Compositional Term Rewriting: An Algebraic Proof of Toyama's Theorem

Christoph Lüth [*]

Universität Bremen — FB 3
Postfach 330440
28334 Bremen
Germany
`cxl@informatik.uni-bremen.de`
Phone: +49 (421) 218 7585, Fax: +49 (421) 218 3054

Abstract. This article proposes a compositional semantics for term rewriting systems, i.e. a semantics preserving structuring operations such as the disjoint union. The semantics is based on the categorical construct of a monad, adapting the treatment of universal algebra in category theory to term rewriting systems.

As an example, the preservation of confluence under the disjoint union of two term rewriting systems is shown, obtaining an algebraic proof of Toyama's theorem, generalised slightly to term rewriting systems introducing variables on the right-hand side of the rules.

1 Introduction

Term rewriting has long been recognised as an important tool for reasoning in algebraic (and other) specifications. Specifications occurring in practice tend to be very large, so structuring operations are used to construct large specifications from smaller ones ([ST88, EM85] etc.). Unfortunately the interaction of the structuring operations with term rewriting systems is not at all clear, since there has been no way to obtain the semantics of a term rewriting system for a structured specification by composing the semantics of the term rewriting systems for its component specifications.

Of particular interest is the question how properties like confluence, termination and completeness are preserved by these structuring operations.

In this article, following some preliminaries, we will show how to obtain a compositional semantics by generalising the categorical treatment of universal algebra, and argue why this semantics can be rightly called "compositional" (Section 2). We will then consider the disjoint union of two term rewriting systems (Section 3), corresponding to the coproduct of two monads, show how to construct this coproduct, and use this construction to show that the coproduct of two confluent monads is confluent.

---

[*] This research was supported by EPSRC grant GR/H73103 and the COMPASS basic research working group while the author was affiliated with Edinburgh University.

1.1 Preliminaries

We assume a working knowledge of term rewriting systems, and category theory as gained from the first five chapters of [Mac71] (the notation and terminology of which will be used here, and to which we will often refer). Although this work involves enriched categories, no knowledge of them is either assumed or even necessary for a basic understanding of what follows;[2] a gentler introduction into enriched category theory than the somewhat demanding standard text [Kel82] is [Bor94, Chapter 6].

This article is an extract of the author's forthcoming thesis [Lü96]. Without referring to it explicitly in the following, the thesis will present the material of the present article, some of which can only be adumbrated due to length limitations, in far more detail.

I would like to thank Don Sannella, Stefan Kahrs and Neil Ghani for many stimulating discussions during the preparation of this work, and Burkhardt Wolff and two of the anonymous referees for helpful comments on the presentation.

2 Using Monads to Model Term Rewriting

2.1 Why Monads?

Recall that 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 $\mu{\cdot}T\mu = \mu{\cdot}\mu_T$.

To motivate the use of monads to model term rewriting systems, consider how a monad $\mathsf{T} = \langle T, \eta, \mu \rangle$ on the category **Set** of all (small) sets and functions between them captures the way terms are built:

- for a set X, we can consider TX to be the term algebra (the set of terms built over the variables X);
- then the unit $\eta_X : X \to TX$ describes how to make elements of X into variables (i.e. terms);
- and the multiplication $\mu : TTX \to TX$ describes how to substitute terms for variables: given a set $Y = \{ y_1, \ldots, y_n \}$ of variables and $t_1, \ldots, t_n \in TX$ (n terms built over X), there is an obvious map $\sigma : Y \to TX$ with $\sigma(y_i) \overset{def}{=} t_i$; then for $s \in TY$ (a term built over Y), the substitution $s[t_i/y_i]$ of y_i with t_i, written $s[t_1, \ldots, t_n]$ in the following, is defined as $s[t_1, \ldots, t_n] \overset{def}{=} \mu_X(T\sigma(s))$.

The monad laws mean that this substitution is associative, and that substituting a term into a variable, and a variable for itself, yields the identity (in an informal notation, $x[t/x] = t$ and $t[x/x] = t$). It turns out this is all one needs for universal algebra (see [Man76]).

By adding a reduction structure, we will extend this to term rewriting systems, but let us first see how to treat signatures.

---

[2] At some points, footnotes will point out technical details and side issues which the casual reader can safely ignore.

2.2 Modelling Signatures

In the following, let Ω be a (single sorted) signature or operator domain, given by a set Ω_0 of operators equipped with a function $ar : \Omega_0 \to \mathbb{N}$ giving for each operator f its *arity*, $ar(f)$. If X is a set, the term algebra $T_\Omega(X)$ contains terms over X, formed either by taking a variable $x \in X$ to the term $\texttt{'}x \in T_\Omega(X)$[3], or by applying an operation $\omega \in \Omega$, then $\omega\texttt{(}t_1, \ldots, t_n\texttt{)} \in T_\Omega(X)$ if $t_1, \ldots, t_n \in T_\Omega(X)$ and $ar(\omega) = n$. (Here and in the following, we use $\omega \in \Omega$ as an abbreviation for $\omega \in \Omega_0$.) A signature Ω gives rise to a monad T_Ω on **Set**:

Definition 1. Given a signature Ω, the monad $\mathsf{T}_\Omega \overset{def}{=} \langle T_\Omega, \eta, \mu \rangle$ is defined as follows:

- the action maps a set X to the term algebra $T_\Omega(X)$, and a morphism $f : X \to Y$ to its *lifting* $f^* : T_\Omega(X) \to T_\Omega(Y)$ defined inductively on the terms as follows:

$$f^*(\texttt{'}x) \overset{def}{=} \texttt{'}fx$$
$$f^*(e\texttt{(}t_1, \ldots, t_n\texttt{)}) \overset{def}{=} e\texttt{(}f^*(t_1), \ldots, f^*(t_n)\texttt{)}$$

- the unit η_X maps $x \in X$ to $\texttt{'}x \in T_\Omega(X)$;
- the substitution $\mu_X : T_\Omega(T_\Omega(X)) \to T_\Omega(X)$ is defined as follows:

$$\mu_X(e\texttt{(}t_1, \ldots, t_n\texttt{)}) \overset{def}{=} e\texttt{(}\mu_X(t_1), \ldots, \mu_X(t_n)\texttt{)}$$
$$\mu_X(\texttt{'}x) \overset{def}{=} x$$

One routinely checks (using familiar structural induction on the terms) the functoriality of the action T_Ω, naturality of η and μ and that they satisfy the monad laws.

2.3 Modelling Term Rewriting Systems

The above extends to an equational presentation (Ω, E) (i.e. a signature Ω with equations E); then $T_{(\Omega,E)}$ maps X to the quotient term algebra (see [Man76] for the details). To model term rewriting systems, we need to extend the set TX with a reduction structure between the elements. To model many-step reductions, this structure needs to be transitive and reflexive. Hence, TX should be a category or preorder (a transitive and reflexive binary relation, or equivalently a category with at most one morphism between any two objects), the objects of which are the terms and the morphisms of which model the reduction; in the former case, we distinguish different reductions between the same terms ("named reductions").

Since the action T is an endofunctor, the variables X have to form a category or preorder as well. This corresponds to reductions between variables (called

---

[3] The reader may wonder about the reason for the notation for variables; later on we will consider terms built over terms, and we will have to distinguish between terms like $\texttt{'}x$ and $\texttt{''}x$, or even $\texttt{'G('}x\texttt{)}$ and $\texttt{G(''}x\texttt{)}$.

variable rewrites). In other words, we should be able to assume more about variables than merely their existence: for if a context like $X = \{ x, y, z \}$ is just a way of modelling the assumption that there are three entities x, y and z to build terms with, then to model reductions there has to be a way of assuming that there are reductions between them.

This entails a more general form of rewrite rules, called *generalised rewrite rules*, in which one can impose conditions on the variables.

Definition 2. A *generalised rewrite rule* in a signature Ω is given by a triple (X, l, r), written as $(X \vdash l \to r)$, where $X = (X_0, \leq)$ is a finite preorder and $l, r \in T_\Omega(X_0)$ are terms.

For example, given the preorder $X \overset{def}{=} (\{ x, y, z \}, \leq)$, ordered as $x \leq z, y \leq z$ and the signature $\Omega = \{ \mathtt{F}, \mathtt{G} \}$ (with `F` binary and `G` unary), the rule $(X \vdash \mathtt{F('}x\mathtt{, '}y\mathtt{)} \to \mathtt{G('}z\mathtt{)})$ means that only if we can instantiate x and y with terms t_1 and t_2 which have a common reduct t_3 (which is the instantiation of z), is there a reduction from $\mathtt{F(}t_1, t_2\mathtt{)}$ to $\mathtt{G(}t_3\mathtt{)}$.

It should be emphasized that this is a conservative extension of the traditional definition of rewrite rules: they are just a special case of Def. 2 with the preorder being the identity relation. The greater generality of Def. 2 is not actually needed until one constructs a mapping from monads to term rewriting systems (see Sec. 2.4). Note also that we do not require the variables occurring in the right-hand side r to occur in the left-hand side l.

We are now going to show how to model the reduction order on the terms generated by a set of generalised rewrite rules by a preorder. The category **Pre** in which we will model term rewriting systems has as its objects preorders, and as morphisms $f : (X, \leq) \to (Y, \preccurlyeq)$ between two preorders maps $f : X \to Y$ respecting the preorder (i.e. $x \leq y \Rightarrow fx \preccurlyeq fy$). Preorder morphisms can be ordered pointwise: given $f, g : (X, \leq) \to (Y, \preccurlyeq)$ then $f \leq g$ iff $\forall x \in X. fx \preccurlyeq gx$. Hence the set of preorder morphisms between two preorders $(X, \leq)$ and $(Y, \preccurlyeq)$ forms in turn a preorder. One says that the category **Pre** is *closed*.[4]

The crucial categorical insight when extending Def. 1 is that it is not sufficient to merely define a monad on **Pre**, but that this monad has to respect the closed structure of **Pre** (i.e. the order on the morphisms); in other words, it has to be an *enriched* monad in the sense of [Kel82].

The analogue of the term algebra for a term rewriting system is the *term reduction algebra*. It is freely generated by three rules (in Table 1 below):

- every variable rewrite is a reduction in the term reduction algebra (rule [VAR]);
- the operations in Ω have to preserve the reduction (rule [PRE]; repeated application of this rule builds contexts);

---

[4] To be precise, *monoidal closed* [Kel82]. This is a particular instance of a general phenomenon: for example, the set of functors between two categories $\mathcal{X}$ and $\mathcal{Y}$ are the objects of a category the morphisms of which are the natural transformations between the functors: the functor category $[\mathcal{X}, \mathcal{Y}]$.

- and the variables in a rewrite rule can be instantiated with terms, provided they satisfy the variable rewrites (rule [Inst]).

Definition 3. Given a term rewriting system $\Theta = (\Omega, R)$ and a preorder $X = (X_0, \leq)$, the *term reduction algebra* on X is the smallest preorder $(T_\Omega(X_0), \leq)$ on the terms over X_0 satisfying the implications in Table 1, where $t[t_1, \ldots, t_n]$ is the substitution of the n variables in $t \in T_\Omega(Y)$ with terms $t_1, \ldots, t_n$ defined above (in Sec. 2.1).

$$[\textsc{Var}] \quad \frac{x \leq y}{\text{'}x \leq \text{'}y} \; x, y \in X_0$$

$$[\textsc{Pre}] \quad \frac{t_1 \leq s_1, \ldots, t_n \leq s_n}{\omega(t_1, \ldots, t_n) \leq \omega(s_1, \ldots, s_n)} \; \omega \in \Omega, ar(\omega) = n$$

$$[\textsc{Inst}] \quad \frac{\begin{array}{l}(Y \vdash l \rightarrow r) \in R,\ Y = (\{\, y_1, \ldots, y_n \,\}, \preccurlyeq) \\ \forall i = 1, \ldots, n \, \forall j = 1, \ldots n.\ y_i \preccurlyeq y_j \Rightarrow t_i \leq t_j\end{array}}{l[t_1, \ldots, t_n] \leq r[t_1, \ldots, t_n]} \; t_1, \ldots, t_n \in T_\Omega(X)$$

Table 1. Definition of the Reduction Preorder.

Definition 4. Given a term rewriting system $\Theta = (\Omega, R)$, the monad $\mathsf{T}_\Theta = \langle T_\Theta, \eta, \mu \rangle$ is defined as follows:

- its action T_Θ maps a preorder $(X, \leq)$ to the term reduction algebra $(T_\Omega(X), \leq)$ from Def. 3, and a preorder morphism f to its lifting f^* from Def. 1;
- its unit η and multiplication μ are as in Def. 1.

It has to be shown (by structural induction) that T_Θ is a **Pre**-enriched functor by showing that f^* is a preorder morphism (i.e. $s \leq t \Rightarrow f^*(s) \leq f^*(t)$), and that the lifting respects the pointwise order on the morphisms (i.e. if $\forall x \in X \,.\, fx \leq gx$ then $\forall t \in T_\Omega(X) \,.\, f^*(t) \leq g^*(t)$); and further that η and μ as defined above are preorder morphisms (the former is trivial, the latter requires another easy induction). Then the monad laws follow from Def. 1, and T_Θ is a **Pre**-enriched monad.

2.4 Compositionality

As observed by Goguen and Burstall [GB92], most structuring operations for algebraic specifications are colimits, either in the category of syntactic presentations (here, term rewriting systems), or in the category of semantic representations (here, monads). "Compositionality of the semantics" means that the mapping from the syntax to the semantics should preserve these colimits; for this, it is sufficient that the mapping is (or extends to) a left adjoint functor, and use the general fact that left adjoints preserve colimits [Mac71, Section V.5]. We will omit the details of this construction here since they are not needed in the following.

2.5 Finiteness

A signature as defined above in which the arities of all the operations are finite is called *finitary.* A monad T_{Ω} arising from a finitary signature Ω preserves a special kind of colimit, called filtered or directed (see [Mac71, Section IX.1]). In fact, we have an even stronger result: the monad T_{Θ} is *strongly finitary*, meaning it preserves *weakly filtered diagrams*, where for any two objects X, Y there is an object Z and morphisms $f : X \to Z$ and $g : Y \to Z$.

3 Disjoint Union and the Preservation of Confluence

The disjoint union of two term rewriting systems is given by the coproduct in the category **TRS** (just as the disjoint union of two sets is given by the coproduct in **Set**); by the compositionality, the theory of the coproduct is the same as the coproduct of the theories, i.e. $\mathsf{T}_{\Theta_1+\Theta_2} \cong \mathsf{T}_{\Theta_1}+\mathsf{T}_{\Theta_2}$. In this section, we are going to construct the coproduct on the right side of this isomorphism — the coproduct of two monads — and will use this to give a categorical account of Toyama's theorem by showing that the coproduct of two confluent monads (monads arising from confluent term rewriting systems) is confluent as well.

We will also switch to the more general case of monads on the category **Cat** of all small categories, of which the case above (preorders) is a special case. Given $\mathsf{T}_1 = \langle T_1, \eta_1, \mu_1 \rangle, \mathsf{T}_2 = \langle T_2, \eta_2, \mu_2 \rangle$ on **Cat**, we define their coproduct by its universal property: a monad $\mathsf{T}_{1+2} = \langle T, \eta, \mu \rangle$ such that there are two monad morphisms $\iota_1 : \mathsf{T}_1 \to \mathsf{T}_{1+2}, \iota_2 : \mathsf{T}_2 \to \mathsf{T}_{1+2}$, and for any other monad $\mathsf{S} = \langle S, \eta_S, \mu_S \rangle$ with monad morphisms $\alpha : \mathsf{T}_1 \to \mathsf{S}$, $\beta : \mathsf{T}_2 \to \mathsf{S}$, there is a unique monad morphism $[\alpha, \beta] : \mathsf{T}_{1+2} \to \mathsf{S}$ such that $\alpha = [\alpha, \beta] \cdot \iota_1$ and $\beta = [\alpha, \beta] \cdot \iota_2$.[5]

This coproduct monad is defined pointwise. For every category $\mathcal{X}$, the action T will map $\mathcal{X}$ to the colimit of a diagram describing the combinations of T_1 and T_2.

To motivate the construction, consider two monads on the category **Set** given by signatures Ω_1, Ω_2. Then the coproduct of T_{Ω_1} and T_{Ω_2} should map a set X to the set of all terms built from operations of $\Omega_1 + \Omega_2$. Terms from $T_{\Omega_1+\Omega_2}(X)$

---

[5] We have omitted the definitions of the category **TRS** of term rewriting systems and *Mon*(**Cat**) of monads on **Cat** here. The morphisms of the latter are natural transformations respecting the unit and multiplication (*monad morphisms*, see [BW85, Section 3.6]). It is actually a **Cat**-enriched category (or 2-category [KS74]), so the colimit has to have an additional colimiting property on 2-cells: given two pairs of monad morphisms $\alpha, \alpha' : \mathsf{T}_1 \to \mathsf{S}$, $\beta, \beta' : \mathsf{T}_1 \to \mathsf{S}$ and two *modifications* (see [KS74]) $\gamma : \alpha \to \alpha', \delta : \beta \to \beta'$, there is a modification $[\gamma, \delta] : [\alpha, \beta] \to [\alpha', \beta']$.

can be decomposed into *layers* of terms from $T_{\Omega_1}(X)$ and $T_{\Omega_2}(X)$ (*aliens* or *principal subterms*) (see e.g. [KM+94] for the definition and the notation). We forego the introduction of holes and principal subterms, and take the notion of a layer as primitive in so far as every application of the functor T_1 or T_2 corresponds to one layer; i.e. it takes $t_1, \ldots, t_n$ to $C[\![t_1, \ldots, t_n]\!]$. This is because we can build term algebras on top of term algebras; for example, the elements of $T_{\Omega_1}(T_{\Omega_2}(X))$ correspond (roughly) to terms of rank two. However, in term algebras like $T_{\Omega_2}(T_{\Omega_1}(T_{\Omega_2}(X)))$ we will have terms from $T_{\Omega_2}(X)$ treated as variables in $T_{\Omega_1}(X)$ inserted into terms of $T_{\Omega_2}(X)$, which should be equivalent to a term from $T_{\Omega_2}(X)$; for example, if $F \in \Omega_2$, then the terms $F\texttt{(''}F\texttt{('}x\texttt{))}$ and $F\texttt{(}F\texttt{('}x\texttt{))}$ should be equivalent. This identification is called "collapsing layers".

Hence, the coproduct will be given by the disjoint union of all the term algebras

$$\begin{aligned} T_{\Omega_1+\Omega_2}(X) \stackrel{def}{=} \; & X + T_{\Omega_1}(X) + T_{\Omega_2}(X) + T_{\Omega_1}(T_{\Omega_2}(X)) + T_{\Omega_2}(T_{\Omega_1}(X)) + \\ & T_{\Omega_1}(T_{\Omega_2}(T_{\Omega_1}(X))) + T_{\Omega_2}(T_{\Omega_1}(T_{\Omega_2}(X))) + \ldots \end{aligned}$$

quotiented by a suitable equivalence relation, affected by the unit and the multiplication: the unit identifies all the variables from X in the different term algebras, and the multiplication collapses layers as described above. We arrive at the definition of the action T as the colimit of a diagram which has all the combinations of T_1 and T_2 as objects, and all morphisms which can be formed using the unit and multiplication of the two monads as morphisms. We are now going to define this diagram formally.

3.1 The Functor $D_{\mathcal{X}}$

In the following, the graph $\mathcal{G}$ will define the diagram giving all possible combinations of η and μ in a generic fashion (i.e. independent of $\mathcal{X}$). The functor $D_{\mathcal{X}}$ will map this scheme to a specific diagram over $\mathcal{X}$. We use the alphabet $\mathcal{L} \stackrel{def}{=} \{1, 2\}$, and the words $W \stackrel{def}{=} \mathcal{L}^*$ over that alphabet. The functor $D_{\mathcal{X}}$ will be a functor from the *free category* $\mathcal{F}(\mathcal{G})$ of the graph $\mathcal{G}$, which has vertices of $\mathcal{G}$ as objects and paths in $\mathcal{G}$ as morphisms (see [Mac71, pg.50]), into **Cat**. First, for all $w \in W$ the functor $T^w : \mathbf{Cat} \to \mathbf{Cat}$ is defined as follows:

$$\begin{aligned} T^{\varepsilon} &\stackrel{def}{=} 1_{\mathbf{Cat}} \\ T^{jv} &\stackrel{def}{=} T_j T^v \quad \text{where} \quad j \in \mathcal{L}, v \in W \end{aligned}$$

The graph $\mathcal{G}$ has two different classes of vertices, $E_E(\mathcal{G})$ and $E_M(\mathcal{G})$ (corresponding to units and multiplication respectively), which are explicitly distinguished since we need to refer to them later on:

$$\begin{array}{lrcl} \text{Vertices:} & V(\mathcal{G}) & \stackrel{def}{=} & W \\ \text{Edges:} & E_E(\mathcal{G}) & \stackrel{def}{=} & \{\, \mathtt{e}^w_{j,v} : wv \to wjv \mid w, v \in W, j \in \mathcal{L} \,\} \\ & E_M(\mathcal{G}) & \stackrel{def}{=} & \{\, \mathtt{m}^w_{j,v} : wjjv \to wjv \mid w, v \in W, j \in \mathcal{L} \,\} \\ & E(\mathcal{G}) & \stackrel{def}{=} & E_E(\mathcal{G}) \cup E_M(\mathcal{G}) \end{array}$$

The graphs $\mathcal{G}_E$ and $\mathcal{G}_M$ are the subgraphs of $\mathcal{G}$ with the same edges, but only $E_E(\mathcal{G})$ and $E_M(\mathcal{G})$, respectively, as vertices.

For a category $\mathcal{X} \in \mathbf{Cat}$, we define the functor $D_{\mathcal{X}} : \mathcal{F}(\mathcal{G}) \to \mathbf{Cat}$ by mapping the vertices and edges of $\mathcal{G}$ to the underlying graph of $\mathbf{Cat}$ as follows:

$$\begin{array}{rrcl} \text{On the vertices:} & D_{\mathcal{X}}(w) & \overset{def}{=} & T^w(\mathcal{X}) \\ \text{On the edges:} & D_{\mathcal{X}}(\mathtt{e}^w_{j,v}) & \overset{def}{=} & T^w(\eta_{j,T^v(\mathcal{X})}) \\ & D_{\mathcal{X}}(\mathtt{m}^w_{j,v}) & \overset{def}{=} & T^w(\mu_{j,T^v(\mathcal{X})}) \end{array}$$

3.2 The Coproduct Monad

The coproduct monad $\mathsf{T}_{1+2} = \langle T, \eta, \mu \rangle$ is defined as follows:

- The action T maps a category $\mathcal{X}$ to the colimit $colim\, D_{\mathcal{X}}$ of the functor $D_{\mathcal{X}}$. Given a functor $F : \mathcal{X} \to \mathcal{Y}$, we can precompose the colimiting cone over $D_{\mathcal{Y}}$ with F, and since all components of $D_{\mathcal{X}}$ and $D_{\mathcal{Y}}$ are natural transformations, this yields a cone over $D_{\mathcal{X}}$, and hence by the colimiting property of $colim D_{\mathcal{X}}$, there is a functor $!_F : colim\, D_{\mathcal{X}} \to colim\, D_{\mathcal{Y}}$. The action maps F to this unique functor $!_F$. Similarly, a natural transformation $\alpha : F \Rightarrow G$ induces a natural transformation $!_\alpha :!_F \Rightarrow !_G$, which describes the action on natural transformations.[6]
- The unit η is given by the component of the colimiting cone over $D_{\mathcal{X}}$ at $\mathcal{X}$;
- The multiplication μ relies on the fact that T_1 and T_2 preserve weakly filtered colimits, and that $D_{\mathcal{X}}$ is weakly filtered. $TT\mathcal{X}$ is a colimit of a functor which maps $w \in W$ to $T^w T\mathcal{X}$, then $T^w T\mathcal{X} = T^w colim\, D_{\mathcal{X}} \cong colim\, T^w D_{\mathcal{X}}$; now $T^w D_{\mathcal{X}}$ is again a part of $D_{\mathcal{X}}$, and we can form a cone over that diagram by using the colimiting cone over $D_{\mathcal{X}}$, which induces a morphism $! : TT\mathcal{X} \to T\mathcal{X}$ (by the colimiting property of $TT\mathcal{X}$).

Verifying the monad laws and the universal property is a case of diagram chasing the details of which would take us well outside the scope of this paper.

Category theory also tells us how the colimit is computed using coproducts and coequalizers. Applying the dual of Theorem 2 in [Mac71, pg. 109], the colimit of $D_{\mathcal{X}}$ is given by the coequalizer of Diagram 1, where on the left side, we have

$$\begin{tikzcd} \displaystyle\coprod_{d:T^u\mathcal{X} \to T^v\mathcal{X} \in D_{\mathcal{X}}} T^u\mathcal{X} \arrow[r, shift left, "F"] \arrow[r, shift right, "G"'] & \displaystyle\coprod_{w \in W} T^w\mathcal{X} \end{tikzcd} \tag{1}$$

for any morphism $d : T^u\mathcal{X} \to T^v\mathcal{X}$ in the image of $D_{\mathcal{X}}$ (with $u, v \in W$) the component $T^u\mathcal{X}$ of the coproduct, and the two functors F and G are defined as $F(T^u\mathcal{X}) \overset{def}{=} \iota_u(T^u\mathcal{X})$, $G(T^u\mathcal{X}) \overset{def}{=} \iota_v(d(T^u\mathcal{X}))$ where ι_u and ι_v are the injections into the coproduct on the right.

In general, given two categories $\mathcal{X}$ and $\mathcal{Y}$, and two functors $F, G : \mathcal{X} \to \mathcal{Y}$ between them, their coequalizer is a category $\mathcal{Z}$, and a functor $Q : \mathcal{Y} \to \mathcal{Z}$, where $\mathcal{Z}$ is defined as follows (see [Gra74, Chapter I.1]):

---

[6] This follows since all components of $\mathcal{G}$ are 2-natural transformations, and from the colimit property of $colim\, D_{\mathcal{X}}$ on 2-cells.

- The objects are the objects of $\mathcal{Y}$, quotiented by the equivalence closure $\equiv$ of the relation $\sim$ defined as $x \sim y \Leftrightarrow \exists z \in \mathcal{X} \,.\, Fz = x, Gz = y$.
- The morphisms are sequences $<f_1, \ldots, f_n>$ of morphisms f_i from $\mathcal{Y}$ such that $\delta_s(f_i) \equiv \delta_t(f_{i-1})$ (where for a morphism α, $\delta_s(\alpha)$ is its source, and $\delta_t(\alpha)$ its target), quotiented by the smallest equivalence relation $\equiv$ compatible with composition in $\mathcal{Y}$ such that $<f, g> \equiv <g{\cdot}f>$ if f, g are composable in $\mathcal{Y}$, and $<Fh> \equiv <Gh>$ for all morphisms h in $\mathcal{X}$.

3.3 Preservation of Confluence for Coequalizers of Functors

We are now going to find general conditions under which the coequalizing category $\mathcal{Z}$ above is confluent. We will obtain a general result which we can subsequently specialise to Diagram 1. First of all, we have to define confluence for categories and monads: the first is a straightforward generalization of the usual definition, the second has to take variable rewrites into account, spans of which have to be completed by variable rewrites.

Definition 5. A category $\mathcal{C}$ is *confluent* if for any two morphisms $\alpha : x \to x_1, \beta : x \to x_2$ there are morphisms $\gamma : x_1 \to z, \delta : x_2 \to z$ such that $\gamma{\cdot}\alpha = \delta{\cdot}\beta$.

A monad $\mathsf{T} = \langle T, \eta, \mu \rangle$ on **Cat** is *confluent* if $T\mathcal{X}$ is confluent whenever $\mathcal{X}$ is.

By an easy induction, one shows that this coincides with the usual definition of confluence (i.e. if a category $\mathcal{X}$ is confluent and the term rewriting system Θ is confluent, then $T_\Theta(\mathcal{X})$ is confluent).

Let us try to characterize sufficient conditions for the confluence of the coequalising category $\mathcal{Z}$. Since the morphisms in $\mathcal{Z}$ are equivalence classes of sequences of morphisms in $\mathcal{Y}$, we can complete any span of these sequences if for any two morphisms in $\mathcal{Y}$ the source of which is equal under the coequalising functor Q we can find another two morphisms in $\mathcal{Y}$ such that their composition is equal under Q — a one-step completion of $\mathcal{Y}$ with respect to the functor Q:

Definition 6. Given a functor $Q : \mathcal{Y} \to \mathcal{Z}$, the category $\mathcal{Y}$ has the *one-step completion property with respect to* Q, written $\mathcal{Y} \models_Q \diamond$, if for all morphisms $\alpha : x \to x'$, $\beta : y \to y'$ in $\mathcal{Y}$ such that $Qx = Qy$ there are morphisms $\gamma : v \to v', \delta : w \to w'$ in $\mathcal{Y}$ such that $Q\gamma{\cdot}Q\alpha = Q\delta{\cdot}Q\beta$.

Lemma 7. *Let $Q : \mathcal{Y} \to \mathcal{Z}$ be the coequalizer of two functors $F, G : \mathcal{X} \to \mathcal{Y}$ in* **Cat**. *If $\mathcal{Y}$ is confluent and $\mathcal{Y} \models_Q \diamond$, then $\mathcal{Z}$ is confluent.*

Proof. Given two morphisms $\alpha = [<\alpha_1, \ldots, \alpha_n>]$ and $\beta = [<\beta_1, \ldots, \beta_m>]$ in $\mathcal{Z}$ with the same source (i.e. $Q(\delta_s(\beta_1)) = Q(\delta_s(\alpha_1))$). Then (since $\mathcal{Y} \models_Q \diamond$) there are β_1', α_1' such that $Q(\beta_1'){\cdot}Q(\alpha_1) = Q(\alpha_1'){\cdot}Q(\beta_1)$. By induction on the length n and m of α and β, respectively, we obtain completions $\alpha' \overset{def}{=} [<\alpha_1^{(m)}, \ldots, \alpha_n^{(m)}>]$, $\beta' \overset{def}{=} [<\beta_1^{(n)}, \ldots, \beta_m^{(n)}>]$ such that $\beta'{\cdot}\alpha = \alpha'{\cdot}\beta$. □

To show $\mathcal{Y} \models_Q \diamond$ and hence confluence of $\mathcal{Z}$, we introduce the notion of a *witness*. The idea is that for two morphisms the sources of which are equal under Q, we find two equivalent morphisms which form a span in $\mathcal{Y}$ which by confluence of $\mathcal{Y}$ can be completed. Given a functor $Q : \mathcal{Y} \to \mathcal{Z}$, we say an object $x \in \mathcal{Y}$ is a *witness* of an object $y \in \mathcal{Y}$ with respect to Q, written $x \operatorname{wit}_Q y$, if $Qx = Qy$ and for all morphisms $\beta : y \to y'$ in $\mathcal{Y}$, there is a morphism $\alpha : x \to x'$ such that $Q\alpha = Q\beta$; in other words, for all morphisms the source of which is y there is an equivalent one the source of which is x. If now any two equivalent objects have a common witness, then $\mathcal{Z}$ will be confluent, thus to prove of confluence it is sufficient to prove the existence of a common witness:

Lemma 8. *Given the coequalizer $Q : \mathcal{Y} \to \mathcal{Z}$ of two functors $F, G : \mathcal{X} \to \mathcal{Y}$. If $\mathcal{Y}$ is confluent and for all $x, y \in \mathcal{Y}$ s.t. $Qx = Qy$ there is a common witness $z \in \mathcal{Y}$ such that z wit$_Q$ x and z wit$_Q$ y then $\mathcal{Z}$ is confluent.*

Proof. We show that $\mathcal{Y} \models_Q \diamond$ and use Lemma 7: given any two morphisms $\alpha : x \to x'$, $\beta : y \to y'$ such that $Qx = Qy$, there is a common witness $z \in \mathcal{Y}$, hence there are $\alpha' : z \to z'$, $\beta' : z \to z''$ such that $Q\alpha' = Q\alpha, Q\beta' = Q\beta$, which by confluence of $\mathcal{Y}$ have a completion $\gamma : z' \to u, \delta : z'' \to u$ such that $\gamma \cdot \alpha' = \delta \cdot \beta'$, hence (since Q preserves composition) $Q\gamma \cdot Q\alpha' = Q\delta \cdot Q\beta'$, and $Q\gamma \cdot Q\alpha = Q\delta \cdot Q\beta$. □

The existence of a common witness is shown by constructing a binary relation $\preccurlyeq$ on the objects of $\mathcal{Y}$, compatible with the equivalence relation, such that $x \preccurlyeq y$ implies that x is a witness of y. Such a relation is called a *witness relation*:

Definition 9. Given the coequalizer $Q : \mathcal{Y} \to \mathcal{Z}$ of two functors $F, G : \mathcal{X} \to \mathcal{Y}$, a binary relation $\preccurlyeq$ on the objects of $\mathcal{Y}$ is a *witness relation* if it satisfies the following four properties:

(i) $x \preccurlyeq y \Rightarrow x \operatorname{wit}_Q y$;
(ii) For all $x \in \mathcal{X}$ there is $y \in \mathcal{Y}$ such that $y \preccurlyeq Fx$ and $y \preccurlyeq Gx$;
(iii) $\preccurlyeq$ is transitive;
(iv) $\preccurlyeq$ has *filtered lower bounds*: if $x \preccurlyeq z$ and $y \preccurlyeq z$, then there is $w \in \mathcal{Y}$ such that $w \preccurlyeq x$ and $w \preccurlyeq y$.

Lemma 10. *Given the coequalizer $Q : \mathcal{Y} \to \mathcal{Z}$ of two functors $F, G : \mathcal{X} \to \mathcal{Y}$, if there is a witness relation $\preccurlyeq$ on the objects of $\mathcal{Y}$, and $\mathcal{Y}$ is confluent, then $\mathcal{Z}$ is confluent.*

Proof. Given $x \equiv y$, we show there is a z such that $z \preccurlyeq x$ and $z \preccurlyeq y$, then by the first property $z \operatorname{wit}_Q x$ and $z \operatorname{wit}_Q y$, and by Lemma 8 $\mathcal{Z}$ is confluent. If $x \equiv y$, then

- either there is a $u \in \mathcal{X}$ such that $Fu = x, Gu = y$, then by the second property, there is $z \in \mathcal{Y}$ such that $z \preccurlyeq x$ and $z \preccurlyeq y$;

- or they are made equivalent by the equivalence closure, of which the symmetric and reflexive closure are trivial, so only the transitive closure remains: there is $z \in \mathcal{Y}$ s.t. $x \equiv z, z \equiv y$, and we can assume that there is $u \in \mathcal{Y}$ s.t. $u \precsim x, u \precsim z$ and $v \in \mathcal{Y}$ s.t. $v \precsim z, v \precsim y$. Since $\precsim$ has filtered lower bounds, there is a $w \in \mathcal{Y}$ such that $w \precsim u$ and $w \precsim v$, and by transitivity $w \precsim x$ and $w \precsim y$, hence $u \precsim x$ and $u \precsim y$.

□

3.4 Confluence of the Coproduct Monad

To show that T_{1+2} is confluent, it now suffices to construct a witness relation for $\coprod_{w \in W} T^w \mathcal{X}$ with respect to the coequalizer Q of F and G in Diagram 1; by an easy induction on w, all $T^w \mathcal{X}$ are confluent and thus the coproduct is confluent as well, and we can apply Lemma 10. We will need to make the following two assumptions about the two monads T_1 and T_2: that they are non-expanding, and that their units are monomorphisms (in **Cat**); in particular, that their object function is injective. The first corresponds to the restriction to term rewriting systems in which the left-hand side of the rules is not a variable; and the second is more technical and means that two terms $'x$ and $'y$ are equal only if x and y are equal.

We first define non-expanding monads:

Definition 11. A functor $F : \mathcal{X} \to \mathcal{Y}$ is *non-expanding*, if for all objects $x \in \mathcal{X}$ and all morphisms $\alpha : Fx \to y'$ in $\mathcal{Y}$ there is a morphism $\beta : x \to x'$ in $\mathcal{X}$ such that $F\beta = \alpha$.

A monad $\mathsf{T} = \langle T, \eta, \mu \rangle$ on **Cat** is non-expanding if all components $\eta_{\mathcal{X}} : \mathcal{X} \to T\mathcal{X}$ of the unit are non-expanding, and the action preserves this: for any non-expanding functor $F : \mathcal{X} \to \mathcal{Y}$, $TF : T\mathcal{X} \to T\mathcal{Y}$ is non-expanding as well.

That a term rewriting system Θ which does not contain expanding rules (a rule $(X \vdash l \to r)$ in which $l = {}'x$) gives rise to a monad T_Θ which is non-expanding in the sense above is shown by easy induction on the rules in Def. 3. Further, the unit is a monomorphism, since by definition (i.e. the freeness of the term algebra) $'x = {}'y$ implies $x = y$.

For convenience, we introduce the notational shortcuts $\eta^w_{j,v} \overset{def}{=} T^w(\eta_{j,T^v})$ and $\mu^w_{j,v} \overset{def}{=} T^w(\mu_{j,T^v})$, for $w, v \in W$ and $j \in \mathcal{L}$. We are now going to construct the witness relation:

Definition 12. The relation $\prec$ on the objects of $\coprod_{w \in W} T^w \mathcal{X}$ is defined to be the smallest relation such that for all $w \in W$ and $x \in T^w \mathcal{X}$,

$$\forall u, v \in W, j \in \mathcal{L} \,.\, w = uv \Rightarrow x \prec \eta^u_{j,v}(x)$$
$$\forall u, v \in W, j \in \mathcal{L} \,.\, w = ujjv \Rightarrow \mu^u_{j,v}(x) \prec x$$

The relation $\preceq$ is defined to be the reflexive-transitive closure of $\prec$.

To show that $\preceq$ is a witness relation, we go through the four properties of Def. 9 in turn.

(i) We show that $x \prec y \Rightarrow x \operatorname{wit}_Q y$, from which $x \preceq y \Rightarrow x \operatorname{wit}_Q y$ follows by a simple induction. (Note that every object trivially witnesses itself, taking care of the reflexive closure.) If $x \prec y$, there are two cases:

  - if $y = \eta^w_{j,v}(x)$, then since $\eta^w_{j,v}$ is non-expanding, there is an $\alpha : x \to x'$ for all $\beta : \eta^w_{j,v} x \to y'$ with $Q\alpha = Q\beta$;
  - if $x = \mu^r_{i,s}(y)$, then there is $\mu^r_{i,s}(\beta)$ for all $\beta : y \to y'$.

  Hence x is a witness of y with respect to Q.

(ii) We have show that for all $d : T^u\mathcal{X} \to T^v\mathcal{X}$ in the image of $D_\mathcal{X}$ and all $x \in T^u\mathcal{X}$, there is a $w \in W$, $y \in T^w\mathcal{X}$ such that $y \operatorname{wit}_Q F(x)$ and $y \operatorname{wit}_Q G(x)$, with $F(x) = \iota_u(x) = x$; and $G(x) = \iota_v(d(x)) = d(x)$.

  By Lemma 14 below, for all morphisms $d : T^u\mathcal{X} \to T^v\mathcal{X}$ in the image of $D_\mathcal{X}$ there are morphisms e in $\mathcal{F}(\mathcal{G}_E)$ and m in $\mathcal{F}(\mathcal{G}_M)$ such that $d = D_\mathcal{X}(e) \cdot D_\mathcal{X}(m)$. Let $d_\eta \overset{def}{=} D_\mathcal{X}(e)$, $d_\mu \overset{def}{=} D_\mathcal{X}(m)$, then $y \overset{def}{=} d_\mu(x)$, and by a simple induction, $y \preceq d_\eta(y)$ and $d_\mu(x) \preceq x$, hence $y \preceq d_\eta d_\mu(x) = d(x) = G(x)$ and $y = d_\mu(x) \preceq x = F(x)$.

(iii) Transitivity is trivial.

(iv) To show the existence of filtered lower bounds for $\preceq$, we show that $\prec$ has filtered lower bounds. From this, we obtain the existence of filtered lower bounds for $\preceq$ by another simple induction (a "tiling" process like the proof of Lemma 7).

  Given x, y, z such that $x \prec z$ and $y \prec z$, we have to show there is u s.t. $u \prec x$ and $u \prec y$. By the definition of $\prec$, there are four cases to consider, two of which are symmetric; in particular, these are (for some $w, v, r, s \in W$, $i, j \in \mathcal{L}$):

  1. $z = \eta^w_{j,v}(x)$ and $z = \eta^r_{i,s}(y)$;
  2. $z = \eta^w_{j,v}(x)$ and $\mu^r_{i,s}(z) = y$ and its symmetric case;
  3. $x = \mu^w_{j,v}(z)$ and $y = \mu^r_{i,s}(z)$.

  The existence of u follows from the three properties of the morphisms in the image of $D_\mathcal{X}$ given by Lemma 13 below. For example, in the second case, $y = \mu^r_{i,s}(\eta^w_{j,v}(x))$; then either $\mu^r_{i,s} \cdot \eta^w_{j,v} = 1_{T^{wv}\mathcal{X}}$ and $x = y$, hence $u \overset{def}{=} x = y$, or there are $w', v', r', s' \in W$ such that $\mu^r_{i,s} \cdot \eta^w_{j,v} = \eta^{w'}_{j,v'} \cdot \mu^{r'}_{i,s'}$, then $u \overset{def}{=} \mu^{r'}_{i,s'}(x)$ with $u \preceq \eta^{w'}_{j,v'} \mu^{r'}_{i,s'}(x) = \mu^r_{i,s} \eta^w_{j,v}(x) = y$ and $u = \mu^{{r'}_{i,s'}}(x) \preceq x$.

It remains to show the two lemmas needed above:

Lemma 13. *In the following, let $w, v, r, s \in W$ and $i, j \in \mathcal{L}$:*

(i) *Given $\mu^w_{i,v}$,$\mu^r_{j,s}$ such that $wiiv = rjjs$, then either $w = r$, $i = j$, $v = s$, or there are $w', v', r', s' \in W$ such that $\mu^{w'}_{j,v'} \cdot \mu^w_{i,v} = \mu^{r'}_{i,s'} \cdot \mu^r_{j,s}$.*

(ii) *Given $\eta^w_{j,v}$, $\eta^r_{i,s}$ such that $wjv = ris$, and given $y \in T^{wv}\mathcal{X}, z \in T^{rs}\mathcal{X}$ such that $\eta^w_{j,v}(y) = \eta^r_{i,s}(z)$, there is $u \in W, x \in T^u\mathcal{X}$, and w', v', r', s' such that $\eta^{w'}_{i,v'}(x) = y$, $\eta^{w'}_{i,v'}(x) = z$.*

(iii) *Given $\eta^w_{j,v}$, $\mu^s_{i,t}$ such that $wjv = siit$, then there are either $r', s', w', v' \in W$ s.t. $\mu^s_{i,t} \cdot \eta^w_{j,v} = \eta^{w'}_{j,v'} \cdot \mu^{s'}_{i,t'}$, or $\mu^s_{i,t} \cdot \eta^w_{j,v} = 1_{wv}$.*

Proof. The idea of the proof is that either the two morphisms commute over each other by the naturality of μ and η, or we can use the monad laws (associativity in the first case, the unit laws in the third case).

The proofs rely on the algebra of words over the language W, and a careful case distinction. Briefly, given a word $u \in W$ such that $u = wxv$ and $u = rys$ (with $r, s, w, v, x, y \in W$) we call the occurrence of x and y *independent* if we can find a word $z \in W$ such that $w = ryz$ and $s = zxv$ (there is of course a symmetric case). Above, x and y correspond to ii and jj in the first case; j and i in the second, and j and ii in the third.

If x and y are independent we can construct a naturality square such that the two morphisms commute over each other. For example, in the first case, assume that there is a $z \in W$ s.t. $w = rjjz$ and $s = ziiv$. Then let $w' \overset{def}{=} r$, $v' \overset{def}{=} ziv$, $r' \overset{def}{=} rjz$ and $s' = v$, and by applying T^r to the naturality square of μ_j (Diagram 2), we obtain $\mu^r_{j,ziv} \cdot \mu^{rjjz}_{i,v} = \mu^{rjz}_{i,v} \cdot \mu^r_{j,ziiv}$, hence (by definition of w',v',r',and s') $\mu^{w'}_{j,v'} \cdot \mu^w_{i,v} = \mu^{r'}_{i,s'} \cdot \mu^r_{j,s}$.

$$\begin{tikzcd}
T^{jjziiv}\mathcal{X} \arrow[r, "\mu_{j,ziiv}"] \arrow[d, "\mu^{jjz}_{i,v}"'] & T^{jziiv}\mathcal{X} \arrow[d, "\mu^{jz}_{i,v}"] \\
T^{jjziv}\mathcal{X} \arrow[r, "\mu_{j,ziv}"'] & T^{jziv}\mathcal{X}
\end{tikzcd} \tag{2}$$

In the second case, this square is completed starting from the target (that would be T^{jzv} for the square in Diagram 2), hence the requirement that the units are monomorphisms (to be more specific, have left inverses).

If on the other hand x and y are not independent, then in the first case we use the associativity of the multiplication; in the second case, they are equal; and in the third case, multplication and unit cancel each other out by the unit law. □

Lemma 14. *For all morphisms d in the image of $D_{\mathcal{X}}$, there are morphisms e in $\mathcal{F}(\mathcal{G}_E)$, m in $\mathcal{F}(\mathcal{G}_M)$ such that $d = D_{\mathcal{X}}(e) \cdot D_{\mathcal{X}}(m)$.*

Proof. For d as above there is a path p in $\mathcal{G}$ such that $D_{\mathcal{X}}(p) = d$. The lemma is shown by induction on the length of p, using the third case of Lemma 13, which here means that if in a path p there is an edge from $E_E(\mathcal{G})$ followed by one from $E_M(\mathcal{G})$, we can always find another path p' s.t. $D_{\mathcal{X}}(p) = D_{\mathcal{X}}(p')$ in which either none of these edges occur (they cancel each other out), or an edge from $E_M(\mathcal{G})$ is followed by an edge from $E_E(\mathcal{G})$. Hence, for all paths p we can find a path q s.t. $D_{\mathcal{X}}(p) = D_{\mathcal{X}}(q)$ and q consists of edges from $E_M(\mathcal{G})$ (the path m above) followed by edges from $E_E(\mathcal{G})$ (the path e). □

We conclude that $\preceq$ is indeed a witness relation for $\coprod_{w \in W} T^w \mathcal{X}$ with respect to the coequalizer Q in Diagram 1, hence $T\mathcal{X}$ is confluent if $\mathcal{X}$ is, and hence T is confluent, under the assumption that T_1 and T_2 are confluent, non-expanding and their units are monomorphisms. Call such monads *regular*, then we have the main theorem:

Theorem 15. *The coproduct of two confluent regular monads is confluent.*

The original theorem by Toyama [Toy87] of course does not consider categories (corresponding to named reductions); it corresponds to Theorem 15 for two confluent, regular monads on **Pre**. It is obtained as a corollary from that theorem by observing that the category **Pre** of preorders is a reflexive subcategory of **Cat**. Then given two monads on **Pre**, we can (under the inclusion) consider them to be monads on **Cat**, apply the construction and proof, and obtain a monad which in turn is a monad on **Pre** for which Theorem 15 holds.

Theorem 15 is more general than Toyama's theorem. Since neither the construction of the monad nor the proof rely on the fact that all variables on the right side of a rewriting rule occur on the left (one readily checks that the monad given by such a term rewriting system is regular, in particular non-expanding, in the sense defined above), the proposition above extends Toyama's theorem to these term rewriting systems. Furthermore, it is valid for generalized rewrite rules (Def. 2), which can be considered a very limited form of conditional rewriting.

The original proof (and its simplified version [KM+94]) proceeds by induction on the rank of terms (hence it would not work for rewriting rules with extra variables on the right where the rank can increase arbitrarily). The proof presented here uses the algebraic properties of the combination of the term rewriting systems (as given by Def. 12 and Lemmas 13 and 14). Hence, although the proof is based on the same ideas as [Toy87] and [KM+94], the actual technique employed is different.

The condition that the term rewriting systems are non-expanding is essential, since Theorem 15 does not hold for expanding term rewriting systems. Consider two systems with rules $\texttt{'}x \rightarrow \texttt{A('}x\texttt{)}$ and $\texttt{B(C('}y\texttt{))} \rightarrow \texttt{C(B('}y\texttt{))}$ (where all operations are unary), then in their combination there is the incompletable span

$$\texttt{B(A(C('}z\texttt{)))} \leftarrow \texttt{B(C('}z\texttt{))} \rightarrow \texttt{C(B('}z\texttt{))}\ .$$

The condition can be replaced by requiring that all expanding rewrites are contractible again.

4 Conclusions and Further Work

We have introduced a semantics for term rewriting systems based on the categorical treatment of universal algebra which is compositional in the sense that it is a free construction. We have shown how to use this semantics to give a categorical account of Toyama's theorem, which slightly generalises the original.

The semantics gives a clean distinction between the structure modelling the process of closing the reduction structure under context and substitution (given by the monad) and the choice of the mathematical structure used to model the reduction (preorders or categories; other possibilities include Sesqui-categories [Ste94], or graphs or binary relations to model the one-step reduction). It can handle some forms of non-standard rewriting as well. For example, combining the treatment of equational presentations and term rewriting systems, we obtain a model for rewriting modulo an arbitrary set of equations by a monad on **Pre**. However, this monad will not preserve all weakly filtered diagrams, and so the coproduct will have to be constructed differently.

Further work to appear in the author's thesis includes consideration of other structuring operations. Shared operations are modelled by the coequalizer of two monads. Restricting ourselves to systems sharing only constructors (this corresponds to coequalizers in which the two functors F, G are non-expanding) we should be able to recover results along the lines of [Ohl94].

References

[Bor94] Francis Borceux. *Handbook of Categorical Algebra 2: Categories and Structures.* Number 51 in Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1994.

[BW85] M. Barr and C. Wells. *Toposes, Triples and Theories.* Number 278 in Grundlehren der mathematischen Wissenschaften. Springer Verlag, 1985.

[EM85] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification 1: Equations and Initial Semantics*, volume 6 of *EATCS Monographs on Theoretical Computer Science.* Springer Verlag, 1985.

[GB92] J. A. Goguen and R. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the ACM*, 39:95– 146, 1992.

[Gra74] John W. Gray. *Formal Category Theory: Adjointness for 2-Categories.* Number 391 in Lecture Notes in Mathematics. Springer Verlag, 1974.

[Kel82] G. M. Kelly. *Basic Concepts of Enriched Category Theory*, volume 64 of *LMS Lecture Note Series.* Cambridge University Press, 1982.

[KM+94] J. W. Klop, A. Middeldorp, Y. Toyama, and R. de Vrijer. A simplified proof of Toyama's theorem. *Information Processing Letters*, 49:101–109, 1994.

[KS74] G. M. Kelly and Ross Street. Review of the elements of 2-categories. In *Category Seminar Sydney 1972/73*, number 420 in Lecture Notes in Mathematics, pages 75–103. Springer Verlag, 1974.

[Lü96] Christoph Lüth. *Compositional Categorical Term Rewriting in Structured Algebraic Specifications.* PhD thesis, University of Edinburgh, 1996. Forthcoming.

[Mac71] S. Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics.* Springer Verlag, 1971.

[Man76] Ernest G. Manes. *Algebraic Theories*, volume 26 of *Graduate Texts in Mathematics.* Springer Verlag, 1976.

[Ohl94] Enno Ohlenbusch. On the modularity of confluence of constructor-sharing term rewriting systems. In Sophie Tison, editor, *Trees in Algebra and Programming — CAAP 94*, LNCS 787. Springer Verlag, April 1994.

[ST88] D. Sannella and A. Tarlecki. Specifications in an arbitrary institution. *Information and Computation*, 76(2/3):165–210, Feb/Mar 1988.

[Ste94] John G. Stell. Modelling term rewriting systems by Sesqui-categories. Technical Report TR94-02, Keele Unversity, January 1994.

[Toy87] Y. Toyama. On the Church-Rosser property for the direct sum of term rewriting systems. *Journal of the ACM*, 34(1):128–143, 1987.

This article was processed using the LaTeX macro package with LLNCS style