Monads and Modular Term Rewriting

Christoph Lüth[1] and Neil Ghani[2]

[1] Bremen Institute for Safe Systems, FB 3, Universität Bremen
Postfach 330440, 28334 Bremen
cxl@informatik.uni-bremen.de

[2] The School Of Computer Science
University of Birmingham, Birmingham, England
nxg@cs.bham.ac.uk

Abstract. Monads can be used to model term rewriting systems by generalising the well-known equivalence between universal algebra and monads on the category **Set**. In [Lü96], this semantics was used to give a purely categorical proof of the modularity of confluence for the disjoint union of term rewriting systems. This paper provides further support for monadic semantics of rewriting by giving a categorical proof of the most general theorem concerning the modularity of strong normalisation. In the process, we improve upon the technical aspects of earlier work.

1 Introduction

Term rewriting systems (TRSs) are widely used throughout computer science as they provide an abstract model of computation while retaining a relatively simple syntax and semantics. Reasoning about large term rewriting systems can be very difficult and an alternative is to define structuring operations which build large term rewriting systems from smaller ones. Of particular interest is whether key properties are *modular*, that is when does a structured term rewriting system inherit properties from its components?

Although most properties are not in general modular, there are a number of results in the literature providing sufficient conditions for the modularity of key properties. Research originally focussed on the *disjoint union*, for which confluence is modular (Toyama's Theorem), whereas strong normalisation is not. However, strong normalisation is modular under a variety conditions, such as both systems are not collapsing (i.e. contain no *collapsing rules*) [Rus87], both systems are not *duplicating* [Rus87], one system is neither duplicating nor collapsing [Mid89], or both systems are *simplifying* [KO92]. Modularity results for conditional term rewriting systems and unions which permit limited sharing of term constructors are rather unsatisfactory and tend to require rather strong syntactic conditions. Overall, although many specific modularity results are known, what is lacking is a coherent framework which explains the underlying principles behind these results.

We believe that part of the problem is the overly concrete, syntactic nature of term rewriting and that a more abstract semantics is needed. *Abstract*

Reduction Systems provide a semantics for term rewriting systems using relations, but relations do not posses enough structure to adequately model key concepts such as *substitution, context, layer structure* etc. Thus the relational model is used mainly as an organisational tool with the difficult results proved directly in the syntax. 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, using structures such as *2-categories* [RS87,See87], *Sesqui-categories* [Ste94] or *confluent categories* [Jay90]. However, despite some one-off results [Gha95,RS87], these approaches have failed to make a lasting impact on term rewriting.

An alternative approach starts from the observation that the categorical treatment of universal algebra is based on the idea of a monad on the category **Set**. Since term rewriting systems can be regarded as a generalisation of universal algebra it is natural to model a term rewriting system by a monad over a more structured base category. The basic theory of monads over categories with more structure than **Set** has been developed by Kelly and Power [KP93] and forms the theoretical basis of this research.

Monads offer a general methodology for the study of modularity in term rewriting. Firstly, one proves that the semantics is *compositional* wrt. the structuring operation in question. For the disjoint union of term rewriting systems, this means proving that if Θ is a term rewriting system and T_Θ is its semantics, then $\mathsf{T}_{\Theta_1+\Theta_2} \cong \mathsf{T}_{\Theta_1} + \mathsf{T}_{\Theta_2}$. Next we express the action of the monad representing the combined term rewriting system as a pointwise colimit over the base category $\mathsf{T}_{\Theta_1+\Theta_2}(X) = \mathit{colim}\,\mathcal{D}_X$. Finally we prove that if the objects of $\mathcal{D}_X$ satisfy the desired property, then so does $\mathit{colim}\,\mathcal{D}_X$.

This methodology is particularly pleasing as the diagram $\mathcal{D}_X$ abstractly represents the fundamental concept in modular term rewriting of the *layer structure* on terms. In addition, the conditions on the use of variables which occur in the literature arise naturally as conditions on the units of the component monads. In [Lü96], Lüth used this approach to give an entirely categorical proof of Toyama's theorem. This paper proves the most general result for the modularity of strong normalisation for disjoint unions and also improves some technical aspects of earlier work.

The paper is divided as follows. Section 2 motivates the use of monads as models of term rewriting systems by recalling the equivalence between universal algebra and finitary monads on **Set**. Section 3 formally introduces term rewriting systems, and Section 4 defines the monadic semantics for term rewriting systems. Section 5 shows how disjoint unions of term rewriting systems are treated semantically while sections 6 and 7 contain the actual modularity results. Section 8 finishes with directions for further research. We would like to thank Don Sannella, Stefan Kahrs and John Power for many stimulating discussions. Glory, glory to the Hibees!

2 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, $\eta : \mathbf{1}_{\mathcal{C}} \Rightarrow T$, called the *unit*, and $\mu : TT \Rightarrow T$, called the *multiplication* of the monad, satisfying the *monad laws*: $\mu{\cdot}T\eta = \mathbf{1}_{\mathcal{C}} = \mu{\cdot}\eta_T$, 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 ([Man76], [Rob94] and [Mac71, 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:

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

Quotes are used to distinguish a variable $x \in X$ from the term $\text{'}x \in T_\Sigma(X)$. This will be important when analysing the layer structure on terms formed from the disjoint union of two signatures. A term of the form $\text{'}x$ will be called a *term variable* while all other terms are called *compound terms*. An element of $T_\Sigma(X)$ will be called a term built over X.

Lemma 4. *The map $X \mapsto T_\Sigma(X)$ defines a monad T_Σ on* **Set**.

Lemma 4 generalises to many-sorted signatures— if S is a set of sorts, then an S-sorted signature defines a monad on $\mathbf{Set}^S$. Monads arising via the term algebra construction satisfy an important continuity condition, namely they are *finitary*. To understand this condition, observe that 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 because all the operators in Σ have a finite arity and thus a term built over X can only contain 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 5 (Finitary Monads). A functor is *finitary* iff it preserves filtered colimits [Mac71]. A monad is *finitary* iff its action is finitary.

Lemma 6. *If Σ is a signature, then T_Σ is finitary [Rob94, Lemma 1.7].*

One can consider signatures with operations of infinite arities in which case the associated monad satisfies a suitably generalised definition of finitariness. All monads we shall consider are finitary — an example of a monad which isn't finitary is the powerset monad on **Set** which forms powersets of arbitrary large sets and hence has "operations" of arbitrary large arity. Monads also model *algebraic theories*:

Definition 7 (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 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**" [Man76].

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.

Definition 8 (Internal Signature). The *internal signature* of a finitary monad $\mathsf{S} = \langle S, \eta, \mu\rangle$ on **Set** is given by

$$\Sigma_S(n) \stackrel{\mathit{def}}{=} \bigcup_{card(X)=n} S(X)$$

We can define a map $\varepsilon_{S,X} : T_{\Sigma_S}(X) \to SX$ interpreting terms from $T_{\Sigma_S}(X)$ in SX. We say that a monad S *admits* an equation (X, l, r) where $l, r \in T_{\Sigma_S}(X)$, written $S \models_X l = r$, if $\varepsilon_{S,X}(l) = \varepsilon_{S,X}(r)$. The set of equations admitted by S, written $\mathcal{E}_S$, is defined as $\mathcal{E}_S \stackrel{\mathit{def}}{=} \{(X, l, r) \mid S \models_X l = r\}$.

Definition 9 (Internal Language). The internal language of a finitary monad S on **Set** is given by $\mathcal{L}_S \stackrel{\mathit{def}}{=} \langle \Sigma_S, \mathcal{E}_S\rangle$.

Crucially, these constructions are adjoint: that is, there is an adjunction $\mathsf{T} \dashv \mathcal{L} : \mathbf{AlgTh} \to \mathbf{Mon}_{Fin}(\mathbf{Set})$ where the categories **AlgTh** of algebraic theories and $\mathbf{Mon}_{Fin}(\mathbf{Set})$ of finitary monads on **Set** are appropriately defined [BW85]. The evaluation ε_S is the counit of this adjunction. Since T is left adjoint, it preserves colimits and hence the semantics is compositional.

In summary, monads provide a semantics for algebraic theories with the concepts of term-algebra, variable and substitution taken as primitive. This semantics is compositional, allowing us to reason about the disjoint union of algebraic theories in terms of the component theories.

3 Term Rewriting Systems

We now briefly review the theory of term rewriting systems — further details may be found in [Klo92]. First, fix a countably infinite set V of variables.

Definition 10 (Term Rewriting Systems). A *term rewriting system* $\Theta = \langle \Sigma, R \rangle$ consists of a signature Σ and a set R of Σ-rewrite rules of the form $r : t \to s$ where $t, s \in T_\Sigma(V)$.

A rewrite rule $r : t \to s$ gives rise to the *one-step reduction* relation $C[\sigma(t)] \to_r C[\sigma(s)]$, where $C[\,]$ is a context and σ is a substitution. The one-step reduction relation $\to_R$ of a term rewriting system $\Theta = \langle \Sigma, R \rangle$ is defined as the union of $\{\to_r\}_{r \in R}$, while the *many-step reduction relation*, denoted $\twoheadrightarrow_R$, is the transitive-reflexive closure of the one-step reduction relation.

A rewrite rule $r : t \to s$ is called *expanding* if t is a variable, and *collapsing* if s is a variable. It is said to *introduce variables* if there is a variable occuring in s which does not occur in t, and be *duplicating* if a variable occurs more often in s than in t. Traditionally, rewrite rules are not to allowed to be expanding or variable-introducing, but semantically these restrictions are unnatural and hence omitted. The two key properties of term rewriting systems are *confluence* and *strong normalisation*.

Definition 11 (Confluence and SN). A term rewriting system is *confluent* iff $\forall x, y_1, y_2.\, x \twoheadrightarrow_R y_1 \wedge x \twoheadrightarrow_R y_2\ \exists z.\, y_1 \twoheadrightarrow_R z \wedge y_2 \twoheadrightarrow_R z$. It is *strongly normalising* (SN, terminating, Noetherian) iff there is no infinite sequence $x_1 \to_R x_2 \to_R x_3 \to_R \ldots$.

A term rewriting system which is both confluent and SN is called *complete*. Modular term rewriting studies how properties of large term rewriting systems are inherited from their component systems. The key definitions are

Definition 12 (Disjointness and Modularity). Given two term rewriting systems $\Theta_1 = \langle \Sigma_1, R_1 \rangle$ and $\Theta_2 = \langle \Sigma_2, R_2 \rangle$ their *disjoint union* is defined as $\langle \Sigma_1 + \Sigma_2, R_1 + R_2 \rangle$. A property P is *modular* if the disjoint union of Θ_1 and Θ_2 satisfies P iff Θ_1 satisfies P and Θ_2 satisfies P.

4 Monads as Models of Term Rewriting Systems

Our semantics for term rewriting systems generalises the treatment of algebraic theories as finitary monads over the category **Set**. We regard term rewriting systems as a generalised signature and hence its semantics is naturally given by a monad over a more structured base category. The choice of the base category depends on the specific aspects of rewriting one is interested in. We start by using the category **Pre** of preorders as a base category because definition 14 is notationally easier[1], although later we switch to **Cat**.

Kelly and Power [KP93] have shown how algebraic theories can be generalised to categories other than **Set** in such a way that the theory of section 2 can be developed at this more abstract level. This general theory requires the arity of operations, variables and term algebra to have the same structure (sets in the case of algebraic theories, preorders here) so as to allow a uniform treatment of term formation by the multiplication of the monad. Thus, each rewrite rule must be given an arity which is a *preorder* and the term algebra construction must map a *preorder* of variables to a *preorder* of rewrites. This leads to a more general form of rewrite rules:

Definition 13 (Generalised Rewrite Rules). A *generalised rewrite rule* in a signature Σ is a triple (X, l, r), written as $(X \vdash l \to r)$, where $X = (X_0, \to_X)$ is a finite preorder and $l, r \in T_\Sigma(X_0)$ are terms.

Thus, in order to instantiate a generalised rewrite rule (X, l, r), one must not only supply terms for the free variables of the rule, but these terms must have rewrites between them which conform to the order structure of X — see rule [INST] of Definition 14. The traditional rewrite rules of definition 10 are of course generalised rewrite rules whose the arities are discrete.

In universal algebra, each signature Σ defines a functor T_Σ whose action is to map a set X to the term algebra $T_\Sigma(X)$ built using the operators of Σ as term constructors and the elements of X as variables. The equivalent construction for term rewriting systems is called a *term reduction algebra*:

Definition 14 (Term Reduction Algebra). Given a term rewriting system $\Theta = \langle \Sigma, R \rangle$ and a preorder $X = (X_0, \to_X)$, the *term reduction algebra* $T_\Theta(X)$ is the smallest preorder $\to_{T_\Theta(X)}$ on the terms $T_\Sigma(X_0)$ satisfying the following inference rules (where $t[t_1, \ldots, t_n]$ is the substitution of the n vari-

[1] The term algebra construction for categories is technically more complicated as extra equations are required to ensure the term algebra is a category.

ables in $t \in T_\Omega(Y)$ with terms $t_1, \ldots, t_n$):

$$[\textsc{Var}] \quad \frac{x \to_X y}{\text{'}x \to_{T_\Theta(X)} \text{'}y}$$

$$[\textsc{Pre}] \quad \frac{t_1 \to_{T_\Theta(X)} s_1, \ldots, t_n \to_{T_\Theta(X)} s_n}{f(t_1, \ldots, t_n) \to_{T_\Theta(X)} f(s_1, \ldots, s_n)} \; f \in \Sigma_n$$

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

So the term reduction algebra $T_\Theta(X)$ has as objects the terms which can be built over X and has as rewrites the transitive-reflexive closure of the union of the rewrites of Θ and the rewrites of X closed under the term constructors. This construction defines a monad and this semantics is compositional.

Lemma 15. *The map $X \mapsto T_\Theta(X)$ defines a finitary,* **Pre***-enriched monad T_Θ. Furthermore, this semantics is compositional.*

Proof. The proofs follow those for algebraic theories. See [Lü97]. □

Our construction of the coproduct of two monads in Section 5 requires the following technical properties.

Definition 16 (Regular Monads). A monad T is *regular* if

1) the action T preserves *weakly filtered colimits* (colimits of weakly filtered diagrams) where a diagram D is weakly filtered if for all $i, j \in D$, there is a $k \in D$ and morphisms $m : i \to k$, $n : j \to k$;
2) the unit is a mono (i.e. every component of the unit is a mono).

Lemma 17. *For a term rewriting system Θ, the monad T_Θ is regular.*

Proof. That the unit is a monomorphism is easy to see. To show that T_Θ preserves weakly filtered diagrams, it is sufficient to show that T_Θ preserves both filtered colimits (because the underlying monad on **Set** is finitary, see lemma 6), and coequalizers (because it does not identify any terms). □

Enriched Monads

The crucial insight behind the constructions of the previous section is the proper *enrichment* [Kel82]. In particular, the base category $\mathcal{A}$ has to be enriched over a closed monoidal category $\mathcal{V}$. Further, $\mathcal{A}$ and $\mathcal{V}$ have to be locally finitely presentable, i.e. have a small set $\mathcal{N}$ of objects representing isomorphism classes of *finitely presentable objects* [KP93]; for **Set**, $\mathcal{N}$ is the natural numbers and the finitely presentable objects are the finite sets.

In the enriched setting, a signature over $\mathcal{A}$ is a map $\Sigma : \mathcal{N} \to \mathcal{A}$, giving for every $c \in \mathcal{N}$ the operations of arity c. The term algebra is then given by a functor $T_\Sigma : \mathcal{A} \to \mathcal{A}$ which maps an $\mathcal{A}$-object of variables X to the $\mathcal{A}$-object of Σ-terms constructed over X. Formally, $T_\Sigma(X)$ is defined as the colimit in $\mathcal{A}$ of the chain $T_0(X) \hookrightarrow T_1(X) \hookrightarrow \dots$ where $T_0(X) = X$ and

$$T_{n+1}(X) = X + \sum_{c \in \mathcal{N}} [c, T_n(X)] \otimes \Sigma(c) \tag{1}$$

Note how the closed structure over which $\mathcal{A}$ enriches occurs in equation 1. We think of $T_{n+1}(X)$ as the terms of depth $n+1$, constructed as operations of arity c applied to c-objects of terms of depth n. Our models of term rewriting systems arise when we take $\mathcal{A} = \mathcal{V} = \mathbf{Pre}$ with the usual cartesian closed structure providing the enrichment.

Each of the rules of Definition 14 arises as a special case of equation 1. For instance the rule [Var] stems from the inclusion of X in $T_1(X)$, while specialising equation 1 to the declaration of rewrite rules gives the following equivalent formulation of [Inst]

$$\frac{\theta \in \mathbf{Pre}(Y, T_\Theta(X)) \quad (Y \vdash l \to r) \in R}{\theta(l) \ \to_{T_\Theta(X)} \ \theta(r)} \ [\text{Inst'}]$$

We can also specialise equation 1 to the declaration of term constructors and hence obtain an equivalent formulation of rule [Pre].

Of course, one can vary not only the base category, but also the choice of the monoidal structure. There are in fact two monoidal closed structures over **Cat** — the usual cartesian structure and a monoidal structure which has as objects the same objects as the cartesian product but whose morphisms are alternating sequences of morphisms from each category. Categories enriched over this alternative monoidal structure are called *Sesqui-categories* and have been used as alternative models for term rewriting [Ste94] since they have a categorical notion of "length". It is our intention to use this observation to compare the Sesqui-category approach within our monadic framework.

Monadic Versions of Rewriting Concepts

In the remainder of the paper, we give semantic proofs of modularity results for confluence and strong normalisation. The first step is to define confluence and strong normalisation for arbitrary monads, and show that these definitions coincide with the traditional definitions of section 3.

Definition 18 (Confluence for Monads). 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.

Definition 19 (SN for Monads). A category $\mathcal{C}$ is *SN*, written $\mathcal{C} \models \mathrm{SN}$, if its *underlying order* $R^{-}(\mathcal{C})$ is SN, where $R^{-}(\mathcal{C})$ is defined as follows:

$$R^{-}(\mathcal{C}) \stackrel{def}{=} (|\mathcal{C}|, \{x > y \mid \exists \alpha : x \to y \wedge \alpha \neq \mathbf{1}_x\}) \tag{2}$$

A monad T on **Cat** is strongly normalising if $T\mathcal{X} \models \mathrm{SN}$ whenever $\mathcal{X} \models \mathrm{SN}$.

The definition of a confluent category is different from Stell's [Ste94] which does not require the completions to form a commuting diagram; it is used by Jay [Jay90] but his confluent functors have a different intention and hence only require the identity and composition to be preserved up to having a common reduct.

Lemma 20. *A TRS Θ is confluent iff T_Θ is a confluent monad. Similarly, Θ is SN iff T_Θ is SN.*

Proof. If X is a preorder and Θ is a term rewriting system, then $\mathsf{T}_\Theta(X)$ is the transitive-reflexive closure of the union of the one-step reduction relation $\to_R$ and the closure of the variable rewrites in X under application of operations. Thus lemma 20 amounts to proving that the addition of variable rewrites to a term rewriting system does not change its properties. See [Lü97]. $\square$

Collapsing and expanding rewrites also have categorical formulations.

Definition 21 (Non-Expanding/Non-Collapsing Monads). 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 $\beta : x \to y$ 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 of the unit η are non-expanding, and the action preserves non-expanding functors, i.e. if $F : \mathcal{X} \to \mathcal{Y}$ is non-expanding, then so is TF.

A functor $F : \mathcal{X} \to \mathcal{Y}$ is *non-collapsing*, if F^{op} is non-expanding. A monad $\mathsf{T} = \langle T, \eta, \mu \rangle$ on **Cat** is non-collapsing if all components of the unit are non-collapsing, and the action preserves non-collapsing functors

One may easily verify that a term rewriting system Θ is non-expanding (non-collapsing) iff T_Θ is non-expanding(non-collapsing).

5 A Monadic Approach to Modularity

We have given a semantics to term rewriting systems in terms of monads on **Cat**. By lemma 20 we can reason about the disjoint union of Θ_1 and Θ_2 by reasoning about its semantics $\mathsf{T}_{\Theta_1+\Theta_2}$, which by lemma 15 is isomorphic to the coproduct of $\mathsf{T}_{\Theta_1} + \mathsf{T}_{\Theta_2}$. This section gives a pointwise construction of the coproduct of two regular monads as the colimit of a diagram. Since this diagram is built solely from the component monads, we can reason about the coproduct monad in terms of the component monads.

Consider terms built in the disjoint union of two signatures Ω, Σ. Such terms have an inherent notion of *layer*, that is one can decompose a term constructed from symbols in the union of two disjoint signatures into a term constructed from symbols in only one signature and strictly smaller subterms whose head symbol is from the other signature. Thus terms built from operations of $\Omega + \Sigma$ are contained in

$$\begin{aligned} T_{\Omega+\Sigma}(X) = \; & X + T_\Omega(X) + T_\Sigma(X) + T_\Omega T_\Sigma(X) + T_\Sigma T_\Omega(X) + \\ & T_\Omega T_\Sigma T_\Omega(X) + T_\Sigma T_\Omega T_\Sigma(X) + \dots \end{aligned} \tag{3}$$

However this disjoint union is too large as each component of the sum in equation 3 contains a separate copy of the variables X. Therefore this sum is quotiented by taking the colimit of a diagram including all arrows formed using the units and multiplications of the monads. Formally, let $\mathsf{T}_1, \mathsf{T}_2$ be two regular monads on **Cat**, let $\mathcal{L} \stackrel{def}{=} \{1, 2\}$, and define $W \stackrel{def}{=} \mathcal{L}^*$ to be the words over $\mathcal{L}$, and for $w \in W$, $T^w : \mathbf{Cat} \to \mathbf{Cat}$ by $T^\varepsilon \stackrel{def}{=} 1_{\mathbf{Cat}}$ and $T^{jv} \stackrel{def}{=} T_j T^v$ where $j \in \mathcal{L}, v \in W$. As notational shortcuts, we also define the natural transformations $\eta^u_{i,v} \stackrel{def}{=} T^u(\eta_{i,T^v})$ and $\mu^u_{j,v} \stackrel{def}{=} T^u(\mu_{j,T^v})$ for $u, v \in W, j \in \mathcal{L}$.

Definition 22 (The Colimit Diagram $\mathcal{D}_\mathcal{X}$). For every category $\mathcal{X}$, the diagram $\mathcal{D}_\mathcal{X}$ has as objects the categories $T^w(\mathcal{X})$ for $w \in W$, and as edges:

$$(\eta^u_{i,v})_\mathcal{X} : T^{uv}(\mathcal{X}) \to T^{uiv}(\mathcal{X}) \qquad (\mu^u_{j,v})_\mathcal{X} : T^{ujjv}(\mathcal{X}) \to T^{ujv}(\mathcal{X})$$

Lemma 23. *The map on categories $\mathcal{X} \mapsto colim\, D_\mathcal{X}$ extends to a monad which is the coproduct of the monads T_1 and T_2.*

Proof. Functoriality follows from the universal property of the colimit, and by the fact that all arrows in the diagram are natural transformations. The unit is simply the inclusion of $\mathcal{X}$ into the colimiting object. The multiplication uses the fact that the diagram $D_\mathcal{X}$ is weakly filtered, and hence preserved by the two functors T_1, T_2. The monad laws and universal property follow from various diagram chases (see [Lü97] for details). □

Note that $D_\mathcal{X}$ is not filtered, since there is e.g. no arrow in the diagram which makes $\eta^{12}_1 \cdot \eta^1_2$ and $\eta_{2,1} \cdot \eta_{1,21}$ equal.

Analysing the Coproduct Monad

By the dual of Theorem 2 in [Mac71, pg. 109], every colimit can be expressed via coproducts and coequalizers. In particular, the colimit of $D_\mathcal{X}$ is given by the coequalizer of Diagram 4, where on the left side, for every morphism

$$\coprod_{d : T^u\mathcal{X} \to T^v\mathcal{X} \in D_\mathcal{X}} T^u\mathcal{X} \underset{G}{\overset{F}{\rightrightarrows}} \coprod_{w \in W} T^w\mathcal{X} \tag{4}$$

$d : T^u\mathcal{X} \to T^v\mathcal{X}$ in $D_\mathcal{X}$ (with $u, v \in W$) there is a component $T^u\mathcal{X}$ in the coproduct, and F and G are defined as $F(T^u\mathcal{X}) \stackrel{def}{=} \iota_u(T^u\mathcal{X})$, $G(T^u\mathcal{X}) \stackrel{def}{=} \iota_v(d(T^u\mathcal{X}))$ where ι_u and ι_v are injections into the coproduct on the right.

Lemma 24. *Given two functors $F, G : \mathcal{X} \to \mathcal{Y}$, their coequalizer is a functor $Q : \mathcal{Y} \to \mathcal{Z}$, where $\mathcal{Z}$ is defined as follows:*

1) *The objects are the objects of $\mathcal{Y}$, quotiented by the equivalence closure $\equiv$ of the relation $\sim$ defined as $x \sim y \Leftrightarrow \exists z \in \mathcal{X} \,.\, Fz = x, Gz = y$.*
2) *Morphisms are sequences $<f_1, \ldots, f_n>$ of morphisms $f_i \in \mathcal{Y}(x_i, y_i)$ such that $y_i \equiv x_{i+1}$, quotiented by the smallest equivalence relation $\equiv$ compatible with composition in $\mathcal{Y}$ s.t. $<f, g> \equiv <g \cdot f>$ if f, g are composable in $\mathcal{Y}$, and $<Fh> \equiv <Gh>$ for all morphisms h in $\mathcal{X}$.*

Deciding the Equivalence: Normal Forms

The terms of the disjoint union of two monads are equivalence classes of objects from $\coprod_{w \in W} T^w \mathcal{X}$. In this section, we improve upon the presentation of [Lü96] by introducing a pair of reduction systems which reduce the objects and morphisms of $\coprod_{w \in W} T^w \mathcal{X}$ to a unique normal form, deciding this equivalence. We stress that these constructions occur at the level of regular monads and nowhere do we use the fact that these monads arise from term rewriting systems.

Definition 25 (The Reduction System $\to_{Ob}$). Define the following reduction systems on the objects of $\coprod_{w \in W} T^w \mathcal{X}$:

$$\begin{aligned}
\to_\mu &\stackrel{def}{=} \{x \to_\mu \mu^u_{j,v}(x) \mid u, v \in W, j \in \mathcal{L}\} \\
\to_\eta &\stackrel{def}{=} \{\eta^u_{j,v}(x) \to_\eta x \mid u, v \in W, j \in \mathcal{L}\} \\
\to_{Ob} &\stackrel{def}{=} \to_\eta \cup \to_\mu
\end{aligned}$$

We show that $\to_{Ob}$, is complete and hence obtain a decision procedure for the associated equality. First, define the *rank* of a term $t \in T^w \mathcal{X}$ as $\mathrm{rank}(t) \stackrel{def}{=} |w|$ (where $|w|$ is the length of the word w).

Lemma 26. *$\twoheadrightarrow_{Ob}$ is complete.*

Proof. For each reduction step $t \to_{Ob} u$, the rank of t is strictly greater than that of u and hence $\to_{Ob}$ is SN. For confluence, we refer to lemma 13 of [Lü96]. Clause (i) of that lemma implies confluence of $\to_\mu$, and clause (ii) implies confluence of $\to_\eta$. Clause (iii) implies that $\to_\eta$ and $\to_\mu$ commute and hence $\twoheadrightarrow_{Ob}$ is confluent. The cited proof also elucidates the necessity for the units η_1, η_2 to be monomorphisms. $\square$

Since $\twoheadrightarrow_{Ob}$ is complete, every object in $t \in \coprod_{w \in W} T^w \mathcal{X}$ reduces to a unique normal form which we denote NF(t). This forms a decision procedure for the equivalence of the objects:

Lemma 27. *Given $t, t' \in \coprod_{w \in W} T^w \mathcal{X}$, $Qt = Qt'$ iff $NF(t) = NF(t')$.*

Proof. $\mathrm{NF}(t) = \mathrm{NF}(t')$ iff t and t' are related is the equational theory on $\coprod_{w \in W} T^w \mathcal{X}$ generated by $\twoheadrightarrow_{Ob}$. This theory is clearly the same as that induced by the coequalizer of diagram 4. $\square$

We now consider morphisms in the coequalizer of diagram 4. Since such morphisms are sequences of morphisms in $\coprod_{w \in W} T^w \mathcal{X}$, we start by considering the normal forms of morphisms in $\coprod_{w \in W} T^w \mathcal{X}$.

Definition 28 (The Reduction System $\rightarrow_{Mor}$). Define the reduction systems on the morphisms of $\coprod_{w \in W} T^w \mathcal{X}$:

$$
\begin{aligned}
\rightarrow_\mu &\stackrel{def}{=} \{\alpha \rightarrow_\mu \mu^u_{j,v}(\alpha) \mid u, v \in W, j \in \mathcal{L}\} \\
\rightarrow_\eta &\stackrel{def}{=} \{\eta^u_{j,v}(\alpha) \rightarrow_\eta \alpha \mid u, v \in W, j \in \mathcal{L}\} \\
\rightarrow_{Mor} &\stackrel{def}{=} \rightarrow_\eta \cup \rightarrow_\mu
\end{aligned}
$$

Lemma 29. *$\rightarrow_{Mor}$ is complete, and every morphism α in $T^w \mathcal{X}$ reduces to a unique normal form $NF(\alpha)$ s.t. for all β, $Q\alpha = Q\beta$ iff $NF(\alpha) = NF(\beta)$.*

Proof. Analogously to lemma 26 and 27. $\square$

The mapping of terms and morphisms to their normal form can not be extended to a functor, since the presence of non-expanding and non-collapsing rewrites means that the normal form need not preserve the source and target of a morphism. For example given a rewrite α : `'`x`→G('`x`)` in $T_1(X)$, then $\mathrm{NF}(\alpha) = \alpha$ which is in $T_1(X)$ while $\mathrm{NF}($`'`$x) = x$.

Definition 30 (Layer-Expanding and Layer-Collapsing). Let $\alpha : s \rightarrow t$ be in $T^w \mathcal{X}$, and $\mathrm{NF}(\alpha) : s' \rightarrow t'$ its normal form. Then α is called *layer-collapsing (layer-expanding) in* T_j if there are $u, v \in W, j \in \mathcal{L}$ and $y \in T^{uv} \mathcal{X}$ s.t. $t' = \eta^u_{j,v}(y)$ $(s' = \eta^u_{j,v}(y))$.

Lemma 31. *$\alpha : s \rightarrow t$ in $T^w \mathcal{X}$ is layer-expanding (layer-collapsing) iff for $NF(\alpha) : s' \rightarrow t'$, $s' \neq NF(s)$ $(t' \neq NF(t))$.*

Proof. We can apply $\rightarrow_\mu$ to a morphism $\alpha : x \rightarrow y$ iff we can apply it to its source x and target y. It is feasible that we can apply $\rightarrow_\eta$ to x (or y) but not to α; namely, if $x = \eta^u_{j,v}(x')$ (or $y = \eta^u_{j,v}(y')$) and for all $\beta : x' \rightarrow y'$, $\eta^u_{j,v}(\beta) \neq \alpha$ (and so $\alpha \not\rightarrow_\eta \beta$), but then α is layer-expanding (or layer-collapsing). $\square$

Note that a rewrite can be expanding or collapsing in both systems at the same time. Further note that there can only be layer-expanding (collapsing) rewrites in T_j if T_j is expanding (collapsing).

For sequences `<`$\alpha_1, \ldots, \alpha_n$`>` in the colimit, we do not really need to decide the equality on them, but merely want to reason about their length (in the light of lemma 39 below). Hence we introduce the notion of minimal length:

Definition 32 (Minimal Length). A sequence $A = \texttt{<}\alpha_1, \ldots, \alpha_n\texttt{>}$ is of *minimal length* iff all elements are normal forms: $\forall i = 1, \ldots, n.\, \alpha_i = \mathrm{NF}(\alpha_i)$, and no equivalent sequence is shorter: $B \equiv A \Rightarrow |A| \leq |B|$.

For an example, consider the two monads given by the following two term rewriting systems $R_1 = \{\texttt{F(F('}x\texttt{))} \to \texttt{H('}x\texttt{)}\}, R_2 = \{\texttt{G('}y\texttt{)} \to \texttt{'}y\}$ Then there are reductions $\alpha_1 : \texttt{F('G('F('}x\texttt{)))} \to \texttt{F(''F('}x\texttt{))}$ in $T_1T_2T_1(X)$ and $\alpha_2 : \texttt{F(F('}x\texttt{))} \to \texttt{H('}x\texttt{)}$ in $T_1(X)$. Since $\mathrm{NF}(\texttt{F(''F('}x\texttt{))}) = \texttt{F(F('}x\texttt{))}$, one can form a sequence $\texttt{<}\alpha_1, \alpha_2\texttt{>}$ although α_1 and α_2 are not composable as they inhabit different components of $\coprod_{w \in W} T^w \mathcal{X}$. This situation (and its dual, where α_2 would be layer-expanding) is prototypical, as the following lemma shows:

Lemma 33. *A sequence $A = \texttt{<}\alpha_1, \ldots, \alpha_n\texttt{>}$ is of minimal length iff for all $l = 1, \ldots, n$, $\alpha_l : x_l \to y_l$ is in normal form, and for all $k = 1, \ldots, n-1$:*

- *α_k is layer-collapsing, with $y_k = \eta^w_{j,v}(z)$ $(w, v \in W, j \in \mathcal{L})$, and there are $r, s \in W, i \in \mathcal{L}$ s.t. $w = ri$, $v = is$, $i \neq j$, $x_{k+1} = \mu^r_{i,s}(z)$ and for all $\beta : z \to z'$, $\mu^r_{i,s}(\beta) \neq \alpha_{k+1}$;*
- *or α_{k+1} is layer-expanding, with $x_{k+1} = \eta^w_{j,v}(z)$ $(w, v \in W, j \in \mathcal{L})$, and there are $r, s \in W, i \in \mathcal{L}$ s.t. $w = ri$, $v = is$, $i \neq j$, $y_k = \mu^r_{i,s}(z)$ and for all $\beta : z' \to z$, $\mu^r_{i,s}(\beta) \neq \alpha_k$.*

Then (α_k, α_{k+1}) are called an incomposable pair*.*

Proof. A is of minimal length iff we cannot compose α_k and α_{k+1}, and there are no β_k, β_{k+1} which are composable and equivalent to α_k, α_{k+1}. In particular, $y_k \neq x_{k+1}$ but $\mathrm{NF}(y_k) = \mathrm{NF}(x_{k+1})$, hence $\mathrm{NF}(y_k) \neq y_k$ or $x_{k+1} \neq \mathrm{NF}(x_{k+1})$. By lemma 31,$\alpha_k$ is layer-collapsing iff $\mathrm{NF}(y_k) \neq y_k$. Then the second part of the first clause ensures that there is no β' which is composable with α_k s.t. $Q(\beta) = Q(\alpha_{k+1})$. The second clause is the dual of the first (with $x_{k+1} \neq \mathrm{NF}(x_{k+1})$). □

We close this section by showing that given two monad morphisms $\kappa : \mathsf{T}_1 \to \mathsf{S}_1$, $\lambda : \mathsf{T}_2 \to \mathsf{S}_2$, their coproduct $\kappa + \lambda : \mathsf{T}_1 + \mathsf{T}_2 \to \mathsf{S}_1 + \mathsf{S}_2$ preserves the normal form with respect to the two reduction systems $\twoheadrightarrow_{Ob}, \twoheadrightarrow_{Mor}$ above. Intuitively $\kappa + \lambda$ replaces every T_1 layer with its image in S_1 under κ and similarly for T_2 layers. Formally the components of $\kappa + \lambda$ are constructed by defining the obvious cone over the diagram whose colimit defines $(\mathsf{T}_1 + \mathsf{T}_2)(X)$. It is however not the case that $\kappa + \lambda$ preserves sequences of minimal length, since in general the conditions of lemma 33 are not preserved.

Lemma 34. *Given $\kappa : \mathsf{T}_1 \to \mathsf{S}_1$, $\lambda : \mathsf{T}_2 \to \mathsf{S}_2$ which are epi, let $M \stackrel{def}{=} \kappa + \lambda$. Then $M(NF(t)) = NF(M(t))$ for $t \in T^w\mathcal{X}$, and $M(NF(\alpha)) = NF(M(\alpha))$ for $\alpha : s \to t$ in $T^w\mathcal{X}$.*

Proof. By induction on the derivations $t \twoheadrightarrow_{Ob} \mathrm{NF}(t)$, $\alpha \twoheadrightarrow_{Mor} \mathrm{NF}(\alpha)$. Essentially, whenever we can reduce $t \rightarrow_{Ob} t'$, then we can reduce $M(t) \rightarrow_{Ob} M(t')$ (by naturality of the unit and multiplication of T_1 and T_2, and κ and λ being monad morphisms.) □

6 Modularity of Confluence

We now prove the modularity of confluence. The first step towards proving confluence is to find conditions under which a functor preserves confluence.

Definition 35 (One-Step Completion). 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 36. *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 $\alpha = [\langle\alpha_1, \ldots, \alpha_n\rangle]$ and $\beta = [\langle\beta_1, \ldots, \beta_m\rangle]$ in $\mathcal{Z}$ with the same source. 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 $\alpha' \stackrel{def}{=} [\langle\alpha_1^{(m)}, \ldots, \alpha_n^{(m)}\rangle]$, $\beta' \stackrel{def}{=} [\langle\beta_1^{(n)}, \ldots, \beta_m^{(n)}\rangle]$ such that $\beta' \cdot \alpha = \alpha' \cdot \beta$. □

To prove that the coequalizer of diagram 4 is confluent we show that $\coprod_{w \in W} T^w X \models_Q \diamond$ where Q is the coequalising functor. In [Lü96], this was done using a *witness relation*. Here, the witnesses are replaced by the conceptually simpler normal forms.

Lemma 37. *The coproduct of two confluent, non-expanding, regular monads is confluent.*

Proof. We first show that if $\mathcal{X}$ is confluent then $\coprod_{w \in W} T^w \mathcal{X} \models_Q \diamond$. Since $\mathcal{X}$, T_1 and T_2 are confluent and coproducts in **Cat** preserve confluence, $\coprod_{w \in W} T^w \mathcal{X}$ is confluent. Given $\alpha : x \to x'$ and $\beta : y \to y'$ in $\coprod_{w \in W} T^w \mathcal{X}$ such that $Qx = Qy$, by lemma 31 ($\mathsf{T}_1, \mathsf{T}_2$ are non-expanding) $\mathrm{NF}(\alpha) : \mathrm{NF}(x) \to x_0$ and $\mathrm{NF}(\beta) : \mathrm{NF}(y) \to y_0$. Further, since $Qx = Qy$ by lemma 27 $\mathrm{NF}(x) = \mathrm{NF}(y)$. Hence there are completions $\gamma : x_0 \to z_0$ and $\delta : y_0 \to z_0$ s.t. $\gamma \cdot \mathrm{NF}(\alpha) = \delta \cdot \mathrm{NF}(\beta)$, and hence $Q\gamma \cdot Q\alpha = Q\delta \cdot Q\beta$.

Then by lemma 36 the colimit of diagram 4 is confluent if $\mathcal{X}$ is confluent, and so the coproduct monad is confluent. □

The modularity of confluence for TRSs follows easily:

Theorem 38 (Toyama). *Confluence is modular for non-expanding TRSs.*

Proof. Let Θ_1 and Θ_2 be confluent TRSs. By lemma 17 the monads T_{Θ_1} and T_{Θ_2} are regular, non-expanding and by lemma 20 confluent, and so is their coproduct (lemma 37). By lemma 15 this coproduct models the disjoint union of Θ_1 and Θ_2 and hence by lemma 20 this TRS is confluent. $\square$

7 Modularity of Strong Normalisation

As mentioned in the introduction, strong normalisation (SN) is *not* a modular property for the disjoint union of term rewriting systems. We will below find conditions under which the disjoint union of two SN monads cannot be strongly normalising, from which several conditions under which the union is SN will be derived. This is an adaption of the minimal counterexamples technique in [Gra92]. We first need a criterion to determine when the disjoint union of two monads is not SN.

Lemma 39. *Given monads $\mathsf{T}_1, \mathsf{T}_2$ on* **Cat** *s.t. $\mathsf{T}_1 \models SN, \mathsf{T}_2 \models SN$, then $\mathsf{T}_1 + \mathsf{T}_2 \not\models SN$ iff for all $n \in \mathbb{N}$ there is a sequence A of minimal length s.t. $|A| > n$ (called an* infinite sequence*).*

Proof. The lemma follows from the observation that every sequence $A = \langle\alpha_1, \ldots, \alpha_n\rangle$ (with $\alpha_i : x_i \to y_i$) of minimal length gives rise to a sequence $[x_1] < [x_2] < \ldots [y_n]$ in the underlying order $R^-(T\mathcal{X})$ of the coproduct monad at $\mathcal{X}$. See [Lü97] for the details. $\square$

A term rewriting system Θ is called *strongly normalising under deterministic collapses* (SNDC or $\mathcal{C}_\mathcal{E}$-terminating) [Ohl94,Gra92] if it is SN and the disjoint union $\Theta + \mathcal{C}_\mathcal{E}$ is SN, where $\mathcal{C}_\mathcal{E}$ is the term rewriting system $\mathcal{C}_\mathcal{E} \stackrel{def}{=} \{\mathtt{G}(\text{'}x, \text{'}y) \to \text{'}x, \mathtt{G}(\text{'}x, \text{'}y) \to \text{'}y\}$. A recent term rewriting result is that the disjoint union is not SN if one system is SNDC and the other collapsing. The term rewriting proof is a rather intricate encoding construction. In this setting, the proof is far simpler: we find a monad $\mathsf{T}_\perp$ representing $\mathcal{C}_\mathcal{E}$ and then analyse its combination with T_1. This combination will be obtained by a universal property of $\mathsf{T}_\perp$. We first define SNDC for monads, and show this definition is equivalent to the one used in term rewriting:

Definition 40 (SNDC). A monad T on **Cat** is called *strongly normalising under deterministic collapses*, $T \models \text{SNDC}$ if $T \models \text{SN} \wedge T + T_\perp \models \text{SN}$

Lemma 41. *The term rewriting system Θ is strongly normalising under deterministic collapses iff. the monad T_Θ is.*

Proof. Using lemma 39, we must show there is an infinite reduction in $\Theta + \mathcal{C}_\mathcal{E}$ iff there is an infinite sequence of minimal length in $T_1 + T_\perp$. One direction is easy, since every non-identity rewrite $\alpha : s \to t$ in $T_1 + T_\perp$ gives rise to at least one rewrite step in $\Theta + \mathcal{C}_\mathcal{E}$. For the other direction, we draw upon [Gra92, Lemma 2], which shows that an infinite derivation in $\Theta + \mathcal{C}_\mathcal{E}$ contains infinitely many rewrites which satisfy the criteria of lemma 33[2]. $\square$

[2] Namely, they are destructive at level 2.

Definition 42 (A Monad called $\mathsf{T}_\perp$). The monad $\mathsf{T}_\perp = \langle T_\perp, \eta_\perp, \mu_\perp \rangle$ on **Cat** is defined as follows: it maps a category $\mathcal{X}$ to the category $T_\perp(\mathcal{X})$, which has as objects $|T_\perp \mathcal{X}| \stackrel{def}{=} \{\perp\} + |\mathcal{X}|$ and as morphisms

$$T_\perp(\mathcal{X})(x, y) \stackrel{def}{=} \begin{cases} \{!_y\} & \text{if } x = \perp \\ \emptyset & \text{if } x \neq \perp, y = \perp \\ \mathcal{X}(x, y) & \text{otherwise} \end{cases}$$

with the evident composition (for $f : x \to y$, $f \cdot !_x = !_y$ etc.). For a functor $F : \mathcal{X} \to \mathcal{Y}$, $T_\perp(F)$ maps $\perp$ to $\perp$, and x (with $x \in \mathcal{X}$) to Fx in $T_\perp(\mathcal{Y})$, and similarly on the morphisms. The unit $\eta_{\perp,\mathcal{X}} : \mathcal{X} \to T_\perp \mathcal{X}$ is the injection of the category $\mathcal{X}$ into $T_\perp \mathcal{X}$, and the multiplication $\mu_{\perp,\mathcal{X}} : T_\perp T_\perp \mathcal{X} \to T_\perp \mathcal{X}$ identifies the two adjoined objects.

From the term rewriting point, this monad can be seen as representing the system $\mathcal{C}_\mathcal{E}$.[3] From the categorical point of view, this monad freely adjoins an initial object $\perp$ to a category. This monad is terminal amongst non-expanding monads on **Cat**:

Lemma 43. *For any non-expanding monad* T *on* **Cat***, there is a unique monad morphism* $!_T : \mathsf{T} \to \mathsf{T}_\perp$*.*

Proof. For a category $\mathcal{X}$, $!_{T,\mathcal{X}} : T\mathcal{X} \to T_\perp \mathcal{X}$ is defined on objects as

$$!_{T,\mathcal{X}}(x) \stackrel{def}{=} \begin{cases} x_0 & \text{if } x = \eta(x_0) \\ \perp & \text{otherwise} \end{cases}$$

and similarly on the morphisms. This is the only definition which makes $!_T$ a monad morphism. Note $!_{T,\mathcal{X}}$ is only a functor if T is non-expanding. $\square$

By lemma 43, for two monads T_1 and T_2, there is a monad morphism $1{+}! : \mathsf{T}_1 + \mathsf{T}_2 \to \mathsf{T}_1 + \mathsf{T}_\perp$, substituting all compound terms from T_2 in $\mathsf{T}_1 + \mathsf{T}_2$ with the object $\perp$ from $T_\perp$. The proof of our main result proceeds as follows: since $\mathsf{T}_1 + \mathsf{T}_2$ is not SN, there is an infinite sequence of minimal length, A. We consider the image of A under the monad morphism $M \stackrel{def}{=} 1{+}! : \mathsf{T}_1 + \mathsf{T}_2 \to \mathsf{T}_1 + T_\perp$. From lemma 33, we know when a sequence has minimal length. We will show that the monad morphism preserves these properties, so there will be an infinite sequence of minimal length in $\mathsf{T}_1 + T_\perp$ as well, showing that T_1 is not SNDC. Recall from lemma 33 the notion of incomposable pairs. Assuming both monads to be non-expanding, only the second case of lemma 33 applies:

Lemma 44. *Given an incomposable pair* (α, β) *where* α *is layer-collapsing in* T_2*, then* $(M\alpha, M\beta)$ *is an incomposable pair as well.*

[3] Although of course $\mathsf{T}_\perp$ is not the monad $\mathsf{T}_{\mathcal{C}_\mathcal{E}}$ given by that system because of its multiplication.

Proof. By lemma 34, $M(\mathrm{NF}(\alpha)) = \mathrm{NF}(M\alpha)$ (as both 1 and ! are epi), so $M(\alpha) : x_1 \to y_1, M(\beta) : x_2 \to y_2$ is in normal form. The other conditions follow since M is given by a cone morphism ν between the two cones over the diagrams defining the colimits: since $\eta^v_{2,w}$ and $\mu^r_{1,s}$ are morphisms of the diagram, ν preserves them. □

Theorem 45. *Given two regular, non-expanding, SN monads* T_1 *and* T_2 *on* **Cat**, *if* $\mathsf{T}_1 + \mathsf{T}_2 \not\models$ *SN, then either* $\mathsf{T}_1 \not\models$ *SNDC and* T_2 *is collapsing or vice versa.*

Proof. By lemma 39, there is an infinite sequence $A = \langle\alpha_1, \ldots, \alpha_n \ldots\rangle$ of minimal length in $\mathsf{T}_1 + \mathsf{T}_2$. By lemma 33, all α_i in A have to be collapsing, so at least one of T_1 or T_2 is collapsing. Further, there are infinitely many rewrites collapsing in T_1, or infinitely many rewrites collapsing in T_2. Wolg. assume the latter, and consider the sequence MA in $\mathsf{T}_1 + \mathsf{T}_\perp$. If we compose all $M\alpha_i$ and $M\alpha_{i+1}$ which can be composed, we obtain a sequence A' of minimal length which is equivalent to MA, but by lemma 44, if α_i is collapsing in T_2, $(M\alpha_i, M\alpha_{i+1})$ will remain an incomposable pair, so A' will be infinite as well. Hence, by lemma 39, $\mathsf{T}_1 + \mathsf{T}_\perp$ is not strongly normalising, so $\mathsf{T}_1 \not\models$ SNDC.
□

Theorem 45 has a host of interesting corollaries such as:

Corollary 46. *The following modularity results follow from Theorem 45:*

1) *SN is modular for non-collapsing systems.*
2) *SN is modular for non-duplicating systems.*
3) *SN is modular if one system is non-collapsing and non-duplicating.*
4) *SN is modular for simplifying systems.*

Proof. The first is obvious. For the rest, non-duplicating and simplifying systems are strongly normalising under deterministic collapses [Gra92].
□

Hence, all of the conditions listed in the introduction follow as corollaries from Theorem 45. [Ohl94] contains further derived criteria.

8 Conclusions and Further Work

We have shown how monads can be used to give a semantics to term rewriting systems by generalising the well-known equivalence between universal algebra and finitary monads on **Set**. Monads are well suited to the study of modular term rewriting as the key concepts have concise monadic formulations. We believe this paper provides ample justification for these claims, and further for the more general claim that category theory provides a useful level of abstraction for the study of rewriting.

We propose to extend this work and tackle open problems in modular term rewriting. Firstly, monads can be used to model more general notions of term rewriting for which current modularity results are less than satisfactory. In particular research on modularity for conditional term rewriting is at an advanced stage. Another area where significant problems remain is that of modularity for unions which permit limited forms of sharing. Categorically, these unions are modelled by push-outs which again have a compositional semantics. This observation allows us to apply the methodology outlined in the introduction to study modularity for these more general structuring operations.

References

[BW85] M. Barr and C. Wells. *Toposes, Triples and Theories.* Springer 1985.

[Gha95] N. Ghani. *Adjoint Rewriting.* PhD thesis, University of Edinburgh, 1995.

[Gra92] B. Gramlich. Generalized sufficient conditions for modular termination of rewriting. In *Proc. 3rd ICALP*, LNCS 632, pages 53–68. Springer, 1992.

[Jay90] C. B. Jay. Modelling reductions in confluent categories. In *Proc. Durham Symposium on Applications of Categories in Computer Science*, 1990.

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

[Klo92] J. W. Klop. Term rewriting systems. In S. Abramsky et.al., eds., *Handbook of Logic in Computer Science* Vol. 2, pages 1–116. OUP, 1992.

[KO92] M. Kurihara and A. Ohuchi. Modularity of simple termination of term rewriting systems with shared constructors. *TCS* 103:273– 282, 1992.

[KP93] G. M. Kelly and A. J. Power. Adjunctions whose counits are coequalizers, and presentations of finitary monads. *JPAA* 89:163– 179, 1993.

[Lü96] C. Lüth. Compositional term rewriting: An algebraic proof of Toyama's theorem. In *RTA'96*, LNCS 1103, pages 261– 275, Springer Verlag, 1996.

[Lü97] C. Lüth. *Categorical Term Rewriting: Monads and Modularity.* PhD thesis, University of Edinburgh, 1997. Forthcoming.

[Mac71] S. Mac Lane. *Categories for the Working Mathematician.* Springer 1971.

[Man76] E. G. Manes. *Algebraic Theories*, Springer Verlag, 1976.

[Mid89] A. Middeldorp. A sufficient condition for the termination of the direct sum of term rewriting systems. In *Proc. 4th LICS*, p. 396–401. June 1989.

[Ohl94] E. Ohlebusch. On the modularity of termination of term rewriting systems. *TCS* 136:333– 360, 1994.

[Rob94] E. Robinson. Variations on algebra: monadicity and generalisations of equational theories. Tech. Rep. 6/94, Sussex Univ. Comp. Sci., 1994.

[RS87] D. E. Rydeheard and J. G. Stell. Foundations of equational deduction: A categorical treatment of equational proofs and unification algorithms. In *CTCS '87*, LNCS 283, pages 114– 139. Springer Verlag, 1987.

[Rus87] M. Rusinowitch. On the termination of the direct sum of term-rewriting systems. *Information Processing Letters*, 26(2):65–70, 1987.

[See87] R. A. G. Seely. Modelling computations: A 2-categorical framework. In *Proc. 2nd LICS*, pages 65–71, 1987.

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