A parameterization process, functorially

César Dominguez, Dominique Duval

A parameterization process, functorially

César Domínguez [*] Dominique Duval [†]

August 31., 2009

Abstract. The parameterization process used in the symbolic computation systems Kenzo and EAT is studied here as a general construction in a categorical framework. This parameterization process starts from a given specification and builds a parameterized specification by adding a parameter as a new variable to some operations. Given a model of the parameterized specification, each interpretation of the parameter, called an argument, provides a model of the given specification. Moreover, under some relevant terminality assumption, this correspondence between the arguments and the models of the given specification is a bijection. It is proved in this paper that the parameterization process is provided by a functor and the subsequent parameter passing process by a natural transformation. Various categorical notions are used, mainly adjoint functors, pushouts and lax colimits.

1 Introduction

Kenzo [8] and its predecessor EAT [16] are software systems developed by F. Sergeraert. They are devoted to Symbolic Computation in Algebraic Topology. In particular, they carry out calculations of homology groups of complex topological spaces, namely iterated loop spaces. By means of EAT and Kenzo, some homology groups that had never been obtained with any other method, neither theoretical nor automatic, have been computed. In view of the obtained results, some years ago, the first author of this paper began the formal study of the programs, in order to reach a good understanding on the internal calculation processes of these software systems. In particular, our study of the data types used in EAT and Kenzo [11, 6, 7] shows that there are two different layers of data structures in the systems. In the first layer, one finds the usual abstract data types, like the type of integers. In the second layer, one deals with algebraic structures, like the structure of groups, which are implemented thanks to the abstract data types belonging to the first layer. In addition, we realized that in a system such as EAT, we do not simply implement one group, but more generally *parameterized* families of groups. In [11] an operation is defined, which is called the *imp* construction because of its role in the implementation process in the system EAT. Starting from a specification Σ in which some operations are labelled as "pure" [7], the *imp* construction builds a new specification Σ_A with a distinguished sort A which is added to the domain of each non-pure operation. It follows that each implementation of Σ_A defines a family of implementations of Σ depending on the choice of a value in the interpretation of A. Besides, working with the *imp* construction in [11] we were able to prove that the implementations of EAT algebraic structures are as general as possible, in the sense that they are ingredients of terminal objects in certain categories of models; this result is called the *exact parameterization property.* Later on, led by this characterization of EAT algebraic structures, in [11] we reinterpreted our results in terms of object-oriented technologies like hidden algebras [9] or coalgebras [17].

This paper deals with generalization by parameterization in the sense of Kenzo and EAT, so that our *parameters* are symbolic constants of a given type, that will be replaced by *arguments* which are elements in

[*] Departamento de Matemáticas y Computación, Universidad de La Rioja, Edificio Vives, Luis de Ulloa s/n, E-26004 Logroño, La Rioja, Spain, cesar.dominguez@unirioja.es.

[†] Laboratoire Jean Kuntzmann, Université de Grenoble, 51 rue des mathématiques, BP 53, F-38041 Grenoble Cédex 9, France, Dominique.Duval@imag.fr.

a given set. The notion of parameterization in programming and specification languages bears several meanings, where the parameter may be a type or a specification. For instance, in object-oriented programming, parametric polymorphism is called generic programming, in C++ it is characterized by the use of template parameters to represent abstract data types. On the other hand, in algebraic specifications, a parameterized specification is defined as a morphism of specifications where the parameter is the source and the parameter passing is defined as a pushout [2].

The framework for this paper is provided by *equational logic*, considered from a categorical point of view. An equational theory, or simply a theory, is a category with chosen finite products. A model M of a theory Θ is a functor $M\colon \Theta \to \mathit{Set}$ which maps the chosen products to cartesian products. A theory Θ can be presented by a specification Σ, this means that Σ generates Θ. In this paper, we are not interested in specifications for themselves, but as presentations of theories. So, specifications are used mainly in the examples, and we feel free to modify a specification whenever needed as long as the presented theory is not changed.

The *parameterization process* studied in this paper is essentially the "*imp* construction" of [11]. Starting from a theory Θ it provides a *parameterized theory* Θ_A by adding a *type of parameters* A and by transforming each term $f\colon X \to Y$ in Θ into a parameterized term $f'\colon A \times X \to Y$ in Θ_A. Then clearly Θ_A generalizes Θ: the models of Θ can be identified to the models of Θ_A which interpret the type of parameters A as a singleton. There is another way to relate Θ and Θ_A, called the *parameter passing process*, which runs as follows. By adding to Θ_A a constant a (called the *parameter*) of type A we get a *theory with parameter* Θ_a, such that for each parameterized term $f'\colon A \times X \to Y$ in Θ_A there is a term $f'(a,-)\colon X \to Y$ in Θ_a. Then the *parameter passing morphism* $j\colon \Theta \to \Theta_a$ maps each term $f\colon X \to Y$ in Θ to $f'(a,-)\colon X \to Y$ in Θ_a. Given a model M_A of Θ_A an *argument* α is an element of the set $M_A(A)$, it provides a model $M_{A,\alpha}$ of Θ_a which extends M_A and satisfies $M_{A,\alpha}(a) = \alpha$. Thanks to the parameter passing morphism, the model $M_{A,\alpha}$ of Θ_a gives rise to a model M of Θ such that $M(f) = M_A(f')(\alpha,-)$ for each term f in Θ. Moreover, under some relevant terminality assumption on M_A, this correspondence between the arguments $\alpha \in M_A(A)$ and the models of Θ is a bijection: this is the *exact parameterization property.*

The parameterization process and its associated parameter passing process have been described for each given theory Θ, but in fact they have the property of preserving the theory structure, which can be stated precisely in a categorical framework: this is the aim of this paper. The parameterization process is defined as a *functor*: the construction of the parameterized theory Θ_A from the given theory Θ is a functor, which in addition is left adjoint to the construction of a coKleisli category. The parameter passing process is defined as a *natural transformation*, along the following lines. First, the construction of the theory with parameter Θ_a from the parameterized theory Θ_A is simply a pushout construction, such that the construction of Θ_a from Θ is a functor. Then, each parameter passing morphism $j : \Theta \to \Theta_A$ is defined from a lax colimit of theories, in such a way that the parameter passing morphisms are (essentially) the components of a natural transformation from the identity functor to this functor.

A first version of this approach can be found in [3], and a more abstract point of view, relying on *diagrammatic logic*, is presented in [5]. With respect to the previous papers like [11], we provide a new interpretation of the parameterization process and in addition an interpretation of the parameter passing process. Moreover, we take into account the fact that there is a pure part in the given theory, and we derive the exact parameterization property from a more general result which does not rely on the existence of a terminal model.

Equational theories are defined in section 2, then the parameterization process and the parameter passing process are studied in section 3. Various examples are presented. Most of the categorical notions used in this paper can be found in [14] or in [1]. We omit the size issues: for instance most colimits should be small. A *graph* is a directed multigraph, and in order to distinguish between various kinds of structures with an underlying graph, we speak about the *objects* and *morphisms* of a category, the *types* and *terms* of a theory or a specification and the *points* and *arrows* of a limit sketch.

2 Definitions

2.1 Equational theories and specifications

In this paper, equational logic is seen from a categorical point of view, as for instance in [15].

Definition 2.1 The category $\mathbf{T}_{eq}$ of *equational theories* is made of the categories with chosen finite products together with the functors which preserve the chosen finite products. In addition, $\mathbf{T}_{eq}$ can be seen as a 2-category with the natural transformations as 2-cells.

Equational theories are called simply *theories*. For instance, the theory *Set* is made of the category of sets with the cartesian products as chosen products.

Remark 2.2 The correspondence between equational theories in the universal algebra style (as in [13]) and equational theories in the categorical style (as defined here) can be found in [15]. Basically, the *sorts* and products of sorts become objects, still called types, the *operations* and *terms* become morphisms, still called terms (the *variables* correspond to projections, as in example 2.6) and the *equations* become equalities: for instance a commutative square $g_1 \circ f_1 = g_2 \circ f_2$ means that there is a term h such that $g_1 \circ f_1 = h$ and $g_2 \circ f_2 = h$. A more subtle point of view on equations is presented in [4].

Definition 2.3 A *(strict) model* M of a theory Θ is a morphism of theories $M\colon \Theta \to \mathit{Set}$ and a *morphism* $m\colon M \to M'$ *of models* of Θ is a natural transformation. This forms the category $\mathit{Mod}(\Theta)$ of models of Θ.

For every morphism of equational theories $\theta\colon \Theta_1 \to \Theta$, we denote by $\theta^*\colon \mathit{Mod}(\Theta) \to \mathit{Mod}(\Theta_1)$ the functor which maps each model M of Θ to the model $\theta^*(M) = M \circ \theta$ of Θ_1 and each morphism $m\colon M \to M'$ to $m \circ \theta$. In addition, for each model M_1 of Θ_1, the category of *models of* Θ *extending* M_1 is denoted $\mathit{Mod}(\Theta)|_{M_1}$, it is the subcategory of $\mathit{Mod}(\Theta)$ made of the models M such that $\theta^*(M) = M_1$ and the morphisms m such that $\theta^*(m) = id_{M_1}$. Whenever θ is surjective on types, the category $\mathit{Mod}(\Theta)|_{M_1}$ is discrete.

A theory Θ can be described by some presentation: a *presentation* of an equational theory Θ is an equational specification Σ which generates Θ; this is denoted $\Theta \dashv \Sigma$. Two specifications are called *equivalent* when they present the same theory. An equational specification can be defined either in the universal algebra style as a signature (made of sorts and operations) together with equational axioms, or equivalently, in a more categorical style, as a finite product sketch, see [12], [1]. The correspondence between the universal algebra and the categorical points of view runs as in remark 2.2.

Definition 2.4 The category $\mathbf{S}_{eq}$ of *equational specifications* is the category of finite product sketches. With (generalized) natural transformations as 2-cells, $\mathbf{S}_{eq}$ can be seen as a 2-category.

Equational specifications are called simply *specifications*. The category $\mathbf{T}_{eq}$ can be identified to a subcategory of $\mathbf{S}_{eq}$ (more precisely, to a reflective subcategory of $\mathbf{S}_{eq}$). When Σ is a presentation of Θ, a model of Θ is determined by its restriction to Σ, which is called a *model* of Σ, and in fact $\mathit{Mod}(\Theta)$ can be identified to the category $\mathit{Mod}(\Sigma)$ of models of Σ.

We will repeatedly use the fact that $\mathbf{T}_{eq}$ and $\mathbf{S}_{eq}$, as well as other categories of theories and of specifications, have colimits, and that left adjoint functors preserve colimits. In addition every specification is the colimit of a diagram of elementary specifications. The *elementary specifications* are the specifications respectively made of: a type, a term, an identity term, a composed term, a n-ary product and a n-ary tuple for all $n \geq 0$, or only for $n = 0$ and $n = 2$, as in figure 1. Let us consider a theory Θ presented by a specification Σ, then Σ is the colimit of a diagram Δ of elementary specifications, and Θ is the colimit of the diagram of theories generated by Δ.

	subscript `E`	$\Sigma_{\mathtt{E}}$
type (or sort)	`Type`	X
term (or operation)	`Term`	$X \xrightarrow{f} Y$
selection of identity	`Selid`	$X \xrightarrow{id_X} X$
composition	`Comp`	$$\begin{tikzcd} X \arrow[r, "f"] \arrow[rr, bend right, "="{above}, "g\circ f"'] & Y \arrow[r, "g"] & Z \end{tikzcd}$$
terminal type	`0-Prod`	1
collapsing	`0-Tuple`	$X \xrightarrow{\langle\,\rangle_X} 1$
binary product	`2-Prod`	$$\begin{tikzcd} X & \\ & X\times Y \arrow[ul, "p_X"'] \arrow[dl, "p_Y"] \\ Y & \end{tikzcd}$$
pairing	`2-Tuple`	$$\begin{tikzcd} & X \arrow[d, phantom, "="] & \\ Z \arrow[ur, "f"] \arrow[dr, "g"'] \arrow[rr, "\langle f,g\rangle" description] & {} & X\times Y \arrow[ul, "p_X"'] \arrow[dl, "p_Y"] \\ & Y \arrow[u, phantom, "="] & \end{tikzcd}$$

Figure 1: Elementary specifications

2.2 Examples

Example 2.5 Let us consider the theory $\Theta_{op,0}$ presented by two types X, Y, and the three following theories extending $\Theta_{op,0}$ (the subscript *op* stands for "operation", since Θ_{op} is presented by the elementary specification for terms or operations $\Sigma_{\mathtt{Term}}$). The unit type is denoted 1 and the projections are not given any name.

$$\Theta_{op,A} \dashv \begin{tikzcd} A & A\times X \arrow[l] \arrow[d] \arrow[dr, "f'"] & \\ & X & Y \end{tikzcd}$$

$$\Theta_{op,a} \dashv \begin{tikzcd} A & A\times X \arrow[l] \arrow[d] \arrow[dr, "f'"] & \\ 1 \arrow[u, "a"] & X & Y \end{tikzcd}$$

$$\Theta_{op} \dashv \begin{tikzcd} X \arrow[r, "f"] & Y \end{tikzcd}$$

These theories are related by various morphisms (all of them preserving $\Theta_{op,0}$): $\theta_{op,A}\colon \Theta_{op,A} \to \Theta_{op}$ maps A to 1 and $\theta_{op,a}\colon \Theta_{op,a} \to \Theta_{op}$ extends $\theta_{op,A}$ by mapping a to id_1, while $j_{op,A}\colon \Theta_{op,A} \to \Theta_{op,a}$ is the inclusion. In addition, here are two other presentations of the theory $\Theta_{op,a}$ (the projections are omitted and $1 \times X$ is identified to X):

$$\begin{tikzcd} A & A\times X \arrow[dr, "f'"] \arrow[d, phantom, "=" near end, shift left=4] & \\ 1 \arrow[u, "a"] & X \arrow[u, "a\times id_X"] \arrow[r, "f''"'] & Y \end{tikzcd}$$

$$\begin{tikzcd} A & A\times X \arrow[r, "f'"] \arrow[dr, phantom, "="] & Y \\ 1 \arrow[u, "a"] & X \arrow[u, "a\times id_X"] \arrow[r, "f''"'] & Y \arrow[u, "id_Y"'] \end{tikzcd}$$

It is clear from these presentations of $\Theta_{op,a}$ that there is a morphism $j_{op}\colon \Theta_{op} \to \Theta_{op,a}$ which maps f to f''. In addition, $\theta_{op,a} \circ j_{op,A} = \theta_{op,A}$ and there is a natural transformation $t_{op}\colon j_{op} \circ \theta_{op,A} \Rightarrow j_{op,A}$ defined by $(t_{op})_X = id_X$, $(t_{op})_Y = id_Y$ and $(t_{op})_A = a\colon 1 \to A$.

$$\begin{tikzcd}
\Theta_{op,A} \arrow[d, "\theta_{op,A}"'] \\
\Theta_{op}
\end{tikzcd}
\qquad
\begin{tikzcd}
\Theta_{op,A} \arrow[dr, "j_{op,A}"] \arrow[dd, "\theta_{op,A}"'] & \\
 & \Theta_{op,a} \arrow[dl, "\theta_{op,a}"] \arrow[l, phantom, "="] \\
\Theta_{op} &
\end{tikzcd}
\qquad
\begin{tikzcd}
\Theta_{op,A} \arrow[dr, "j_{op,A}"] \arrow[dd, "\theta_{op,A}"'] & \\
 & \Theta_{op,a} \arrow[l, phantom, "t_{op}\Uparrow"] \\
\Theta_{op} \arrow[ur, "j_{op}"'] &
\end{tikzcd}$$

Parameterization process (construction of $\Theta_{op,A}$ from Θ_{op}). The theory $\Theta_{op,A}$ is obtained from Θ_{op} by adding a type A, called the *type of parameters*, to the domain of the unique term in Θ_{op}. Then $\Theta_{op,A}$ can be seen as a *generalization* of Θ_{op}, since each model M of Θ_{op} can be identified to a model of $\Theta_{op,A}$ where $M(A)$ is a singleton.

Parameter passing process (construction of $\Theta_{op,a}$ from $\Theta_{op,A}$ and of a morphism from Θ_{op} to $\Theta_{op,a}$). The theory $\Theta_{op,a}$ is obtained from $\Theta_{op,A}$ by adding a constant term $a\colon 1 \to A$, called the *parameter*. A model M_a of $\Theta_{op,a}$ is made of a model M_A of $\Theta_{op,A}$ together with an element $\alpha = M_a(a) \in M_A(A)$, so that we can denote $M_a = (M_A, \alpha)$. Now, let M_A be some fixed model of $\Theta_{op,A}$, then the models M_a of $\Theta_{op,a}$ extending M_A correspond bijectively to the elements of $M_A(A)$ by $M_a \mapsto M_a(a)$, so that we get the *parameter adding* bijection (the category $\mathit{Mod}(\Theta_{op,a})|_{M_A}$ is discrete):

$$\mathit{Mod}(\Theta_{op,a})|_{M_A} \xrightarrow{\cong} M_A(A) \quad \text{by} \quad M_a = (M_A, \alpha) \mapsto M_a(a) = \alpha \ .$$

On the other hand, each model $M_a = (M_A, \alpha)$ of $\Theta_{op,a}$ gives rise to a model ${j_{op}}^*(M_a)$ of Θ_{op} such that ${j_{op}}^*(M_a)(X) = M_a(X) = M_A(X)$, ${j_{op}}^*(M_a)(Y) = M_a(Y) = M_A(Y)$ and ${j_{op}}^*(M_a)(f) = M_a(f'') = M_A(f')(\alpha, -)$. Now, let M_A be some fixed model of $\Theta_{op,A}$ and M_0 its restriction to $\Theta_{op,0}$, then for each model $M_a = (M_A, \alpha)$ of $\Theta_{op,a}$ extending M_A the model ${j_{op}}^*(M_a)$ of Θ_{op} extends M_0. This yields the *parameter passing* function (the categories $\mathit{Mod}(\Theta_{op,a})|_{M_A}$ and $\mathit{Mod}(\Theta_{op})|_{M_0}$ are discrete):

$$\mathit{Mod}(\Theta_{op,a})|_{M_A} \to \mathit{Mod}(\Theta_{op})|_{M_0} \quad \text{by} \quad M_a \mapsto {j_{op}}^*(M_a) \ .$$

Exact parameterization. Let M_0 be any fixed model of $\Theta_{op,0}$, it is made of two sets $\mathbb{X} = M_0(X)$ and $\mathbb{Y} = M_0(Y)$. Let M_A be the model of $\Theta_{op,A}$ extending M_0 such that $M_A(A) = \mathbb{Y}^{\mathbb{X}}$ and $M_A(f')\colon \mathbb{Y}^{\mathbb{X}} \times \mathbb{X} \to \mathbb{Y}$ is the application. It can be noted that M_A is the terminal model of $\Theta_{op,A}$ extending M_0. Then the parameter passing function is a bijection, and composing it with the parameter adding bijection we get (where $\ulcorner M(f) \urcorner \in \mathbb{Y}^{\mathbb{X}}$ corresponds by currying to $M(f)\colon \mathbb{X} \to \mathbb{Y}$):

$$\mathit{Mod}(\Theta_{op})|_{M_0} \cong M_A(A) \quad \text{by} \quad M \leftrightarrow \ulcorner M(f) \urcorner \ .$$

Example 2.6 Let Θ_{sgp} be the theory for semigroups presented by one type G, one term $\mathit{prd}\colon G^2 \to G$ and one equation $\mathit{prd}(x, \mathit{prd}(y,z)) = \mathit{prd}(\mathit{prd}(x,y),z)$ where x, y, z are variables of type G. As usual with the categorical point of view, in fact the *variables* are projections; here, $x, y, z\colon G^3 \to G$ are the three projections and $\mathit{prd}(x,y)$ is $\mathit{prd} \circ \langle x, y \rangle\colon G^3 \to G$, composed of the pair $\langle x, y \rangle\colon G^3 \to G^2$ and of $\mathit{prd}\colon G^2 \to G$, and so on.

Parameterization process. In order to get parameterized families of semigroups, we consider the theory $\Theta_{sgp,A}$ presented by two types A and G, one term $\mathit{prd}'\colon A \times G^2 \to G$ and one equation $\mathit{prd}'(p, x, \mathit{prd}'(p, y, z)) = \mathit{prd}'(p, \mathit{prd}'(p, x, y), z)$ where x, y, z are variables of sort G and p is a variable of sort A.

Parameter passing process. The theory $\Theta_{sgp,a}$ is $\Theta_{sgp,A}$ together with a parameter $a\colon 1 \to A$, hence with $\mathit{prd}'' = \mathit{prd}' \circ (a \times id_{G^2})\colon G^2 \to G$ (where $1 \times G^2$ is identified to G^2). Each model M_A of $\Theta_{sgp,A}$ gives rise to a family of models of $\Theta_{sgp,a}$, all of them with the same underlying set $M_A(G)$ but with different interpretations of a in $M_A(A)$. Mapping prd to prd'' defines a morphism from Θ_{sgp} to $\Theta_{sgp,a}$. So, each model M_a of $\Theta_{sgp,a}$ gives rise to a model M of Θ_{sgp} such that $M(G) = M_a(G)$ and $M(\mathit{prd})(x,y) = M_a(\mathit{prd}')(\alpha, x, y)$ for each $x, y \in M_a(G)$, where $\alpha = M_a(a)$ is called the argument.

Example 2.7 This example motivates the existence of pure terms in the given theory. Let us consider the theory Θ_{nat} "of naturals" presented by a type N and two terms $z\colon 1 \to N$ and $s\colon N \to N$, and let us say that z is pure. Let $\Theta_{nat,0}$ be the subtheory presented by N and z, it is called the pure subtheory of Θ_{nat}. We define the theory $\Theta_{nat,A}$ as made of two types A and N and two terms $z\colon 1 \to N$ and $s'\colon A \times N \to N$. It should be noted that $\Theta_{nat,A}$ contains $\varepsilon_1\colon A \times 1 \to 1$ and $z' = z \circ \varepsilon_1\colon A \times 1 \to N$. Then $\Theta_{nat,A}$ is a theory "of lists of A", with z for the empty list and s' for concatenating an element to a list. In this way, the theory of lists of A is built as a generalization of the theory of naturals; indeed the naturals can be identified to the lists over a singleton.

Example 2.8 Here is another example where pure terms are required, this is a simplified version of many structures in Kenzo/EAT. Let Θ_{mon} be the theory for monoids presented by one type G, two terms $prd\colon G^2 \to G$ and $e\colon \to G$, and the equations $prd(x, prd(y,z)) = prd(prd(x,y),z)$, $prd(x,e) = x$, $prd(e,x) = x$ where x, y, z are variables of type G. Let Θ_{dm} be the theory for *differential monoids*, presented by Θ_{mon} together with one term $dif\colon G \to G$ and the equations $dif(prd(x,y)) = prd(dif(x), dif(y))$, $dif(e) = e$, $dif(dif(x)) = e$, and with the terms in Θ_{mon} as its pure terms. In order to get parameterized families of differential structures on one monoid, we define the theory $\Theta_{dm,A}$ presented by two types G and A, three terms $prd\colon G^2 \to G$, $e\colon 1 \to G$ and $dif'\colon A \times G \to G$, the three equations from Θ_{mon} and the equations $dif'(p, (prd(x,y))) = prd(dif'(p,x), dif'(p,y))$, $dif'(p,e) = e$, $dif'(p, dif'(p,x)) = e$. Each model M_A of $\Theta_{dm,A}$ gives rise to a family of models of Θ_{dm}, all of them with the same underlying monoid $(M_A(G), M_A(prd), M_A(e))$: there is a model M_a of Θ_{dm} extending M_A for each element α in $M_A(A)$, with its differential structure defined by $M_a(dif) = M_A(dif')(\alpha, -)$.

Example 2.9 In the next sections we will use the theories with the following presentations:

$$\Pi_A \dashv \boxed{A} \qquad \Pi_a \dashv \boxed{\begin{matrix} A \\ a\uparrow \\ 1 \end{matrix}} \qquad \Pi \dashv \boxed{1}$$

These theories are related by several morphisms: $\pi_A\colon \Pi_A \to \Pi$ maps A to 1, both $i\colon \Pi \to \Pi_a$ and $i_A\colon \Pi_A \to \Pi_a$ are the inclusions, and $\pi_a\colon \Pi_a \to \Pi$ extends π_A by mapping a to id_1, so that π_A and π_a are epimorphisms. In addition, $\pi_a \circ i_A = \pi_A$ and there is a natural transformation $p\colon i \circ \pi_A \Rightarrow i_A$ defined by $p_A = a\colon 1 \to A$. The diagram below on the right is the *lax colimit of* π_A, which means that it enjoys the following universal property: for each Π'_a with $i'_A\colon \Pi_A \to \Pi'_a$, $i'\colon \Pi \to \Pi'_a$ and $p'\colon i' \circ \pi_A \Rightarrow i'_A$, there is a unique $h\colon \Pi_a \to \Pi'_a$ such that $h \circ i_A = i'_A$, $h \circ i = i'$ and $h \circ p = p'$. For instance, given Π, $\pi_A\colon \Pi_A \to \Pi$, $id_\Pi\colon \Pi \to \Pi$ and $id_{\pi_A}\colon \pi_A \Rightarrow \pi_A$, then $\pi_a\colon \Pi_a \to \Pi$ is the unique morphism such that $\pi_a \circ i_A = \pi_A$, $\pi_a \circ i = id_\Pi$ and $\pi_a \circ p = id_{\pi_A}$.

$$\begin{tikzcd}
\Pi_A \arrow[dd, "\pi_A"'] \\
\\
\Pi
\end{tikzcd}
\qquad
\begin{tikzcd}
\Pi_A \arrow[dd, "\pi_A"'] \arrow[drr, "i_A"] & & \\
 & = & \Pi_a \arrow[dll, "\pi_a"] \\
\Pi & &
\end{tikzcd}
\qquad
\begin{tikzcd}
\Pi_A \arrow[dd, "\pi_A"'] \arrow[drr, "i_A"] & & \\
 & p \Uparrow & \Pi_a \\
\Pi \arrow[urr, "i"'] & &
\end{tikzcd}$$

2.3 Some other kinds of theories

For every theory Θ, the coslice category of *theories under* Θ is denoted $\Theta{\downarrow}\mathbf{T}_{eq}$. It can be seen as a 2-category, with the natural transformations which extend the identity on Θ as 2-cells.

Definition 2.10 A *parameterized theory* Θ_A is a theory Θ with a distinguished type, called the *type of parameters* and usually denoted A. The 2-category of parameterized theories is the coslice 2-category $\mathbf{T}_A = \Pi_A{\downarrow}\mathbf{T}_{eq}$ of theories under Π_A. A *theory with a parameter* Θ_a is a parameterized theory with a distinguished constant of type A, called the *parameter* and usually denoted $a\colon 1 \to A$. The 2-category of theories with a parameter is the coslice 2-category $\mathbf{T}_a = \Pi_a{\downarrow}\mathbf{T}_{eq}$ of theories under Π_a.

According to the context, Θ_A denotes either the parameterized theory $\gamma_A\colon \Pi_A \to \Theta_A$, or the equational theory Θ_A itself. Similarly for Θ_a, which denotes either $\gamma_a\colon \Pi_a \to \Theta_a$ or Θ_a itself. In addition, it can be noted that Π is the initial theory (which may also be presented by the empty specification) so that $\Pi{\downarrow}\mathbf{T}_{eq}$ is isomorphic to $\mathbf{T}_{eq}$. The 2-categories $\mathbf{S}_A$ and $\mathbf{S}_a$ of *parameterized specifications* and *specifications with a parameter*, respectively, are defined in a similar way.

On the other hand, the input of the parameterization process is a theory Θ together with a wide subtheory Θ_0 (*wide* means: with the same types), such a structure is called a decorated theory.

Definition 2.11 A *decorated theory* is made of a theory Θ with a wide subtheory Θ_0 called the *pure* subtheory of Θ. A morphism of decorated theories is a morphism of theories $\theta\colon \Theta \to \Theta'$ which maps the pure part of Θ to the pure part of Θ'. This forms the category $\mathbf{T}_{dec}$ of decorated theories.

So, a decorated theory Θ is endowed with a distinguished family of terms, called the *pure* terms, such that all the identities and projections are pure and every composition or tuple of pure terms is pure. Pure terms are denoted with "$\rightsquigarrow$". When there is no ambiguity we often use the same notation Θ for the theory Θ itself and for the decorated theory made of Θ and Θ_0. The decorated specifications are defined in a straightforward way. For instance, we may consider the decorated specification made of a type N, a pure term $z\colon 1 \rightsquigarrow N$ and a term $s\colon N \to N$ (see example 2.7).

3 Constructions

3.1 The parameterization process is a functor

In this section we prove that the parameterization process is functorial, by defining a functor $F_{par}\colon \mathbf{T}_{dec} \to \mathbf{T}_A$, called the *parameterization functor*, which adds the type of parameters to the domain of every non-pure term. In addition, theorem 3.2 states that F_{par} is left adjoint to the functor $G_{par}\colon \mathbf{T}_A \to \mathbf{T}_{dec}$, which builds the coKleisli category of the comonad $A \times -$.

In order to define the functor F_{par} we use the fact that it should preserve colimits. It has been seen in section 2.1 that every specification is the colimit of a diagram of elementary specifications. Similarly, every decorated specification is the colimit of a diagram of elementary decorated specifications, denoted $\Sigma_{\mathtt{E.x}}$ where $\mathtt{x} = \mathtt{p}$ for "pure" or $\mathtt{x} = \mathtt{g}$ for "general". Informally, the functor F_{par} explicits the fact that every general feature in a decorated specification gets parameterized, while every pure feature remains unparameterized. Figure 2 defines the parameterized specification $F_{par}(\Sigma_{\mathtt{E.x}})$ for each elementary decorated specification $\Sigma_{\mathtt{E.x}}$ (many projection arrows are omitted, when needed the projections from $A \times X$ are denoted $pr_X\colon A \times X \to A$ and $\varepsilon_X\colon A \times X \to X$). The morphisms of parameterized specifications $F_{par}(\sigma)$, for σ between elementary decorated specifications, are straightforward. For instance, let $\sigma_{\mathtt{c}}\colon \Sigma_{\mathtt{Term.g}} \to \Sigma_{\mathtt{Term.p}}$ be the conversion morphism, which corresponds to the fact that every pure term can be seen as a general term, then $F_{par}(\sigma_{\mathtt{c}})$ maps $f'\colon A \times X \to Y$ in $F_{par}(\Sigma_{\mathtt{Term.g}})$ to $f \circ \varepsilon_X\colon A \times X \to Y$ in $F_{par}(\Sigma_{\mathtt{Term.p}})$. Now, given a decorated theory Θ presented by the colimit of a diagram Δ of elementary decorated specifications, we define $F_{par}(\Theta)$ as the parameterized theory presented by the colimit of the diagram $F_{par}(\Delta)$ of parameterized specifications.

Definition 3.1 The functor $F_{par} : \mathbf{T}_{dec} \to \mathbf{T}_A$ defined above is called the *parameterization functor*.

Clearly the parameterization functor preserves colimits. In addition, let Θ_A be the parameterized theory $F_{par}(\Theta)$, it follows from the definition of F_{par} that the equational theory Θ_A is a theory under Θ_0.

Now the functor G_{par} is defined independently from F_{par}. Let Θ_A be a parameterized theory. The endofunctor of product with A forms a comonad on Θ_A with the counit ε made of the projections $\varepsilon_X\colon A \times X \to X$ and the comultiplication made of the terms $\delta_X\colon A \times X \to A \times A \times X$ induced by the diagonal on A. Let

	index `E.x`	$\Sigma_{\texttt{E.x}}$	$F_{par}\Sigma_{\texttt{E.x}}$
type	`Type.p`	X	X
pure term	`Term.p`	$\begin{tikzcd} X \arrow[r, squiggly, "f"] & Y \end{tikzcd}$	$\begin{tikzcd} X \arrow[r, "f"] & Y \end{tikzcd}$
term	`Term.g`	$\begin{tikzcd} X \arrow[r, "f"] & Y \end{tikzcd}$	$\begin{tikzcd} A\times X \arrow[r, "f'"] & Y \end{tikzcd}$
selection of identity	`Selid.p`	$\begin{tikzcd} X \arrow[r, squiggly, "id_X"] & X \end{tikzcd}$	$\begin{tikzcd} X \arrow[r, "id_X"] & X \end{tikzcd}$
pure composition	`Comp.p`	$\begin{tikzcd} X \arrow[r, squiggly, "f"] \arrow[rr, squiggly, bend right, "=", "g\circ f"'] & Y \arrow[r, squiggly, "g"] & Z \end{tikzcd}$	$\begin{tikzcd} X \arrow[r, "f"] \arrow[rr, bend right, "=", "g\circ f"'] & Y \arrow[r, "g"] & Z \end{tikzcd}$
composition	`Comp.g`	$\begin{tikzcd} X \arrow[r, "f"] \arrow[rr, bend right, "=", "g\circ f"'] & Y \arrow[r, "g"] & Z \end{tikzcd}$	$\begin{tikzcd} A\times X \arrow[r, "\langle pr_X, f'\rangle"] \arrow[rr, bend right, "=", "g'\circ\langle pr_X, f'\rangle"'] & A\times Y \arrow[r, "g'"] & Z \end{tikzcd}$
terminal type	`0-Prod.p`	1	1
pure collapsing	`0-Tuple.p`	$\begin{tikzcd} X \arrow[r, squiggly, "\langle\,\rangle_X"] & 1 \end{tikzcd}$	$\begin{tikzcd} X \arrow[r, "\langle\,\rangle_X"] & 1 \end{tikzcd}$
binary product	`2-Prod.p`	$\begin{tikzcd} X & \\ & X\times Y \arrow[ul, squiggly, "p_X"'] \arrow[dl, squiggly, "p_Y"] \\ Y & \end{tikzcd}$	$\begin{tikzcd} X & \\ & X\times Y \arrow[ul, "p_X"'] \arrow[dl, "p_Y"] \\ Y & \end{tikzcd}$
pure pairing	`2-Tuple.p`	$\begin{tikzcd} & X \arrow[d, phantom, "="] & \\ Z \arrow[ur, squiggly, "f"] \arrow[dr, squiggly, "g"'] \arrow[rr, squiggly, "\langle f,g\rangle" description] & & X\times Y \arrow[ul, squiggly, "p_X"'] \arrow[dl, squiggly, "p_Y"] \\ & Y \arrow[u, phantom, "="] & \end{tikzcd}$	$\begin{tikzcd} & X \arrow[d, phantom, "="] & \\ Z \arrow[ur, "f"] \arrow[dr, "g"'] \arrow[rr, "\langle f,g\rangle" description] & & X\times Y \arrow[ul, "p_X"'] \arrow[dl, "p_Y"] \\ & Y \arrow[u, phantom, "="] & \end{tikzcd}$
pairing	`2-Tuple.g`	$\begin{tikzcd} & X \arrow[d, phantom, "="] & \\ Z \arrow[ur, "f"] \arrow[dr, "g"'] \arrow[rr, "\langle f,g\rangle" description] & & X\times Y \arrow[ul, squiggly, "p_X"'] \arrow[dl, squiggly, "p_Y"] \\ & Y \arrow[u, phantom, "="] & \end{tikzcd}$	$\begin{tikzcd} & X \arrow[d, phantom, "="] & \\ A\times Z \arrow[ur, "f'"] \arrow[dr, "g'"'] \arrow[rr, "\langle f',g'\rangle" description] & & X\times Y \arrow[ul, "p_X"'] \arrow[dl, "p_Y"] \\ & Y \arrow[u, phantom, "="] & \end{tikzcd}$

Figure 2: The functor F_{par} on elementary decorated specifications

Θ be the coKleisli category of this comonad: it has the same types as Θ_A and a term $[f]\colon X \to Y$ for each term $f\colon A \times X \to Y$ in Θ_A. There is a functor from Θ_A to Θ which is the identity on types and maps every $g\colon X \to Y$ in Θ_A to $[g \circ \varepsilon_X]\colon X \to Y$ in Θ. Then every finite product in Θ_A is mapped to a finite product in Θ, which makes Θ a theory. Let Θ_0 denote the image of Θ_A in Θ, it is a wide subtheory of Θ. In this way, any parameterized theory yields a decorated theory. The definition of G_{par} on morphisms is straightforward, and the next result follows easily.

Theorem 3.2 *The parameterization functor F_{par} and the functor G_{par} form an adjunction $F_{par} \dashv G_{par}$:*

$$\begin{tikzcd} \mathbf{T}_{dec} \arrow[r, bend left, "F_{par}"] \arrow[r, phantom, "\perp"] & \mathbf{T}_A \arrow[l, bend left, "G_{par}"] \end{tikzcd}$$

The next result states that Θ can be easily recovered from Θ_A, by mapping A to 1.

Proposition 3.3 *Let Θ be a decorated theory with pure subtheory Θ_0 and $\gamma_A\colon \Pi_A \to \Theta_A$ the parameterized theory $F_{par}(\Theta)$. Let $\gamma\colon \Pi \to \Theta$ be the unique morphism from the initial theory Π to the theory Θ. Then there* *is a morphism $\theta_A\colon \Theta_A \to \Theta$ under Θ_0 such that the following square is a pushout:*

$$\begin{tikzcd}
\Pi_A \arrow[r, "\gamma_A"] \arrow[d, "\pi_A"'] \arrow[dr, phantom, "{[P.O.]}"] & \Theta_A \arrow[d, "\theta_A"] \\
\Pi \arrow[r, "\gamma"] & \Theta
\end{tikzcd}$$

Proof. It can easily be checked that this property is satisfied by each elementary specification. Then the result follows by commuting two colimits: on the one hand the colimit that defines the given theory from its elementary components, and on the other hand the pushout. □

When there is an epimorphism of theories $\theta\colon \Theta_1 \to \Theta_2$, we say that Θ_1 is *the generalization of* Θ_2 *along* θ. Indeed, since θ is an epimorphism, the functor $\theta^*\colon \mathit{Mod}(\Theta_2) \to \mathit{Mod}(\Theta_1)$ is a monomorphism, which can be used for identifying $\mathit{Mod}(\Theta_2)$ to a subcategory of $\mathit{Mod}(\Theta_1)$.

Corollary 3.4 *With notations as in proposition 3.3, Θ_A is the generalization of Θ along θ_A.*

Proof. Clearly $\pi_A\colon \Pi_A \to \Pi$ is an epimorphism. Since epimorphisms are stable under pushouts, proposition 3.3 proves that $\theta_A\colon \Theta_A \to \Theta$ is also an epimorphism. □

Let $F_{par} : \mathbf{T}_{dec} \to \mathbf{T}_A$ be the parameterization functor and let $U\colon \mathbf{T}_A \to \mathbf{T}_{eq}$ be the functor which simply forgets that the type A is distinguished, so that $U \circ F_{par}\colon \mathbf{T}_{dec} \to \mathbf{T}_{eq}$ maps the decorated theory Θ to the equational theory Θ_A.

$$\begin{tikzcd}
\mathbf{T}_{dec} \arrow[r, "F_{par}"] & \mathbf{T}_A \arrow[r, "U"] & \mathbf{T}_{eq}
\end{tikzcd}$$

Every theory Θ can be seen as a decorated theory where the pure terms are defined inductively as the identities, the projections, and the compositions and tuples of pure terms. Let $I\colon \mathbf{T}_{eq} \to \mathbf{T}_{dec}$ denote the corresponding inclusion functor. Then the endofunctor $U \circ F_{par} \circ I\colon \mathbf{T}_{eq} \to \mathbf{T}_{eq}$ corresponds to the "*imp* construction" of [11], which transforms each term $f\colon X \to Y$ in Θ into $f'\colon A \times X \to Y$ for a new type A.

3.2 The parameter passing process is a natural transformation

A theory Θ_a with a parameter is built simply by adding a constant a of type A to a parameterized theory Θ_A. Obviously, this can be seen as a pushout.

Definition 3.5 Let $\gamma_A\colon \Pi_A \to \Theta_A$ be a parameterized theory. The theory with parameter *extending* γ_A is $\gamma_a\colon \Pi_a \to \Theta_a$ given by the pushout of γ_A and i_A:

$$\begin{tikzcd}
\Pi_A \arrow[r, "\gamma_A"] \arrow[d, "i_A"'] \arrow[dr, phantom, "{[P.O.]}"] & \Theta_A \arrow[d, "j_A"] \\
\Pi_a \arrow[r, "\gamma_a"] & \Theta_a
\end{tikzcd}$$

This pushout of theories gives rise to a pullback of categories of models, hence for each model M_A of Θ_A the function which maps each model M_a of Θ_a extending M_A to the element $M_a(a) \in M_A(A)$ defines a bijection:

$$\mathit{Mod}(\Theta_a)|_{M_A} \xrightarrow{\simeq} M_A(A) \ . \qquad (1)$$

Let us assume that the parameterized theory $\gamma_A\colon \Pi_A \to \Theta_A$ is $F_{par}(\Theta)$ for some decorated theory Θ with pure subtheory Θ_0. Then the pushout property in definition 3.5 ensures the existence of a unique $\theta_a\colon \Theta_a \to \Theta$ such that $\theta_a \circ \gamma_a = \gamma \circ \pi_a$ and $\theta_a \circ j_A = \theta_A$, which means that θ_a maps A to 1 and a to id_1 and the θ_a extends θ_A. Then Θ_A is a theory under Θ_0 and the composition by j_A makes Θ_a a theory under Θ_0 with j_A preserving Θ_0.

$$\begin{tikzcd}
\Theta_A \arrow[dd, "\theta_A"'] \arrow[dr, "j_A"] & \\
= & \Theta_a \arrow[dl, "\theta_a"] \\
\Theta &
\end{tikzcd}$$

Definition 3.6 For each decorated theory Θ with pure subcategory Θ_0, let $\Theta_A = F_{par}(\Theta)$ and $\theta_A\colon \Theta_A \to \Theta$ as in proposition 3.3, and let Θ_a and $j_A\colon \Theta_A \to \Theta_a$ as in definition 3.5. Then $j\colon \Theta \to \Theta_a$ is the morphism under Θ_0 which maps each type X to X and each term $f\colon X \to Y$ to $f' \circ (a \times id_X)\colon X \to Y$. And $t\colon j \circ \theta_A \Rightarrow j_A$ is the natural transformation under Θ_0 such that $t_A = a\colon 1 \to A$.

Lax cocones and lax colimits in 2-categories generalize cocones and colimits in categories, so that the following diagram is a *lax cocone with base* θ_A in the 2-category $\Theta_0 \downarrow \mathbf{T}_{eq}$, for short it is denoted (Θ_a, j_A, j, t), and it is called *the lax colimit associated to* Θ because of lemma 3.7.

$$\begin{tikzcd}
\Theta_A \arrow[dd, "\theta_A"'] \arrow[dr, "j_A"] & \\
t \Uparrow & \Theta_a \\
\Theta \arrow[ur, "j"'] &
\end{tikzcd}$$

Lemma 3.7 *Let Θ be a decorated theory with pure subcategory Θ_0. The lax cocone (Θ_a, j_A, j, t) with base θ_A defined above is a lax colimit in the 2-category of theories under Θ_0.*

Proof. This means that the given lax cocone is initial among the lax cocones with base θ_A in $\Theta_0 \downarrow \Theta$, in the following sense: for every lax cocone $(\Theta'_a, j'_A, j', t')$ with base θ_A under Θ_0 there is a unique morphism $h\colon \Theta_a \to \Theta'_a$ such that $h \circ j_A = j'_A$, $h \circ j = j'$ and $h \circ t = t'$. Indeed, h is defined from the pushout in definition 3.5 by $h \circ j_A = j'_A$, so that $h(A) = A$, and $h \circ \gamma_a(a) = t'_A : 1 \to A$. □

For instance, given Θ, $\theta_A\colon \Theta_A \to \Theta$, $id_\Theta\colon \Theta \to \Theta$ and $id_{\theta_A}\colon \theta_A \Rightarrow \theta_A$, then θ_a is the unique morphism such that $\theta_a \circ j_A = \theta_A$, $\theta_a \circ j = id_\Theta$ and $\theta_a \circ t = id_{\theta_A}$.

Let Θ be a decorated theory with pure subtheory Θ_0 and let (Θ_a, j_A, j, t) be its associated lax colimit, with base $\theta_A\colon \Theta_A \to \Theta$. Let M_A be a model of Θ_A and M_0 its restriction to Θ_0, and let $\{(M, m) \mid m\colon {\theta_A}^* M \to M_A\}|_{M_0}$ (where as before ${\theta_A}^* M = M \circ \theta_A$) denote the set of pairs (M, m) with M a model of Θ extending M_0 and m a morphism of models of Θ_A extending id_{M_0}. A consequence of the lax colimit property is that the function which maps each model M_a of Θ_a extending M_A to the pair $(j^* M_a, t^* M_a) = (M_a \circ j, M_a \circ t)$ defines a bijection:

$$Mod(\Theta_a)|_{M_A} \cong \{(M, m) \mid m\colon {\theta_A}^* M \to M_A\}|_{M_0} \ . \tag{2}$$

The bijections 1 and 2 provide the next result, which does not involve Θ_a.

Proposition 3.8 *Let Θ be a decorated theory with pure subtheory Θ_0 and let $\Theta_A = F_{par}(\Theta)$ and $\theta_A\colon \Theta_A \to \Theta$. Then for each model M_A of Θ_A, with M_0 denoting the restriction of M_A to Θ_0, the function which maps each element $\alpha \in M_A(A)$ to the pair (M, m), where M is the model of Θ such that $M(f) = M_A(f')(\alpha, -)$ and where $m : {\theta_A}^* M \to M_A$ is the morphism of models of Θ_A such that $m_A : M(1) \to M_A(A)$ is the constant function α, defines a bijection:*

$$M_A(A) \cong \{(M, m) \mid m\colon {\theta_A}^* M \to M_A\}|_{M_0} \ . \tag{3}$$

As an immediate consequence, we get the *exact parameterization* property from [11].

Corollary 3.9 *Let Θ be a decorated theory with pure subcategory Θ_0, and let $\Theta_A = F_{par}(\Theta)$. Let M_0 be a model of Θ_0 and M_A a terminal model of Θ_A extending M_0. Then there is a bijection:*

$$M_A(A) \cong Mod(\Theta)|_{M_0} \tag{4}$$

which maps each $\alpha \in M_A(A)$ to the model $M_{A,\alpha}$ of Θ defined by $M_{A,\alpha}(X) = M_0(X)$ for each type X and $M_{A,\alpha}(f) = M_A(f')(\alpha, -)$ for each term f, so that $M_{A,\alpha}(f) = M_A(f)$ for each pure term f.

The existence of a terminal model of Θ_A extending M_0 is a consequence of [17] and [10]. Corollary 3.9 corresponds to the way algebraic structures are implemented in the systems Kenzo/EAT. In these systems the parameter set is encoded by means of a record of Common Lisp functions, which has a field for each operation in the algebraic structure to be implemented. The pure terms correspond to functions which can be obtained from the fixed data and do not require an explicit storage. Then, each particular instance of the record gives rise to an algebraic structure.

Clearly the construction of γ_a from γ_A is a functor, which is left adjoint to the functor which simply forgets that the constant a is distinguished. So, by composing this adjunction with the adjunction $F_{par} \dashv G_{par}$ from theorem 3.2 we get an adjunction $F'_{par} \dashv G'_{par}$ where F'_{par} maps each decorated theory Θ to Θ_a, as defined above:

$$\begin{tikzcd}
\mathbf{T}_{dec} \arrow[r, bend left, "F'_{par}"] \arrow[r, phantom, "\perp"] & \mathbf{T}_a \arrow[l, bend left, "G'_{par}"]
\end{tikzcd}$$

Let $U' \colon \mathbf{T}_a \to \mathbf{T}_{eq}$ be the functor which simply forgets that the type A and the constant a are distinguished. Then the functor $U' \circ F'_{par} \colon \mathbf{T}_{dec} \to \mathbf{T}_{eq}$ maps the decorated theory Θ to the equational theory Θ_a.

$$\begin{tikzcd}
\mathbf{T}_{dec} \arrow[r, "F'_{par}"] & \mathbf{T}_a \arrow[r, "U'"] & \mathbf{T}_{eq}
\end{tikzcd}$$

The morphism of theories $j \colon \Theta \to \Theta_a$ from definition 3.6 depends on the decorated theory Θ, let us denote it $j = J_\Theta$. Let $H \colon \mathbf{T}_{dec} \to \mathbf{T}_{eq}$ be the functor which maps each decorated theory Θ to the equational theory Θ. The next result is easy to check.

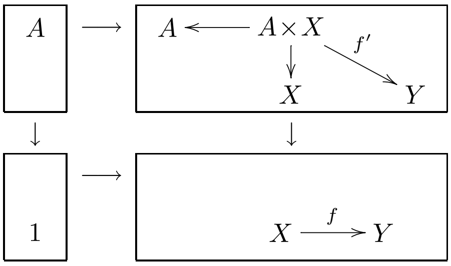
Theorem 3.10 *The morphisms of theories $J_\Theta \colon \Theta \to \Theta_a$ form the components of a natural transformation $J \colon H \Rightarrow U' \circ F'_{par} \colon \mathbf{T}_{dec} \to \mathbf{T}_{eq}$.*

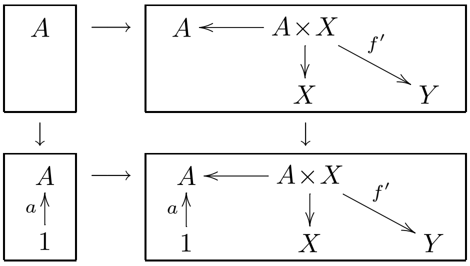
$$\begin{tikzcd}
\mathbf{T}_{dec} \arrow[r, "F'_{par}"] \arrow[rr, bend right, "H"'] & \mathbf{T}_a \arrow[r, "U'"] \arrow[d, phantom, "\Uparrow J"] & \mathbf{T}_{eq} \\
& {} &
\end{tikzcd}$$

Definition 3.11 The natural transformation $J \colon H \Rightarrow U' \circ F'_{par} \colon \mathbf{T}_{dec} \to \mathbf{T}_{eq}$ in theorem 3.10 is called the *parameter passing natural transformation.*

3.3 Examples

Example 3.12 Starting from Θ_{op} and $\Theta_{op,0}$ as in example 2.5, the pushouts of theories from proposition 3.3 and definition 3.5 are respectively:





We have seen in example 2.5 two other presentations of the vertex $\Theta_{op,a}$ of the second pushout, with $f'' = f' \circ (a \times id_X) : X \to Y$. For each decorated theory Θ, the morphism of equational theories $j_{op} = J_{\Theta_{op}}$: $\Theta \to \Theta_a$ maps f to f'', as in example 2.5.

A model M_0 of $\Theta_{op,0}$ is simply made of two sets $\mathbb{X} = M_0(X)$ and $\mathbb{Y} = M_0(Y)$. On the one hand, a model of Θ extending M_0 is characterized by a function $\varphi: \mathbb{X} \to \mathbb{Y}$. On the other hand, the terminal model M_A of $\Theta_{op,A}$ extending M_0 is such that $M_A(A) = \mathbb{Y}^{\mathbb{X}}$ and $M_A(f'): \mathbb{Y}^{\mathbb{X}} \times \mathbb{X} \to \mathbb{Y}$ is the application. The bijection $Mod(\Theta)|_{M_0} \cong M_A(A)$ then corresponds to the currying bijection $\varphi \mapsto \ulcorner\varphi\urcorner$.

Example 3.13 Let Θ_{dm} be the theory for differential monoids from example 2.8, with the pure subtheory $\Theta_{dm,0} = \Theta_{mon}$ of monoids. They generate the parameterized theory $\Theta_{dm,A}$ as in example 2.8. Let M_0 be some fixed monoid and M_A any model of $\Theta_{dm,A}$ extending M_0, then each element of $M_A(A)$ corresponds to a differential structure on the monoid M_0. If in addition M_A is the terminal model of $\Theta_{dm,A}$ extending M_0, then this correspondence is bijective.

Example 3.14 When dealing with an imperative language, the states for the memory are endowed with an operation *lookup* for observing the state and an operation *update* for modifying it. There are two points of view on this situation: either the state is hidden, or it is explicit. Let us check that the parameterization process allows to generate the theory with explicit state from the theory with hidden state.

First, let us focus on observation: the theory Θ_{st} is made of two types L and Z (for locations and integers, respectively) and a term $v: L \to Z$ for observing the values of the variables. The pure subtheory $\Theta_{st,0}$ is made of L and Z. We choose a model M_0 of $\Theta_{st,0}$ made of a countable set of locations (or adresses, or "variables") $\mathbb{L} = M_0(L)$ and of the set of integers $\mathbb{Z} = M_0(Z)$. Let $\mathbb{A} = \mathbb{Z}^{\mathbb{L}}$, then as in example 3.12 the terminal model M_A of $\Theta_{st,A}$ extending M_0 is such that $M_A(A) = \mathbb{A}$ and $M_{st,A}(v'): \mathbb{A} \times \mathbb{L} \to \mathbb{Z}$ is the application, denoted *lookup*. The terminal model M_A does correspond to an "optimal" implementation of the state.

Now, let us look at another model N_A of $\Theta_{st,A}$ extending M_0, defined as follows: $N_A(A) = \mathbb{A} \times \mathbb{L} \times \mathbb{Z}$ and $N_A(v'): \mathbb{A} \times \mathbb{L} \times \mathbb{Z} \times \mathbb{L} \to \mathbb{Z}$ maps (p, x, n, y) to n if $x = y$ and to $lookup(p, y)$ otherwise. The terminality property of M_A ensures that there is a unique function $update: \mathbb{A} \times \mathbb{L} \times \mathbb{Z} \to \mathbb{A}$ such that $lookup(update(p, x, n), y)$ is n if $x = y$ and is $lookup(p, y)$ otherwise. So, the operation *update* is defined coinductively from the operation *lookup*.

4 Conclusion

This paper provides a neat categorical formalization for the parameterization process in Kenzo and EAT. An additional level of abstraction allows to see the parameterization process as a morphism of logics and the parameter passing process as a 2-morphism of logics, in a relevant 2-category of logics [5]. Future work includes the generalization of this approach from equational theories to other families of theories, like distributive categories, and to more general kinds of parameters, like data types.

References

[1] Michael Barr, Charles Wells. Category Theory for Computing Science. Centre de Recherches Mathématiques (CRM) Publications, 3rd Edition, 1999.

[2] Hartmut Ehrig, Hans-Jörg Kreowski, James Thatcher, Eric Wagner, Jesse Wright. Parameterized Data Types in Algebraic Specification Languages. Springer. Lecture Notes in Computer Science 85, p. 157–168 (1980).

[3] César Domínguez, Dominique Duval, Laureano Lambán, Julio Rubio. Towards Diagrammatic Specifications of Symbolic Computation Systems. In: Mathematics, Algorithms, Proofs. T. Coquand, H. Lombardi, M. Roy (Eds.). Dagstuhl Seminar 05021 (2005). `http://drops.dagstuhl.de/portals/index.php?semnr=05021`.

[4] César Domínguez, Dominique Duval. A parameterization process as a categorical construction. `arXiv:0908.3634` (2009).

[5] César Domínguez, Dominique Duval. Diagrammatic logic applied to a parameterization process. `arXiv:0908.3737` (2009).

[6] César Domínguez, Laureano Lambán, Julio Rubio. Object-Oriented Institutions to Specify Symbolic Computation Systems. Rairo - Theoretical Informatics and Applications 41, p. 191–214 (2007).

[7] César Domínguez, Julio Rubio, Francis Sergeraert. Modeling Inheritance as Coercion in the Kenzo System. Journal of Universal Computer Science 12 (12), p. 1701–1730 (2006).

[8] Xavier Dousson, Francis Sergeraert, Yvon Siret. The Kenzo Program. Institut Fourier, Grenoble (1999). `http://www-fourier.ujf-grenoble.fr/~sergerar/Kenzo`.

[9] Joseph Goguen, Grant Malcolm. A Hidden Agenda. Theoretical Computer Science 245 (1), p. 55–101 (2000).

[10] Ulrich Hensel, Horst Reichel Defining Equations in Terminal Coalgebras. In Recentr Trends in Data Type Specifications, Springer. Lecture Notes in Computer Science 906, p. 307–318 (1995).

[11] Laurenano Lambán, Vico Pascual, Julio Rubio. An Object-Oriented Interpretation of the EAT System. Applicable Algebra in Engineering, Communication and Computing, 14 (3), p. 187–215 (2003).

[12] S.K. Lellahi. Categorical Abstract Data Type (CADT). Diagrammes 21, SKL1-SKL23 (1989).

[13] Jacques Loeckx, Hans-Dieter Ehrich, Markus Wolf. Specification of Abstract Data Types. Wiley and Teubner, New York (1996).

[14] Saunders Mac Lane. Categories for the Working Mathematician. Springer, 2th edition, 1998.

[15] Andrew M. Pitts. Categorical Logic. Chapter 2 of S. Abramsky and D. M. Gabbay and T. S. E. Maibaum (Eds). Handbook of Logic in Computer Science, Volume 5. Algebraic and Logical Structures. Oxford University Press, 2000.

[16] Julio Rubio, Francis Sergeraert, Yvon Siret. EAT: Symbolic Software for Effective Homology Computation. Institut Fourier, Grenoble (1997). `ftp://fourier.ujf-grenoble.fr/pub/EAT`.

[17] J.J.M.M. Rutten. Universal Coalgebra: a Theory of Systems. Theoretical Computer Science 249 (1), p. 3–80 (2000).