Modular Specification of Monads Through Higher-Order Presentations

Benedikt Ahrens
University of Birmingham, United Kingdom
B.Ahrens@cs.bham.ac.uk

André Hirschowitz
Université Côte d'Azur, CNRS, LJAD, Nice, France
ah@unice.fr

Ambroise Lafont
IMT Atlantique, Inria, LS2N CNRS, Nantes, France
ambroise.lafont@inria.fr

Marco Maggesi
Università degli Studi di Firenze, Italy
marco.maggesi@unifi.it

Abstract

In their work on second-order equational logic, Fiore and Hur have studied presentations of simply typed languages by generating binding constructions and equations among them. To each pair consisting of a binding signature and a set of equations, they associate a category of "models", and they give a monadicity result which implies that this category has an initial object, which is the language presented by the pair.

In the present work, we propose, for the untyped setting, a variant of their approach where monads and modules over them are the central notions. More precisely, we study, for monads over sets, presentations by generating ("higher-order") operations and equations among them. We consider a notion of 2-signature which allows to specify a monad with a family of binding operations subject to a family of equations, as is the case for the paradigmatic example of the lambda calculus, specified by its two standard constructions (application and abstraction) subject to β- and η-equalities. Such a 2-signature is hence a pair (Σ,E) of a binding signature Σ and a family E of equations for Σ. This notion of 2-signature has been introduced earlier by Ahrens in a slightly different context.

We associate, to each 2-signature (Σ, E), a category of "models of (Σ, E)"; and we say that a 2-signature is "effective" if this category has an initial object; the monad underlying this (essentially unique) object is the "monad specified by the 2-signature". Not every 2-signature is effective; we identify a class of 2-signatures, which we call "algebraic", that are effective.

Importantly, our 2-signatures together with their models enjoy "modularity": when we glue (algebraic) 2-signatures together, their initial models are glued accordingly.

We provide a computer formalization for our main results.

2012 ACM Subject Classification Theory of computation → Algebraic language theory

Keywords and phrases free monads, presentation of monads, initial semantics, signatures, syntax, monadic substitution, computer-checked proofs

Digital Object Identifier 10.4230/LIPIcs.FSCD.2019.6

Supplement Material Computer-checked proofs with compilation instructions on `https://github.com/UniMath/largecatmodules/tree/50fd617`.

Funding This work has partly been funded by the CoqHoTT ERC Grant 637339. This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-17-1-0363.
Benedikt Ahrens: Ahrens acknowledges the support of the Centre for Advanced Study (CAS) in Oslo, Norway, which funded and hosted the research project *Homotopy Type Theory and Univalent Foundations* during the 2018/19 academic year.
Marco Maggesi: Supported by GNSAGA-INdAM and MIUR.

4th International Conference on Formal Structures for Computation and Deduction (FSCD 2019).
Editor: Herman Geuvers; Article No. 6; pp. 6:1–6:19
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Acknowledgements We thank Paige R. North for a valuable hint regarding preservation of epimorphisms. We also thank the referees for their careful reading and thoughtful and constructive criticism.

1 Introduction

The present work is devoted to the study of presentations of monads on the category of sets. More precisely, there is a well established theory of presentations of monads through generating (first-order) operations equipped with relations among the corresponding derived operations. Here we propose a counterpart of this theory, where we consider generation of monads by *binding operations*. Various algebraic structures generated by binding operations have been considered by many, going back at least to Fiore, Plotkin, and Turi [10], Gabbay and Pitts [12], and Hofmann [18]. Every such operation has a binding arity, which is a sequence of non-negative integers. For example, the binding arity of the application operation of the lambda calculus is $(0, 0)$: it takes two arguments without binding any variable in them, while the abstraction operation on the monad of the lambda calculus has binding arity (1), as it binds one variable in its single argument. For each family Σ of binding arities, there is a generated "free" monad $\hat{\Sigma}$ on Set which maps a set of free variables X to the set of terms $\hat{\Sigma}(X)$ taking variables in X.

If $p : \hat{\Sigma} \to R$ is a monad epimorphism, we understand that R is generated by a family of operations whose binding arities are given by Σ, subject to suitable identifications. In particular, for $\Sigma := ((0, 0), (1))$, $\hat{\Sigma}$ may be understood as the monad LC of syntactic terms of the lambda calculus, and we have an obvious epimorphism $p : \hat{\Sigma} \to \mathsf{LC}_{\beta\eta}$, where $\mathsf{LC}_{\beta\eta}$ is the monad of lambda-terms modulo β and η. In order to manage such equalities, the approach in the first-order case suggests to identify p as the coequalizer of a double arrow from T to $\hat{\Sigma}$ where T is again a "free" monad. Let us see what comes out when we attempt to find such an encoding for the β-equality of the monad $\mathsf{LC}_{\beta\eta}$. It should say that for each set X, the following two maps from $\hat{\Sigma}(X + \{*\}) \times \hat{\Sigma}(X)$ to $\hat{\Sigma}(X)$,

- $(t, u) \mapsto \mathsf{app}(\mathsf{abs}(t), u)$
- $(t, u) \mapsto t[* \mapsto u]$

are equal. Here a problem occurs, namely that the above collections of maps, which can be understood as a morphism of functors, cannot be understood as a morphism of monads. Notably, they do not send variables to variables.

On the other hand, we observe that the members of our equations, which are not morphisms of monads, commute with substitution, and hence are more than morphisms of functors: indeed they are morphisms of *modules over* $\hat{\Sigma}$. (In Section 2, we recall briefly what modules over a monad are.) Accordingly, a (second-order) presentation for a monad R could be a diagram

$$T \overset{f}{\rightrightarrows} \hat{\Sigma} \xrightarrow{p} R \tag{1}$$

where Σ is a binding signature, $\hat{\Sigma}$ is the associated free monad, T is a module over $\hat{\Sigma}$, f is a pair of morphisms of modules over $\hat{\Sigma}$, and p is a monad epimorphism. And now we are faced with the task of finding a condition meaning something like "p is the coequalizer of f"[1]. To this end, we introduce the category Mon^{Σ} "of models of Σ", whose objects are monads "equipped with an action of Σ". Of course $\hat{\Sigma}$ is equipped with such an action which turns it into the initial object. Next, we define the full subcategory of models satisfying the equation

[1] This cannot be the case stricto sensu since f is a pair of module morphisms while p is a monad morphism.

f, and require R to be the initial object therein. Our definition is suited for the case where the equation f is parametric in the model: this means that now T and f are functions of the model S, and $f(S) = (u(S), v(S))$ is a pair of S-module morphisms from $T(S)$ to S. We say that S satisfies the equation f if $u(S) = v(S)$. Generalizing the case of one equation to the case of a family of equations yields the notion of 2-signature already introduced by Ahrens [1] in a slightly different context.

Now we are ready to formulate our main problem: given a 2-signature (Σ, E), where E is a family of parametric equations as above, does the subcategory of models of Σ satisfying the family of equations E admit an initial object?

We answer positively for a large subclass of 2-signatures which we call *algebraic* 2-signatures (see Theorem 32).

This provides a construction of a monad from an algebraic 2-signature, and we prove furthermore (see Theorem 27) that this construction is *modular*, in the sense that merging two extensions of 2-signatures corresponds to building an amalgamated sum of initial models. This is analogous to our previous result for 1-signatures shown in [2, Theorem 32].

As expected, our initiality property generates a recursion principle which is a recipe allowing us to specify a morphism from the presented monad to any given other monad.

We give various examples of monads arising "in nature" that can be specified via an algebraic 2-signature (see Section 6), and we also show through a simple example how our recursion principle applies (see Section 7).

Computer-checked formalization. This work is accompanied by a computer-checked formalization of the main results, based on the formalization of our previous work [2]. We work over the UniMath library [27], which is implemented in the proof assistant Coq [23]. The formalization consists of about 9,500 lines of code, and can be consulted on `https://github.com/UniMath/largecatmodules`. A guide is given in the README, and a summary of our formalization is available at `https://initialsemantics.github.io/doc/50fd617/Modules.SoftEquations.Summary.html`.

For the purpose of this article, we refer to a fixed version of our library, with the short hash 50fd617. This version compiles with version 10839ee of `UniMath`.

Throughout the article, statements are annotated with their corresponding identifiers in the formalization. These identifiers are also hyperlinks to the online documentation stored at `https://initialsemantics.github.io/doc/50fd617/index.html`.

Related work. The present work follows a previous work of ours [2] where we study a slightly different kind of presentation of monads. Specifically, in [2], we treat a class of 1-signatures which can be understood as quotients of algebraic 1-signatures. This should amount to considering a specific kind of equations, as suggested in Section 6.2, where we recover, in the current setting, all the examples given there.

Ahrens [1] introduces the notion of 2-signature which we consider here, in the slightly different context of (relative) monads on preordered sets, where the preorder models the reduction relation. In some sense, our result tackles the technical issue of quotienting the initial (relative) monad constructed in [1] by the preorder.

In a classical paper, Barr [3] explained the construction of the "free monad" generated by an endofunctor[2]. In another classical paper, Kelly and Power [19] explained how any finitary

[2] Fiore and Saville [11] give an enlightening generalization of the construction by Barr.

monad can be presented as a coequalizer of free monads[3]. There, free monads correspond to our initial models of an algebraic 1-signature without any binding construction.

As mentioned above, the present work is also closely related to that of Fiore and collaborators:

- Our notion of equations and that of model for them seem very close to the notion of equational systems and that of algebra for them in [7]: in particular, the preservation of epimorphisms, which occurs in their construction of inductive free algebras for equational systems, appears here in our definition of elementary equation. It would be interesting to understand formal connections between the two approaches.
- In [8], Fiore and Hur introduce a notion of equation based on syntax with *meta-variables*: essentially, a specific syntax, say, $T := T(M, X)$ considered there depends on two contexts: a meta-context M, and an object-context X. The terms of the actual syntax are then those terms $t \in T(\emptyset, X)$ in an empty meta-context. An equation for T is, simply speaking, a pair of terms in the same pair of contexts. Transferring an equation to any model of the underlying algebraic 1-signature is done by induction on the syntax with meta-variables. The authors show a monadicity theorem which straightforwardly implies an initiality result very similar to ours. That monadicity result is furthermore an instance of a more general theorem by Fiore and Mahmoud [9, Theorem 6.2].
- Translations between languages similar to the translation we present in Section 7 are also studied in [9]. Here again, it would be interesting to understand formal connections.
- At this stage, our work only concerns untyped syntax, but we anticipate it will generalize to the sorted setting as in [8] (see also the more general [6]).

Furthermore, Hamana [15] proposes initial algebra semantics for “binding term rewriting systems”, based on Fiore, Plotkin, and Turi’s presheaf semantics of variable binding and Lüth and Ghani’s monadic semantics of term rewriting systems [21].

The alternative *nominal* approach to binding syntax initiated by Gabbay and Pitts [12] has been actively studied[4]. We highlight some contributions:

- Clouston [4] discusses signatures, structures (a.k.a. models), and equations over signatures in nominal style.
- Fernández and Gabbay [5] study signatures and equational theories as well as rewrite theories over signatures.
- Kurz and Petrisan [20] study closure properties of subcategories of algebras under quotients, subalgebras, and products. They characterize full subcategories closed under these operations as those that are definable by equations. They also show that the signature of the lambda calculus is effective, and study the subcategory of algebras of that signature specified by the β- and η-equations.

2 Categories of modules over monads

In this section, we recall the notions of monad and module over a monad, as well as some constructions of modules. We restrict our attention to the category Set of sets, although most definitions are straightforwardly generalizable. See [17] for a more extensive introduction.

A **monad** (over Set) is a triple $R = (R, \mu, \eta)$ given by a functor $R\colon \mathsf{Set} \longrightarrow \mathsf{Set}$, and two natural transformations $\mu\colon R \cdot R \longrightarrow R$ and $\eta\colon I \longrightarrow R$ such that the well-known monadic laws hold. A **monad morphism** to another such monad (R', μ', η') is a natural

[3] Their work has been applied to various more general contexts (e.g. [25]).

[4] The approaches by Fiore and collaborators and Gabbay and Pitts [12] are nicely compared by Power [24], who also comments on some generalization of the former approach.

transformation $f : R \to R'$ that commutes with the monadic structure. The category of monads is denoted by Mon.

Let R be a monad. A **(left) R-module**[5] is given by a functor $M\colon \mathsf{Set} \longrightarrow \mathsf{Set}$ equipped with a natural transformation $\rho\colon M \cdot R \longrightarrow M$, called *module substitution*, which is compatible with the monad composition and identity:

$$\rho \circ \rho R = \rho \circ M\mu, \qquad \rho \circ M\eta = 1_M.$$

Let $f\colon R \longrightarrow S$ be a morphism of monads and M an S-module. The module substitution $M \cdot R \xrightarrow{Mf} M \cdot S \xrightarrow{\rho} M$ turns M into an R-module f^*M, called **pullback of M along f**.

A natural transformation of R-modules $\varphi\colon M \longrightarrow N$ is **linear** if it is compatible with module substitution on either side, that is, if $\varphi \circ \rho^M = \rho^N \circ \varphi R$. Modules over R and their morphisms form a category denoted $\mathsf{Mod}(R)$, which is complete and cocomplete: limits and colimits are computed pointwise.

We define the **total module category** $\int_R \mathsf{Mod}(R)$ as follows: its objects are pairs (R, M) of a monad R and an R-module M. A morphism from (R, M) to (S, N) is a pair (f, m) where $f\colon R \longrightarrow S$ is a morphism of monads, and $m\colon M \longrightarrow f^*N$ is a morphism of R-modules. The category $\int_R \mathsf{Mod}(R)$ comes equipped with a forgetful functor to the category of monads, given by the projection $(R, M) \mapsto R$. This functor is a Grothendieck fibration with fiber $\mathsf{Mod}(R)$ over R. In particular, any monad morphism $f : R \longrightarrow S$ gives rise to a functor $f^*\colon \mathsf{Mod}(S) \longrightarrow \mathsf{Mod}(R)$ which preserves limits and colimits.

▶ **Example 1.** We give some important examples of modules:
1. Every monad R is a module over itself, which we call the **tautological module**.
2. For any functor $F\colon \mathsf{Set} \longrightarrow \mathsf{Set}$ and any R-module $M\colon \mathsf{Set} \longrightarrow \mathsf{Set}$, the composition $F \cdot M$ is an R-module (in the evident way).
3. For every set W we denote by $\underline{\mathrm{W}}\colon \mathsf{Set} \longrightarrow \mathsf{Set}$ the constant functor $\underline{\mathrm{W}} := X \mapsto W$. Then $\underline{\mathrm{W}}$ is trivially an R-module since $\underline{\mathrm{W}} = \underline{\mathrm{W}} \cdot R$.
4. Given an R-module M, the R-module M' is defined on objects by $M'(X) := M(X + \{*\})$, and with the obvious module structure. Derivation yields an endofunctor on $\mathsf{Mod}(R)$ that is right adjoint to the functor $M \mapsto M \times R$, "product with the tautological module". Details are given, e.g., in [2, Section 2.3].
5. Derivation can be iterated. Given a list of non negative integers $(a) = (a_1, \ldots, a_n)$ and a left module M over a monad R, we denote by $M^{(a)} = M^{(a_1,\ldots,a_n)}$ the module $M^{(a_1)} \times \cdots \times M^{(a_n)}$, with $M^{()} = 1$ the final module.

3 1-signatures and their models

In this section, we review the notion of 1-signature studied in detail in [2] – there only called "signature".

A **1-signature** is a section of the forgetful functor from the category $\int_R \mathsf{Mod}(R)$ to the category Mon. A **morphism between two 1-signatures** $\Sigma_1, \Sigma_2\colon \mathsf{Mon} \longrightarrow \int_R \mathsf{Mod}(R)$ is a natural transformation $m\colon \Sigma_1 \longrightarrow \Sigma_2$ which, post-composed with the projection $\int_R \mathsf{Mod}(R) \longrightarrow \mathsf{Mon}$, is the identity. The category of 1-signatures is denoted by $\mathsf{1\text{-}Sig}$.

Limits and colimits of 1-signatures can be easily constructed pointwise: the category of 1-signatures is complete and cocomplete.

[5] The analogous notion of *right* R-module is not used in this work, we hence simply write "R-module" instead of "left R-module" for brevity.

Table 1 Examples of 1-signatures.

Hypotheses	On objects	Notation of the 1-signature
	$R \mapsto R$	Θ
Σ 1-signature, F functor	$R \mapsto F \cdot \Sigma(R)$	$F \cdot \Sigma$
	$R \mapsto 1_R$	1
Σ, Ψ 1-signatures	$R \mapsto \Sigma(R) \times \Psi(R)$	$\Sigma \times \Psi$
Σ, Ψ 1-signatures	$R \mapsto \Sigma(R) + \Psi(R)$	$\Sigma + \Psi$
	$R \mapsto R'$	Θ'
$n \in \mathbb{N}$	$R \mapsto R^{(n)}$	$\Theta^{(n)}$
$(a) = (a_1, \ldots, a_n) \in \mathbb{N}^n$	$R \mapsto R^{(a)} = R^{(a_1)} \times \ldots \times R^{(a_n)}$	$\Theta^{(a)}$ **elementary signatures**

Table 1 lists important examples of 1-signatures. An **algebraic 1-signature** is a (possibly infinite) coproduct of elementary signatures (defined in Table 1). For instance, the algebraic 1-signature of the lambda calculus is $\Sigma_{\mathsf{LC}} = \Theta^2 + \Theta'$.

Given a monad R over Set, we define an **action of the 1-signature $\boldsymbol{\Sigma}$ in $\boldsymbol{R}$** to be a module morphism from $\Sigma(R)$ to R. For example, the application $\mathsf{app}\colon \mathsf{LC}^2 \longrightarrow \mathsf{LC}$ is an action of the elementary 1-signature Θ^2 into the monad LC of syntactic lambda calculus. The abstraction $\mathsf{abs}\colon \mathsf{LC}' \longrightarrow \mathsf{LC}$ is an action of the elementary 1-signature Θ' into the monad LC. Then $[\mathsf{app}, \mathsf{abs}] : \mathsf{LC}^2 + \mathsf{LC}' \longrightarrow \mathsf{LC}$ is an action of the algebraic 1-signature of the lambda-calculus $\Theta^2 + \Theta'$ into the monad LC.

Given a 1-signature Σ, we build the category Mon^{Σ} of **models of $\boldsymbol{\Sigma}$** as follows. Its objects are pairs (R, r) of a monad R equipped with an action $r : \Sigma(R) \to R$ of Σ. A morphism from (R, r) to (S, s) is a morphism of monads $m : R \to S$ making the following diagram of R-modules commutes:

$$\begin{tikzcd}
\Sigma(R) \arrow[r, "r"] \arrow[d, "\Sigma(m)"'] & R \arrow[d, "m"] \\
m^*(\Sigma(S)) \arrow[r, "m^* s"'] & m^* S
\end{tikzcd}$$

Let $f\colon \Sigma \longrightarrow \Psi$ be a morphism of 1-signatures and $\mathcal{R} = (R, r)$ a model of Ψ. The linear morphism $\Sigma(R) \xrightarrow{f(R)} \Psi(R) \xrightarrow{r} R$ defines an action of Σ in R. The induced model of Σ is called **pullback** of $\mathcal{R}$ along f and noted $f^*\mathcal{R}$.

The **total category** $\int_\Sigma \mathsf{Mon}^{\Sigma}$ of models is defined as follows:

- An object of $\int_\Sigma \mathsf{Mon}^{\Sigma}$ is a triple (Σ, R, r) where Σ is a 1-signature, R is a monad, and r is an action of Σ in R.
- A morphism in $\int_\Sigma \mathsf{Mon}^{\Sigma}$ from (Σ_1, R_1, r_1) to (Σ_2, R_2, r_2) consists of a pair (i, m) of a 1-signature morphism $i : \Sigma_1 \longrightarrow \Sigma_2$ and a morphism m of Σ_1-models from (R_1, r_1) to $(R_2, i^*(r_2))$.

The forgetful functor $\int_\Sigma \mathsf{Mon}^{\Sigma} \to \mathsf{Sig}$ is a Grothendieck fibration.

Given a 1-signature Σ, the initial object in Mon^{Σ}, if it exists, is denoted by $\hat{\Sigma}$. In this case, the 1-signature Σ is said **effective**[6].

▶ **Theorem 2** ([16, Theorems 1 and 2]). *Algebraic 1-signatures are effective.*

[6] In our previous work [2], we call **representable** any 1-signature Σ that has an initial model, called a **representation** of Σ, or **syntax generated by** Σ.

4 2-Signatures and their models

In this section we study *2-signatures* and *models of 2-signatures.* A 2-signature is a pair of a 1-signature and a family of *equations* over it.

4.1 Equations

Our equations are those of Ahrens [1]: they are parallel module morphisms parametrized by the models of the underlying 1-signature. The underlying notion of 1-model is essentially the same as in [1], even if, there, such equations are interpreted instead as *inequalities.*

Throughout this subsection, we fix a 1-signature Σ, that we instantiate in the examples.

▶ **Definition 3.** *We define a* ***Σ-module*** *to be a functor T from the category of models of Σ to the category $\int_R \mathsf{Mod}(R)$ commuting with the forgetful functors to the category* Mon *of monads,*

$$\begin{tikzcd} \mathsf{Mon}^{\Sigma} \arrow[rr, "T"] \arrow[dr] & & \int_R \mathsf{Mod}(R) \arrow[dl] \\ & \mathsf{Mon} & \end{tikzcd}$$

▶ **Example 4.** To each 1-signature Ψ is associated, by precomposition with the projection from Mon^{Σ} to Mon, a Σ-module still denoted Ψ. All the Σ-modules occurring in this work arise in this way from 1-signatures; in other words, they do not depend on the action of the 1-model. In particular, we have the **tautological Σ-module** Θ, and, more generally, for any natural number $n \in \mathbb{N}$, a Σ-module $\Theta^{(n)}$. Also we have another fundamental Σ-module (arising in this way from) Σ itself.

▶ **Definition 5.** *Let S and T be Σ-modules. We define a* ***morphism of Σ-modules*** *from S to T to be a natural transformation from S to T which becomes the identity when postcomposed with the forgetful functor $\int_R \mathsf{Mod}(R) \to \mathsf{Mon}$.*

▶ **Example 6.** Each 1-signature morphism $\Psi \to \Phi$ upgrades into a morphism of Σ-modules. Further in that vein, there is a morphism of Σ-modules $\tau^{\Sigma} : \Sigma \to \Theta$. It is given, on a model (R, m) of Σ, by $m : \Sigma(R) \to R$. (Note that it does not arise from a morphism of 1-signatures.) When the context is clear, we write simply τ for this morphism, and call it the **tautological morphism of Σ-modules**.

▶ **Proposition 7.** *Our Σ-modules and their morphisms, with the obvious composition and identity, form a category.*

▶ **Definition 8.** *We define a Σ-equation to be a pair of parallel morphisms of Σ-modules. We also write $e_1 = e_2$ for the Σ-equation $e = (e_1, e_2)$.*

▶ **Example 9** (Commutativity of a binary operation)**.** Here we instantiate our fixed 1-signature as follows: $\Sigma := \Theta \times \Theta$. In this case, we say that τ is the (tautological) binary operation. Now we can formulate the usual law of commutativity for this binary operation.

We consider the morphism of 1-signatures $\mathsf{swap} : \Theta^2 \longrightarrow \Theta^2$ that exchanges the two components of the direct product. Again by Example 6, we have an induced morphism of Σ-modules, still denoted swap.

Then, the Σ-equation for commutativity is given by the two morphisms of Σ-modules

$$\Theta^2 \xrightarrow{\mathsf{swap}} \Theta^2 \xrightarrow{\tau} \Theta$$
$$\Theta^2 \xrightarrow[\tau]{\qquad\qquad} \Theta$$

See also Section 6.1 where we explain in detail the case of monoids.

For the example of the lambda calculus with β- and η-equality (given in Example 11), we need to introduce *currying*:

▶ **Definition 10.** *By abstracting over the base monad R the adjunction in the category of R-modules of Example 1, item 4, we can perform* ***currying*** *of morphisms of 1-signatures: given a morphism of signatures $\Sigma_1 \times \Theta \to \Sigma_2$ it produces a new morphism $\Sigma_1 \to \Sigma_2'$. By Example 4, currying acts also on morphisms of Σ-modules.*

Conversely, given a morphism of 1-signatures (resp. Σ-modules) $\Sigma_1 \to \Sigma_2'$, we can define the ***uncurryied*** *map $\Sigma_1 \times \Theta \to \Sigma_2$.*

▶ **Example 11** **(β- and η-conversions).** Here we instantiate our fixed 1-signature as follows: $\Sigma_{\mathsf{LC}} := \Theta \times \Theta + \Theta'$. This is the 1-signature of the lambda calculus. We break the tautological Σ-module morphism into its two pieces, namely $\mathsf{app} := \tau \circ \mathsf{inl} : \Theta \times \Theta \longrightarrow \Theta$ and $\mathsf{abs} := \tau \circ \mathsf{inr} : \Theta' \longrightarrow \Theta$. Applying currying to app yields the morphism $\mathsf{app}_1 : \Theta \longrightarrow \Theta'$ of Σ_{LC}-modules. The usual β and η relations are implemented in our formalism by two Σ_{LC}-equations that we call e_β and e_η respectively:

$$e_\beta : \begin{array}{l} \Theta' \xrightarrow{\mathsf{abs}} \Theta \xrightarrow{\mathsf{app}_1} \Theta' \\ \Theta' \xrightarrow[1]{\qquad\qquad} \Theta' \end{array} \qquad \text{and} \qquad e_\eta : \begin{array}{l} \Theta \xrightarrow{\mathsf{app}_1} \Theta' \xrightarrow{\mathsf{abs}} \Theta \\ \Theta \xrightarrow[1]{\qquad\qquad} \Theta \end{array}$$

4.2 2-signatures and their models

▶ **Definition 12.** *A* ***2-signature*** *is a pair (Σ, E) of a 1-signature Σ and a family E of Σ-equations.*

▶ **Example 13.** The 2-signature for a commutative binary operation is $(\Theta^2, \tau \circ \mathsf{swap} = \tau)$ (cf. Example 9).

▶ **Example 14.** The 2-signature of the lambda calculus modulo β- and η-equality is $\Upsilon_{\mathsf{LC}_{\beta\eta}} = (\Theta \times \Theta + \Theta', \{e_\beta, e_\eta\})$, where e_β, e_η are the Σ_{LC}-equations defined in Example 11.

▶ **Definition 15** **(`satisfies_equation`).** *We say that a model M of Σ* ***satisfies the Σ-equation $e = (e_1, e_2)$*** *if $e_1(M) = e_2(M)$. If E is a family of Σ-equations, we say that a model M of Σ* ***satisfies E*** *if M satisfies each Σ-equation in E.*

▶ **Definition 16.** *Given a monad R and a 2-signature $\Upsilon = (\Sigma, E)$, an* ***action of Υ in*** *R is an action of Σ in R such that the induced 1-model satisfies all the equations in E.*

▶ **Definition 17** **(`category_model_equations`).** *For a 2-signature (Σ, E), we define the* ***category*** *$\mathsf{Mon}^{(\Sigma,E)}$* ***of models of*** *$(\Sigma, E)$ to be the full subcategory of the category of models of Σ whose objects are models of Σ satisfying E, or equivalently, monads equipped with an action of (Σ, E).*

▶ **Example 18.** A model of the 2-signature $\Upsilon_{\mathsf{LC}_{\beta\eta}} = (\Theta \times \Theta + \Theta', \{e_\beta, e_\eta\})$ is given by a model $(R, \mathsf{app}^R : R \times R \to R, \mathsf{abs}^R : R' \to R)$ of the 1-signature Σ_{LC} such that $\mathsf{app}_1^R \cdot \mathsf{abs}^R = 1_{R'}$ and $\mathsf{abs}^R \cdot \mathsf{app}_1^R = 1_R$ (see Example 11).

▶ **Definition 19.** *A 2-signature* (Σ, E) *is said to be* ***effective*** *if its category of models* $\mathsf{Mon}^{(\Sigma,E)}$ *has an initial object, denoted* $\widehat{(\Sigma, E)}$.

In Section 4.4, we aim to find sufficient conditions for a 2-signature (Σ, E) to be effective.

4.3 Modularity for 2-signatures

In this section, we define the category 2Sig of 2-signatures and the category 2Mod of models of 2-signatures, together with functors that relate them with the categories of 1-signatures and 1-models. The situation is summarized in the commutative diagram of functors

$$\begin{tikzcd}
\mathsf{2Mod} \arrow[r, bend left, "U_{\mathsf{Mod}}"] \arrow[d, "2\pi"'] \arrow[r, phantom, "\top"] & \mathsf{Mod} \arrow[l, bend left, "F_{\mathsf{Mod}}"] \arrow[d, "\pi"] \\
\mathsf{2Sig} \arrow[r, bend left, "U_{\mathsf{Sig}}"] \arrow[r, phantom, "\top"] & \mathsf{Sig} \arrow[l, bend left, "F_{\mathsf{Sig}}"]
\end{tikzcd}$$

where

- 2π is a Grothendieck fibration;
- π is the Grothendieck fibration defined in [2, Section 5.2];
- U_{Sig} is a coreflection and preserves colimits; and
- U_{Mod} is a coreflection.

As a simple consequence of this data, we obtain, in Theorem 27, a *modularity* result in the sense of Ghani, Uustalu, and Hamana [13]: it explains how the initial model of an amalgamated sum of 2-signatures is the amalgamation of the initial model of the summands[7].

We start by defining the category 2Sig of 2-signatures:

▶ **Definition 20** (`TwoSig_category`)**.** *Given 2-signatures* (Σ_1, E_1) *and* (Σ_2, E_2)*, a* ***morphism of 2-signatures from*** $\boldsymbol{(\Sigma_1, E_1)}$ ***to*** $\boldsymbol{(\Sigma_2, E_2)}$ *is a morphism of 1-signatures* $m : \Sigma_1 \to \Sigma_2$ *such that for any model* M *of* Σ_2 *satisfying* E_2*, the* Σ_1*-model* m^*M *satisfies* E_1.

These morphisms, together with composition and identity inherited from 1-signatures, form the category 2Sig.

We now study the existence of colimits in 2Sig. We know that Sig is cocomplete, and we use this knowledge in our study of 2Sig, by relating the two categories:

Let $F_{\mathsf{Sig}} : \mathsf{Sig} \to \mathsf{2Sig}$ be the functor which associates to any 1-signature Σ the empty family of equations, $F_{\mathsf{Sig}}(\Sigma) := (\Sigma, \emptyset)$. Call $U_{\mathsf{Sig}} : \mathsf{2Sig} \to \mathsf{Sig}$ the forgetful functor defined on objects as $U_{\mathsf{Sig}}(\Sigma, E) := \Sigma$.

▶ **Lemma 21** (`TwoSig_OneSig_is_right_adjoint`, `OneSig_TwoSig_fully_faithful`)**.** *We have* $F_{\mathsf{Sig}} \dashv U_{\mathsf{Sig}}$*. Furthermore,* U_{Sig} *is a coreflection.*

We are interested in specifying new languages by "gluing together" simpler ones. On the level of 2-signatures, this is done by taking the coproduct, or, more generally, the pushout of 2-signatures:

---

[7] As noticed by an anonymous referee, this definition of "modularity" does not seem related to the specific meaning it has in the rewriting community (see, for example, [14]).

▶ **Theorem 22** (`TwoSig_PushoutsSET`). *The category* $\mathsf{2Sig}$ *has pushouts.*

Coproducts are computed by taking the union of the equations and the coproducts of the underlying 1-signatures. Coequalizers are computed by keeping the equations of the codomain and taking the coequalizer of the underlying 1-signatures. Thus, by decomposing any colimit into coequalizers and coproducts, we have this more general result:

▶ **Proposition 23.** *The category* $\mathsf{2Sig}$ *is cocomplete and* U_{Sig} *preserves colimits.*

We now turn to our modularity result, which states that the initial model of a coproduct of two 2-signatures is the coproduct of the initial models of each 2-signature. More generally, the two languages can be amalgamated along a common "core language", by considering a pushout rather than a coproduct.

For a precise statement of that result, we define a "total category of models of 2-signatures":

▶ **Definition 24.** *The category* $\int_{(\Sigma,E)} \mathsf{Mon}^{(\Sigma,E)}$*, or* $\mathsf{2Mod}$ *for short, has, as objects, pairs* $((\Sigma,E),M)$ *of a 2-signature* (Σ,E) *and a model* M *of* (Σ,E)*.*

A morphism from $((\Sigma_1,E_1),M_1)$ *to* $((\Sigma_2,E_2),M_2)$ *is a pair* (m,f) *consisting of a morphism* $m:(\Sigma_1,E_1)\to(\Sigma_2,E_2)$ *of 2-signatures and a morphism* $f:M_1\to m^*M_2$ *of* (Σ_1,E_1)*-models (or, equivalently, of* Σ_1*-models).*

This category of models of 2-signatures contains the models of 1-signatures as a coreflective subcategory. Let $F_{\mathsf{Mod}}:\mathsf{Mod}\to\mathsf{2Mod}$ be the functor which associates to any 1-model (Σ,M) the empty family of equations, $F_{\mathsf{Mod}}(\Sigma,M):=(F_{\mathsf{Sig}}(\Sigma),M)$. Conversely, the forgetful functor $U_{\mathsf{Mod}}:\mathsf{2Mod}\to\mathsf{Mod}$ maps $((\Sigma,E),M)$ to (Σ,M).

▶ **Lemma 25** (`TwoMod_OneMod_is_right_adjoint`, `OneMod_TwoMod_fully_faithful`). *We have* $F_{\mathsf{Mod}}\dashv U_{\mathsf{Mod}}$*. Furthermore,* U_{Mod} *is a coreflection.*

The modularity result is a consequence of the following technical result:

▶ **Proposition 26** (`TwoMod_cleaving`). *The forgetful functor* $2\pi:\mathsf{2Mod}\to\mathsf{2Sig}$ *is a Grothendieck fibration.*

The *modularity result* below is analogous to the modularity result for 1-signatures [2, Theorem 32]:

▶ **Theorem 27** (Modularity for 2-signatures, `pushout_in_big_rep`). *Suppose we have a pushout diagram of effective 2-signatures, as on the left below. This pushout gives rise to a commutative square of morphisms of models in* $\mathsf{2Mod}$ *as on the right below, where we only write the second components, omitting the (morphisms of) signatures. This square is a pushout square.*

$$
\begin{tikzcd}
\Upsilon_0 \arrow[r] \arrow[d] & \Upsilon_1 \arrow[d] \\
\Upsilon_2 \arrow[r] & \Upsilon \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd}
\qquad
\begin{tikzcd}
\widehat{\Upsilon}_0 \arrow[r] \arrow[d] & \widehat{\Upsilon}_1 \arrow[d] \\
\widehat{\Upsilon}_2 \arrow[r] & \widehat{\Upsilon} \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd}
$$

Intuitively, the 2-signatures Υ_1 and Υ_2 specify two extensions of the 2-signature Υ_0, and Υ is the smallest extension containing both these extensions. By Theorem 27 the initial model of Υ is the "smallest model containing both the languages generated by Υ_1 and Υ_2".

4.4 Initial Semantics for 2-Signatures

We now turn to the problem of constructing the initial model of a 2-signature (Σ, E). More specifically, we identify sufficient conditions for (Σ, E) to admit an initial object $\widehat{(\Sigma, E)}$ in the category of models. Our approach is very straightforward: we seek to construct $\widehat{(\Sigma, E)}$ by applying a suitable quotient construction to the initial object $\hat{\Sigma}$ of Mon^{Σ}.

This leads immediately to our first requirement on (Σ, E), which is that Σ must be an effective 1-signature. (For instance, we can assume that Σ is an algebraic 1-signature, see Theorem 2.) This is a very natural hypothesis, since in the case where E is the empty family of Σ-equations, it is obviously a necessary and sufficient condition.

Some Σ-equations are never satisfied. In that case, the category $\mathsf{Mon}^{(\Sigma,E)}$ is empty. For example, given any 1-signature Σ, consider the Σ-equation $\mathsf{inl}, \mathsf{inr} : \Theta \rightrightarrows \Theta + \Theta$ given by the left and right inclusion. This is obviously an unsatisfiable Σ-equation. We have to find suitable hypotheses to rule out such unsatisfiable Σ-equations. This motivates the notion of *elementary* equations.

▶ **Definition 28.** *Given a 1-signature Σ, a Σ-module S is* ***nice*** *if S sends pointwise epimorphic Σ-model morphisms to pointwise epimorphic module morphisms.*

▶ **Definition 29** (`elementary_equation`)**.** *Given a 1-signature Σ, an* ***elementary Σ-equation*** *is a Σ-equation such that*

- *the target is a finite derivative of the tautological 2-signature Θ, i.e., of the form $\Theta^{(n)}$ for some $n \in \mathbb{N}$, and*
- *the source is a nice Σ-module.*

▶ **Example 30** (`BindingSigAreEpiSig`)**.** Any algebraic 1-signature is nice [2, Example 45]. Thus, any Σ-equation between an algebraic 1-signature and $\Theta^{(n)}$, for some natural number n, is elementary.

▶ **Definition 31.** *A 2-signature (Σ, E) is said* ***algebraic*** *if Σ is algebraic and E is a family of elementary equations.*

▶ **Theorem 32** (`elementary_equations_on_alg_preserve_initiality`)**.** *Any algebraic 2-signature has an initial model.*

The proof of Theorem 32 is given in Section 5.

▶ **Example 33.** The 2-signature of lambda calculus modulo β and η equations given in Example 14 is algebraic. Its initial model is precisely the monad $\mathsf{LC}_{\beta\eta}$ of lambda calculus modulo $\beta\eta$ equations.

The instantiation of the formalized Theorem 32 to this 2-signature is done in `LCBetaEta`[8].

Let us mention finally that, using the axiom of choice, we can take a similar quotient on all the 1-models of Σ:

▶ **Proposition 34** (`ModEq_Mod_is_right_adjoint`, `ModEq_Mod_fully_faithful`)**.** *Here we assume the axiom of choice. The forgetful functor from the category $\mathsf{Mon}^{(\Sigma,E)}$ of 2-models of (Σ, E) to the category Mon^{Σ} of Σ-models has a left adjoint. Moreover, the left adjoint is a reflector.*

[8] An initiality result for this particular case was also previously discussed and proved formally in the Coq proof assistant in [17].

5 Proof of Theorem 32

Our main technical result on effectiveness is the following Lemma 35. In Theorem 32, we give a much simpler criterion that encompasses all the examples we give.

The main technical result is encapsulated in the following lemma.

▶ **Lemma 35** (`elementary_equations_preserve_initiality`). *Let (Σ, E) be a 2-signature such that:*

1. *Σ sends epimorphic natural transformations to epimorphic natural transformations,*
2. *E is a family of elementary equations,*
3. *the initial 1-model of Σ exists,*
4. *the initial 1-model of Σ preserves epimorphisms,*
5. *the image by Σ of the initial 1-model of Σ preserves epimorphisms.*

Then, the category of 2-models of (Σ, E) has an initial object.

Before tackling the proof of Lemma 35, we discuss how to derive Theorem 32 from it, and we prove some auxiliary results.

We start with a lemma about preservation of epimorphisms:

▶ **Lemma 36** (`algebraic_model_Epi` and `BindingSig_on_model_isEpi`). *Let Σ be an algebraic 1-signature. Then $\hat{\Sigma}$ and $\Sigma(\hat{\Sigma})$ preserve epimorphisms.*

Now we have everything we need to prove Theorem 32:

Proof of Theorem 32. The “epimorphism” hypotheses of Lemma 35 are used to transfer structure from the initial model $\hat{\Sigma}$ of the 1-signature Σ onto a suitable quotient. There are different ways to prove these hypotheses:

- The axiom of choice implies conditions 4 and 5 since, in this case, any epimorphism in Set is split and thus preserved by any functor.
- Condition 5 is a consequence of condition 4 if Σ sends monads preserving epimorphisms to modules preserving epimorphisms.
- If Σ is algebraic, then conditions 1, 3, 4 and 5 are satisfied, cf. Example 30 and Lemma 36.

From the remarks above, we derive the simpler and weaker statement of Theorem 32 that covers all our examples, which are algebraic. ◀

The rest of this section is dedicated to the proof of the main technical result, Lemma 35. The reader inclined to do so may safely skip this part, and rely on the correctness of the machine-checked proof instead.

The proof of Lemma 35 uses some quotient constructions that we present now:

▶ **Proposition 37** (`u_monad_def`). *Given a monad R preserving epimorphisms and a collection of monad morphisms $(f_i : R \to S_i)_{i \in I}$, there exists a* quotient *monad $R/(f_i)$ together with a* projection *$p^R \colon R \longrightarrow R/(f_i)$, which is a morphism of monads such that each f_i factors through p.*

Proof. The set $R/(f_i)(X)$ is computed as the quotient of $R(X)$ with respect to the relation $x \sim y$ if and only if $f_i(x) = f_i(y)$ for each $i \in I$. This is a straightforward adaptation of Lemma 47 of [2]. ◀

Note that the epimorphism preservation is implied by the axiom of choice, but can be proven for the monad underlying the initial model $\hat{\Sigma}$ of an algebraic 1-signature Σ even without resorting to the axiom of choice.

The above construction can be transported on Σ-models:

▶ **Proposition 38** (`u_rep_def`). *Let Σ be a 1-signature sending epimorphic natural transformations to epimorphic natural transformations, and let R be a Σ-model such that R and $\Sigma(R)$ preserve epimorphisms. Let $(f_i : R \to S_i)_{i\in I}$ be a collection of Σ-model morphisms. Then the monad $R/(f_i)$ has a natural structure of Σ-model and the quotient map $p^R\colon R \longrightarrow R/(f_i)$ is a morphism of Σ-models. Any morphism f_i factors through p^R in the category of Σ-models.*

The fact that R and $\Sigma(R)$ preserve epimorphisms is implied by the axiom of choice. The proof follows the same line of reasoning as the proof of Proposition 37.

Now we are ready to prove the main technical lemma:

Proof of Lemma 35. Let Σ be an effective 1-signature, and let E be a set of elementary Σ-equations. The plan of the proof is as follows:

1. Start with the initial model $(\hat{\Sigma}, \sigma)$, with $\sigma : \Sigma(\hat{\Sigma}) \to \hat{\Sigma}$.
2. Construct the quotient model $\hat{\Sigma}/(f_i)$ according to Proposition 38 where $(f_i : \hat{\Sigma} \to S_i)_i$ is the collection of all initial Σ-morphisms from $\hat{\Sigma}$ to any Σ-model satisfying the equations. We denote by $\sigma/(f_i) : \Sigma(\hat{\Sigma}/(f_i)) \to \hat{\Sigma}/(f_i)$ the action of the quotient model.
3. Given a model M of the 2-signature (Σ, E), we obtain a morphism $i_M : \hat{\Sigma}/(f_i) \to M$ from Proposition 38. Uniqueness of i_M is shown using epimorphicity of the projection $p : \hat{\Sigma} \to \hat{\Sigma}/(f_i)$. For this, it suffices to show uniqueness of the composition $i_M \circ p : \hat{\Sigma} \to M$ in the category of 1-models of Σ, which follows from initiality of $\hat{\Sigma}$.
4. The verification that $\big(\hat{\Sigma}/(f_i), \sigma/(f_i)\big)$ satisfies the equations is given below. Actually, it follows the same line of reasoning as in the proof of Proposition 37 that $\hat{\Sigma}/(f_i)$ satisfies the monad equations.

Let $e = (e_1, e_2) : U \to \Theta^{(n)}$ be an elementary equation of E. We want to prove that the two arrows

$$e_{1,\hat{\Sigma}/(f_i)}, e_{2,\hat{\Sigma}/(f_i)}\colon U(\hat{\Sigma}/(f_i)) \longrightarrow (\hat{\Sigma}/(f_i))^{(n)}$$

are equal. As p is an epimorphic natural transformation, $U(p)$ also is by definition of an elementary equation. It is thus sufficient to prove that

$$e_{1,\hat{\Sigma}/(f_i)} \circ U(p) = e_{2,\hat{\Sigma}/(f_i)} \circ U(p) \ ,$$

which, by naturality of e_1 and e_2, is equivalent to $p^{(n)} \circ e_{1,\hat{\Sigma}} = p^{(n)} \circ e_{2,\hat{\Sigma}}$.

Let x be an element of $U(\hat{\Sigma})$ and let us show that $p^{(n)}(e_{1,\hat{\Sigma}}(x)) = p^{(n)}(e_{2,\hat{\Sigma}}(x))$. By definition of $\hat{\Sigma}/(f_i)$ as a pointwise quotient (see Proposition 37), it is enough to show that for any j, the equality $f_j^{(n)}(e_{1,\hat{\Sigma}}(x)) = f_j^{(n)}(e_{2,\hat{\Sigma}}(x))$ is satisfied. Now, by naturality of e_1 and e_2, this equation is equivalent to $e_{1,S_j}(U(f_j)(x))) = e_{2,S_j}(U(f_j)(x)))$ which is true since S_j satisfies the equation $e_1 = e_2$. ◀

6 Examples of algebraic 2-signatures

We already illustrated our theory by looking at the paradigmatic case of lambda calculus modulo β- and η-equations (Examples 11 and 33). This section collects further examples of application of our results.

In our framework, complex signatures can be built out of simpler ones by taking their coproducts. Note that the class of algebraic 2-signatures encompasses the algebraic 1-signatures and is closed under arbitrary coproducts: the prototypical examples of algebraic 2-signatures given in this section can be combined with any other algebraic 2-signature, yielding an effective 2-signature thanks to Theorem 32.

6.1 Monoids

We begin with an example of monad for a first-order syntax with equations. Given a set X, we denote by $M(X)$ the free monoid built over X. This is a classical example of monad over the category of (small) sets. The monoid structure gives us, for each set X, two maps $m_X : M(X) \times M(X) \longrightarrow M(X)$ and $e_X : 1 \longrightarrow M(X)$ given by the product and the identity respectively. It can be easily verified that $m : M^2 \longrightarrow M$ and $e : 1 \longrightarrow M$ are M-module morphisms. In other words, $(M, \rho) = (M, [m, e])$ is a model of the 1-signature $\Sigma = \Theta \times \Theta + 1$.

We break the tautological morphism of Σ-modules (cf. Example 6) into constituent pieces, defining $\mathsf{m} := \tau \circ \mathsf{inl} : \Theta \times \Theta \to \Theta$ and $\mathsf{e} := \tau \circ \mathsf{inr} : 1 \to \Theta$.

Over the 1-signature Σ we specify equations postulating *associativity* and *left and right unitality* as follows:

$$\begin{array}{lll} \Theta^3 \xrightarrow{\Theta \times \mathsf{m}} \Theta^2 \xrightarrow{\mathsf{m}} \Theta & \quad \Theta \xrightarrow{\mathsf{e} \times \Theta} \Theta^2 \xrightarrow{\mathsf{m}} \Theta & \quad \Theta \xrightarrow{\Theta \times \mathsf{e}} \Theta^2 \xrightarrow{\mathsf{m}} \Theta \\ \Theta^3 \xrightarrow[\mathsf{m} \times \Theta]{} \Theta^2 \xrightarrow[\mathsf{m}]{} \Theta & \quad \Theta \xrightarrow[1]{} \Theta & \quad \Theta \xrightarrow[1]{} \Theta \end{array}$$

and we denote by E the family consisting of these three Σ-equations. All are elementary since their codomain is Θ, and their domain a product of Θs.

One checks easily that $(M, [m, e])$ is the initial model of (Σ, E).

Several other classical (equational) algebraic theories, such as groups and rings, can be treated similarly, see Section 6.3 below. However, at the present state we cannot model theories with partial construction (e.g., fields).

6.2 Colimits of algebraic 2-signatures

In this section, we argue that our framework encompasses any colimit of algebraic 2-signatures.

Actually, the class of algebraic 2-signatures is not stable under colimits, as this is not even the case for algebraic 1-signatures. However, we can weaken this statement as follows:

▶ **Proposition 39.** *Given any colimit of algebraic 2-signatures, there is an algebraic 2-signature yielding an isomorphic category of models.*

Proof. As the class of algebraic 2-signatures is closed under arbitrary coproducts, using the decomposition of colimits into coproducts and coequalizers, any colimit Ξ of algebraic 2-signatures can be expressed as a coequalizer of two morphisms f, g between some algebraic 2-signatures (Σ_1, E_1) and (Σ_2, E_2),

$$(\Sigma_1, E_1) \overset{f}{\underset{g}{\rightrightarrows}} (\Sigma_2, E_2) \xrightarrow{p} \Xi = (\Sigma_3, E_2)\ .$$

where Σ_3 is the coequalizer of the 1-signatures morphisms f and g. Note that the set of equations of Ξ is E_2, by definition of the coequalizer in the category of 2-signatures. Now, consider the algebraic 2-signature $\Xi' = (\Sigma_2, E_2 + (2))$ consisting of the 1-signature Σ_2 and the equations of E_2 plus the following elementary equation (see Example 30):

$$\begin{array}{l} \Sigma_1 \xrightarrow{f} \Sigma_2 \xrightarrow{\tau^{\Sigma_2}} \Theta \\ \Sigma_1 \xrightarrow[g]{} \Sigma_2 \xrightarrow[\tau^{\Sigma_2}]{} \Theta \end{array} \tag{2}$$

We show that Mon^{Ξ} and $\mathsf{Mon}^{\Xi'}$ are isomorphic. A model of Ξ' is a monad R together with an R-module morphism $r : \Sigma_2(R) \to R$ such that $r \circ f_R = r \circ g_R$ and that the equations of E_2 are satisfied. By universal property of the coequalizer, this is exactly the same as giving an R-module morphism $\Sigma_3(R) \to R$ satisfying the equations of E_2, i.e., giving R an action of $\Xi = (\Sigma_3, E_2)$.

It is straightforward to check that this correspondence yields an isomorphism between the category of models of Ξ and the category of models of Ξ'. ◀

This proposition, together with the following corollary, allow us to recover all the examples presented in [2], as colimits of algebraic 1-signatures: syntactic commutative binary operator, maximum operator, application à la differential lambda calculus, syntactic closure operator, integrated substitution operator, coherent fixpoint operator.

▶ **Corollary 40.** *If F is a finitary endofunctor on* Set*, then there is an algebraic 2-signature whose category of models is isomorphic to the category of 1-models of the 1-signature $F \cdot \Theta$.*

Proof. It is enough to prove that $F \cdot \Theta$ is a colimit of algebraic 1-signatures.

As F is finitary, it is isomorphic to the coend $\int^{n\in\mathbb{N}} F(n) \times \_^n$ where $\mathbb{N}$ is the full subcategory of Set of finite ordinals (see, e.g., [26, Example 3.19]). As colimits are computed pointwise, the 1-signature $F \cdot \Theta$ is the coend $\int^{n\in\mathbb{N}} F(n) \times \Theta^n$, and as such, it is a colimit of algebraic 2-signatures. ◀

However, we do not know whether we can recover our theorem [2, Theorem 35] stating that any presentable 1-signature is effective.

6.3 Algebraic theories

From the categorical point of view, several fundamental algebraic structures in mathematics can be conveniently and elegantly described using finitary monads. For instance, the category of monoids can be seen as the category of Eilenberg–Moore algebras of the monad of lists. Other important examples, like groups and rings, can be treated analogously. A classical reference on the subject is the work of Manes, where such monads are significantly called *finitary algebraic theories* [22, Definition 3.17].

We want to show that such "algebraic theories" fit in our framework, in the sense that they can be incorporated into an algebraic 2-signature, with the effect of enriching the initial model with the operations of the algebraic theory, subject to the axioms of the algebraic theory.

For a finitary monad T, Corollary 40 says how to encode the 1-signature $T \cdot \Theta$ as an algebraic 2-signature (Σ_T, E_T). Models are monads R together with an R-linear morphism $r : T \cdot R \to R$.

Now, for any model (R, m) of $T \cdot \Theta$, we would like to enforce the usual T-algebra equations on the action m. This is done thanks to the following equations, where τ denotes the tautological morphism of $T \cdot \Theta$-modules:

$$\begin{array}{ll} \Theta \xrightarrow{\eta_T \cdot \Theta} T \cdot \Theta \xrightarrow{\tau} \Theta & \qquad T \cdot T \cdot \Theta \xrightarrow{\mu_T \cdot \Theta} T \cdot \Theta \xrightarrow{\tau} \Theta \\ \Theta \xrightarrow[1]{\hspace{6em}} \Theta & \qquad T \cdot T \cdot \Theta \xrightarrow[T\tau]{} T \cdot \Theta \xrightarrow[\tau]{} \Theta \end{array} \tag{3}$$

The first equation is clearly elementary. The second one is elementary thanks to the following lemma:

▶ **Lemma 41.** *Let F be a finitary endofunctor on* Set*. Then F preserves epimorphisms.*

Proof. An anonymous referee remarked that this is a consequence of the axiom of choice, because then any epimorphism in the category of Set is split, and thus preserved by any functor. Here we provide an alternative proof which does not rely on the axiom of choice. (However, it may require the excluded middle, depending on the chosen definition of finitary functor.)

As F is finitary, it is isomorphic to the coend $\int^{n\in\mathbb{N}} F(n) \times \_^n$ [26, Example 3.19]. By decomposing it as a coequalizer of coproducts, we get an epimorphism $\alpha : \coprod_{n\in\mathbb{N}} F(n) \times \_^n \to F$. Now, let $f : X \to Y$ be a surjective function between two sets. We show that $F(f)$ is epimorphic. By naturality, the following diagram commutes:

$$\begin{tikzcd}
\coprod_{n\in\mathbb{N}} F(n) \times X^n \arrow[r, "F(n)\times f^n"] \arrow[d, "\alpha_X"'] & \coprod_{n\in\mathbb{N}} F(n) \times Y^n \arrow[d, "\alpha_Y"] \\
F(X) \arrow[r, "F(f)"'] & F(Y)
\end{tikzcd}$$

The composition along the top-right is epimorphic by composition of epimorphisms. Thus, the bottom left is also epimorphic, and so is $F(f)$ as the last morphism of this composition. ◀

In conclusion, we have exhibited the algebraic 2-signature (Σ_T, E'_T), where E'_T extends the family E_T with the two elementary equations of Diagram 3. This signature allows to enrich any other algebraic 2-signature with the operations of the algebraic theory T, subject to the relevant equations.

6.4 Fixpoint operator

Here, we show the algebraic 2-signature corresponding to a fixpoint operator. In [2, Section 8.4] we studied fixpoint operators in the context of 1-signatures. In that setting, we treated a *syntactic* fixpoint operator called *coherent* fixpoint operator, somehow reminiscent of mutual letrec. We were able to impose many natural equations to this operator but we were not able to enforce the fixpoint equation. In this section, we show how a fixpoint operator can be fully specified by an algebraic 2-signature. We restrict our discussion to the unary case; the coherent family of multi-ary fixpoint operators presented in [2, Section 8.4], now including the fixpoint equations, can also be specified, in an analogous way, via an algebraic 2-signature.

Let us start by recalling the following

▶ **Definition 42.** *A **unary fixpoint operator for a monad R** [2, Definition 40] is a module morphism f from R' to R that makes the following diagram commute, where σ is the substitution morphism defined as the uncurrying (see Definition 10) of the identity morphism on Θ':*

$$\begin{tikzcd}
R' \arrow[rr, "(id_{R'}, f)"] \arrow[dr, "f"'] & & R' \times R \arrow[dl, "\sigma_R"] \\
& R &
\end{tikzcd}$$

In order to rephrase this definition, we introduce the obviously algebraic 2-signature Υ_{fix} consisting of the 1-signature $\Sigma_{\mathsf{fix}} = \Theta'$ and the family E_{fix} consisting of the single following Σ_{fix}-equation:

$$e_{\mathsf{fix}} : \begin{array}{l} \Theta' \xrightarrow{\langle 1,\tau\rangle} \Theta' \times \Theta \xrightarrow{\sigma} \Theta \\ \Theta' \xrightarrow[\tau]{\qquad\qquad\qquad} \Theta \end{array} \tag{4}$$

This allows us to rephrase the previous definition as follows: a unary fixpoint operator for a monad R is just an action of the 2-signature Υ_{fix} in R.

The name "fixpoint operator" is motivated by the following proposition:

▶ **Proposition 43** ([2, Proposition 41]). *Fixpoint combinators are in one-to-one correspondence with actions of Υ_{fix} in the monad $\mathsf{LC}_{\beta\eta}$ of the lambda calculus modulo β- and η-equality.*

Recall that fixpoint combinators are lambda terms Y satisfying, for any (possibly open) term t, the equation

$$\mathsf{app}(t, \mathsf{app}(Y, t)) = \mathsf{app}(Y, t) \ .$$

Explicitly, such a combinator Y induces a fixpoint operator $\hat{Y} : \mathsf{LC}'_{\beta\eta} \to \mathsf{LC}_{\beta\eta}$ which associates, to any term t depending on an additional variable $*$, the term $\hat{Y}(t) := \mathsf{app}(Y, \mathsf{abs}\ t)$.

7 Recursion

In this section, we explain how a recursion principle can be derived from our initiality result, and give an example of a morphism – a *translation* – between monads defined via the recursion principle.

7.1 Principle of recursion

In our context, the recursion principle is a recipe for constructing a morphism from the monad underlying the initial model of a 2-signature to an arbitrary monad.

▶ **Proposition 44** (Recursion principle). *Let S be the monad underlying the initial model of the 2-signature Υ. To any action a of Υ in T is associated a monad morphism $\hat{a} : S \to T$.*

Proof. The action a defines a 2-model M of Υ, and $\hat{a}$ is the monad morphism underlying the initial morphism to M. ◀

Hence the recipe consists in the following two steps:

1. give T an action of the 1-signature Σ;
2. check that all the equations in E are satisfied for the induced model.

In the next section, we illustrate this principle.

7.2 Translation of lambda calculus with fixpoint to lambda calculus

In this section, we consider the 2-signature $\Upsilon_{\mathsf{LC}_{\beta\eta,\mathsf{fix}}} := \Upsilon_{\mathsf{LC}_{\beta\eta}} + \Upsilon_{\mathsf{fix}}$ where the two components have been introduced above (see Example 18 and Section 6.4).

As a coproduct of algebraic 2-signatures, $\Upsilon_{\mathsf{LC}_{\beta\eta,\mathsf{fix}}}$ is itself algebraic, and thus the initial model exists. The underlying monad $\mathsf{LC}_{\beta\eta,\mathsf{fix}}$ of the initial model can be understood as the monad of lambda calculus modulo β and η enriched with an *explicit* fixpoint operator $\mathsf{fix} : \mathsf{LC}'_{\beta\eta,\mathsf{fix}} \longrightarrow \mathsf{LC}_{\beta\eta,\mathsf{fix}}$. Now we build by recursion a monad morphism from this monad to the "bare" monad $\mathsf{LC}_{\beta\eta}$ of lambda calculus modulo β and η.

As explained in Section 7.1, we need to define an action of $\Upsilon_{\mathsf{LC}_{\beta\eta,\mathsf{fix}}}$ in $\mathsf{LC}_{\beta\eta}$, that is to say an action of $\Upsilon_{\mathsf{LC}_{\beta\eta}}$ plus an action of Υ_{fix}. For the action of $\Upsilon_{\mathsf{LC}_{\beta\eta}}$, we take the one yielding the initial model.

Now, in order to find an action of Υ_{fix} in $\mathsf{LC}_{\beta\eta}$, we choose a fixpoint combinator Y (say the one of Curry) and take the action $\hat{Y}$ as defined at the end of Section 6.4.

In more concrete terms, our translation is a kind of compilation which replaces each occurrence of the explicit fixpoint operator $\mathsf{fix}(t)$ with $\mathsf{app}(Y, \mathsf{abs}\ t)$.

References

1 Benedikt Ahrens. Modules over relative monads for syntax and semantics. *Mathematical Structures in Computer Science*, 26:3–37, 2016. doi:10.1017/S0960129514000103.

2 Benedikt Ahrens, André Hirschowitz, Ambroise Lafont, and Marco Maggesi. High-Level Signatures and Initial Semantics. In Dan Ghica and Achim Jung, editors, *27th EACSL Annual Conference on Computer Science Logic (CSL 2018)*, volume 119 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 4:1–4:22, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.CSL.2018.4.

3 Michael Barr. Coequalizers and free triples. *Mathematische Zeitschrift*, 116(4):307–322, December 1970. doi:10.1007/BF01111838.

4 Ranald Clouston. Binding in Nominal Equational Logic. *Electr. Notes Theor. Comput. Sci.*, 265:259–276, 2010. doi:10.1016/j.entcs.2010.08.016.

5 Maribel Fernández and Murdoch J. Gabbay. Closed nominal rewriting and efficiently computable nominal algebra equality. In Karl Crary and Marino Miculan, editors, *Proceedings 5th International Workshop on Logical Frameworks and Meta-languages: Theory and Practice, LFMTP 2010, Edinburgh, UK, 14th July 2010.*, volume 34 of *EPTCS*, pages 37–51, 2010. doi:10.4204/EPTCS.34.5.

6 Marcelo P. Fiore and Makoto Hamana. Multiversal Polymorphic Algebraic Theories: Syntax, Semantics, Translations, and Equational Logic. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*, pages 520–529. IEEE Computer Society, 2013. doi:10.1109/LICS.2013.59.

7 Marcelo P. Fiore and Chung-Kil Hur. On the construction of free algebras for equational systems. *Theor. Comput. Sci.*, 410(18):1704–1729, 2009. doi:10.1016/j.tcs.2008.12.052.

8 Marcelo P. Fiore and Chung-Kil Hur. Second-Order Equational Logic (Extended Abstract). In Anuj Dawar and Helmut Veith, editors, *CSL*, volume 6247 of *Lecture Notes in Computer Science*, pages 320–335. Springer, 2010. doi:10.1007/978-3-642-15205-4_26.

9 Marcelo P. Fiore and Ola Mahmoud. Second-Order Algebraic Theories (Extended Abstract). In Petr Hlinený and Antonín Kucera, editors, *MFCS*, volume 6281 of *Lecture Notes in Computer Science*, pages 368–380. Springer, 2010. doi:10.1007/978-3-642-15155-2_33.

10 Marcelo P. Fiore, Gordon D. Plotkin, and Daniele Turi. Abstract Syntax and Variable Binding. In *14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy, July 2-5, 1999*, pages 193–202, 1999. doi:10.1109/LICS.1999.782615.

11 Marcelo P. Fiore and Philip Saville. List Objects with Algebraic Structure. In Dale Miller, editor, *2nd International Conference on Formal Structures for Computation and Deduction (FSCD 2017)*, volume 84 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:18, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.FSCD.2017.16.

12 Murdoch J. Gabbay and Andrew M. Pitts. A New Approach to Abstract Syntax Involving Binders. In *14th Annual Symposium on Logic in Computer Science*, pages 214–224, Washington, DC, USA, 1999. IEEE Computer Society Press. doi:10.1109/LICS.1999.782617.

13 Neil Ghani, Tarmo Uustalu, and Makoto Hamana. Explicit substitutions and higher-order syntax. *Higher-Order and Symbolic Computation*, 19(2-3):263–282, 2006. doi:10.1007/s10990-006-8748-4.

14 Bernhard Gramlich. Modularity in term rewriting revisited. *Theoretical Computer Science*, 464:3–19, 2012. New Directions in Rewriting (Honoring the 60th Birthday of Yoshihito Toyama). doi:10.1016/j.tcs.2012.09.008.

15 Makoto Hamana. Term Rewriting with Variable Binding: An Initial Algebra Approach. In *Proceedings of the 5th ACM SIGPLAN International Conference on Principles and Practice of Declaritive Programming*, PPDP '03, pages 148–159, New York, NY, USA, 2003. ACM. doi:10.1145/888251.888266.

16 André Hirschowitz and Marco Maggesi. Modules over Monads and Linearity. In D. Leivant and R. J. G. B. de Queiroz, editors, *WoLLIC*, volume 4576 of *Lecture Notes in Computer Science*, pages 218–237. Springer, 2007. `doi:10.1007/978-3-540-73445-1_16`.

17 André Hirschowitz and Marco Maggesi. Modules over monads and initial semantics. *Information and Computation*, 208(5):545–564, May 2010. Special Issue: 14th Workshop on Logic, Language, Information and Computation (WoLLIC 2007). `doi:10.1016/j.ic.2009.07.003`.

18 Martin Hofmann. Semantical Analysis of Higher-Order Abstract Syntax. In *14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy, July 2-5, 1999*, pages 204–213. IEEE Computer Society, 1999. `doi:10.1109/LICS.1999.782616`.

19 G. Maxwell Kelly and A. John Power. Adjunctions whose counits are coequalizers, and presentations of finitary enriched monads. *Journal of Pure and Applied Algebra*, 89(1):163–179, 1993. `doi:10.1016/0022-4049(93)90092-8`.

20 Alexander Kurz and Daniela Petrisan. On universal algebra over nominal sets. *Mathematical Structures in Computer Science*, 20(2):285–318, 2010. `doi:10.1017/S0960129509990399`.

21 Christoph Lüth and Neil Ghani. Monads and Modular Term Rewriting. In Eugenio Moggi and Giuseppe Rosolini, editors, *Category Theory and Computer Science, 7th International Conference, CTCS '97*, volume 1290 of *Lecture Notes in Computer Science*, pages 69–86. Springer, 1997. `doi:10.1007/BFb0026982`.

22 Ernest Manes. *Algebraic Theories*, volume 26 of *Graduate Texts in Mathematics*. Springer, 1976.

23 The Coq development team. *The Coq Proof Assistant, version 8.9*, 2019. Version 8.9. URL: `http://coq.inria.fr`.

24 A. John Power. Abstract Syntax: Substitution and Binders: Invited Address. *Electr. Notes Theor. Comput. Sci.*, 173:3–16, 2007. `doi:10.1016/j.entcs.2007.02.024`.

25 Sam Staton. An Algebraic Presentation of Predicate Logic (Extended Abstract). In Frank Pfenning, editor, *Foundations of Software Science and Computation Structures - 16th International Conference, FOSSACS 2013*, volume 7794 of *Lecture Notes in Computer Science*, pages 401–417. Springer, 2013. `doi:10.1007/978-3-642-37075-5_26`.

26 Jiří Velebil and Alexander Kurz. Equational presentations of functors and monads. *Mathematical Structures in Computer Science*, 21(2):363–381, 2011. `doi:10.1017/S0960129510000575`.

27 Vladimir Voevodsky, Benedikt Ahrens, Daniel Grayson, et al. UniMath — a computer-checked library of univalent mathematics. Available at `https://github.com/UniMath/UniMath`.