-0715 and NSF Grant CCR-0234524.

H.-J. Kreowski et al. (Eds.): Formal Methods (Ehrig Festschrift), LNCS 3393, pp. 220–235, 2005.

categories of algebras within a single *functorial semantics* framework, so that a theory morphism $H : T \longrightarrow T'$ becomes a product-preserving functor and can therefore be regarded as a T-algebra interpreted in the target Lawvere theory T'. In a completely similar way, rewrite theories can be viewed as Lawvere 2-theories and form a category **RWTh**, so that rewrite theory morphisms and models of rewrite theories (which intuitively correspond to "true concurrency" models of the concurrent system specified by the given rewrite theory) are again unified within a single 2-functorial semantics as 2-product preserving 2-functors. Furthermore, the semantics of a parameterized module as the left adjoint of the forgetful functor associated to the inclusion of the parameter theory into the body theory typical of algebraic specifications has an exact analogue for parameterized rewrite theories (supported for example by the Maude language [7, 11]) as a corresponding left adjoint of the forgetful functor in the 2-functorial semantics.

This paper is mostly based on two appendices of the SRI Technical report [24] now not easily available. The paper is organized as follows. The inference rules and model theory of (unconditional) rewriting logic is summarized in Section 2. The functorial semantics of rewrite theories is then presented in Section 3. Rewrite theory morphisms and parameterization are then discussed in Section 4. I then discuss related work and give some concluding remarks in Section 5.

2 Rewriting Logic and Its Models

2.1 Inference Rules and Their Meaning

A *signature* in rewriting logic is an equational theory[1] (Σ, E), where Σ is an equational signature and E is a set of Σ-equations. Rewriting will operate on equivalence classes of terms modulo E. In this way, we free rewriting from the syntactic constraints of a term representation and gain a much greater flexibility in deciding what counts as a *data structure*; for example, string rewriting is obtained by imposing an associativity axiom, and multiset rewriting by imposing associativity and commutativity. Of course, standard term rewriting is obtained as the particular case in which the set of equations E is empty. Techniques for rewriting modulo equations have been studied extensively [10] and can be used to implement rewriting modulo many equational theories of interest.

Given a signature (Σ, E), *sentences* of rewriting logic are sequents of the form

$$[t]_E \longrightarrow [t']_E,$$

where t and t' are Σ-terms possibly involving some variables X, and $[t]_E$, or $[t]$ for short, denotes the equivalence class of the term t modulo the equations E, that is, an element of the free (Σ, E)-algebra $T_{\Sigma,E}(X)$ generated by the variables X. A *rewrite theory* $\mathcal{R}$ is a 4-tuple $\mathcal{R} = (\Sigma, E, L, R)$ where Σ is a ranked alphabet

[1] Rewriting logic is parameterized by the choice of its underlying equational logic, that can be unsorted, many-sorted, order-sorted, membership equational logic, and so on. To ease the exposition I give an *unsorted* presentation.