HAL open science

A parameterization process as a categorical construction

Dominique Duval, César Dominguez

▶ To cite this version:

Dominique Duval, César Dominguez. A parameterization process as a categorical construction. 2009. hal-00411061

HAL Id: hal-00411061
https://hal.science/hal-00411061v1

Preprint submitted on 25 Aug 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A parameterization process as a categorical construction

César Domínguez * Dominique Duval †

August 25., 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 transforming some operations into parameterized operations, which depend on one additional variable called the parameter. 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 free 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 [20] 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 [14, 5, 6] 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 [14] 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" [6], 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 [14] 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*

*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.

property. Later on, led by this characterization of EAT algebraic structures, in [14] we reinterpreted our results in terms of object-oriented technologies like hidden algebras [13] or coalgebras [21].

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 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 [14]. 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 left adjoint to the construction of a coKleisli category, and more precisely it is a free functor in the sense of section 4. 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 [7], it relies on *diagrammatic logic* [9, 10]. In this paper, the explicit use of diagrammatic logic is postponed to the appendix. With respect to the previous papers like [14], 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.

In section 2 equational theories are defined and several examples are presented. The parameterization process and the parameter passing process are defined categorically in section 3. In section 4 free functors are defined as left adjoint functors associated to morphisms of limit sketches, and it is proved that the parameterization

functor is free. The diagrammatic point of view on equational logic is presented in appendix A. Most of the categorical notions used in this paper can be found in [17] or in [1]. We omit the size issues: for instance most colimits should be small. A *graph* always means 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 Examples and definitions

2.1 Equational theories and specifications

In this paper, equational logic is seen from a categorical point of view, as for instance in [19].

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 [16]) and equational theories in the categorical style (as defined here) can be found in [19]. 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 $g_2 \circ f_2 = h$. However a more subtle point of view on equations is presented in appendix A.

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

For every morphism of equational theories $\theta\colon \Theta_1 \to \Theta$, we denote by $\theta^*\colon Mod(\Theta) \to Mod(\Theta_1)$ 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 *over* M_1 is denoted $Mod(\Theta)|_{M_1}$, it is the subcategory of $Mod(\Theta)$ 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 $Mod(\Theta)|_{M_1}$ 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 [15], [1], and also section 4.1 and appendix A.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 $Mod(\Theta)$ can be identified to the category $Mod(\Sigma)$ 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

	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`	$\begin{tikzcd} X \arrow[r, "f"] \arrow[rr, bend right, "g \circ f"'] \arrow[rr, phantom, "="] & Y \arrow[r, "g"] & Z \end{tikzcd}$
binary product	`2-Prod`	$\begin{tikzcd} X & & \\ & X \times Y \arrow[ul, "p_X"'] \arrow[dl, "p_Y"] & \\ Y & & \end{tikzcd}$
pairing (or binary tuple)	`2-Tuple`	$\begin{tikzcd} & X \arrow[dr, phantom, "="] & \\ Z \arrow[ur, "f"] \arrow[rr, "\langle f,g \rangle"] \arrow[dr, "g"'] & & X \times Y \arrow[ul, "p_X"'] \arrow[dl, "p_Y"] \\ & Y \arrow[ur, phantom, "="] & \end{tikzcd}$

Figure 1: Elementary specifications

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 $n \geq 0$, as in figure 1 (where only $n = 2$ is represented). 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.

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.

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

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

$\Theta_{op} \dashv$ $X \xrightarrow{f} Y$

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):

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

It is clear from anyone of these new 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 $(t_{op})_A = a\colon 1 \to A$.

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

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. We will also say that $\Theta_{op,A}$ is the *expansion* of Θ_{op}.

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 $\Theta_{op,a}$ over 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 $Mod(\Theta_{op,a})|_{M_A}$ is discrete):

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

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 $\Theta_{op,a}$ over M_A the model ${j_{op}}^*(M_a)$ of Θ_{op} is over M_0. This yields the *parameter passing* function (the categories $Mod(\Theta_{op,a})|_{M_A}$ and $Mod(\Theta_{op})|_{M_0}$ are discrete):

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

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 $\Theta_{op,A}$ over 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 $\Theta_{op,A}$ over 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 $M(f)\colon \mathbb{X} \to \mathbb{Y}$):

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

Example 2.6 Let Θ_{sgp} be the theory for semigroups presented by one type G, one term $prd\colon G^2 \to G$ and one equation $prd(x, prd(y, z)) = prd(prd(x, y), z)$ 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 for instance $prd(x, y)$ is $prd \circ \langle x, y\rangle\colon G^3 \to G$, composed of the pair $\langle x, y\rangle\colon G^3 \to G^2$ and of $prd\colon G^2 \to G$ (more details are given in the appendix, example A.3).

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 $prd'\colon A\times G^2 \to G$ and one equation $prd'(p,x,prd'(p,y,z)) = prd'(p,prd'(p,x,y),z)$ 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 $prd'' = prd' \circ (a \times id_{G^2})\colon G^2 \to G$ (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 $M(prd)(x,y) = M_a(prd')(\alpha,x,y)$ for each $x,y \in M_a(G)$, where $\alpha = M_a(a)$ is 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, A and three terms $prd\colon G^2 \to G$, $e\colon 1 \to G$ and $dif'\colon A\times G \to G$ and the equations $prd(x,prd(y,z)) = prd(prd(x,y),z)$, $prd(x,e) = x$, $prd(e,x) = x$, $dif'(p,(prd(x,y))) = prd(dif'(p,x), dif'(p,y))$, $dif'(p,e) = e$, $dif'(p,dif'(p,x)) = e$. 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} over 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:

$$\Pi_A \dashv \boxed{A} \qquad\qquad \Pi_a \dashv \boxed{\begin{array}{c} A \\ {\scriptstyle a}\uparrow \\ 1 \end{array}}$$

$$\Pi \dashv \boxed{1}$$

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 $\pi_a \circ p = id_{\pi_A}$.

$$\begin{tikzcd} \Pi_A \arrow[d, "\pi_A"'] \\ \Pi \end{tikzcd} \qquad \begin{tikzcd} \Pi_A \arrow[dr, "i_A"] \arrow[dd, "\pi_A"'] \arrow[dr, phantom, shift right=4ex, "="] & \\ & \Pi_a \arrow[dl, "\pi_a"] \\ \Pi & \end{tikzcd} \qquad \begin{tikzcd} \Pi_A \arrow[dr, "i_A"] \arrow[dd, "\pi_A"'] \arrow[dr, phantom, shift right=4ex, "p \Uparrow"] & \\ & \Pi_a \\ \Pi \arrow[ur, "i"'] & \end{tikzcd}$$

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 usually denotes the parameterized theory $\gamma_A\colon \Pi_A \to \Theta_A$, and sometimes it denotes the equational theory Θ_A itself. Similarly for Θ_a, which usually denotes $\gamma_a\colon \Pi_a \to \Theta_a$ and sometimes 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 $F_{exp}\colon \mathbf{T}_{dec} \to \mathbf{T}_A$ ("*exp*" for "expansion") which adds the type of parameters to the domain of every non-pure term. In addition, theorem 3.2 states that F_{exp} is left adjoint to the functor $G_{exp}\colon \mathbf{T}_A \to \mathbf{T}_{dec}$, which builds the coKleisli category of the comonad $A \times -$. Moreover, we will see in section 4 that F_{exp} is a free functor associated to a morphism of limit sketches, and in appendix A that this morphism of limit sketches underlies a morphism of diagrammatic logics.

In order to define the functor F_{exp} 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_{exp} 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 $F_{exp}\Sigma_{\mathtt{E.x}}$ for each elementary decorated

	index E.x	$\Sigma_{\texttt{E.x}}$	$F_{exp}\Sigma_{\texttt{E.x}}$
type	Type.p	X	X
pure term	Term.p	$X \overset{f}{\rightsquigarrow} Y$	$X \xrightarrow{f} Y$
term	Term.g	$X \xrightarrow{f} Y$	$A \times X \xrightarrow{f'} Y$
sel. of identity	Selid.p	$X \overset{id_X}{\rightsquigarrow} X$	$X \xrightarrow{id_X} X$
pure composition	Comp.p	$\begin{tikzcd} X \arrow[r, squiggly, "f"] \arrow[rr, squiggly, bend right, "g\circ f"'] & Y \arrow[r, squiggly, "g"] \arrow[d, phantom, "="] & Z \\ & {} & \end{tikzcd}$	$\begin{tikzcd} X \arrow[r, "f"] \arrow[rr, bend right, "g\circ f"'] & Y \arrow[r, "g"] \arrow[d, phantom, "="] & Z \\ & {} & \end{tikzcd}$
composition	Comp.g	$\begin{tikzcd} X \arrow[r, "f"] \arrow[rr, bend right, "g\circ f"'] & Y \arrow[r, "g"] \arrow[d, phantom, "="] & Z \\ & {} & \end{tikzcd}$	$\begin{tikzcd} A\times X \arrow[r, "\langle pr_X, f'\rangle"] \arrow[rr, bend right, "g'\circ\langle pr_X, f'\rangle"'] & A\times Y \arrow[r, "g'"] \arrow[d, phantom, "="] & Z \\ & {} & \end{tikzcd}$
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	$\begin{tikzcd} & X \arrow[dr, phantom, "="] & \\ Z \arrow[ur, squiggly, "f"] \arrow[rr, squiggly, "\langle f,g\rangle"] \arrow[dr, squiggly, "g"'] & & X\times Y \arrow[ul, squiggly, "p_X"'] \arrow[dl, squiggly, "p_Y"] \\ & Y \arrow[dr, phantom, "="] & \\ & {} & \end{tikzcd}$	$\begin{tikzcd} & X \arrow[dr, phantom, "="] & \\ Z \arrow[ur, "f"] \arrow[rr, "\langle f,g\rangle"] \arrow[dr, "g"'] & & X\times Y \arrow[ul, "p_X"'] \arrow[dl, "p_Y"] \\ & Y \arrow[dr, phantom, "="] & \\ & {} & \end{tikzcd}$
pairing	2-Tuple.g	$\begin{tikzcd} & X \arrow[dr, phantom, "="] & \\ Z \arrow[ur, "f"] \arrow[rr, "\langle f,g\rangle"] \arrow[dr, "g"'] & & X\times Y \arrow[ul, squiggly, "p_X"'] \arrow[dl, squiggly, "p_Y"] \\ & Y \arrow[dr, phantom, "="] & \\ & {} & \end{tikzcd}$	$\begin{tikzcd} & X \arrow[dr, phantom, "="] & \\ A\times Z \arrow[ur, "f'"] \arrow[rr, "\langle f',g'\rangle"] \arrow[dr, "g'"'] & & X\times Y \arrow[ul, "p_X"'] \arrow[dl, "p_Y"] \\ & Y \arrow[dr, phantom, "="] & \\ & {} & \end{tikzcd}$

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

specification $\Sigma_{\texttt{E.x}}$ (many projection arrows are omitted, and 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 $F_{exp}\sigma$, for σ between elementary decorated specifications, are straightforward. For instance, let $c\colon \Sigma_{\texttt{Term.g}} \to \Sigma_{\texttt{Term.p}}$ be the conversion morphism, which corresponds to the fact that every pure term can be seen as a general term, then $F_{exp}c$ maps $f'\colon A \times X \to Y$ in $F_{exp}\Sigma_{\texttt{Term.g}}$ to $f \circ \varepsilon_X\colon A \times X \to Y$ in $F_{exp}\Sigma_{\texttt{Term.p}}$. Now, given a decorated theory Θ presented by the colimit of a diagram Δ of elementary decorated specifications, we define $F_{exp}\Theta$ as the parameterized theory presented by the colimit of the diagram $F_{exp}\Delta$ of parameterized specifications.

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

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

Now the functor G_{exp} is defined independently from F_{exp}. 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 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_{exp} on morphisms is straightforward. The next result can be derived directly, or as a consequence of theorem 4.5.

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

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

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 $F_{exp}\Theta$. 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 $\theta^*\colon Mod(\Theta_2) \to Mod(\Theta_1)$ is a monomorphism, which can be used for identifying $Mod(\Theta_2)$ to a subcategory of $Mod(\Theta_1)$.

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 $F_{exp} : \mathbf{T}_{dec} \to \mathbf{T}_A$ 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 $U \circ F_{exp}\colon \mathbf{T}_{dec} \to \mathbf{T}_{eq}$ maps the decorated theory Θ to the equational theory Θ_A.

$$\mathbf{T}_{dec} \xrightarrow{F_{exp}} \mathbf{T}_A \xrightarrow{U} \mathbf{T}_{eq}$$

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 $U \circ F_{exp} \circ I\colon \mathbf{T}_{eq} \to \mathbf{T}_{eq}$ corresponds to the "*imp* construction" of [14], 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:

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

The pushout of theories in definition 3.5 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 over M_A to the element $M_a(a) \in M_A(A)$ defines a bijection:

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

Let us assume that the parameterized theory $\gamma_A\colon \Pi_A \to \Theta_A$ is $F_{exp}\Theta$ 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 $\theta_a \circ \gamma_a = \gamma \circ \pi_a$ (which means that θ_a maps A to 1 and a to id_1) and $\theta_a \circ j_A = \theta_A$. 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.

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

Lax cocones and lax colimits in 2-categories generalize cocones and colimits in categories. For each decorated theory Θ with pure subcategory Θ_0, let $\Theta_A = F_{exp}\Theta$ and $\theta_A\colon \Theta_A \to \Theta$ be as in section 3.1, and let Θ_a and $j_A\colon \Theta_A \to \Theta_a$ be as above. Let $j\colon \Theta \to \Theta_a$ be the morphism under Θ_0 which maps each type X to X and each term $f\colon X \to Y$ to $f' \circ (a \times id_X)\colon X \to Y$. Let $t\colon j \circ \theta_A \Rightarrow j_A$ be the natural transformation under Θ_0 such that $t_A = a\colon 1 \to A$. Then 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 3.6.

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

Lemma 3.6 *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 $h \circ t = t'$, it is defined from the pushout in definition 3.5 by $h \circ j_A = j'_A$, so that $h(A) = A$, and $h \circ \gamma_a(a) = t'_A : 1 \to A$. $\square$

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 over M_0

and m a morphism of models of Θ_A over M_0. A consequence of the lax colimit property is that the function which maps each model M_a of Θ_a over 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 3.7 *Let Θ be a decorated theory with pure subtheory Θ_0 and let $\Theta_A = F_{exp}\Theta$ 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 $\star \to \alpha$, 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 [14].

Corollary 3.8 *Let Θ be a decorated theory with pure subcategory Θ_0, and let $\Theta_A = F_{exp}\Theta$. Let M_0 be a model of Θ_0 and M_A a terminal model of Θ_A over 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 over M_0 is a consequence of [21] and [11]. Corollary 3.8 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 $F_{exp} \dashv G_{exp}$ from theorem 3.2 we get an adjunction $F'_{exp} \dashv G'_{exp}$ where F'_{exp} maps each decorated theory Θ to Θ_a, as defined above:

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

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 $U' \circ F'_{exp}\colon \mathbf{T}_{dec} \to \mathbf{T}_{eq}$ maps the decorated theory Θ to the equational theory Θ_a.

$$\mathbf{T}_{dec} \xrightarrow{F'_{exp}} \mathbf{T}_a \xrightarrow{U'} \mathbf{T}_{eq}$$

The morphism of theories $j\colon \Theta \to \Theta_a$ 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 3.9 *The morphisms of theories $J_\Theta\colon \Theta \to \Theta_a$ form the components of a natural transformation $J\colon H \Rightarrow U' \circ F'_{exp}\colon \mathbf{T}_{dec} \to \mathbf{T}_{eq}$.*

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

Definition 3.10 The natural transformation $J\colon H \Rightarrow U' \circ F'_{exp}\colon \mathbf{T}_{dec} \to \mathbf{T}_{eq}$ in theorem 3.9 is called the *parameter passing natural transformation.*

Example 3.11 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:

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

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 over M_0 is characterized by a function $\varphi\colon \mathbb{X} \to \mathbb{Y}$. On the other hand, the terminal model M_A of $\Theta_{op,A}$ over M_0 is 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. The bijection $Mod(\Theta)|_{M_0} \cong M_A(A)$ then corresponds to the currying bijection $\varphi \mapsto \ulcorner\varphi\urcorner$.

Example 3.12 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 $\Theta_{dm,A}$ over 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 $\Theta_{dm,A}$ over M_0, then this correspondence is bijective.

Example 3.13 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 $v\colon L \to Z$ 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 addresses, 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 3.11 the terminal model M_A of $\Theta_{st,A}$ over M_0 is such that $M_A(A) = \mathbb{A}$ and $M_{st,A}(v')\colon \mathbb{A} \times \mathbb{L} \to \mathbb{Z}$ 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 $\Theta_{st,A}$ over M_0, defined as follows: $N_A(A) = \mathbb{A}\times\mathbb{L}\times\mathbb{Z}$ and $N_A(v')\colon \mathbb{A}\times \mathbb{L} \times \mathbb{Z} \times \mathbb{L} \to \mathbb{Z}$ 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 $update\colon \mathbb{A} \times \mathbb{L} \times \mathbb{Z} \to \mathbb{A}$ such that $lookup(update(p, x, n), y)$ is n if $x = y$ and $lookup(p, y)$ otherwise. So, the updating operation *update* is defined coinductively from the observation operation *lookup*.

4 Free functors

In this section some basic facts about limit sketches and their associated adjunction are mentioned, and it is proved that the parameterization functor F_{exp} from section 3.1 is a free functor, in the sense that it is the left adjoint associated to a morphism of limits sketches.

4.1 Limit sketches

It is quite usual to define a *free* functor as the left adjoint of a forgetful functor, but there is no unique definition of a forgetful functor. In this section forgetful functors are defined from morphisms of limit sketches, they are not always faithful.

There are several definitions of limit sketches (also called projective sketches) in the litterature, see for instance [3] or [1]. These definitions are different but all of them serve the same purpose: each limit sketch generates a category with limits, so that limit sketches generalize equational specifications in allowing some interdependence between the variables. In this paper, limit sketches are used at the meta level, in order to describe each category of theories or specifications as the category of realizations (or models) of a limit sketch. While a category with limits is a graph with identities, composition, limit cones and tuples, satisfying a bunch of axioms, a limit sketch is a graph with *potential* identities, composition, limit cones and tuples, which are not required to satisfy any axiom. Potential limit cones, or simply *potential limits*, may also be called *specified limits* or *distinguished cones*.

Definition 4.1 A *limit sketch* is a graph where some points X have an associated potential identity arrow $id_X\colon X \to X$, some pairs of consecutive arrows $f\colon X \to Y$, $g\colon Y \to Z$ have an associated potential composed arrow $g \circ f\colon X \to Z$, some diagrams Δ have an associated potential limit, which is a cone with base Δ, and when there is a potential limit with base Δ then some cones with base Δ have an associated potential tuple, which is a morphism of cones with base Δ from the given cone to the potential limit cone. A morphism of limit sketches is a morphism of graphs which preserves the potential features. This yields the category of limit sketches.

Whenever this definition is restricted to potential limits with a finite discrete base (called potential finite products), we get the category of *finite product sketches*: this is the category $\mathbf{S}_{eq}$ of equational specifications, from section 2.1.

Definition 4.2 Given a limit sketch $\mathbf{E}$ and a category $\mathbf{C}$, a *realization* (or *loose model*) of $\mathbf{E}$ with values in $\mathbf{C}$ is a graph homomorphism which maps the potential features of $\mathbf{E}$ to real features of $\mathbf{C}$. A morphism of realizations is (an obvious generalization of) a natural transformation. This gives rise to the category $\mathit{Real}(\mathbf{E}, \mathbf{C})$ of realizations of $\mathbf{E}$ with values in $\mathbf{C}$. By default, $\mathbf{C}$ is the category of sets.

By default, $\mathbf{C}$ is the category of sets. A category is called *locally presentable* if it is equivalent to the category of set-valued realizations of a limit sketch $\mathbf{E}$; then $\mathbf{E}$ is called a limit sketch *for* this category.

Let $\overline{\mathbf{E}}$ denote the category generated by $\mathbf{E}$ such that every potential potential feature of $\mathbf{E}$ becomes a real feature of $\overline{\mathbf{E}}$. The *Yoneda contravariant realization* $\mathcal{Y}_{\mathbf{E}}$ of $\mathbf{E}$ is the contravariant realization of $\mathbf{E}$ with values in $\mathit{Real}(\mathbf{E})$ such that $\mathcal{Y}_{\mathbf{E}}(E) = \mathit{Hom}_{\overline{\mathbf{E}}}(E, -)$ for every point or arrow E in $\mathbf{E}$. Then for each theory Θ and each point E in $\mathbf{E}$, the set $\Theta(E)$ is in bijection with $\mathit{Hom}_{\mathit{Real}(\mathbf{E})}(\mathcal{Y}_{\mathbf{E}}(E), \Theta)$. The Yoneda contravariant realization is injective on objects and faithful. In addition it is *dense*: although $\mathit{Real}(\mathbf{E})$ may be "much larger" than $\mathbf{E}$, every realization of $\mathbf{E}$ is the vertex of a colimit with its base in the image of $\mathcal{Y}_{\mathbf{E}}$.

Let $\mathbf{e}\colon \mathbf{E}_1 \to \mathbf{E}_2$ be a morphism of limit sketches and $G_{\mathbf{e}}\colon \mathit{Real}(\mathbf{E}_2) \to \mathit{Real}(\mathbf{E}_1)$ the precomposition with $\mathbf{e}$. A fundamental result due to Ehresmann states that there is an adjunction, that will be called the adjunction *associated with* $\mathbf{e}$:

$$\begin{tikzcd} \mathit{Real}(\mathbf{E}_1) \arrow[r, bend left, "F_{\mathbf{e}}"] \arrow[r, phantom, "\perp"] & \mathit{Real}(\mathbf{E}_2) \arrow[l, bend left, "G_{\mathbf{e}}"] \end{tikzcd}$$

Moreover, the functor $F_{\mathbf{e}}$ *(contravariantly) extends* $\mathbf{e}$ via the Yoneda contravariant realizations, in the sense that there is an isomorphism:

$$F_{\mathbf{e}} \circ \mathcal{Y}_{\mathbf{E}_1} \cong \mathcal{Y}_{\mathbf{E}_2} \circ \mathbf{e}\ .$$

Our definition of forgetful and free functors relies on this adjunction.

Definition 4.3 A *forgetful* functor is a functor of the form $G = - \circ \mathbf{e}\colon \mathit{Real}(\mathbf{E}_2) \to \mathit{Real}(\mathbf{E}_1)$ for a morphism of limit sketches $\mathbf{e}\colon \mathbf{E}_1 \to \mathbf{E}_2$. A *free* functor is a left adjoint to a forgetful functor (as every adjoint functor, it is unique up to a natural isomorphism).

Remark 4.4 It is easy to describe the forgetful functor $G_{\mathbf{e}}$, using its definition: for each realization R_2 of $\mathbf{E}_2$, the realization $R_1 = G_{\mathbf{e}}(R_2)$ of $\mathbf{E}_1$ is such that $R_1(E) = R_2(\mathbf{e}(E))$ for every point or arrow E in $\mathbf{E}_1$. It is also quite easy to describe the left adjoint functor $F_{\mathbf{e}}$, using the fact that $F_{\mathbf{e}}$ extends $\mathbf{e}$: let R_1 be a realization of $\mathbf{E}_1$ and $R_2 = F_{\mathbf{e}}(R_1)$, if $R_1 = \mathcal{Y}_{\mathbf{E}_1}(E)$ for some point E in $\mathbf{E}_1$ then $R_2 = \mathcal{Y}_{\mathbf{E}_2}(\mathbf{e}(E))$, and the general case follows thanks to the density of $\mathcal{Y}_{\mathbf{E}_1}$ and to the fact that $F_{\mathbf{e}}$ preserves colimits (since it is a left adjoint).

4.2 A limit sketch for equational theories

The construction of various "sketches of categories" and "sketches of sketches" is a classical exercise about sketches [3, 4, 1]. Here we build (a significant part of) a limit sketch $\mathbf{E}_{eq}$ for the category $\mathbf{T}_{eq}$ of equational theories, i.e., for the category of categories with chosen products.

• Graphs

Let us start from the following limit sketch $\mathbf{E}_{gr}$ for the category of graphs, simply made of two points `Type` and `Term` (for types and terms) and two arrows `dom` and `codom` (for domain and codomain):

$$\begin{tikzcd} \mathtt{Type} & \mathtt{Term} \arrow[l, shift right, "\mathtt{dom}"'] \arrow[l, shift left, "\mathtt{codom}"] \end{tikzcd}$$

The image of $\mathbf{E}_{gr}$ by its Yoneda contravariant realization is the following diagram of graphs:

$$\boxed{X} \quad \begin{array}{c} \xrightarrow{X \mapsto X} \\ \xrightarrow[X \mapsto Y]{} \end{array} \quad \boxed{X \xrightarrow{f} Y}$$

• Categories

First, let us build a limit sketch $\mathbf{E}'_{gr}$ by adding to $\mathbf{E}_{gr}$ a point `Cons` for consecutive terms, as the vertex of the following potential limit, where the projections `fst` and `snd` stand for the first and second component of a pair of consecutive terms and `middle` stands for its "middle type" (codomain of the first component and domain of the second one):

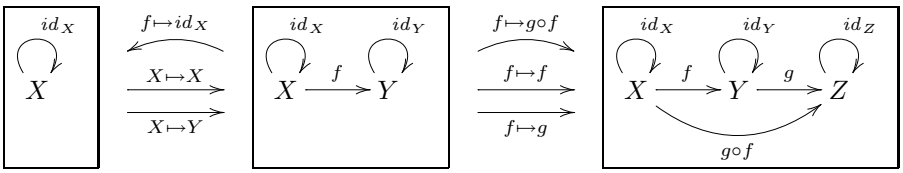
$$\begin{tikzcd} & \mathtt{Cons} \arrow[dl, "\mathtt{fst}"'] \arrow[d, "\mathtt{middle}"] \arrow[dr, "\mathtt{snd}"] & \\ \mathtt{Term} \arrow[r, "\mathtt{codom}"'] & \mathtt{Type} & \mathtt{Term} \arrow[l, "\mathtt{dom}"] \end{tikzcd}$$

hence the equations $\mathtt{codom} \circ \mathtt{fst} = \mathtt{middle}$, $\mathtt{dom} \circ \mathtt{snd} = \mathtt{middle}$ hold, so that `middle` may be omitted. Adding such a potential limit, with new vertex and projections over a known base, is an equivalence of limit sketches: the realizations of $\mathbf{E}'_{gr}$ are still the graphs. Now, a limit sketch $\mathbf{E}_{cat}$ for categories is obtained by adding to $\mathbf{E}'_{gr}$ two arrows `selid` for the selection of identities and `comp` for the composition and several equations:

$$\begin{tikzcd} \mathtt{Type} \arrow[r, bend left, "\mathtt{selid}"] & \mathtt{Term} \arrow[l, shift right, "\mathtt{dom}"'] \arrow[l, shift left, "\mathtt{codom}"] & \mathtt{Cons} \arrow[l, bend right, "\mathtt{comp}"'] \arrow[l, shift right, "\mathtt{fst}"'] \arrow[l, shift left, "\mathtt{snd}"] \end{tikzcd}$$

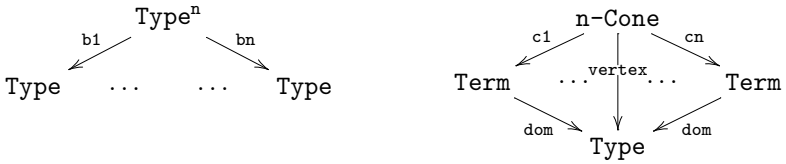
$$\mathtt{dom} \circ \mathtt{selid} = \mathtt{id}_{\mathtt{Type}}\,,\ \mathtt{codom} \circ \mathtt{selid} = \mathtt{id}_{\mathtt{Type}}\,,\ \mathtt{dom} \circ \mathtt{comp} = \mathtt{dom} \circ \mathtt{fst}\,,\ \mathtt{codom} \circ \mathtt{comp} = \mathtt{codom} \circ \mathtt{snd}\,,$$

and with the equations which ensure that the three axioms of categories are satisfied. The image of this part of $\mathbf{E}_{cat}$ by its Yoneda contravariant realization is the following diagram of categories:



• **Theories**

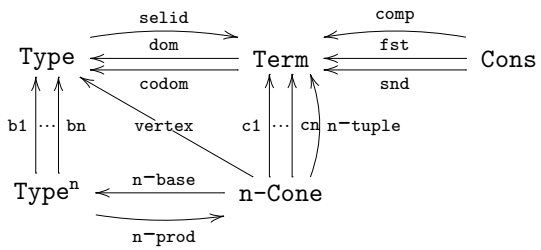
We build a limit sketch $\mathbf{E}'_{cat}$ by adding to $\mathbf{E}_{cat}$ for each $n \in \mathbb{N}$ the following potential limits, with vertex $\texttt{Type}^{\texttt{n}}$ for n-tuples of types and `n-Cone` for n-ary discrete cones:



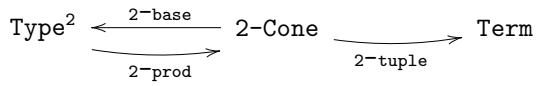
The arrow `vertex` may be omitted when $n > 0$. When $n = 0$, the potential limits mean that $\texttt{Type}^0$ is a unit type (also denoted `Unit`) and `0-Cone` is isomorphic to `Type`. We also add the tuple $\texttt{n-base} = \langle \texttt{codom} \circ \texttt{c1}, \ldots, \texttt{codom} \circ \texttt{cn} \rangle$ which maps each cone to its base:

$$\texttt{Type}^{\texttt{n}} \xleftarrow{\texttt{n-base}} \texttt{n-Cone}$$

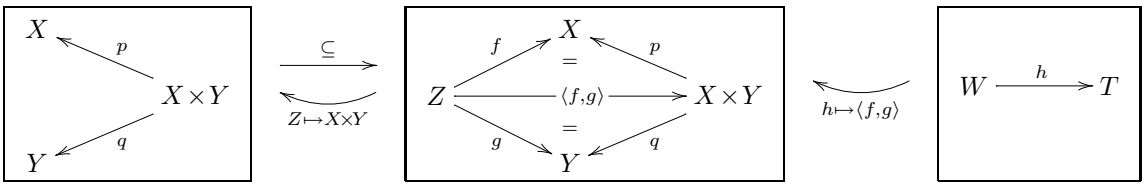
The realizations of $\mathbf{E}'_{cat}$ are still the categories. Now, a limit sketch $\mathbf{E}_{th}$ for equational theories is obtained by adding to $\mathbf{E}'_{cat}$ the following features, for each $n \in \mathbb{N}$. First an arrow $\texttt{n-prod}\colon \texttt{Type}^{\texttt{n}} \to \texttt{n-Cone}$ together with the equation $\texttt{n-base} \circ \texttt{n-prod} = \texttt{id}_{\texttt{Type}^{\texttt{n}}}$, for building the product cone of each family of n types. Then for building tuples, an arrow $\texttt{n-tuple}\colon \texttt{n-Cone} \to \texttt{Term}$ together with the equations $\texttt{dom} \circ \texttt{n-tuple} = \texttt{vertex}$ and $\texttt{codom} \circ \texttt{n-tuple} = \texttt{vertex} \circ \texttt{n-prod} \circ \texttt{n-base}$ and with several additional equations for ensuring that the universal property of a product is satisfied. So, here is a relevant part of this limit sketch $\mathbf{E}_{th}$ for theories (equations are omitted, and only one arity n is represented):



Let us focus on the following part of $\mathbf{E}_{th}$:



and its image by the Yoneda contravariant realization (only presentations are given):



4.3 The parameterization process is a free functor

• Parameterized theories

A limit sketch $\mathbf{E}_A$ for parameterized theories is obtained by adding to $\mathbf{E}_{th}$ an arrow $\mathtt{A}\colon \mathtt{Unit} \to \mathtt{Type}$.

• Decorated theories

A limit sketch $\mathbf{E}_{dec}$ for decorated theories comes with a morphism $\mathbf{e}_{undec}\colon \mathbf{E}_{dec} \to \mathbf{E}_{th}$ which forgets about the decorations ("*undec*" for "undecoration"). Here are two slightly different choices, the first one is simpler but the second one better reflects the idea of decoration.

A limit sketch $\mathbf{E}_{dec}$ for decorated theories is made of two related copies of $\mathbf{E}_{th}$: one copy $\mathbf{E}.\mathtt{p}$ for the pure features and another copy $\mathbf{E}.\mathtt{g}$ for the general features, together with a monomorphic transition arrow $\mathtt{t}_{\mathtt{E}}\colon \mathtt{E.p} \to \mathtt{E.g}$ for each point $\mathtt{E}$ in $\mathbf{E}_{th}$ with $\mathtt{t}_{\mathtt{Type}}$ an identity and with the transition equations $\mathtt{t}_{\mathtt{E}'} \circ \mathtt{e.p} = \mathtt{e.g} \circ \mathtt{t}_{\mathtt{E}}$ for each arrow $\mathtt{e}\colon \mathtt{E} \to \mathtt{E}'$ in $\mathbf{E}_{th}$. The morphism $\mathbf{e}_{undec}$ maps both copies $\mathbf{E}.\mathtt{p}$ and $\mathbf{E}.\mathtt{g}$ to $\mathbf{E}$.

Another limit sketch for decorated theories, still denoted $\mathbf{E}_{dec}$, is the *sketch of elements* (similar to the more usual *category of elements*) of a model Δ of $\mathbf{E}_{th}$ with values in $\mathbf{T}_{eq}$, then the morphism $\mathbf{e}_{undec}$ is provided by the construction. This model Δ formalizes the fact that identities and projections are always pure while the composition or pairing of pure terms is pure. Precisely, the theory $\Delta(\mathtt{Type})$ is generated by one type D and the theory $\Delta(\mathtt{Term})$ by two types p, g (for "pure" and "general", respectively) and a monomorphism $p \to g$ (for "every pure term can be seen as a general term"). As for the functors, $\Delta(\mathtt{selid})$ maps D to p, $\Delta(\mathtt{n\text{-}prod})$ maps $\langle D, \ldots, D\rangle$ to p, while $\Delta(\mathtt{comp})$ maps $\langle p, p\rangle$ to p and $\langle p, g\rangle, \langle g, p\rangle, \langle g, g\rangle$ to g and $\Delta(\mathtt{n\text{-}tuple})$ maps $\langle p, \ldots, p\rangle$ to p and everything else to g. The resulting sketch of elements $\mathbf{E}_{dec}$ is made of one point $\mathtt{Type.D}$ over the point $\mathtt{Type}$ of $\mathbf{E}_{th}$, two points $\mathtt{Term.p}$ and $\mathtt{Term.g}$ over the point $\mathtt{Term}$ of $\mathbf{E}_{th}$, four points over $\mathtt{Cons}$, 2^n over $\mathtt{Type}^{\mathtt{n}}$ and $\mathtt{n\text{-}Cone}$, a monomorphic arrow $\mathtt{Term.p} \to \mathtt{Term.g}$, and so on.

• From decorated theories to parameterized theories

Let us consider the functors $F_{exp}\colon \mathbf{T}_{dec} \to \mathbf{T}_A$ and $G_{exp}\colon \mathbf{T}_A \to \mathbf{T}_{dec}$ from section 3.1. We can now prove that G_{exp} is a forgetful functor and F_{exp} is its associated free functor, in the sense of definition 4.3.

Theorem 4.5 *There is a morphism of limit sketches $\mathbf{e}_{exp}\colon \mathbf{E}_{dec} \to \mathbf{E}_A$ such that the associated adjunction is $F_{exp} \dashv G_{exp}$ from section 3.1.*

Proof. In section 3.1 the functor F_{exp} has been defined on $\mathcal{Y}(\mathbf{E}_{dec})$ (see figure 2). Since F_{exp} extends $\mathbf{e}_{exp}$ via the Yoneda contravariant realizations, this provides the definition of a unique morphism $\mathbf{e}_{exp}$ with associated left adjoint F_{exp}. For instance, the point $\mathtt{Term.p}$ is mapped to $\mathtt{Term}$ and the point $\mathtt{Term.g}$ to the point of $\mathbf{E}_A$ characterized by the fact that its image by Yoneda is presented by X, Y and $f'\colon A \times X \to Y$. Then it is easy to check that the precomposition with $\mathbf{e}_{exp}$ is the functor G_{exp}. $\square$

• **A span of limit sketches**

Altogether, the following span of limit sketches provides a framework for the process that starts from an equational theory, choose the pure terms, and forms the corresponding parameterized theory:

$$\mathbf{E}_{eq} \xleftarrow{\mathbf{e}_{undec}} \mathbf{E}_{dec} \xrightarrow{\mathbf{e}_{exp}} \mathbf{E}_{A}$$

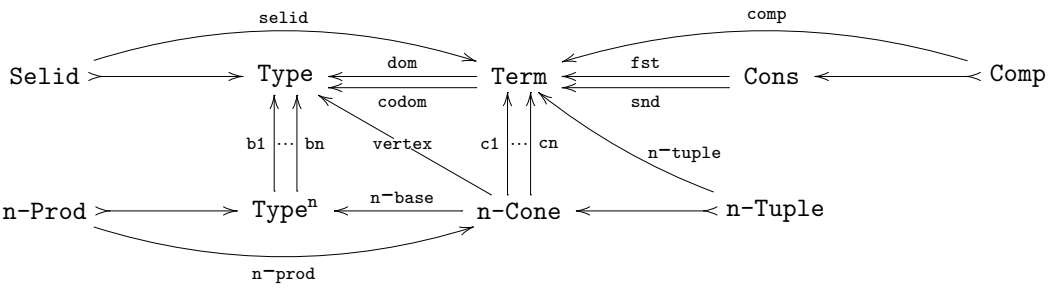
4.4 A limit sketch for equational specifications

In this section we build a limit sketch $\mathbf{E}_{sp}$ for equational specifications from the limit sketch $\mathbf{E}_{th}$ for theories, thus providing another point of view on the elementary specifications in section 2.1. This construction can be seen as an illustration of the factorization theorem in [9]. A direct detailed construction of a limit sketch for equational specifications can be found in appendix A.2.

In the part of $\mathbf{E}_{th}$ shown in section 4.2 there are four arrows that are neither in $\mathbf{E}_{gr}$ nor projections in a potential limit: `selid`, `comp`, `n-prod`, `n-tuple`. These arrows stand for features that are always defined in a theory but only partially defined in a specification. So, $\mathbf{E}_{sp}$ is obtained by replacing each of these arrows $\mathtt{e}\colon \mathtt{E}_1 \to \mathtt{E}_2$ by a span:

$$\mathtt{E}_1 \xleftarrow{\mathtt{e}'_1} \mathtt{E}'_1 \xrightarrow{\mathtt{e}'} \mathtt{E}_2$$

where the arrow "$\rightarrowtail$" stands for a potential monomorphism (which can be expressed as a potential limit). So, here is (a relevant part of) $\mathbf{E}_{sp}$:



The elementary specifications from section 2.1 are the images by the Yoneda contravariant realization of the points in $\mathbf{E}_{sp}$ which are not vertices of potential limits, namely: `Type`, `Term`, `Selid`, `Comp`, `n-Prod`, `n-Tuple`: our notations are such that $\Sigma_{\mathtt{E}} = \mathcal{Y}(\mathtt{E})$ for each of these points $\mathtt{E}$.

5 Conclusion

This paper provides a neat categorical formalization for the parameterization process in Kenzo and EAT. 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] Laurent Coppey, Christian Lair. Leçons de Théorie des Esquisses. Diagrammes 12 (1984).

[4] Laurent Coppey, Christian Lair. Leçons de Théorie des Esquisses. Diagrammes 19 (1988).

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

[6] 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).

[7] 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`.

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

[9] Dominique Duval. Diagrammatic Specifications. Mathematical Structures in Computer Science 13, p. 857–890 (2003).

[10] Dominique Duval. Diagrammatic Inference. arXiv:0710.1208v1 (2007).

[11] 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).

[12] Peter Gabriel, Michel Zisman. Calculus of Fractions and Homotopy Theory. Springer (1967).

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

[14] 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).

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

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

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

[18] Michael Makkai. Generalized Sketches as a Framework for Completeness Theorems (I). Journal of Pure and Applied Algebra 115, p. 49–79 (1997).

[19] 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.

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

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

A Diagrammatic logics

In this paper we have introduced the equational logic in a categorical way, considering equational theories as categories with chosen finite products. An equational theory can be presented by an equational specification, which means that this specification generates the theory. In section 4 we have outlined the construction first of a limit sketch for the equational theories and then of a limit sketch for the equational specifications. This appendix provides a detailed description of these limit sketches, with slightly more subtle definitions of equational theories and specifications, which are better suited for formalizing equational proofs. In addition, as often in the framework of algebraic specifications (as for instance in [16] and in [7]) we consider first the specifications, then we get the theories by using the inference rules of the equational logic. Finally, the parameterization process is presented from this point of view. The framework of diagrammatic logics [9, 10] is well suited for dealing with "usual" logics like the equational logic as well as with more "unusual" ones like the decorated equational logic, and also for dealing with various morphisms of logics, for instance we will see that the parameterizing functor stems from a morphism of logics. This appendix can be seen as an introduction to diagrammatic logics, based on section 4.1 about limit sketches.

A.1 Equational logic, revisited

As in section 2, instead of the algebraic definition of equational specifications given for instance in [16], we define equational specifications from finite product sketches. In the main text we have defined equational specifications exactly as finite product sketches, so that the equations become equalities of arrows: this is all right for defining models but this makes every proof trivial. In this appendix we give a more subtle definition of equational specifications as finite product sketches *with equations*; then definition 2.4 is easily recovered by mapping equations to equalities. In spite of this minor difference, we use the same notations ($\mathbf{S}_{eq}$, $\mathbf{T}_{eq}$) in this appendix as in the main part of the paper.

Definition A.1 An *equational specification* is a limit sketch (definition 4.1) where all the potential limits are potential finite products, together with a set of pairs of parallel terms called the *equations* and denoted $t_1 \equiv t_2$. A morphism of equational specifications is a morphism of limit sketches which preserves the equations. This yields the category of equational specifications $\mathbf{S}_{eq}$.

Similarly, in this appendix, the equations in an equational theory need not be equalities. Roughly speaking, an equational theory is an equational specification where the equations form an equivalence relation and all the potential features become real up to equations. For this reason the relation $\equiv$ is called a *congruence*. So, an equational theory is not a category, it is only a bicategory (the congruence defines its 2-cells), but it becomes a category with chosen finite products, as in definition 2.1, as soon as both members in each equation get identified. Conversely, every category with chosen finite products can be seen as an equational theory where the equations are the equalities.

Definition A.2 An *equational theory* is an equational specification where each type has a potential identity, each pair of consecutive terms has a potential composition, each list of types has a potential product, each list of terms with a common domain has a potential tuple, and in addition the equations form a *congruence*, which means that the relation $\equiv$ is an equivalence relation compatible with composition and that the usual axioms for categories with products are satisfied up to $\equiv$. The *category of equational theories* $\mathbf{T}_{eq}$ is the full subcategory of $\mathbf{S}_{eq}$ with objects the equational theories.

It may be noted that the inclusion of $\mathbf{T}_{eq}$ in $\mathbf{S}_{eq}$ is faithful. In fact, for products and tuples, only the arities $n = 2$ and $n = 0$ will be considered: the general case may easily be guessed, or alternatively one can use the fact that all finite products may be recovered from the binary products and a terminal object. A set of *inference rules* for the equational logic, for generating an equational theory from an equational specification, is presented in figure 3. When there is no ambiguity, we often omit "equational" and "potential".

name	rules
composition	$\frac{f{:}X\to Y \quad g{:}Y\to Z}{g\circ f{:}X\to Z}$
identity	$\frac{X}{id_X{:}X\to X}$
equivalence	$\frac{f}{f\equiv f} \qquad \frac{f\equiv g}{g\equiv f} \qquad \frac{f\equiv g \quad g\equiv h}{f\equiv h}$
substitution	$\frac{f{:}X\to Y \quad g_1\equiv g_2{:}Y\to Z}{g_1\circ f\equiv g_2\circ f{:}X\to Z}$
replacement	$\frac{f_1\equiv f_2{:}X\to Y \quad g{:}Y\to Z}{g\circ f_1\equiv g\circ f_2{:}X\to Z}$
associativity	$\frac{f{:}X\to Y \quad g{:}Y\to Z \quad h{:}Z\to W}{(h\circ g)\circ f\equiv h\circ(g\circ f)}$
unit rules	$\frac{f{:}X\to Y}{f\circ id_X\equiv f} \qquad \frac{f{:}X\to Y}{id_Y\circ f\equiv f}$
binary product	$\frac{Y_1 \quad Y_2}{Y_1\times Y_2} \qquad \frac{Y_1 \quad Y_2}{pr_1{:}Y_1\times Y_2\to Y_1} \qquad \frac{Y_1 \quad Y_2}{pr_2{:}Y_1\times Y_2\to Y_2}$
pairing	$\frac{f_1{:}X\to Y_1 \quad f_2{:}X\to Y_2}{\langle f_1,f_2\rangle{:}X\to Y_1\times Y_2} \qquad \frac{f_1{:}X\to Y_1 \quad f_2{:}X\to Y_2}{pr_1\circ\langle f_1,f_2\rangle\equiv f_1} \qquad \frac{f_1{:}X\to Y_1 \quad f_2{:}X\to Y_2}{pr_2\circ\langle f_1,f_2\rangle\equiv f_2}$
pairing uniqueness	$\frac{f_1{:}X\to Y_1 \quad f_2{:}X\to Y_2 \quad f{:}X\to Y_1\times Y_2 \quad pr_1\circ f\equiv f_1 \quad pr_2\circ f\equiv f_2}{\langle f_1,f_2\rangle\equiv f}$
terminal type	$\frac{}{1}$
collapsing	$\frac{X}{\langle\,\rangle_X{:}X\to 1}$
collapsing uniqueness	$\frac{f{:}X\to 1}{\langle\,\rangle_X\equiv f}$

Figure 3: The rules for equational logic

The *models* of a specification Σ with values in a theory Θ are defined as the morphisms of specifications from Σ to Θ. In addition, the *morphisms of models* of Σ with values in Θ can be defined in the usual natural way, so that there is a *category of models* $Mod(\Sigma,\Theta)$ of Σ with values in Θ. The *category of set-valued models* of Σ is the category of models of Σ with values in the category of sets seen as an equational theory, with the cartesian products as potential products and the equalities of functions as equations.

Each equational specification in the algebraic sense Sp gives rise to an equational specification Σ: each sort of Sp becomes a type of Σ, each list of sorts $X_1,\ldots,X_n$ of Sp becomes a type $X_1\times\ldots\times X_n$ of Σ, each operation or term $f{:}\,X_1\ldots X_n\to Y$ of Sp becomes a term $f{:}\,X_1\times\ldots\times X_n\to Y$ of Σ, and each equation $f_1\equiv f_2$ of Sp becomes an equation $f_1\equiv f_2$ of Σ; for this purpose, both terms f_1 and f_2 in Sp must be considered as terms in all the variables that appear in f_1 or in f_2, as explained for instance in [1]. Then, the category of models of Sp in the algebraic sense is isomorphic to the category of set-valued models of Σ.

Example A.3 The equational specification Σ_{sgp} for semigroups can be represented as a graph with an equation:

$$\Sigma_{sgp} = \boxed{G^2 \xrightarrow{prd} G \qquad prd\circ\langle x, prd\circ\langle y,z\rangle\rangle \equiv prd\circ\langle prd\circ\langle x,y\rangle, z\rangle}$$

However, many details are implicit in this illustration. More precisely, the equational specification Σ_{sgp} can be built as follows. First an equational specification Σ_{mgm} for magmas (a *magma* is simply a set with a binary operation) is made of two types G and G^2, three terms $u,v,prd : G^2\to G$ and one potential

product $G \xleftarrow{u} G^2 \xrightarrow{v} G$. Then, a second equational specification Σ'_{mgm} is obtained by adding to Σ_{mgm} a type G^3, terms $x : G^3 \to G$, $w : G^3 \to G^2$, a potential product $G \xleftarrow{x} G^3 \xrightarrow{w} G^2$, and also the terms $f_1 = prd \circ \langle x, prd \circ w\rangle : G^3 \to G$ and $f_2 = prd \circ \langle prd \circ \langle x, u \circ w\rangle, v \circ w\rangle : G^3 \to G$. We also add $y = u \circ w$ and $z = v \circ w$, and the equations $w \equiv \langle y, z\rangle$, $f_1 \equiv prd \circ \langle x, prd \circ \langle y, z\rangle\rangle$ and $f_2 \equiv prd \circ \langle prd \circ \langle x, y\rangle, z\rangle$. Then Σ'_{mgm} is equivalent to Σ_{mgm}. Finally Σ_{sgp} is made of Σ'_{mgm} with the equation $f_1 \equiv f_2$, or equivalently with the equation $prd \circ \langle x, prd \circ \langle y, z\rangle\rangle \equiv prd \circ \langle prd \circ \langle x, y\rangle, z\rangle$.

In sections A.2 and A.3, these notions are embedded in the definition of a *diagrammatic equational logic* $\mathcal{L}_{eq}$ [10]. This means that we build a limit sketch $\mathbf{E}_{eq,S}$ for $\mathbf{S}_{eq}$, a limit sketch $\mathbf{E}_{eq,T}$ for $\mathbf{T}_{eq}$, and a morphism of limit sketches $\mathbf{e}_{eq} : \mathbf{E}_{eq,S} \to \mathbf{E}_{eq,T}$ such that the inclusion functor $G_{eq} : \mathbf{T}_{eq} \to \mathbf{S}_{eq}$ is the precomposition with $\mathbf{e}_{eq}$ and its left adjoint $F_{eq} : \mathbf{S}_{eq} \to \mathbf{T}_{eq}$ (as we saw in section 4.1) maps each equational specification to its generated theory.

A.2 Equational specifications

In this section we provide a detailed construction of a limit sketch $\mathbf{E}_{eq,S}$ for the category $\mathbf{S}_{eq}$ of equational specifications; except for equations, we will get essentially the same sketch as $\mathbf{E}_{sp}$ in section 4.3. We begin with the sketch $\mathbf{E}_{gr}$ for graphs:

$$\begin{tikzcd} \mathtt{Type} & \mathtt{Term} \arrow[l, shift right, "\mathtt{dom}"'] \arrow[l, shift left, "\mathtt{codom}"] \end{tikzcd}$$

Then, we extend $\mathbf{E}_{gr}$ for each kind of potential features; each limit sketch is followed by its image by its Yoneda contravariant realization. Finally, by glueing together these extensions of $\mathbf{E}_{gr}$ (by a colimit of limit sketches) we get the limit sketch $\mathbf{E}_{eq,S}$.

• Composites

A sketch $\mathbf{E}_{gr\_comp}$ for graphs with potential composites is obtained by extending $\mathbf{E}_{gr}$ as follows, with its potential limit and equalities:

$$\begin{tikzcd} & & \mathtt{Comp} \arrow[dl, "\mathtt{comp}"'] \arrow[d, tail, "\mathtt{i}"] \\ \mathtt{Type} & \mathtt{Term} \arrow[l, shift right, "\mathtt{dom}"'] \arrow[l, shift left, "\mathtt{codom}"] & \mathtt{Cons} \arrow[l, shift right, "\mathtt{fst}"'] \arrow[l, shift left, "\mathtt{snd}"] \end{tikzcd}$$

$$\begin{tikzcd} & \mathtt{Cons} \arrow[dl, "\mathtt{fst}"'] \arrow[d, "\mathtt{middle}"] \arrow[dr, "\mathtt{snd}"] & \\ \mathtt{Term} \arrow[r, "\mathtt{codom}"'] & \mathtt{Type} & \mathtt{Term} \arrow[l, "\mathtt{dom}"] \end{tikzcd}$$

$$\mathtt{dom} \circ \mathtt{comp} = \mathtt{dom} \circ \mathtt{fst} \circ \mathtt{i}$$

$$\mathtt{codom} \circ \mathtt{comp} = \mathtt{codom} \circ \mathtt{snd} \circ \mathtt{i}$$

The point `Comp` stands for the set of composable terms, the potential mono `i` for the inclusion, and the arrow `comp` for the composition of composable terms. The image of $\mathbf{E}_{gr\_comp}$ by its Yoneda contravariant realization is the following morphism of realizations of $\mathbf{E}_{gr\_comp}$; as required, the image of the mono `i` is an epimorphism.

$$\begin{tikzcd} & & & \boxed{X \xrightarrow{f} Y \xrightarrow{g} Z,\ X \xrightarrow{g \circ f} Z} \\ \boxed{X} \arrow[r, shift left, "X \mapsto X"] \arrow[r, shift right, "X \mapsto Y"'] & \boxed{X \xrightarrow{f} Y} \arrow[r, shift left, "f \mapsto f"] \arrow[r, shift right, "f \mapsto g"'] & \boxed{X \xrightarrow{f} Y \xrightarrow{g} Z} \arrow[ur, "f \mapsto g \circ f"] & \end{tikzcd}$$

$$\boxed{X \xrightarrow{f} Y \xrightarrow{g} Z,\ X \xrightarrow{g \circ f} Z} \xleftarrow{\cup\!\mathrm{I}} \boxed{X \xrightarrow{f} Y \xrightarrow{g} Z}$$

• **Identities**

A sketch $\mathbf{E}_{gr\_id}$ for graphs with potential identities is obtained by extending $\mathbf{E}_{gr}$ as follows:

$$\begin{tikzcd}
\mathtt{Selid} \arrow[d, tail, "\mathtt{i0}"'] \arrow[dr, "\mathtt{selid}"] & \\
\mathtt{Type} & \mathtt{Term} \arrow[l, shift right, "\mathtt{dom}"'] \arrow[l, shift left, "\mathtt{codom}"]
\end{tikzcd}$$

$$\mathtt{dom} \circ \mathtt{selid} = \mathtt{id}_{\mathtt{Selid}}$$
$$\mathtt{codom} \circ \mathtt{selid} = \mathtt{id}_{\mathtt{Selid}}$$

The point `Selid` stands for the set of types with a selected identity, the potential mono `i0` for the inclusion, and the arrow `selid` for the selection of the identities.

$$\begin{tikzcd}
\boxed{{}^{id}X \circlearrowright X} & \\
\boxed{X} \arrow[u, two heads, "\cup\!\mathsf{I}"] \arrow[r, shift left, "X \mapsto X"] \arrow[r, shift right, "X \mapsto Y"'] & \boxed{X \xrightarrow{f} Y} \arrow[ul, "f \mapsto {}^{id}X"']
\end{tikzcd}$$

• **Binary products**

A sketch $\mathbf{E}_{gr\_prod}$ for graphs with potential binary products is obtained by extending $\mathbf{E}_{gr}$ as follows:

$$\begin{tikzcd}
 & & \mathtt{2\text{-}Prod} \arrow[dl, "\mathtt{2\text{-}prod}"'] \arrow[d, tail, "\mathtt{j}"] & \\
\mathtt{Type} & \mathtt{Term} \arrow[l, shift right, "\mathtt{dom}"'] \arrow[l, shift left, "\mathtt{codom}"] & \mathtt{2\text{-}Cone} \arrow[l, shift right, "\mathtt{c1}"'] \arrow[l, shift left, "\mathtt{c2}"] \arrow[r, "\mathtt{2\text{-}base}"] & \mathtt{Type}^2 \arrow[ull, shift right, "\mathtt{b1}"'] \arrow[ull, shift left, "\mathtt{b2}"]
\end{tikzcd}$$

$$\begin{tikzcd}
 & \mathtt{Type}^2 \arrow[dl, "\mathtt{b1}"'] \arrow[dr, "\mathtt{b2}"] & \\
\mathtt{Type} & & \mathtt{Type}
\end{tikzcd}$$

$$\begin{tikzcd}
 & \mathtt{2\text{-}Cone} \arrow[dl, "\mathtt{c1}"'] \arrow[d, "\mathtt{vertex}"] \arrow[dr, "\mathtt{c2}"] & \\
\mathtt{Term} \arrow[r, "\mathtt{dom}"'] & \mathtt{Type} & \mathtt{Term} \arrow[l, "\mathtt{dom}"]
\end{tikzcd}$$

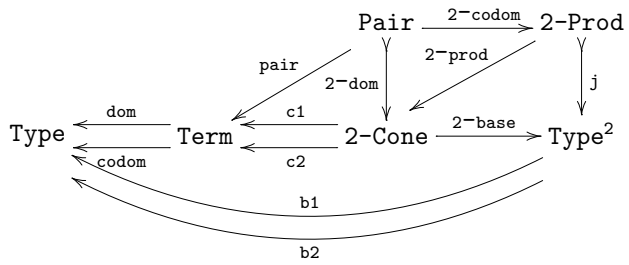
$$\mathtt{b1} \circ \mathtt{2\text{-}base} = \mathtt{codom} \circ \mathtt{c1}$$
$$\mathtt{b2} \circ \mathtt{2\text{-}base} = \mathtt{codom} \circ \mathtt{c2}$$

The point `2-Prod` stands for the set of binary products, the mono `j` for the inclusion, and the arrow `2-prod` for the operation which maps a binary product to its underlying binary cone.

$$\begin{tikzcd}
 & & & \boxed{\begin{tikzcd} & Y_1 \times Y_2 \arrow[dl, "pr_1"'] \arrow[dr, "pr_2"] & \\ Y_1 & & Y_2 \end{tikzcd}} \\
\boxed{X} \arrow[r, shift left, "X \mapsto X"] \arrow[r, shift right, "X \mapsto Y"'] & \boxed{X \xrightarrow{f} Y} \arrow[r, shift left, "f \mapsto f_1"] \arrow[r, shift right, "f \mapsto f_2"'] & \boxed{\begin{tikzcd} & X \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & \\ Y_1 & & Y_2 \end{tikzcd}} & \boxed{\begin{tikzcd} Y_1 & & Y_2 \end{tikzcd}} \arrow[l, "\supseteq"'] \arrow[u, two heads, "\cup\!\mathsf{I}"] \arrow[ul, "X \mapsto Y_1 \times Y_2", "f_i \mapsto pr_i"']
\end{tikzcd}$$

• **Pairing**

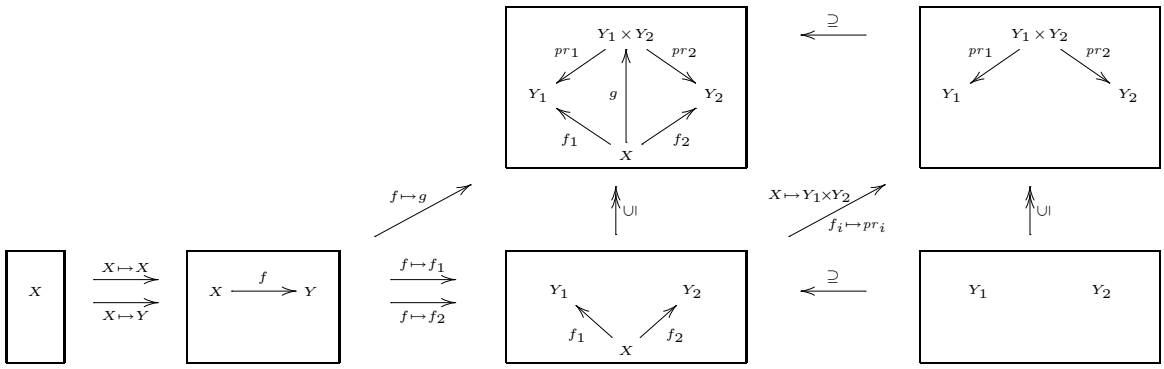
Now, a sketch $\mathbf{E}_{gr\_pair}$ for graphs with potential binary products and with potential pairings (or 2-tuples) is obtained by extending $\mathbf{E}_{gr\_prod}$ as follows:



$\mathtt{j} \circ \mathtt{2\text{-}codom} = \mathtt{2\text{-}base} \circ \mathtt{2\text{-}dom}$

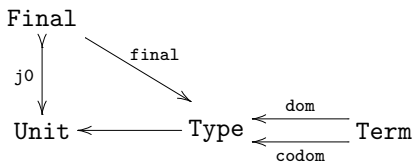
$\mathtt{dom} \circ \mathtt{pair} = \mathtt{vertex} \circ \mathtt{2\text{-}dom}$

$\mathtt{codom} \circ \mathtt{pair} = \mathtt{vertex} \circ \mathtt{2\text{-}prod} \circ \mathtt{2\text{-}codom}$



• **Terminal type**

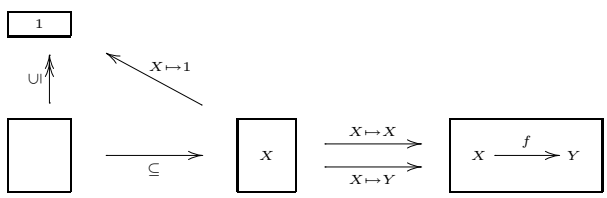
A terminal (or final) type is a nullary product, and a nullary cone is simply a type (its vertex). With this correspondence in mind, the construction of $\mathbf{E}_{gr\_fin}$ below is similar to the construction of $\mathbf{E}_{gr\_prod}$ above, with $\mathtt{0\text{-}Cone} = \mathtt{Type}$, $\mathtt{Type}^0 = \mathtt{Unit}$ and $\mathtt{0\text{-}base} : \mathtt{Type} \to \mathtt{Unit}$, where $\mathtt{Unit}$ is the vertex of a potential limit cone with an empty base, and in addition $\mathtt{0\text{-}Prod} = \mathtt{Final}$ and $\mathtt{0\text{-}prod} = \mathtt{final}$



Unit

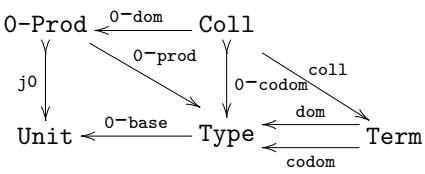
(empty base)

The point $\mathtt{Unit}$ stands for a singleton, the point $\mathtt{Final}$ with the arrow $\mathtt{j0}$ for a set with at most one element, and the arrow $\mathtt{final}$ stands for the selection of the terminal type.



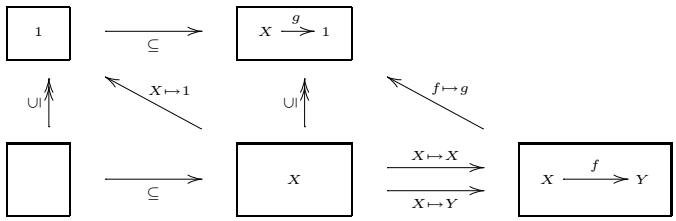
• **Collapsing**

Now, a sketch $\mathbf{E}_{gr\_coll}$ for graphs with a potential terminal type and with potential collapsings (or 0-tuples) is obtained by extending $\mathbf{E}_{gr\_fin}$ as follows:



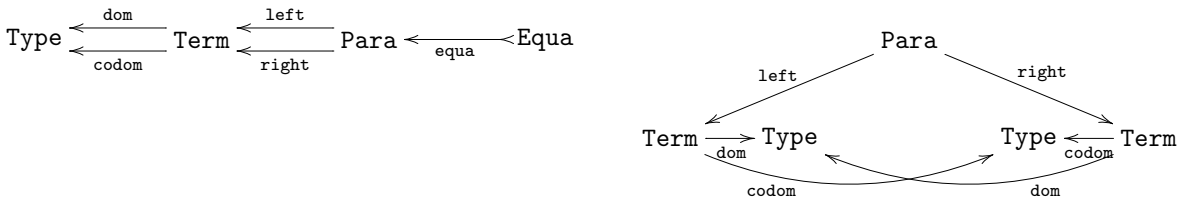
$\texttt{dom} \circ \texttt{coll} = \texttt{0-codom}$

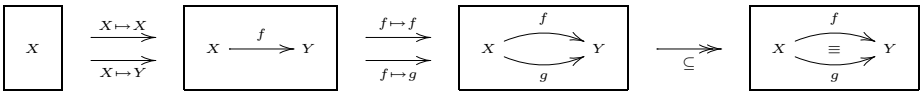
$\texttt{0-prod} \circ \texttt{0-dom}$



• **Equations**

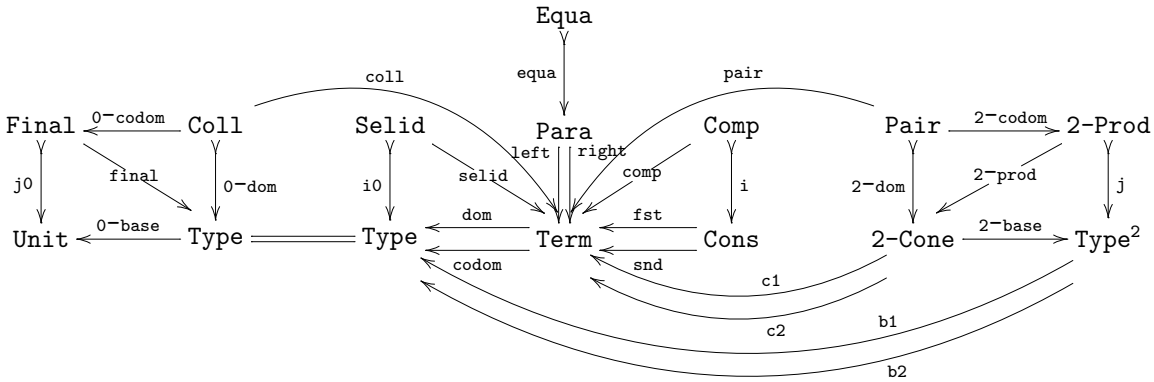
A sketch $\mathbf{E}_{gr\_eq}$ for graphs with equations is obtained by extending $\mathbf{E}_{gr}$ with two points `Para` and `Equa` which stand for the set of pairs of parallel arrows and the set of equations, respectively. The arrows `left` and `right` extract the two terms from a pair of parallel terms. The potential limit establishes that `Para` represents pairs of parallel terms.





• **Equational specifications**

Finally, a sketch $\mathbf{E}_{eq,S}$ for equational specifications is obtained as the colimit of the sketches $\mathbf{E}_{gr\_comp}$, $\mathbf{E}_{gr\_id}$, $\mathbf{E}_{gr\_eq}$, $\mathbf{E}_{gr\_pair}$ and $\mathbf{E}_{gr\_coll}$ over $\mathbf{E}_{gr}$. Here is its underlying graph (with `Type` repeated twice for readablity), in addition it has all the potential limits and all the equalities from the component sketches.



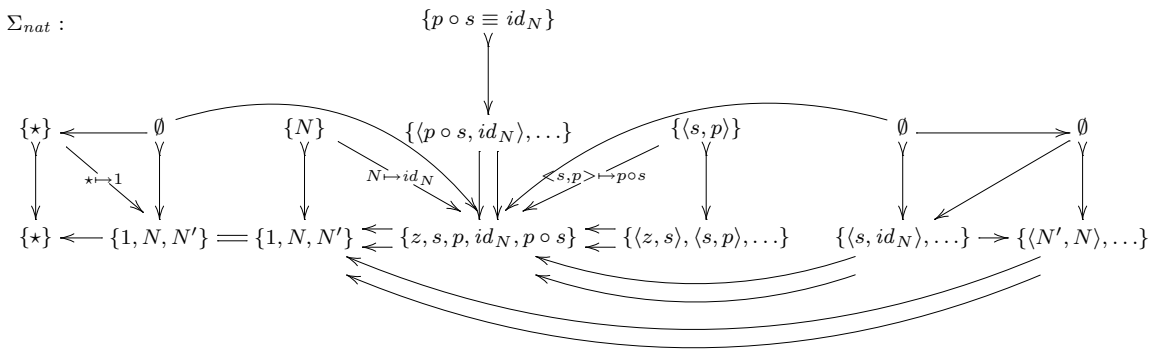
Example A.4 Let us consider the equational specification Σ_{nat}:

$$\Sigma_{nat}: \quad \begin{tikzcd} & N' \arrow[d, bend left, "p"] \\ 1 \arrow[r, "z"] & N \arrow[u, bend left, "s"] \end{tikzcd} \qquad \begin{array}{l} \text{terminal type}: 1 \\ \text{equation: } p \circ s \equiv id_N \end{array}$$

The specification Σ_{nat} has a model "of naturals" M_{nat} which maps the type 1 to a singleton $\{\star\}$, the type N to the set $\mathbb{N}$ of non-negative integers, the type N' to the set $\mathbb{N}^*$ of positive integers, the term z to the constant function $\star \mapsto 0$, and the terms s and p to the functions $x \mapsto x+1$ and $x \mapsto x-1$, respectively. So, the model M_{nat} of Σ_{nat} is illustrated by a diagram in the equational theory of sets, which has the same form as the diagram for Σ_{nat}:

$$M_{nat}: \quad \begin{tikzcd} & \mathbb{N}^* \arrow[d, bend left, "x\mapsto x-1"] \\ \{\star\} \arrow[r, "\star\mapsto 0"] & \mathbb{N} \arrow[u, bend left, "x\mapsto x+1"] \end{tikzcd}$$

Besides, Σ_{nat} can be seen as a set-valued realization of $\mathbf{E}_{eq,S}$:



A.3 Equational logic

In section A.2, a limit sketch $\mathbf{E}_{eq,S}$ for equational specifications has been defined. Now, we describe simultaneously a limit sketch $\mathbf{E}_{eq,T}$ for equational theories and a morphism $\mathbf{e}_{eq} : \mathbf{E}_{eq,S} \to \mathbf{E}_{eq,T}$, by translating at the sketch level the fact that the equational theories are the equational specifications which satisfy the rules of the equational logic, as described in section A.1. It happens that this can be done simply by mapping

some arrows in $\mathbf{E}_{eq,S}$ to identities, thereby some pairs of points in $\mathbf{E}_{eq,S}$ get identified in $\mathbf{E}_{eq,T}$. Let us call *(syntactic) entailment* any arrow t in $\mathbf{E}_{eq,S}$ which will become an identity in $\mathbf{E}_{eq,T}$. Let us look more closely at the rules of the equational logic (figure 3). Each rule may be considered as a *fraction* in the sense of [12], i.e., as a span $r = s/t$ from the hypothesis H to the conclusion C, where the *denominator* t is an entailment, which is illustrated as follows:

$$H \xleftarrow[t]{} H' \xrightarrow{s} C$$

The image of a fraction $r = s/t$ by the Yoneda contravariant realization is a cospan $\rho = \sigma\backslash\tau$ in the category $\mathbf{S}_{eq}$ of equational specifications, where $\tau = \mathcal{Y}(t)$, which will become an idenitity in $\mathbf{T}_{eq}$, is also called an *entailment*; this is illustrated in a similar way:

$$\mathcal{H} \xrightarrow[\tau]{} \mathcal{H}' \xleftarrow{\sigma} \mathcal{C}$$

The numerators of the rules are used for easily composing the rules, but here only the denominators matter, and in addition it may happen that several rules have the same denominator. Let $F_{eq} \dashv G_{eq}$ be the adjunction associated to $\mathbf{e}_{eq}$, then the functor F_{eq} is obtained by mapping the denominators of the rules to identities. A similar approach can be found in [18].

For instance, the reflexivity rule means that "for each term f there is an equation $f \equiv f$", the corresponding fraction is:

$$\texttt{Term} \xleftarrow[t]{} \texttt{Refl} \xrightarrow{s} \texttt{Equa}$$

where `Refl`, s and t are defined by the potential limit:

$$\begin{tikzcd}
 & \texttt{Refl} \arrow[dl, "t"'] \arrow[d] \arrow[dr, "s"] & \\
\texttt{Term} \arrow[r, shift right, "\texttt{right}"'] & \texttt{Para} \arrow[l, shift right, "\texttt{left}"'] & \texttt{Equa} \arrow[l, "\texttt{equa}"]
\end{tikzcd}$$

The specification $\mathcal{Y}(\texttt{Refl})$ is made of a term $f : X \to Y$ and an equation $f \equiv f$, which is represented as $X \xrightarrow{f} \equiv \to Y$. The image of this rule by Yoneda is:

$$\boxed{X \xrightarrow{f} Y} \xleftarrow[f \mapsto f]{} \boxed{X \xrightarrow{f} \equiv \to Y} \xleftarrow[g \mapsto f]{f \mapsto f} \boxed{\begin{tikzcd} X \arrow[r, bend left, "f"] \arrow[r, bend right, "g"'] \arrow[r, phantom, "\equiv"] & Y \end{tikzcd}}$$

In figure 4, for several rules of the equational logic we give the corresponding denominator in $\mathbf{E}_{eq,S}$ (on the left) and its image by $\mathcal{Y}$ in $\mathbf{S}_{eq}$ (on the right). This entailment has the form $\mathcal{H} \xrightarrow[\tau]{} \mathcal{H}'$ and can be read as: "as soon as there is an occurrence of $\mathcal{H}$ in a specification Σ, it may be extended (up to equivalence) as an occurrence of $\mathcal{H}'$".

entailment	image by $\mathcal{Y}$
composition: "each pair of consecutive terms is composable"	
$\texttt{Cons} \xleftarrow[\texttt{i}]{} \texttt{Comp}$	$X \xrightarrow{f} Y \xrightarrow{g} Z \;\xleftrightarrow[\subseteq]{}\; \begin{tikzcd} X \arrow[r, "f"] \arrow[rr, bend right, "g \circ f"'] & Y \arrow[r, "g"] & Z \end{tikzcd}$
identity: "each type is a type with identity"	
$\texttt{Type} \xleftarrow[\texttt{i0}]{} \texttt{Selid}$	$X \;\xleftrightarrow[\subseteq]{}\; \begin{tikzcd} X \arrow[loop right, "id_X"] \end{tikzcd}$
reflexivity: "for each term f there is an equation $f \equiv f$"	
$\texttt{Term} \xleftarrow[t]{} \texttt{Refl}$	$X \xrightarrow{f} Y \;\xleftrightarrow[\subseteq]{}\; X \xrightarrow{f \equiv} Y$
binary product: "each pair of types has a product"	
$\texttt{Type}^2 \xleftarrow[\texttt{j}]{} \texttt{2-Prod}$	$Y_1 \quad Y_2 \;\xleftrightarrow[\subseteq]{}\; \begin{tikzcd} & Y_1 \times Y_2 \arrow[dl, "pr_1"'] \arrow[dr, "pr_2"] & \\ Y_1 & & Y_2 \end{tikzcd}$
pairing: "each binary cone has a pairing"	
$\texttt{2-Cone} \xleftarrow[\texttt{2-dom}]{} \texttt{Pair}$	$\begin{tikzcd} Y_1 & & Y_2 \\ & X \arrow[ul, "f_1"] \arrow[ur, "f_2"'] & \end{tikzcd} \;\xleftrightarrow[\subseteq]{}\; \begin{tikzcd} & Y_1 \times Y_2 \arrow[dl, "pr_1"'] \arrow[dr, "pr_2"] & \\ Y_1 & \equiv \langle f_1, f_2 \rangle \equiv & Y_2 \\ & X \arrow[ul, "f_1"] \arrow[ur, "f_2"'] \arrow[uu] & \end{tikzcd}$
terminal type: "there is a terminal type"	
$\texttt{Unit} \xleftarrow[\texttt{j0}]{} \texttt{0-Prod}$	$\;\xleftrightarrow[\subseteq]{}\; 1$
collapsing: "each type is collapsing"	
$\texttt{Type} \xleftarrow[\texttt{0-codom}]{} \texttt{Coll}$	$X \;\xleftrightarrow[\subseteq]{}\; X \xrightarrow{g} 1$

Figure 4: Some rules for equational logic, diagrammatically