Rewriting the Conditions in Conditional Rewriting

Neil Ghani
University of Leicester
ng13@mcs.le.ac.uk

Christoph Lüth
Universität Bremen
cxl@informatik.uni-bremen.de

Stefan Kahrs
University of Canterbury
S.Kahrs@ukc.ac.uk

August 30, 2000

Abstract

Category theory has been used to provide a semantics for term rewriting systems at an intermediate level of abstraction between the actual syntax and the relational model. Recently we have developed a semantics for TRSs using monads which generalises the equivalence between algebraic theories and finitary monads on the category **Set**. This semantics underpins the recent categorical proofs of state-of-the-art results in modular rewriting.

We believe that our methods can be applied to modularity for conditional rewriting where several open problems exist. Any results we achieve here would be highly significant as, for the first time, substantial open problems in rewriting would have been solved using categorical techniques. This paper reports on the first step in this project, namely the construction of a semantics for CTRS using monads. [1]

1 Introduction

Term Rewriting Systems (TRSs) are widely used throughout computer science as they form an abstract model of computation while retaining a relatively simple and concrete syntax. TRSs originally arose in the study of the λ-calculus and combinatory logic and have, more recently, played a prominent role in the design and implementation of functional programming languages and in the theory of algebraic data types. In a different direction, TRSs are widely used in formal reasoning as TRSs which are *strongly normalising* and *confluent* provide decision procedures for equational theories.

A TRS consists of a set of rules for generating transformations, or *rewrites*, of the terms of a formal language and intuitively each rewrite is conceptually simple and easy to reason about while combinations of rewrites can model complex phenomena. The concreteness of TRSs has led to a tendency to concentrate on the technical details of specific problems to the detriment of a wider understanding of the subject and a proper interaction with other disciplines in computer science. Abstract models of TRSs have been developed to

[1] Technical Report No. 2000/89, Department of Mathematics & Computer Science, University of Leicester, University Road, Leicester LE1 7RH, England.

address this problem with the simplest being the *Abstract Reduction Systems* (ARSs) based upon relations. While useful for certain results, eg Newmann's Lemma and the Commutation Lemma, ARSs do not posses enough structure to adequately model key concepts such as substitution, context, layer structure etc which arise in more complex problems such as modular rewriting. Hence ARSs are mainly used as an organisational tool with the difficult results proved directly at the syntactic level.

Category theory has been used to provide a semantics for term rewriting systems at an intermediate level of abstraction between the actual syntax and the relational model. Research originally focused on structures such as *2-categories* [16, 17], *Sesqui-categories* [18] and *confluent categories* [3]. However, despite some one-off results [2, 16], these approaches have yet to make a lasting impact on term rewriting. Part of the problem seems to be that the semantics is actually constructed "set-theoretically" and hence reasoning about the semantics often involves set-theoretic reasoning. This often looks like translating the problem back into the syntax!

An alternative approach starts from the observation that term rewriting systems can be regarded as generalisations of algebraic theories. Syntactically, algebraic theories and TRSs are both presented via signatures — the former declares term constructors while TRSs declare term constructors and rewrite rules. Semantically, algebraic theories are modelled by sets with extra structure while TRSs are modelled by pre-orders with extra structure. Sice the categorical treatment of algebraic theories is based upon finitary monads on the category **Set**, it is natural to model a TRS by a monad over a more structured base category. Lüth and Ghani have used this monadic semantics to give categorical proofs of the modularity of confluence and strong normalization [9, 8].

Conditional term rewriting systems (CTRSs) are generalisations of TRSs which allow restrictions to be imposed on when a rewrite is permissible. An example of a conditional rewrite rule is the following

$$l \Rightarrow^* r \vdash s \Rightarrow t$$

which allows s to rewrite to t *providing we have already established that* l *rewrites to* r. CTRSs arise naturally in the semantics of logic and functional programming languages and in structural operational semantics. We believe that our proofs of modularity for TRSs using monads can be generalised to CTRSs where several open problems exist. Any results we achieve would be highly significant as, for the first time, substantial open problems in rewriting would have been solved using categorical techniques. Towards our overall goal, this paper asks the question *can conditional term rewriting systems be modelled by monads*. The main contributions of this paper are:

- We answer the above question affirmatively. This is rather unexpected as, at first glance, CTRSs have the flavor of *essentially algebraic theories* which are strictly more general than theories modelled by monads. The key idea is to replace monads over **Pre** with monads over ordered Σ-algebras (where Σ is the signature of the CTRS). The conditionality in CTRSs is then replaced by the use of more complex arities for operations which are afforded by the use of this more complex base category.

- The usual semantics of a CTRS is the rewrite relation generated by the CTRS.

Unfortunately this semantics is *not* compositional as the following example shows. Let A, B, C, D be constants. Consider the following conditional rewrite rules

$$\begin{array}{lll} R_1: & & \vdash A \Rightarrow B \\ R_2: & A \Rightarrow^* B & \vdash C \Rightarrow D \\ R_3: & A \Rightarrow^* C & \vdash C \Rightarrow D \end{array}$$

Then R_2 and R_3 have the same interpretations, ie the empty relation. But the CTRS $R_1 \cup R_2$ has $C \Rightarrow D$, while the CTRS $R_1 \cup R_3$ does not. Such a non-compositionality is clearly undesirable. Most strikingly, a non-compositional semantics makes a compositional investigation of properties very awkward.

- There are several different notions of CTRS in the literature, each corresponding to a different definition of what form a condition may have. The categorical methodology suggests a new notion of condition which subsumes most of those occurring in the literature. Our semantics therefore provides a unifying framework for studying CTRSs.

- Finally the theoretical basis of this research is the work of Kelly and Power in generalizing the equivalence between algebraic theories and finitary monads on the category of **Set** to more general categories. This theory was originally developed to study categories with extra structure. By providing a natural application within computer science, we hope to make their work more accessible to the computer science community.

The rest of the paper is divided as follows. In section 2 we standardise notion for the rest of the paper while in section 3 we briefly review the equivalence between algebraic theories and finitary monads on the category **Set**. Section 4 presents TRSs and gives a concrete semantics for TRSs using monads over the category **Pre** and this semantics is explained in section 5 in terms of the Kelly Power framework. Section 6 introduces CTRSs and section 7 discusses their semantics. We finish in section 8 with our conclusions and further work.

2 Notation and Preliminaries

The natural numbers are denoted $\mathbb{N}$. A pre-order is a pair $\langle X, \leq \rangle$ where X is a set and $\leq \subseteq X \times X$ is a reflexive, transitive relation. We use the following conventions for variables: P, Q for pre-orders, X, Y, Z for sets, A, B for algebras, $\mathcal{C}, \mathcal{D}$ for categories, and Σ for signatures. If P is a pre-order, its carrier set is denoted $|P|$ and its order relation is denoted $\Rightarrow_P$ to emphasise the intuition that pre-orders model terms reducing to other terms. For the sake of legibility, and also to emphasise the logical nature of our inductive constructions over pre-orders, we write $P \models x \Rightarrow y$ if $x, y \in P$ and $x \Rightarrow_P y$. We write $\mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots$ for the finite discrete pre-orders, the elements of which we denote as $x_1, x_2, \ldots$. When we need to draw pre-orders we will enclose the pre-order in brackets, either using letters or the symbol $\circ$ to represent elements of the carrier set and use arrows to denote when one element is related to another. Arrows which can be inferred from the reflexivity and transitivity of the order relation are omitted. The symbol $\downarrow$ denotes the preorder with two elements which are related in the order.

We assume the reader is familiar with the basic concepts of category theory and enriched category theory. The basic theory of monads over categories with more structure than **Set** has been developed by Kelly and Power [6] and forms the theoretical basis of this research. However, we dont consider familiarity with their work a pre-requisite for this paper. Rather, the concrete construction of representing monads for algebraic theories and for TRSs given in the next two sections should allow the reader to follow the rest of the paper.

3 Universal Algebra and Monads

Definition 1 (Monad) *A monad $\mathsf{T} = \langle T, \eta, \mu \rangle$ on a category $\mathcal{C}$ is given by an endofunctor $T : \mathcal{C} \to \mathcal{C}$, called the action, and two natural transformations, $\eta : 1_\mathcal{C} \Rightarrow T$, called the unit, and $\mu : TT \Rightarrow T$, called the multiplication of the monad, satisfying the monad laws: $\mu{\cdot}T\eta = 1_\mathcal{C} = \mu{\cdot}\eta_T$, and $\mu{\cdot}T\mu = \mu{\cdot}\mu_T$.*

The monadic approach to term rewriting generalises the well known equivalence between (finitary) monads on the category **Set** and universal algebra. Thus, in order to motivate our constructions, we begin with a brief account of this equivalence. However, since this material is standard category theory, we omit most proofs and instead refer the reader to the standard references ([11], [15] and [10, Section VI]).

Every algebraic theory defines a monad on **Set** whose action maps a set to the free algebra over this set. The unit maps a variable to the associated term, while the multiplication describes the process of substitution. The monad laws ensure that substitution behaves correctly, i.e. substitution is associative and the variables are left and right units. Thus monads form an abstract calculus for equational reasoning where *variables, substitution* and *term algebra* (represented by the unit, multiplication and action of the monad) are the primitive concepts. We now make these ideas precise.

Definition 2 (Signature) *A (single-sorted)* signature *consists of a function $\Sigma : \mathbb{N} \to \mathbf{Set}$. The set of n-*ary operators *of Σ is defined $\Sigma_n \stackrel{def}{=} \Sigma(n)$*

Definition 3 (Term Algebra) *Given a signature Σ and a set of variables X, the* term algebra *$T_\Sigma(X)$ is defined inductively:*

$$\textsc{Var}\ \frac{x \in X}{\text{'}x \in T_\Sigma(X)} \qquad \textsc{Op}\ \frac{f \in \Sigma_n \quad t_1, \ldots t_n \in T_\Sigma(X)}{f(t_1, \ldots, t_n) \in T_\Sigma(X)}$$

Quotes are used to distinguish a variable $x \in X$ from the term $\text{'}x \in T_\Sigma(X)$. This categorically inspired idea turns out to considerably simplify modularity proofs and hence is a nice example of a benefit to rewriting arising from its categorical semantics.

Lemma 1 *The map $X \mapsto T_\Sigma(X)$ defines a monad T_Σ on* **Set**.

The monads arising via the term algebra construction satisfy an important continuity condition, namely they are *finitary*. To understand this condition, the term algebra $T_\Sigma(X)$ built over on an infinite set X of variables can be given as

$$T_\Sigma(X) = \bigcup_{X_0 \subset X \text{ is finite}} T_\Sigma(X_0)$$

This equation holds since all the operators in Σ have a finite arity and so a term built over X contains only a finite number of variables — such terms are therefore built over a finite subset of X. Categorically this is expressed by saying the functor T_Σ is *finitary*:

Definition 4 (Finitary Monads) *A functor is* finitary *iff it preserves filtered colimits [10]. A monad is* finitary *iff its action is finitary.*

Lemma 2 *If Σ is a signature, then T_Σ is finitary [15, Lemma 1.7].*

Monads also model *algebraic theories*:

Definition 5 (Equations and Algebraic Theories) *Given a signature Σ, a Σ-*equation *is of the form $X \vdash t = s$ where X is a set and $t, s \in T_\Sigma(X)$. An* algebraic theory $\langle \Sigma, E \rangle$ *consists of a signature Σ and a set E of Σ-equations.*

The term algebra construction generalises from signatures to algebraic theories by mapping a set X to the term algebra quotiented by the equivalence relation generated from the equations

$$T_A(X) = T_\Sigma(X)/_{\ E}$$

and hence we again obtain a finitary monad over **Set**. The category of algebras of this monad is equivalent to the category of models of $\mathcal{A}$, justifying the correctness of the monadic semantics: "universal algebra is the study of finitary monads over **Set**" [11].

One key property of this monadic semantics for algebraic theories is that it is *compositional*. For the disjoint union of algebraic theories, this means $\mathsf{T}_{\mathcal{A}_1 + \mathcal{A}_2} \cong \mathsf{T}_{\mathcal{A}_1} + \mathsf{T}_{\mathcal{A}_2}$. This compositionality property is established by showing that every finitary monad arises from an algebraic theory called the *internal language* of the monad. Since the monadic semantics is left adjoint to the internal language construction, compositionality follows from the fact that left-adjoints preserve colimits.

In summary, monads provide a semantics for algebraic theories with the concepts of term-algebra, variable and substitution taken as primitive.

4 Categorical Rewriting

Rewriting systems consist of rules for transforming the terms of a formal language. Intuitively, each rule is "local", conceptually simple and easy to reason about while sequences of rewrites can model complex effects. We now briefly review the theory of TRSs — our presentation differs slightly from traditional presentations [7] so as to allow a smoother development of the categorical semantics. The main differences are as follows:

- Traditionally one uses a *set* of variables but, as our semantics for TRSs is based upon monads over **Pre**, our variables must form a *preorder*. The carrier set of this preorder are the usual variables which are used to form terms while the order relation allows variables to rewrite to other variables.

- In addition, one usually fixes the variables at the outset. However, as the action of the representing monad of a TRS should map *any* pre-order (thought of as variables and variable rewrites) to a pre-order (thought of as terms and rewrites between them), we allow the pre-order of variables to vary.

- The last point also applies to rewrite rules, which are usually defined as $l \Rightarrow r$ (with the set of variables fixed). Here, every rewrite rule has a *context* (which is of course not merely a set, but a pre-order of variables). The context is the arity of the rewrite rule and emphasizes our view of rewrite rules as operations, rather then directed equations.

- Traditionally, the semantics of a TRS is a relation, either the *one-step* rewrite relation or the many *many-step* rewrite relation. Rather than give the semantics of a TRS as a relation, our monadic semantics interprets a TRS as a special kind of function (ie a monad) which maps pre-orders (thought of as variables and variable rewrites) to pre-orders (thought of as terms and rewrites between them).

Each of these categorically inspired generalisations of rewriting has been applied to syntactic formulations of rewriting and, in particular, has lead to a significant simplification of modularity proofs. This is strong empirical evidence that our categorical models really do advance our understanding of rewriting.

Definition 6 (Rewrite Rules and Term Rewriting Systems) *Let Σ be a signature. A Σ-*rewrite rule *is of the form $X \vdash l \Rightarrow r$ where X is a finite pre-order and $l, r \in T_\Sigma(|X|)$. A TRS $\Theta = \langle \Sigma, R \rangle$ consists of a signature Σ and a set R of Σ-rewrite rules.*

Example 1 (Addition) *The TRS for addition presented in the introduction formally consists of the signature Σ where*

$$\Sigma(0) = \{0\} \qquad \Sigma(1) = \{S\} \qquad \Sigma(2) = \{+\}$$

and has as Σ-rewrite rules $\{r_1, r_2\}$ where

$$\begin{array}{llllll} r_1 : & \{x\} & \vdash & 0 + x & \Rightarrow & x \\ r_2 : & \{x, y\} & \vdash & Sx + y & \Rightarrow & S(x + y) \end{array}$$

Given a TRS one usually defines the *one-step reduction relation* and then the *many step reduction relation* which is the reflexive-transitive closure of the one-step reduction relation. As indicate above, our presentation of rewriting does not interpret a TRS as a relation, but as a function on relations. In addition, we take the many-step reduction relation as being of primary importance since it is usually this relation that one wants to reason about. Thus we define a *term reduction algebra* in Table 1 which takes a pre-order P of variables and rewrites between them and returns a pre-order $T_\Theta(P)$ of terms and rewrites between them. This term reduction algebra is the direct generalisation of the term algebra construction (Def. 3) which takes a set of variables X and returns a set of terms $T_\Sigma(X)$. Just as the term algebra construction defines a monad over **Set**, so the term algebra construction defines a monad over **Pre**.

Table 1: The Term Reduction Algebra for a TRS $\Theta = \langle \Sigma, \mathcal{R} \rangle$

Let X be a pre-order. Then define the pre-order $\mathsf{T}_\Theta(X)$ as follows

- $\mathsf{T}_\Theta(X)$ has the carrier set $\mathsf{T}_\Sigma(|X|)$
- $\mathsf{T}_\Theta(X)$ has the least pre-order generated by the following rules:

$$[\textsc{Var}] \quad \frac{X \models x \Rightarrow y}{\mathsf{T}_\Theta(X) \models \text{'}x \Rightarrow \text{'}y}$$

$$[\textsc{Cong}] \quad \frac{\mathsf{T}_\Theta(X) \models t_1 \Rightarrow s_1, \ldots, t_n \Rightarrow s_n}{\mathsf{T}_\Theta(X) \models f(t_1, \ldots, t_n) \Rightarrow f(s_1, \ldots, s_n)} \; f \in \Sigma_n$$

$$[\textsc{Inst}] \quad \frac{(Y \vdash l \Rightarrow r) \in R, \qquad \sigma : Y \to T_\Theta(X)}{\mathsf{T}_\Theta(X) \models l\sigma \Rightarrow r\sigma}$$

Lemma 3 *The map $X \mapsto T_\Theta(X)$ defines a finitary,* **Pre***-enriched monad T_Θ.*

Proof. The proofs follow those for algebraic theories, but we also have to show that η and μ preserve the order. To show T_Θ is **Pre**-enriched, we have to show it preserves the pointwise order on morphisms. See [9] for more details. $\square$

Lemma 3 shows that we can model TRSs by monads over **Pre**. Although this semantics is powerful enough to prove state-of-the-art results in modular rewriting, it is unsatisfactory in a number of ways. Firstly, the semantics is a set-theoretic construction and proofs about the semantics will consequently use induction on the term structure; secondly, the semantics assumed various choices, eg the choice of base category, and if we change any of these, lemma 3 needs reproving. Thirdly, although we have given a mathematical account of the similarity of algebraic theories and TRSs at the semantics level, we have so far failed to justify our claim that they are similar at the syntactic level. Finally, given that many semantics exist for TRSs, is there any reason to believe the monadic semantics is preferable. In addition to the empirical evidence given by the results in modular rewriting we have achieved using monads, we would also prefer some more theoretical reasons to believe in the "canonicity" of the monadic semantics. These questions are answered by the theory of enriched monads to which we turn now.

5 TRSs and Enriched Monads

As we have seen, algebraic theories are modelled by monads over **Set** while TRSs are modelled by monads over **Pre**. These results are instances of a general theory developed by Kelly and Power [6] which shows how monads over categories other than **Set** give rise to a generalised notion of algebraic theories. For any such category $\mathcal{C}$, this theory provides us with suitably generalised notions of *signature*, *term algebra*, and *equations*. In this section we briefly explain the Kelly-Power framework using algebraic theories and TRSs to motivate the general constructions. Then in section 7 we apply the Kelly-Power framework to derive a monadic semantics for CTRSs.

5.1 Generalised Signatures

Recall from definition 2 that a signature is a function $\Sigma : \mathbb{N} \to \mathbf{Set}$ — that is for each arity n, there is a set of operations $\Sigma(n)$. To define signatures for a category $\mathcal{C}$ it is natural to replace the idea of a *set* of operations by a $\mathcal{C}$ object of operations, i.e. change the codomain of Σ to $\mathcal{C}$. As for the domain of Σ, the key point about the natural numbers is that they represent the isomorphism classes of finite sets presentable objects. Finite sets correspond to the finitely presentable objects $\mathcal{C}_{fp}$ (or fp-objects) of $\mathcal{C}$. Hence, we replace $\mathbb{N}$ by a *set* $\mathcal{N} \subseteq \mathcal{C}_{fp}$ of fp-objects which represent isomorphism classes of fp-objects of $\mathcal{C}$:

Definition 7 (Generalised Signatures) *If $\mathcal{C}$ is a category, then a $\mathcal{C}$-signature is a map $\Sigma : \mathcal{N} \to \mathcal{C}$.*

A pre-order is finitely presentable iff its carrier set is finite; the set $\mathcal{N}$ are essentially finite ordinals which may additionally have an order on the objects. Hence if we present a TRS as a generalised signature Σ, the signature will map a pre-order $P \in \mathcal{N}$ to a pre-order $\Sigma(P)$ of operations of that arity. The carrier set of $\Sigma(P)$ can be thought of as term constructors and the order relation as rewrite rules. This is another instance where the categorical approach offers a new perspective of treating rewrite rules as constructors in their own right as opposed to the more traditional view of rewrite rules as oriented equations.

Example 2 (Addition cont'd) *The addition TRS has the following operations with associated arities*

Operation	0	S	$+$	r_1	r_2
Arity	**0**	**1**	**2**	**1**	**2**

The associated **Pre**-*signature* $\mathcal{R} : |\mathbf{Pre}_{fp}| \to \mathbf{Pre}$ *is*

$$\begin{array}{rcll} \mathcal{R}(\mathbf{0}) & = & (0) & \\ \mathcal{R}(\mathbf{1}) & = & (S & s_1 \xrightarrow{r_1} t_1) \\ \mathcal{R}(\mathbf{2}) & = & (+ & s_2 \xrightarrow{r_3} t_2) \\ \mathcal{R}(P) & = & () & \textit{otherwise} \end{array}$$

Since the arity of the rewrite rule r_1 is the discrete 1-element pre-order, we have represented it by an arrow in $\mathcal{R}(\mathbf{1})$. But of course arrows need sources and targets, and hence the carrier set of $\mathcal{R}(\mathbf{1})$ has three elements and thus declares three term constructors. To summarise, when we model a TRS as a generalised signature we extend the signature of the TRS with a term constructor for the source and target of every rewrite rule r_i. Equality of e.g. s_1 with the source of the actual rule $0 + x$ will be enforced by equations.

5.2 Generalised Term Algebras

The generalised term algebra construction should map an object of $\mathcal{C}$, thought of as variables, to another object of $\mathcal{C}$ which we think of the terms built over the variables. Recall that the term algebra of an algebraic theory was defined inductively by the clauses VAR and OP in Def. 3. We can replace $t_1, \ldots t_m \in T_\Sigma(X)$ in OP by a function $\theta : m \to T_\Sigma(X)$ where m is the m element set or, equivalently, an element of $[m, T_\Sigma(X)]$. Further, as the term algebra is defined inductively, it is limit of the chain $T_0(X) \hookrightarrow T_1(X) \hookrightarrow \ldots$ where

$$T_0(X) = X \qquad\qquad \frac{f \in \Sigma(m) \quad \theta : [m, T_n(X)]}{f\theta \in T_{n+1}(X)}$$

Writing this more abstractly, and crucially isolating the rôle of **Set**:

$$T_0(X) = X \qquad T_{n+1}(X) = X + \textstyle\sum_{m \in Nat} [m, T_n(X)] \times \Sigma(m)$$

This formulation of the term algebra construction makes explicit the structure of **Set** which is used and, consequently, suggests that we should define the generalised term algebra as follows

Definition 8 (Generalised Term Algebra:) *Let Σ be a $\mathcal{C}$-signature and X an object of X. Then define $T_\Sigma(X) = colim \;\; T_0(X) \hookrightarrow T_1(X) \hookrightarrow \ldots$*

$$\begin{array}{rcl} T_0(X) & = & X \\ T_{n+1}(X) & = & X + \displaystyle\sum_{c \in \mathcal{C}_{fp}} \mathcal{C}(c, T_n(X)) \otimes \Sigma(c) \end{array}$$

In its most general form, the Kelly-Power framework considers a category $\mathcal{C}$ enriched over a category $\mathcal{V}$. Then, the map $X \mapsto T_\Sigma(X)$ extends to a finitary, $\mathcal{V}$-enriched monad on $\mathcal{C}$. In this case, $\otimes$ is the *co-power* or *tensor*, that is we have $\mathcal{C}(U \otimes A, B) \cong [U, \mathcal{C}(A, B)]$

natural in B in $\mathcal{C}$. Thus the term algebra $T_\Sigma(X)$ depends on not just the category $\mathcal{C}$, but also the choice of enrichment. When modeling TRSs, we can consider **Pre** as enriched over **Set** or **Pre** as being enriched over **Pre**. To understand the consequences of the different choices of enrichments, consider a term constructor f of arity c and c-tuples of terms $\theta_1, \theta_2 : c \to T_n(X)$:

- With enrichment over **Set**, $[c, T_n(X)]$ is the *set* of monotone functions from the preorder c to the pre-order $T_n(X)$. Thus, even if $\theta_1 \Rightarrow \theta_2$, there is no forced relationship between the terms $f\theta_1$ and $f\theta_2$. Consequently, reduction is not forced to be a congruence.

- With enrichment over **Pre**, $[c, T_n(X)]$ is a *pre-order* of monotone functions. Consequently, if $\theta_1 \leq \theta_2$, then $f\theta_1 \Rightarrow f\theta_2$. Therefore, enrichment over **Pre** forces reduction to be a congruence.

In most rewriting (including the version presented in this paper) reduction is considered to be a congruence and hence we always work with **Pre**-enriched structures. But it is re-assuring to know that if we wanted to study a non-congruent reduction relation, eg in the theory of eta-expansions [2, 4], we need only change the enrichment. Of course $\mathcal{C}$ and $\mathcal{V}$ must satisfy certain properties for Def. 8 to make sense. However, we delay a discussion of this until after we have introduced the generalisation of equations.

5.3 Generalised Equations and Generalised Algebraic Theories

Thus, every signature $\Sigma : \mathcal{N} \to \mathcal{C}$ gives us a monad. Conversely, every finitary monad gives a signature by precomposing the action with the inclusion $\mathcal{N} \hookrightarrow \mathcal{C}$. These two mappings form an adjunction

$$[\mathcal{N}, \mathcal{C}] \underset{\longleftarrow}{\overset{\mathsf{T}_{-}}{\underset{\perp}{\longrightarrow}}} \mathbf{Mon}_{Fin}(\mathcal{C}) \tag{1}$$

between the category of signatures (functors from $\mathcal{N}$ to $\mathcal{C}$) and finitary monads over $\mathcal{C}$.

Definition 9 (Generalised Algebraic Theory) *A generalised algebraic theory $\mathcal{A} = \langle \Sigma, E \rangle$ is given by a generalised signature Σ, and equations which are given by a function $E : \mathcal{N} \to \mathcal{C}$ together with a family of pairs of maps*

$$\left\{ E(c) \underset{\tau_c}{\overset{\sigma_c}{\rightrightarrows}} T_\Sigma(c) \right\}_{c \in \mathcal{N}}$$

The idea is that E specifies the shape of the equations while the maps σ and τ specify the left-hand side and right hand side of the equations respectively. In the case of TRSs, we want to equate the formal sources and targets introduced by the generalised signature with the actual sources and targets of the rewrite rules. Intuitively this means writing

$$s_1(x) = 0 + x \quad t_1(x) = x \quad s_2(x, y) = Sx + y \quad t_2(x, y) = S(x + y)$$

To translate these intuitive equations into the generalised equations of Def. 9 note that the carrier set of $E(\mathbf{1})$ will contain one element for every equation of arity $\mathbf{1}$ between terms while the order relation of $E(\mathbf{1})$ will specify equations of arity $\mathbf{1}$ between the rewrites. Similar comments hold for $E(\mathbf{2})$. Since there are two equations of arity $\mathbf{1}$ between terms, and two equations of arity $\mathbf{2}$ between terms, we set $E(\mathbf{1}) = E(\mathbf{2}) = \mathbf{2}$. Consequently, the maps σ_1 and τ_1 consist of a pair of elements of $T_\Sigma(1)$, and similarly for σ_2 and τ_2:

$$\begin{array}{rcll} \sigma_1 & = & (s_1(x_1) & t_1(x_1)) \\ \tau_1 & = & (0 + x_1 & x_1) \\ \sigma_2 & = & (s_2(x_1, x_2) & t_2(x_1, x_2)) \\ \tau_2 & = & (Sx_1 + x_2 & S(x_1 + x_2)) \end{array}$$

Given an algebraic theory, the maps $\sigma, \tau : E(c) \to T_\Sigma(c)$ in $[\mathcal{N}, \mathcal{C}]$ have conjugates $\sigma', \tau' : T_E(c) \to T_\Sigma(c)$ in $\mathbf{Mon}_{Fin}(\mathcal{C})$. The monad $T_\mathcal{A}$ representing is the coequaliser of σ' and τ' in $\mathbf{Mon}_{Fin}(\mathcal{C})$. The main theorem of the Kelly and Power [6] is that every finitary monad can be represented as a coequaliser of two such morphisms, and thus every monad arises from a generalised algebraic theory.

Theorem 1 (Kelly-Power) *Let $\mathcal{C}$ be a locally finitely presentable category enriched over the locally finitely presentable symmetric monoidal closed category $\mathcal{V}$. Then*

- *Every $\mathcal{C}$-algebraic theory $\mathcal{A} = \langle \Sigma, E \rangle$ is represented by a finitary $\mathcal{V}$-enriched monad $\mathsf{T}_\mathcal{A}$.*
- *Every finitary $\mathcal{V}$-enriched monad on $\mathcal{C}$ arises from a $\mathcal{C}$-algebraic theory.*

We leave the reader to consult the literature [5] for the precise definition of a locally finitely presentable category in the enriched context. For our purposes, **Pre** is both lfp, and lfp as a closed category.

We finish this section with a crucial observation about our semantics, namely that it is constructed *categorically*, and hence we can reason about it categorically. This contrasts, for example, with the 2-categorical semantics of a TRS where the theory of a TRS is constructed syntactically. It is also more amenable to generalisations, e.g. the constructors (both terms and rewrites) may have non-discrete arities and we could write equations between any two terms, thus modeling *equational rewriting.*

6 Conditional Term Rewriting Systems (CTRSs)

Conditional rewriting allows conditions to be imposed on when a rewrite is permissible. There are various different types of CTRS depending on the exact nature of the conditions which may be imposed. We now briefly review the theory of CTRSs — more details may be found in [7]. Again our presentation differs slightly from traditional formulations in that we parameterise each conditional rewrite rule by a pre-order of variables.

Definition 10 (Conditional Term Rewrite Systems) *Let Σ be a signature. A Σ-conditional rewrite rule is of the form*

$$X,\ \ l_1 \sim r_1, \ldots, l_n \sim r_n \vdash l \Rightarrow r$$

where X is a finite pre-order and l_i, l, r_i, r are all members of $T_\Sigma(|X|)$. A Σ-CTRS consists of a set of Σ-conditional rewrite rules.

In the literature there are several standard interpretations of $\sim$, each of which leads to a different notion of CTRS.

- **Join-CTRSs:** The relation $\sim$ is called *joinability* and $t \sim u$ iff t and u have a common reduct. In this case we write $t \Downarrow u$.
- **Semi-equational CTRSs:** The relation $\sim$ is interpreted as being equivalent in the associated equational theory. In such circumstances we write $t \Leftrightarrow^* u$
- **Reduction CTRSs:** The relation $\sim$ is interpreted as many-step reduction. In such circumstances we write $t \Rightarrow^* u$

Of course, there are many other possible interpretations of the relation $\sim$. Our semantics covers most interpretations of $\sim$ and hence provides a unifying framework for the study of many of the CTRSs which occur in the literature. In order to avoid committing ourselves to a particular notion of CTRS we will in future write Σ-conditional rewrite rules as

$$Y, C \vdash l \Rightarrow r$$

where Y is a pre-order, $l, r \in T_\Sigma(|Y|)$ and C is some relation on $T_\Sigma(|Y|)$

Defining the reduction relation of a CTRS is slightly more delicate than in the unconditional case, since it is not enough to close each rewrite rule is closed under substitutions and contexts. We further have to look at the rewrites already constructed, check whether the conditions of a rewrite rule are met, and if so add the conclusion of the rewrite rule to the rewrite relation. Thus, we take a rewrite relation, and add in more rewrites, if the conditions are satisfied; this gives the whole semantics a rather functional flavor.

Definition 11 (Reduction for CTRSs) *Given a Σ-CTRS Θ, and a pre-order X, the conditional reduction algebra $T_\Theta(X)$ is defined in Table 2.*

7 Can CTRSs be Modelled by Monads?

Since TRSs can be modelled by monads, a natural question is whether a CTRS can be modelled by a monad. To answer this question, recall that each rewrite rule of a TRS is treated as an operation of a generalised signature and that for the TRSs one usually considers, these arities are discrete. However the theory of enriched monads allows arities for rewrite rules to be non-discrete pre-orders and this leads to a limited form of CTRSs *where each term in the conditions is a variable.*

Example 3 *Let Σ be the signature consisting of a nullary function symbol T and a binary function symbol eq. The Σ-conditional rewrite rule*

$$x \Rightarrow z, y \Rightarrow z \vdash eq(x, y) \Rightarrow T$$

Table 2: The Conditional Term Reduction Algebra for a Σ-CTRS Θ

Let X be a pre-order. Then define the pre-order $\mathsf{T}_\Theta(X)$ as follows

- $\mathsf{T}_\Theta(X)$ has the carrier set $\mathsf{T}_\Sigma(|X|)$
- $\mathsf{T}_\Theta(X)$ has the least pre-order generated by

$$[\textsc{Var}]\ \frac{X \models x \Rightarrow y}{\mathsf{T}_\Theta(X) \models' x \Rightarrow' y}$$

$$[\textsc{Cong}]\ \frac{\mathsf{T}_\Theta(X)(t_1 \Rightarrow s_1), \ldots, \mathsf{T}_\Theta(X)(t_n \Rightarrow s_n)}{\mathsf{T}_\Theta(X)(f(t_1, \ldots, t_n) \Rightarrow f(s_1, \ldots, s_n))}\ f \in \Sigma_n$$

$$[\textsc{Inst}]\ \frac{(Y, C \vdash l \Rightarrow r) \in R, \qquad \sigma : Y \rightarrow T_\Theta(X), \qquad T_\Theta(X)(C)}{\mathsf{T}_\Theta(X)(l\sigma \Rightarrow r\sigma)}$$

can be given as an operation with arity the pre-order $(x \Rightarrow z \Leftarrow y)$. *This is because a monotone map* $\theta : (x \Rightarrow z \Leftarrow y) \rightarrow T_\Theta(X)$ *will consist of three terms* $\theta(x), \theta(y)$ *and* $\theta(z)$ *and monotonicity will force* $\theta(x)$ *and* $\theta(y)$ *both to rewrite to* $\theta(z)$.

To model more general forms, where the conditions in the rewrite rules pertain not merely to variables but rather to arbitrary terms, one could imagine trying to use equations to model CTRSs with arbitrary conditions as a quotient of CTRSs whose conditions pertain solely to variables. However, this would betray a fundamental misunderstanding of the *universality* of universal algebra. The key point is that a conditional rewrite rule is an operation whose domain is restricted by the conditions, just like e.g. composition of morphisms in category is an operation which is defined *only* if the target of the first morphism equals the source of the second. Theories containing operations whose domains are (equationally) restricted are examples of *essentially algebraic theories.*

The construction of a monadic semantics for CTRSs will proceed in three essential steps. First, we give a straightforward semantics of CTRSs as *functions* over *rewrite relations.* Since rewrite relations are not suitable as a base category, we reformulate this semantics as a function over ordered algebras, which can then be made into a monad. Continuing the analogy above, categories can be presented by a monad, but over the category **Grph** of graphs, not over **Set**; and CTRSs can also be modelled by a monad, but over a base category which is rich enough to model the conditions uniformly.

7.1 CTRSs as Ordered Σ-Algebra Transformers

Although not defining a monad, we can view a CTRS Θ as a function $[\![\Theta]\!]$ which maps a relation on terms to a relation on terms. In order to meaningful evaluate the conditions of the conditional rewrite rules, these relations need to closed under substitution and contexts. Such relations are called *rewrite relations* [1], and we write $Rew(\Sigma, X)$ for all rewrite relations over $T_\Sigma(X)$.

But even if we can define a function $[\![\Theta]\!] : Rew(\Sigma, X) \rightarrow Rew(\Sigma, X)$ for a CTRS Θ, this does not define a monad. Moreover, the rewrite relations do not form a good base category for our purposes: firstly they are very syntactic, since they are relations on terms; secondly, their categorical properties are not explored very well; and finally, we still need to deal with the conditional clause INST in Table 2. In order to meet these problems, we replace rewrite relations with ordered Σ-algebras.

First we need to defined ordered Σ-algebras. There are three equivalent definitions — proofs of the equivalence can be found in e.g. [14].

Definition 12 (Ordered Σ-Algebras) *Let Σ be a signature. An ordered Σ-algebra is one of:*

- *A pre-order P and, for every $f \in \Sigma(n)$, a monotone map $f_A : \mathcal{A}^n \rightarrow \mathcal{A}$. If A is an ordered Σ-algebra, then its carrier set is written $|A|$ and is defined to be the carrier set of P.*
- *An algebra $T_\Sigma A \xrightarrow{\alpha} A$ of the* **Pre***-enriched monad T_Σ :* **Pre**$\rightarrow$**Pre***, which is the representing monad of the TRS $\Sigma = \langle \Sigma, \emptyset \rangle$ whose signature is Σ and which contains no rewrite rules.*
- *A FP-functor from $Th(\Sigma)$ to* **Pre***, where $Th(\Sigma)$ is the* **Pre***-enriched Lawvere theory of Σ.*

The category of ordered Σ-algebras is denoted **Σ-OAlg**. The free and forgetful functors between **Σ-OAlg** and **Pre** are denoted F_Σ and U_Σ

$$\mathbf{Pre} \underset{U_\Sigma}{\overset{F_\Sigma}{\rightleftarrows}} \mathbf{\Sigma\text{-}OAlg} \qquad \perp \tag{2}$$

where F_Σ maps any preorder X to the free ordered Σ-algebra, and U_Σ maps any algebra to its carrier. Crucially, ordered Σ-algebras have enough structure to interpret the conditions arising in a CTRS. For this, if A is an ordered Σ-algebra $T_\Sigma A \xrightarrow{\alpha} A$, and P a pre-order, then every map $\sigma : P \rightarrow A$ (thought of as a variable assignment) gives (under adjunction 2 rise to a *homomorphic extension* $\sigma^* : T_\Sigma(P)$, $\sigma = \alpha \cdot T_\Sigma(\sigma)$. For $l \in |T_\Sigma(P)|$ we denote the action of ϕ^* on l as $l\phi$.

Definition 13 (Validation) *An ordered Σ-algebra A* validates *a Σ-conditional rewrite rule $(Y, C \vdash l \Rightarrow r)$, if there exists $\sigma : Y \rightarrow A$ s.t. for each $l \Rightarrow r \in C$ we have $A \models l\sigma \Rightarrow r\sigma$; this is written as $A \models_\sigma C$.*

We can now give a semantics to a CTRS as a map of ordered Σ-algebras in Table 3. Again, reflexivity of $[\![\Theta]\!](A)$ follows from the rule [VAR]. This semantics can now be turned into a monad, if we can remove the condition from rule INST.

Table 3: A Semantics for CTRSs as ordered Σ-algebra-transformers

For a CTRS $\Theta = \langle \Sigma, R \rangle$, $[\![\Theta]\!]$ maps an ordered Σ-algebra A to the ordered Σ-algebra defined as follows:

- The carrier of $[\![\Theta]\!](A)$ is the carrier of A.
- The order of $[\![\Theta]\!](A)$ is defined inductively by

$$[\textsc{Var}] \quad \frac{A \text{ is an ordered } \Sigma\text{-algebra}}{A \subseteq [\![\Theta]\!](A)}$$

$$[\textsc{Inst}] \quad \frac{X, C \vdash l \Rightarrow r \qquad \sigma : X \to A \qquad [\![\Theta]\!](A) \models_\sigma C}{[\![\Theta]\!](A) \models l\sigma \Rightarrow r\sigma}$$

$$[\textsc{Cong}] \quad \frac{[\![\Theta]\!](A) \models l_1 \Rightarrow r_1, \ldots, [\![\Theta]\!](A) \models l_n \Rightarrow r_n}{[\![\Theta]\!](A) \models f(l_1, \ldots, l_n) \Rightarrow f(r_1, \ldots r_n)} \; f \in \Sigma(n)$$

7.2 A Monadic Semantics for CTRSs

So far we have tried to give a semantics for CTRSs in terms of finitary monads over **Pre** and, when we found out this was not possible, we subsequently gave a semantics for CTRSs as ordered Σ-algebra-transformers. We now want to turn the ordered Σ-algebra-transformer into a *monad over the category of ordered Σ-algebras.* By changing to a more complex base category for the representing monad, the arities for operations now have a sufficiently rich structure to model CTRSs. However, instead of reasoning about pre-orders, we must now reason about ordered Σ-algebras.

A direct proof that a CTRS Θ can be modelled by a monad $\mathsf{T}_\Theta : \Sigma\textbf{-OAlg} \to \Sigma\textbf{-OAlg}$ would first extend the transformer $[\![\Theta]\!]$ to an endofunctor $[\![\Theta]\!] : \Sigma\textbf{-OAlg} \to \Sigma\textbf{-OAlg}$, construct the unit $\eta_A : A \to [\![\Theta]\!](A)$ (from the rule $\textsc{Var}$) and the multiplication $\mu_A : [\![\Theta]\!]^2)A \to\to [\![\Theta]\!](A)$, and then verify that these definitions satisfy the monad laws, and that $[\![\Theta]\!]$ is finitary. This is a lot of work — not difficult but tedious.

Hence, a better alternative is to use the theory of enriched monads as described in Sect. 5. Therefore we will proceed as follows. First (7.2.1), we show that the category of ordered Σ-algebras satisfies the premises of the Kelly-Power framework as described in theorem 1. Then (7.2.2), we present each CTRS as an ordered Σ-algebra-theory. This amounts to defining each conditional rewrite rule as an operation whose arity is a finitely presentable ordered Σ-algebra. By theorem 1 this will define a **Pre**-enriched monad. Finally (7.2.3), we prove that the monadic semantics is actually correct, ie that it agrees with the ordered Σ-algebra-transformer semantics of section 7.1.

7.2.1 Preliminaries

Most of the results we require are (**Pre**-enriched versions of) well known results — we quote them here and provided references for proofs. Recall that we require properties of the category we are enriching over (here **Pre**) and properties of the base category of the monad (here Σ-**OAlg**). In addition, in order to associate arities to operations we need to know what the finitely presentable objects of Σ-**OAlg** are.

Lemma 4 *The category* **Pre** *is a symmetric monoidal closed category and is lfp with respect to this closed structure.*

Lemma 5 *The category of* Σ-**OAlg** *is lfp. The set* $\mathcal{N}_{\Sigma\text{-}\mathbf{OAlg}}$ *representing isomorphism classes of fp-objects is represented as the coequaliser of* $Fm \rightrightarrows Fn$ *where* m, n *are representing isomorphism classes of fp-objects in* **Pre**.

Finally, tensors in Σ-**OAlg** are constructed with the following isomorphism:

$$P \otimes F_\Sigma(P') \cong F_\Sigma(P \times P') \tag{3}$$

7.2.2 CTRSs as Generalised Theories

We now present a CTRS as a Σ-**OAlg**-theory. For the sake of clarity of our exposition, we focus for the time being on the CTRS containing the single conditional rewrite rule

$$R \stackrel{def}{=} (P, l \Rightarrow r \vdash u \Rightarrow v)$$

At the end of this section we discuss the modifications required to treat more complex CTRSs. The most important question is refers to the arity of R which we shall henceforth denote $ar(R)$. Intuitively, $ar(R)$ should be the smallest ordered Σ-algebra such that the carrier of $ar(R)$ is the carrier of $F_\Sigma(P)$, and that $ar(R) \models l \Rightarrow r$.

We now have construct such an ordered Σ-algebra, and moreover show it is finitely presentable.

Lemma 6 $ar(R)$ *is a finitely presentable ordered* Σ*-algebra presented as the coequalizer in* Σ-**OAlg** *of*

$$F_\Sigma(\mathbf{2}) \underset{\tau}{\overset{\sigma}{\rightrightarrows}} F_\Sigma(P + \downarrow) \tag{4}$$

where $\sigma \stackrel{def}{=} F_\Sigma(i)$ *and* i *is the obvious composition of inclusions* $\mathbf{2} \to \downarrow \to P + \downarrow$. *To define* τ, *note that the pair of terms* (l, r) *define a map in* **Pre** $(l, r) : \mathbf{2} \to T_\Sigma(P) = U_\Sigma F_\Sigma(P)$. *Using the natural isomorphism of adjunction 2 this defines a map* $(l, r)^* : F_\Sigma(\mathbf{2}) \to F_\Sigma(P)$. *Finally define* $\tau = F_\Sigma(\iota) \circ (l, r)^*$ *where* $\iota : P \to P + \downarrow$ *is the obvious inclusion.*

Note that since $ar(R)$ is presented by (4), by lemma 6 $ar(R)$ is in $\mathcal{N}_{\Sigma\text{-}\mathbf{OAlg}}$.

There are two points to remember when generalising the presentation of a TRS as a **Pre**-signature to the presentation of a CTRS as a Σ-**OAlg**-signature:

- Firstly, the use of Σ-**OAlg** as the base category of our monads means that the term constructors of the language have already been declared. This is in contrast to presentations of TRSs as **Pre**-signatures where the signature declares both the term and rewrite constructors.

- Secondly, given a TRS, each rewrite rule of arity P contributes the pre-order $(\longrightarrow)$ to the operations of arity P. Unfortunately here we cannot simply write $S(ar(R)) = (\longrightarrow)$ as the operations of a given arity must form an ordered Σ-algebra, and so have to use the free functor F_Σ.

Hence, the signature $S : \mathcal{N}_{Osalg} \to \Sigma\text{-}\mathbf{OAlg}$ for the CTRS Θ is given by

$$\begin{array}{rcll} S(A) & = & F_\Sigma(\downarrow) & A = ar(R) \\ & = & F_\Sigma(\mathbf{0}) & \text{otherwise} \end{array}$$

Again the need to have the operations of a given arity form a ordered Σ-algebra forces us to use the free functor F_Σ. The equations of the theory are used to ensure that no actually new term constructors are declared. These equations fall into two classes: firstly, those setting constructors introduced by F_Σ equal to their interpretation in their arity; secondly, those setting the source and target of rewrite rules must be set equal to u and v respectively.

Formally, the *shape* of the equations are defined by $E : \mathcal{N}_{\Sigma\text{-}\mathbf{OAlg}} \to \Sigma\text{-}\mathbf{OAlg}$ where

$$\begin{array}{rcll} E(A) & = & F_\Sigma(\mathbf{2}) & A = ar(R) \\ & = & F_\Sigma(\mathbf{0}) & \text{otherwise} \end{array}$$

which is the obvious generalisations of the equations used in the presentation of a TRS, incorporating the free functor F_Σ. Next, we define the maps $\sigma_A, \tau_A : E(A) \Rightarrow T_S(A)$ for $A \in \mathcal{N}_{\Sigma\text{-}\mathbf{OAlg}}$ by the following case distinction:

1. $A = ar(R)$. Recall that our signature has declared term constructors s_R and t_R which we want to set to be the terms l and r. Since l, r are elements of the carrier set of $T_\Sigma(P)$, they define a map $(l, r) : \mathbf{2} \to T_\Sigma(P)$ in **Pre**. This in turn defines a map $(l, r)^* : F_\Sigma(\mathbf{2}) \to F_\Sigma(P)$ which when postcomposed with the maps $F_\Sigma(P) \to ar(R) \to T_S(ar(R))$ defines σ.

   Since σ picks out the real source and target l and r of R, τ have to pick out the formal source and target constructors s_R and t_R. We construct a map $F_\Sigma(\mathbf{2}) \to [ar(R), ar(R)] \otimes S(ar(R))$ and then define τ to be the postcomposition of this map with the evident inclusion into $T_S(ar(R))$. Since $S(ar(R)) = F_\Sigma(\downarrow)$, by equation 3 this amounts to giving a map $F_\Sigma(\mathbf{2}) \to F_\Sigma([ar(R), ar(R)] \times \downarrow)$. This map is the image under F_Σ of

   $$\mathbf{2} \cong \mathbf{1} \times \mathbf{2} \longrightarrow [ar(R), ar(R)] \times \downarrow$$

2. $A \neq ar(R)$. Recall that the operations of a given arity have to form an ordered Σ-algebra and this forced us to declare the theory of Σ again. Now we get rid of them by setting these term constructors to be their interpretation the arity. We therefore define σ as above, but replacing $(l, r) : \mathbf{2} \to T_\Sigma(P)$ with $! : \mathbf{0} \to T_\Sigma(P)$.
   As for τ, by similar reasoning to above, we need a map $\mathbf{0} \to [ar(R), A] \times \mathbf{0}$ which we obviously take as the identity (which is the unique map from the initial object)

7.2.3 Correctness

We have shown how to present a CTRS with a single rewrite rule as a generalised algebraic theory over **Σ-OAlg**. The representation was slightly complicated by the need to add in lots of term constructors and then throw them out again, but nevertheless the presentation is still fairly concise. By Theorem 1, this theory has a representing monad which defines the monadic semantics for CTRSs as promised.

However, we still have one thing to do. although we have a monad, is it correct? That is, is the action of the representing monad the same as ordered Σ-algebra-transformer semantics of section 7.1? We prove that this is the case in two stages.

Correctness of the Underlying Monad: Recall that the ordered Σ-algebra-transformer semantics maps an ordered Σ-algebra A to an ordered Σ-algebra $[\![\Theta]\!](A)$ such that the carrier of A equals the carrier of $[\![\Theta]\!](A)$. Therefore, correctness of the representing monad of a CTRS Θ requires the underlying monad $|\mathsf{T}_\Theta|$ on the category of Σ-algebras is the identity. This underlying monad arises as a representing monad of a theory $|\Theta|$ which underlies the theory Θ. To show that this underlying theory is the identity we introduce the notion of a *flat theory.*

Definition 14 (Flat Theories) *A theory $\mathcal{A} = \langle \Sigma, E\rangle$ is flat iff $E(c) = \Sigma(c)$ and*

- *There is a map $\sigma_0 : \Sigma(c) \to c$ and σ arises by postcomposition with the unit of $\mathsf{T}_\mathcal{A}$. That is σ is*
$$E(c) = \Sigma(c) \xrightarrow{\sigma_0} c \xrightarrow{\eta_c} T_\mathcal{A}(c)$$
- *The map τ arises as the composition of the following maps*
$$E(c) = \Sigma(c) \cong 1 \otimes \Sigma(c) \xrightarrow{1_c \otimes 1} [c, c] \otimes \Sigma(c) \longrightarrow T_\mathcal{A}(c)$$

Lemma 7 *Given a flat theory $\mathcal{A}$, its representing monad is the identity*

Proof. We must construct a map $T(c) \to c$ which coequalises σ and τ. Such a map will of course be a cone over the ω-chain defining $T(c)$ and is constructed inductively. The map $\psi_0 : T_0(c) \to c$ is the identity on c. Given a map $\psi_n : T_n(c) \to c$ we now construct a map $\psi_{n+1} : T_{n+1}(c) \to c$. The maps σ_0 and ψ_n can be used to build a map

$$\Sigma[e, T_n(c)] \otimes S(e) \to \Sigma[e, c] \otimes e$$

Next, the canonical map $[e, c] \otimes e \to c$ can be used to build a map

$$\Sigma[e, c] \otimes e \to \Sigma c \to c$$

When taken with the identity on c, this gives our map $T_{n+1}(c) \to c$. That this map coequalises σ and τ is a routine verification. □

Notice that the underlying theory of a CTRS Θ is flat and hence by Lemma 7 is represented by the identity monad.

Correctness of the Rewrites: Next we verify that the representing monad T_Θ adds in the correct number of rewrites. Since the idea of an equation between rewrites is vacuous we consider those rewrites derived by the term algebra construction. This adds a rewrite to the ordered Σ-algebra A for every map $ar(R) \to A$. By definition of $ar(R)$ such maps are maps $F_\Sigma(X+\downarrow) \to A$ which coequalise the diagram defining $ar(R)$. In turn this consists of a map $X+\downarrow \to U_\Sigma(A)$ whose conjugate satisfies the condition mentioned above. Finally such maps are given by a pair of maps

$$\sigma : X \to U_\Sigma(X) \quad \sigma' : \downarrow U_\Sigma(X)$$

But if these maps are to satisfy coequalise the arrows defining $ar(R)$, then σ' is completely determined by σ and we must have $\sigma^*(l) \Rightarrow \sigma^*(r)$ which is exactly when a rewrite is added by the ordered Σ-algebra-transformer semantics.

8 Conclusions and Further Work

This paper has carried on a line of successful research in categorical models of rewriting. We have presented a semantics for CTRSs using **Pre**-enriched monads over categories of ordered Σ-algebras, and can now use our previous results to study modularity of CTRSs from a categorical perspective. The semantics was obtained by systematically instantiating the general theory of Kelly and Power about enriched monads.

There are two natural continuations of this research

- Firstly, the semantics, obtained on a very general level, could be more tailored to the specific needs of CTRSs, along the lines of the term reduction algebra semantics for TRSs in Sect. 4.
- Secondly, we can now investigate modularity problems, using the tools developed in [9, 8]. This proceeds in the following three steps:
  1. First, we needs to develop notions of the properties (e.g. confluence or strong normalisation) we want to investigate on the level of our monads.
  2. Define a semantic notion of structuring operations — in particular, the coproduct of two monads modeling the disjoint union of CTRSs.
  3. Show modularity semantically, e.g. that the coproduct of two confluent monads on **Σ-OAlg** is confluent.

Any modularity results that could be proven here would potentially be a significant improvement, since previous modularity results for CTRSs are rather scarce [12, 13]. Here, categorical rewriting has the potential to lead the way.

References

[1] Franz Baader and Tobias Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.

[2] Neil Ghani. *Adjoint Rewriting.* PhD thesis, University of Edinburgh, 1995.

[3] C. B. Jay. Modelling reductions in confluent categories. In *Proceedings of the Durham Symposium on Applications of Categories in Computer Science*, 1990.

[4] C. Barry Jay and Neil Ghani. The virtues of η-expansion. *Journal for Functional Programming*, 5(2):135– 154, April 1995.

[5] G. M. Kelly. Structures defined by finite limits in the enriched context I. *Cahiers de Topologie et Géométrie Différentielle*, XXIII(1):3–40, 1982.

[6] G. M. Kelly and A. J. Power. Adjunctions whose counits are coequalizers, and presentations of finitary monads. *Journal for Pure and Applied Algebra*, 89:163–179, 1993.

[7] J. W. Klop. Term rewriting systems. In S. Abramsky, Dov M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2 (Background: Computational Structures), pages 1–116. Oxford University Press, 1992.

[8] C. Lüth and N. Ghani. Monads and modular term rewriting. In E. Moggi and G. Rosolini, editors, *Category Theory and Computer Science*, number 1290 in Lecture Notes in Computer Science, pages 69– 86, Santa Margherita Ligure, Italy, September 1997. Springer Verlag.

[9] Christoph Lüth. *Categorical Term Rewriting: Monads and Modularity.* PhD thesis, University of Edinburgh, 1998.

[10] S. Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics.* Springer Verlag, 1971.

[11] Ernest G. Manes. *Algebraic Theories*, volume 26 of *Graduate Texts in Mathematics.* Springer Verlag, 1976.

[12] Aart Middeldorp. Modular properties of conditional term rewriting systems. *Information and Computation*, 104:110– 158, 1993.

[13] Aart Middeldorp. Completeness of combinations of conditional constructor systems. *Journal for Symbolic Computation*, 17:3– 21, 1994.

[14] A. J. Power. Enriched Lawvere theories. Available at `http://www.dcs.ed.ac.uk/ ajp`, 1999.

[15] E. Robinson. Variations on algebra: monadicity and generalisations of equational theories. Technical Report 6/94, Sussex Computer Science Technical Report, 1994.

[16] D. E. Rydeheard and J. G. Stell. Foundations of equational deduction: A categorical treatment of equational proofs and unification algorithms. In *Category Theory and Computer Science*, number 283 in Lecture Notes in Computer Science, pages 114–139. Springer Verlag, 1987.

[17] R. A. G. Seely. Modelling computations: A 2-categorical framework. In *Proceedings of the Second Annual Symposium on Logic in Computer Science*, pages 65–71, 1987.

[18] John G. Stell. Modelling term rewriting systems by Sesqui-categories. Technical Report TR94-02, Keele Unversity, January 1994.