To appear in: *Fundamenta Informaticae.*

Parameters and Parametrization in Specification, using Distributive Categories

Bart JACOBS
CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

Abstract. A specification, as we shall use it here, consists of a signature together with a collection of (non-conditional) equations; these equations involve terms in the 'distributive type theory' which is built on top of the signature. This type theory has finite product $(\times, 1)$ and coproduct $(+, 0)$ types. Particular simple examples of such specifications are Hagino specifications, which are used to describe inductively defined types. Models of specifications are described in arbitrary distributive categories.
In a more categorical approach, one describes models as structure preserving functors. It enables us to define in general what are (a) models of parametrized spefications (in terms of Kan extensions) and (b) models with parameters (in terms of so-called 'simple slice' categories). It is shown that in the special case of Hagino specifications, these general definitions specialize to ones in terms of algebras or coalgebras for associated 'strong' polynomial functors. Models with parameters of Hagino specifications were described earlier by Cockett and Spencer.

1 Introduction

Computer scientists have introduced many more data types, like stacks or queues, than the ones which are traditionally used by mathematicians, like groups or rings or vector spaces. A specification is taken here to be a linguistic description of such a data type. Early semantical studies of specifications—see for example [8] for an overview—are based on universal algebra. Often these are single-sorted, or single-typed, as we shall say. Category theory is sometimes used, but mostly in a superficial way. Here we intend to use categorical tools properly: we think category theory provides the appropriate framework to study data types. In set theory one proves a result by 'opening sets' and reasoning with their elements. A bit provocatively, one reasons from an implementation. In contrast, arguments in category theory mostly rely on *universal properties* which characterize the objects under consideration. This involves reasoning from a specification (used in an informal sense here, not as defined formally below). The latter approach is closer to the way one operates in computer science.

In this paper we restrict ourselves to equational specifications over a distributive type theory. These are both expressive and simple enough to illustrate the categorical approach. Such a distributive type theory involves finite product $(\times, 1)$ and coproduct $(+, 0)$ types. Especially it involves a lift operation $1+(\_)$ on types which can be used to express partiality of terms. Also, one can use coproduct types $+$ to express overloading.

Bob Walters [28, 29] started describing data types in distributive categories—which are categories with finite products and coproducts satisfying a certain distributivity property, see Definition 3.1 below. Since specifications are linguistic objects written in some type theoretic language, we think it is better first to describe the relevant syntax and describe Walters' examples as *models* in distributive categories. Therefore we start in the next section with distributive type theory. Models are then described in the category **Sets** of sets and functions; they involve assignments of sets to atomic types and of actual functions to function symbols, in such a way that given equations hold. Later the interpretation is extended to arbitrary distributive categories. The picture of models we give at this stage is still rather traditional and much in line with universal algebra. We have a first question at this point. So-called Hagino-specifications are very simple specifications that are used specifically to define a type $\sigma(X)$, either inductively via a constructor $\mathsf{constr}\colon \sigma(X) \to X$ or co-inductively via a destructor $\mathsf{destr}\colon X \to \sigma(X)$. Models of these can be described as algebras or coalgebras of a polynomial functor associated with σ. It is *a priori* not clear that such a description coincides with the traditional picture that we just mentioned.

To tackle this problem appropriately we need to know what a model is of a parametrized specification—since a Hagino specification is parametrized by the set of atomic types which occur. There is a neat approach to the semantics of parametrized specifications using Kan extensions. In order to exploit this categorical notion, we need an alternative description of models as certain functors. It is the functorial semantics of Lawvere. It will be shown how a 'traditional' model of a specification corresponds to a certain functor from what is called the classifying category of the specification. With these tools we can show in Theorem 6.3 that the (co)algebra semantics of Hagino specifications is the same as the traditional semantics. This answers our first question and fills a gap in the literature.

A second question involves models with parameters. Cockett and Spencer [5, 6, 27] give a concrete description of what it means for an algebra of a strong functor to be initial with parameters. We ask ourselves whether there is a general notion of 'model with parameters' for arbitrary specifications, which specializes for Hagino specifications to the one of Cockett and Spencer. We give an affirmative answer in Section 7, see especially Theorem 7.12, making essential use of what we call 'simple slice' categories $\mathbb{B}/\!\!/I$.

The weight of this paper lies in the last two sections 6 and 7. The sections 2 – 5 contain preparatory expositions and some results, which we believe are mostly folklore. However, it is hard—if not impossible—to find all the material in the literature. Readers who feel comfortable with the description of a specification as a signature Σ together with a set $\mathcal{E}$ of equations, and of a model of this specification $(\Sigma, \mathcal{E})$ in a distributive category $\mathbb{B}$ as a distributive functor $\mathcal{C}\ell(\Sigma, \mathcal{E}) \to \mathbb{B}$ from the classifying (or free distributive) category of $(\Sigma, \mathcal{E})$ to $\mathbb{B}$, may wish to skip much of these sections 2 – 5.

There are only few categorical prerequisites necessary to follow the main lines of the expositions. But there are some side remarks (like 5.7, 7.3 or 7.14) which do involve more advanced category theory. There are no deep results; the paper is of a more expository nature.

2 Syntax

In this section it will be explained what a distributive signature is and how it gives rise to a distributive type theory (in which one has finite product and coproduct types). Further, one finds a description of equational logic in such a setting.

Let S be a set, elements of which will be seen as **atomic** or **basic types**[1]. We shall write $\overline{S}$ for the closure of S under finite products $(1, \times)$ and coproducts $(0, +)$. Thus $\overline{S}$ is the least set satisfying

$$S \cup \{1, 0\} \subseteq \overline{S} \qquad \text{and} \qquad \sigma, \tau \in \overline{S} \;\Rightarrow\; (\sigma \times \tau), (\sigma + \tau) \in \overline{S}$$

where 1 and 0 are (new) symbols for the empty product and coproduct. They can be understood as singleton and empty set. Elements of $\overline{S}$ will be called **types**.

For a function $f: S \to T$ there is a corresponding function $\overline{f}: \overline{S} \to \overline{T}$ by

$$\begin{array}{lll} \overline{f}(s) = f(s), \quad \text{for } s \in S & \overline{f}(1) = 1 & \overline{f}(0) = 0 \\ \overline{f}(\sigma \times \tau) = \overline{f}(\sigma) \times \overline{f}(\tau) & \overline{f}(\sigma + \tau) = \overline{f}(\sigma) + \overline{f}(\tau) & \end{array}$$

In this way we obtain a functor $\overline{(\_)}$ from the category **Sets** of sets and functions to itself (i.e. an endofunctor on **Sets**).

2.1. Definition. (i) A **distributive signature** Σ (or just a signature, in this paper) consists of a set S of basic types together with sets of function symbols $F: \sigma \to \tau$, where $\sigma, \tau \in \overline{S}$. Formally Σ consists of a pair $(S, \mathcal{F})$ where $\mathcal{F}$ is an indexed collection $\{\mathcal{F}_{\sigma,\tau}\}_{\sigma,\tau \in \overline{S}}$ of sets of function symbols. We write $F: \sigma \to \tau$ for $F \in \mathcal{F}_{\sigma,\tau}$. A signature will be called **single-typed** (or **single-sorted**) if its set of basic types is a singleton.

(ii) A **morphism of signatures** from $(S, \mathcal{F})$ to $(T, \mathcal{G})$ consists of a function $\phi: S \to T$ together with an indexed collection of functions $\phi_{\sigma,\tau}: \mathcal{F}_{\sigma,\tau} \to \mathcal{G}_{\overline{\phi}(\sigma), \overline{\phi}(\tau)}$. Using the same symbol ϕ here is convenient: one has

$$F: \sigma \longrightarrow \tau \text{ in } (S, \mathcal{F}) \;\Rightarrow\; \phi(F): \overline{\phi}(\sigma) \longrightarrow \overline{\phi}(\tau) \text{ in } (T, \mathcal{G})$$

In this way one obtains a category of signatures, which will be written as **Sign**.

The notion of morphism of signature used above is a very restricted one: atomic types are mapped to atomic types (and not to arbitrary types) and function symbols are mapped to function symbols (and not to arbitrary terms, see below). A category without such restrictions can be obtained as a Kleisli-category on the above category **Sign**, see Remark 5.7 (ii). We won't need this extra generality because essentially all morphisms $\phi: \Sigma \to \Sigma'$ of signatures that we consider will be inclusions.

What we call a 'distributive signature' is called a 'distributive graph' in [30].

2.2. Examples. (i) Any set S (of atomic types) forms a signature with no function symbols. This signature will also be denoted by S. Further, there is a direct correspondence between functions $f: S \to T$ and signature morphisms $f: S \to T$. This gives a full and faithful inclusion functor **Sets** $\to$ **Sign**.

(ii) Let S be a set (of atomic types again) and X be a symbol not in S, which serves as type variable. A **Hagino signature** (after [10, 9]) has $S \cup \{X\}$ as atomic types and one function symbol, namely, either

$$\mathsf{constr}: \sigma \longrightarrow X \qquad \text{or} \qquad \mathsf{destr}: X \longrightarrow \sigma$$

[1] We use 'type' for what is sometimes called 'sort'.

B. Jacobs/Parameters and parametrization

where σ is a type in the closure $\overline{S \cup \{X\}}$ of $S \cup \{X\}$ under finite product and coproduct types. Thus X may occur in σ, which may be made explicit by writing $\sigma(X)$ instead of σ. In the first case we speak of an **inductive** Hagino signature and of the function symbol constr as a **constructor**. In the second case the signature will be called **co-inductive** and destr will be called a **destructor**.

In case σ is of the form $\sigma_1 + \cdots + \sigma_n$ the constructor constr may be understood as an n-tuple of function symbols $\mathsf{constr}_i: \sigma_i \to X$. Dually, if σ is of the form $\sigma_1 \times \cdots \times \sigma_n$ the destructor destr corresponds to an n-tuple $\mathsf{destr}_i: X \to \sigma_i$. This will become clear once the associated term calculus is described below.

Examples of Hagino signatures are

$$\begin{array}{rcll} 1 + X & \to & X & \text{for natural numbers} \\ 1 + \alpha \times X & \to & X & \text{for lists of type } \alpha \\ X & \to & \alpha \times X & \text{for streams (or infinite lists) of type } \alpha \end{array}$$

These examples will be further worked out in due course.

(iii) There are obvious morphisms of signatures from S in (i) above to $(S \cup \{X\}, \mathsf{constr}: \sigma \to X)$ and to $(S \cup \{X\}, \mathsf{destr}: X \to \sigma)$ in (ii). Later we shall understand this as expressing that the set S of atomic types 'parametrizes' the latter two signatures. For example, in the last two examples in (ii) one has $S = \{\alpha\}$ as parameter signature.

Next we describe how a signature Σ as described above forms the basis of a term calculus which we call **distributive type theory**. Its sequents have the form

$$\Gamma \vdash M: \sigma$$

expressing that term M is of type $\sigma \in \overline{S}$ in context Γ, consisting of variable declarations $x_1: \sigma_1, \ldots, x_n: \sigma_n$. These contexts Γ will be written as sequences (with an order), but its assignments $x_i: \sigma_i$ may be permuted freely. We shall write $M[N/x]$ for the result of substituting a term $N: \tau$ for a variable $x: \tau$ in M. As the two most important rules one has context projection and application of functions symbols.

$$\frac{}{x_1: \sigma_1, \ldots, x_n: \sigma_n \vdash x_i: \sigma_i} \qquad \frac{\Gamma \vdash M: \sigma}{\Gamma \vdash F(M): \tau} \, (\text{for } F: \sigma \to \tau \text{ in } \Sigma)$$

Further, there are rules for forming finite tuples and projections of terms

$$\frac{\Gamma \vdash M: \sigma \qquad \Gamma \vdash N: \tau}{\Gamma \vdash \langle M, N \rangle: \sigma \times \tau} \qquad \frac{\Gamma \vdash P: \sigma \times \tau}{\Gamma \vdash \pi P: \sigma} \qquad \frac{\Gamma \vdash P: \sigma \times \tau}{\Gamma \vdash \pi' P: \tau} \qquad \frac{}{\Gamma \vdash \langle\rangle: 1}$$

with conversions

$$\frac{\Gamma \vdash M: \sigma \qquad \Gamma \vdash N: \tau}{\Gamma \vdash \pi \langle M, N \rangle = M: \sigma} \qquad \frac{\Gamma \vdash M: \sigma \qquad \Gamma \vdash N: \tau}{\Gamma \vdash \pi' \langle M, N \rangle = N: \tau}$$

$$\frac{\Gamma \vdash P: \sigma \times \tau}{\Gamma \vdash \langle \pi P, \pi' P \rangle = P: \sigma \times \tau} \qquad \frac{\Gamma \vdash M: 1}{\Gamma \vdash M = \langle\rangle: 1}$$

And similarly one has rules for finite cotuples and coprojections.

$$\frac{\Gamma \vdash M: \sigma}{\Gamma \vdash \kappa M: \sigma + \tau} \qquad \frac{\Gamma \vdash N: \tau}{\Gamma \vdash \kappa' N: \sigma + \tau}$$

$$\frac{\Gamma \vdash P: \sigma + \tau \qquad \Gamma, x: \sigma \vdash Q: \rho \qquad \Gamma, y: \tau \vdash R: \rho}{\Gamma \vdash \left\{ \begin{array}{lcl} x: \sigma & \mapsto & Q \\ y: \tau & \mapsto & R \end{array} \right\} (P): \rho} \qquad \frac{}{\Gamma, z: 0 \vdash \{\}: \rho}$$

The variables x and y are bound in Q and R in the 'case' term $\left\{\begin{array}{lcl} x{:}\,\sigma & \mapsto & Q \\ y{:}\,\tau & \mapsto & R \end{array}\right\}(P)$. It can be understood as follows. Look at $P{:}\,\sigma+\tau$; if P is in σ, then do Q with P for x; else if P is in τ, do R with P for y. This explains the conversions

$$\frac{\Gamma \vdash M{:}\,\sigma \qquad \Gamma, x{:}\,\sigma \vdash Q{:}\,\rho \qquad \Gamma, y{:}\,\tau \vdash R{:}\,\rho}{\Gamma \vdash \left\{\begin{array}{lcl} x{:}\,\sigma & \mapsto & Q \\ y{:}\,\tau & \mapsto & R \end{array}\right\}(\kappa M) = Q[M/x]{:}\,\rho}$$

$$\frac{\Gamma \vdash N{:}\,\tau \qquad \Gamma, x{:}\,\sigma \vdash Q{:}\,\rho \qquad \Gamma, y{:}\,\tau \vdash R{:}\,\rho}{\Gamma \vdash \left\{\begin{array}{lcl} x{:}\,\sigma & \mapsto & Q \\ y{:}\,\tau & \mapsto & R \end{array}\right\}(\kappa' N) = R[N/y]{:}\,\rho}$$

$$\frac{\Gamma \vdash P{:}\,\sigma+\tau \qquad \Gamma, z{:}\,\sigma+\tau \vdash R{:}\,\rho}{\Gamma \vdash \left\{\begin{array}{lcl} x{:}\,\sigma & \mapsto & R[(\kappa x)/z] \\ y{:}\,\tau & \mapsto & R[(\kappa' y)/z] \end{array}\right\}(P) = R[P/z]{:}\,\rho}$$

$$\frac{\Gamma, z{:}\,0 \vdash M{:}\,\rho}{\Gamma, z{:}\,0 \vdash M = \{\}{:}\,\rho}$$

We thus understand $\{\}$ as the unique 'value' of the empty function $0 \to \rho$.

These conversion rules induce a conversion relation $=$ on terms, which is the least equivalence relation containing the above conversions, and which is compatible with the term forming operations.

2.3. Notation and examples. For a function symbol $F{:}\,\sigma_1 \times \cdots \times \sigma_n \longrightarrow \tau$ we write

$$x_1{:}\,\sigma_1, \ldots, x_n{:}\,\sigma_n \vdash F(x_1, \ldots, x_n){:}\,\tau$$

instead of the formally correct

$$x_1{:}\,\sigma_1, \ldots, x_n{:}\,\sigma_n \vdash F(\langle x_1, \ldots, x_n \rangle){:}\,\tau$$

As a special case, for $n = 0$ and $F{:}\,1 \longrightarrow \tau$ we simply write

$$\vdash F{:}\,\tau \qquad \text{for} \qquad \vdash F(\langle\rangle){:}\,\tau$$

In distributive type theory one has boolean logic built in. We write $\mathsf{B} = 1+1$ for the type of booleans; it comes equipped with two constants

$$\mathsf{ff} \stackrel{\text{def}}{=} \kappa(\langle\rangle){:}\,\mathsf{B} \qquad \text{and} \qquad \mathsf{tt} \stackrel{\text{def}}{=} \kappa'(\langle\rangle){:}\,\mathsf{B}$$

for false and true. For a boolean term (or formula) $\Gamma \vdash \varphi{:}\,\mathsf{B}$ we define its negation $\Gamma \vdash \neg\varphi{:}\,\mathsf{B}$ to be

$$\neg\varphi \stackrel{\text{def}}{=} \left\{\begin{array}{lcl} x{:}\,1 & \mapsto & \mathsf{tt} \\ y{:}\,1 & \mapsto & \mathsf{ff} \end{array}\right\}(\varphi)$$

Obviously $\neg\mathsf{ff} = \mathsf{tt}$ and $\neg\mathsf{tt} = \mathsf{ff}$. Also $\neg\neg\varphi = \varphi$, which follows using Lemma 2.4 below.

For boolean terms $\Gamma \vdash \varphi, \psi{:}\,\mathsf{B}$, one can define their conjunct by

$$\varphi \wedge \psi \stackrel{\text{def}}{=} \left\{\begin{array}{lcl} x{:}\,1 & \mapsto & \mathsf{ff} \\ y{:}\,1 & \mapsto & \left\{\begin{array}{lcl} z{:}\,1 & \mapsto & \mathsf{ff} \\ w{:}\,1 & \mapsto & \mathsf{tt} \end{array}\right\}(\psi) \end{array}\right\}(\varphi)$$

B. Jacobs/Parameters and parametrization

which can be read as

if $\varphi = \mathsf{ff}$ then ff
 elif $\psi = \mathsf{ff}$ then ff
 else tt fi

In a similar way one defines disjunction $\vee$. It can be shown that $(\mathsf{B}, \wedge, \mathsf{tt}, \vee, \mathsf{ff}, \neg)$ forms a boolean algebra (in the type theory); in particular, the Morgan laws hold.

A further useful aspect of distributive type theory is that it is well-fitted to deal with partially defined operations via error values. For every type σ there is the lifted type

$$\perp\!\sigma \stackrel{\text{def}}{=} 1+\sigma$$

which comes equipped with an error value $\perp = \kappa(\langle\rangle){:}\,\perp\!\sigma$. Thus a term $\Gamma \vdash M{:}\,\perp\!\sigma$ can be seen as a partially defined operation of type σ. It is undefined if $M = \perp$[2].

The following result is often useful when calculating with coproducts.

2.4. Lemma. *In distributive type theory one has the following equation.*

$$\frac{\Gamma \vdash P{:}\,\sigma+\tau \qquad \Gamma, x{:}\,\sigma \vdash Q{:}\,\rho \qquad \Gamma, y{:}\,\tau \vdash R{:}\,\rho \qquad \Gamma, z{:}\,\rho \vdash L{:}\,\mu}{\Gamma \vdash L[\left\{\begin{array}{lcl} x{:}\,\sigma & \mapsto & Q \\ y{:}\,\tau & \mapsto & R \end{array}\right\}(P)/z] = \left\{\begin{array}{lcl} x{:}\,\sigma & \mapsto & L[Q/z] \\ y{:}\,\tau & \mapsto & L[R/z] \end{array}\right\}(P){:}\,\mu}$$

Informally, case commutes with substitution[3].

Proof. Because

$$\begin{array}{rcl} L[\left\{\begin{array}{lcl} x{:}\,\sigma & \mapsto & Q \\ y{:}\,\tau & \mapsto & R \end{array}\right\}(P)/z] & = & L[\left\{\begin{array}{lcl} x{:}\,\sigma & \mapsto & Q \\ y{:}\,\tau & \mapsto & R \end{array}\right\}(w)/z][P/w] \\ & = & \left\{\begin{array}{lcl} x{:}\,\sigma & \mapsto & L[\left\{\begin{array}{lcl} x{:}\,\sigma & \mapsto & Q \\ y{:}\,\tau & \mapsto & R \end{array}\right\}(\kappa x)/z] \\ y{:}\,\tau & \mapsto & L[\left\{\begin{array}{lcl} x{:}\,\sigma & \mapsto & Q \\ y{:}\,\tau & \mapsto & R \end{array}\right\}(\kappa' y)/z] \end{array}\right\}(P) \\ & = & \left\{\begin{array}{lcl} x{:}\,\sigma & \mapsto & L[Q/z] \\ y{:}\,\tau & \mapsto & L[R/z] \end{array}\right\}(P). \end{array}$$

□

The next lemma justifies the name 'distributive type theory' for the above calculus: it shows that the binary product $\times$ distributes over the finite coproducts $(+, 0)$.

2.5. Lemma. *Consider distributive type theory (over some signature) as described above.*

(i) *There are terms P and Q in*

$$u{:}\,(\sigma\times\tau)+(\sigma\times\rho) \vdash P{:}\,\sigma\times(\tau+\rho) \qquad v{:}\,\sigma\times(\tau+\rho) \vdash Q{:}\,(\sigma\times\tau)+(\sigma\times\rho)$$

which are each other's inverse, i.e.

$$\begin{array}{rcl} u{:}\,(\sigma\times\tau)+(\sigma\times\rho) & \vdash & Q[P/v] \;=\; u{:}\,(\sigma\times\tau)+(\sigma\times\rho) \\ v{:}\,\sigma\times(\tau+\rho) & \vdash & P[Q/u] \;=\; v{:}\,\sigma\times(\tau+\rho) \end{array}$$

[2] One can show that this lift operation forms a strong monad on the classifying categories described in Definition 3.3 below. It forms an example of a 'computational monad', see [24].

[3] The more categorically oriented reader may have recognized cotupling $[f, g]{:}\,A+B \to C$ of $f{:}\,A \to C$ and $g{:}\,B \to C$ in the case term. The lemma then says that $h \circ [f, g] = [h \circ f, h \circ g]$.

(ii) *Similarly, there are terms*

$$z{:}0 \vdash M{:}\sigma\times 0 \qquad\qquad w{:}\sigma\times 0 \vdash N{:}0$$

which are other's inverse.

In a more concise formulation one could write the above result as $(\sigma\times\tau)+(\sigma\times\rho)\cong\sigma\times(\tau+\rho)$ and $\sigma\times 0\cong 0$.

Proof. (i) For P one can take

$$P(u) \stackrel{\text{def}}{=} \left\{\!\!\left|\begin{array}{lcl} x{:}\sigma\times\tau & \mapsto & \langle \pi x, \kappa(\pi' x)\rangle \\ y{:}\sigma\times\rho & \mapsto & \langle \pi y, \kappa'(\pi' y)\rangle \end{array}\right|\!\!\right\}(u)$$

In order to define Q notice that we have a term

$$x{:}\sigma, w{:}\tau+\rho \;\vdash\; \left\{\!\!\left|\begin{array}{lcl} y{:}\tau & \mapsto & \kappa\langle x,y\rangle \\ z{:}\rho & \mapsto & \kappa'\langle x,z\rangle \end{array}\right|\!\!\right\}(w){:}\,(\sigma\times\tau)+(\sigma\times\rho)$$

and thus for $v{:}\sigma\times(\tau+\rho)$ we can substitute $[\pi v/x]$ and $[\pi' v/w]$, which yields

$$Q(v) \stackrel{\text{def}}{=} \left\{\!\!\left|\begin{array}{lcl} y{:}\tau & \mapsto & \kappa\langle \pi v, y\rangle \\ z{:}\rho & \mapsto & \kappa'\langle \pi v, z\rangle \end{array}\right|\!\!\right\}(\pi' v)$$

Then

$$\begin{array}{rcll}
P[Q/u] & = & \left\{\!\!\left|\begin{array}{lcl} y{:}\tau & \mapsto & P[\kappa\langle \pi v, y\rangle/u] \\ z{:}\rho & \mapsto & P[\kappa'\langle \pi v, z\rangle/u] \end{array}\right|\!\!\right\}(\pi' v) & \text{by the previous lemma} \\
& = & \left\{\!\!\left|\begin{array}{lcl} y{:}\tau & \mapsto & \langle \pi v, \kappa y\rangle \\ z{:}\rho & \mapsto & \langle \pi v, \kappa' z\rangle \end{array}\right|\!\!\right\}(\pi' v) & \\
& = & \langle \pi v, \pi' v\rangle & \text{by conversion} \\
& = & v. & \\
Q[P/v] & = & \left\{\!\!\left|\begin{array}{lcl} x{:}\sigma\times\tau & \mapsto & Q[\langle \pi x, \kappa(\pi' x)\rangle/v] \\ y{:}\sigma\times\rho & \mapsto & Q[\langle \pi y, \kappa'(\pi' y)\rangle/v] \end{array}\right|\!\!\right\}(u) & \text{by the previous lemma} \\
& = & \left\{\!\!\left|\begin{array}{lcl} x{:}\sigma\times\tau & \mapsto & \kappa\langle \pi x, \pi' x\rangle \\ y{:}\sigma\times\rho & \mapsto & \kappa'\langle \pi y, \pi' y\rangle \end{array}\right|\!\!\right\}(u) & \\
& = & \left\{\!\!\left|\begin{array}{lcl} x{:}\sigma\times\tau & \mapsto & \kappa x \\ y{:}\sigma\times\rho & \mapsto & \kappa' y \end{array}\right|\!\!\right\}(u) & \\
& = & u. &
\end{array}$$

(ii) We have

$$w{:}\sigma\times 0 \vdash \pi' w{:}0 \qquad \text{and} \qquad z{:}0 \vdash \{\}{:}\sigma\times 0$$

Then

$$z{:}0 \vdash (\pi'\{\})[\{\}/w] \;=\; \pi'\{\} \;=\; z{:}0$$

Further,

$$y{:}\sigma, z{:}0 \vdash \langle y,z\rangle \;=\; \{\}{:}\sigma\times 0$$

Therefore

$$w{:}\sigma\times 0 \vdash w \;=\; \langle y,z\rangle[\pi w/y][\pi' w/z] \;=\; \{\}[\pi' w/z]{:}\sigma\times 0. \qquad \square$$

The above distributivity comes for free since we have used appropriate formulations of finite coproduct types with contexts as parameters; it is given by the syntax.

2.6. Example (Overloading with coproducts). Suppose we have types N and R for natural numbers and reals and two associated function symbols for addition,

$$\mathsf{plus_N}{:}\,\mathsf{N}\times\mathsf{N}\longrightarrow\mathsf{N} \qquad \text{and} \qquad \mathsf{plus_R}{:}\,\mathsf{R}\times\mathsf{R}\longrightarrow\mathsf{R}$$

One can then form an overloaded operation plus such that $\mathsf{plus}(x,y)$ will be $\mathsf{plus_N}(x,y)$ if x,y are both in N, $\mathsf{plus_R}(x,y)$ if x,y are both in R and undefined otherwise. We regard this as a runtime choice. Thus we will have as type of the overloaded addition,

$$\mathsf{plus}{:}\,(\mathsf{N}+\mathsf{R})\times(\mathsf{N}+\mathsf{R})\longrightarrow\bot(\mathsf{N}+\mathsf{R})$$

where $\bot(-)=1+(-)$. It can be obtained by making suitable case distinctions. We leave the type theoretic description to the reader, but will provide a diagrammatic formulation of plus in Example 3.6 in the next section.

Next we turn to equations in distributive type theory.

2.7. Definition. Let Σ be a distributive signature. A **Σ-equation** is a sequent of the form

$$\Gamma \vdash M =_\sigma M'$$

where M, M' are terms of type σ in context Γ. A **specification** is a pair $(\Sigma,\mathcal{E})$ where Σ is a signature and $\mathcal{E}$ is a collection of Σ-equations, considered as axioms of a theory.

It may be clear that we are restricting ourselves to equational specifications involving only non-conditional equations.

2.8. Examples. (i) Every signature Σ determines a specification which has no equations. In particular—see Example 2.2 (i)—every set of basic types determines a specification and so does every Hagino signature. In the latter case we shall speak of a 'Hagino specification'.

(ii) Monoids can be specified by a signature with one type, say Ω, and two function symbols,

$$m{:}\,\Omega\times\Omega\longrightarrow\Omega \qquad\qquad e{:}\,1\longrightarrow\Omega$$

for multiplication and unit. These are required to satisfy the familiar equations

$$x{:}\Omega \vdash m(x,e) =_\Omega x \qquad\qquad x{:}\Omega \vdash m(e,x) =_\Omega x$$
$$x{:}\Omega, y{:}\Omega, z{:}\Omega \vdash m(m(x,y),z) =_\Omega m(x,m(y,z)).$$

This constitutes the specification *MONOID*. A specification *GROUP* of groups is obtained by adding an extra function symbol

$$i{:}\,\Omega\longrightarrow\Omega$$

intended as inverse operation, and two extra equations

$$x{:}\Omega \vdash m\,(x,i(x)) =_\Omega x \qquad\qquad x{:}\Omega \vdash m\,(i(x),x) =_\Omega x$$

(iii) Natural numbers may be specified in various ways, for example as Hagino specification,

$$0{:}\,1\longrightarrow\mathsf{N} \qquad\qquad \mathsf{S}{:}\,\mathsf{N}\longrightarrow\mathsf{N}$$

that is, by one sort N, two function symbols and no equations. Formally—see Example 2.2 (ii)—we should say there is one function symbol $\mathsf{constr} = [0, \mathsf{S}] \colon 1 + \mathsf{N} \to \mathsf{N}$.

One may wish in an alternative specification an explicit predecessor operation, with besides the above function symbols, also

$$\mathsf{P} \colon \mathsf{N} \longrightarrow 1 + \mathsf{N} = \bot\mathsf{N}$$

The equations involved are

$$\begin{array}{rcl} n \colon \mathsf{N} & \vdash & \left\{\!\!\left\{ \begin{array}{lcl} x \colon 1 & \mapsto & 0 \\ y \colon \mathsf{N} & \mapsto & \mathsf{S}(y) \end{array} \right\}\!\!\right\}(\mathsf{P}(n)) =_{\mathsf{N}} n \\ z \colon \bot\mathsf{N} & \vdash & \mathsf{P}(\left\{\!\!\left\{ \begin{array}{lcl} x \colon 1 & \mapsto & 0 \\ y \colon \mathsf{N} & \mapsto & \mathsf{S}(y) \end{array} \right\}\!\!\right\}(z)) =_{\bot\mathsf{N}} z. \end{array}$$

One can further extend this specification with function symbols like

$$\mathsf{plus} \colon \mathsf{N} \times \mathsf{N} \longrightarrow \mathsf{N} \qquad\qquad \mathsf{min} \colon \mathsf{N} \times \mathsf{N} \longrightarrow \bot\mathsf{N}$$

(iv) The following example of queues give a type theoretic version of Walters' categorical specification in [28, 29]. The specification has two atomic types: α and $\mathcal{Q}(\alpha)$; there are three function symbols

$$\mathsf{nil} \colon 1 \longrightarrow \mathcal{Q}(\alpha) \qquad \mathsf{push} \colon \alpha \times \mathcal{Q}(\alpha) \longrightarrow \mathcal{Q}(\alpha) \qquad \mathsf{pop} \colon \mathcal{Q}(\alpha) \longrightarrow 1 + \mathcal{Q}(\alpha) \times \alpha$$

satisfying a single equation,

$$\begin{array}{l} w \colon 1 + \alpha \times \mathcal{Q}(\alpha) \vdash \mathsf{pop}\Big(\left\{\!\!\left\{ \begin{array}{rcl} x \colon 1 & \mapsto & \mathsf{nil} \\ \langle a, q\rangle \colon \alpha \times \mathcal{Q}(\alpha) & \mapsto & \mathsf{push}(a, q) \end{array} \right\}\!\!\right\}(w)\Big) =_{1+\alpha\times\mathcal{Q}(\alpha)} \\ \left\{\!\!\left\{ \begin{array}{rcl} x \colon 1 & \mapsto & \bot \\ \langle a, q\rangle \colon \alpha \times \mathcal{Q}(\alpha) & \mapsto & \left\{\!\!\left\{ \begin{array}{rcl} y \colon 1 & \mapsto & \kappa'\langle \mathsf{nil}, a\rangle \\ \langle b, p\rangle \colon \alpha \times \mathcal{Q}(\alpha) & \mapsto & \kappa'\langle \mathsf{push}(a, p), b\rangle \end{array} \right\}\!\!\right\}(\mathsf{pop}(q)) \end{array} \right\}\!\!\right\}(w) \end{array}$$

In somewhat different informal notation this says

$$\begin{array}{rcl} & \vdash & \mathsf{pop}(\mathsf{nil}) \;=\; \bot \\ a \colon \alpha, q \colon \mathcal{Q}(\alpha) & \vdash & \mathsf{pop}(\mathsf{push}(a, q)) \;=\; \begin{cases} \langle \mathsf{nil}, a\rangle & \text{if } \mathsf{pop}(q) = \bot \\ \langle \mathsf{push}(a, p), b\rangle & \text{if } \mathsf{pop}(q) = \langle p, b\rangle \end{cases} \end{array}$$

We shall refer to this specification as $QUEUE(\alpha)$.

2.9. Remark. A signature may be called **algebraic** if it does not involve finite coproduct types, i.e. if for each function symbol $F \colon \sigma \longrightarrow \tau$, the types σ, τ are built with finite products only (from atomic types). Similarly, a specification $(\Sigma, \mathcal{E})$ is **algebraic** if Σ is algebraic and terms in equations in $\mathcal{E}$ do not involve any finite coproducts, cotuples or coprojections. Thus the above specifications of monoids and groups are algebraic and also the Hagino specification of natural numbers in terms of $0 \colon 1 \longrightarrow \mathsf{N}$ and $\mathsf{S} \colon \mathsf{N} \longrightarrow \mathsf{N}$.

Next we describe the derivation rules for equational logic over the distributive type theory associated with a signature. We emphasize that the following is *not* a rule of (many-typed) equational logic

$$(\textbf{strengthening}) \quad \frac{\Gamma, x \colon \sigma \vdash M =_\tau M'}{\Gamma \vdash M =_\tau M'} \text{ (if } x \text{ not in } M, M')$$

simply because it does not hold in general: σ may be empty; then the assumption $x \colon \sigma$ that σ is inhabited leads to absurdities. In single-typed equational logic there is only one atomic type whose interpretation is usually required to be non-empty. Then the strengthening rule is valid. In many-typed logic one does not require the interpretation of each type to be non-empty[4].

We do use the following five rules.

$$\frac{\Gamma \vdash M = M' \colon \sigma}{\Gamma \vdash M =_\sigma M'} \qquad\qquad \frac{\Gamma \vdash M =_\sigma M' \qquad \Gamma \vdash M' =_\sigma M''}{\Gamma \vdash M =_\sigma M''} \text{(trans)}$$

$$\frac{\Gamma \vdash M =_\sigma M'}{\Gamma \vdash M' =_\sigma M} \text{(sym)} \qquad\qquad \frac{\Gamma \vdash M =_\sigma M' \qquad \Gamma, x \colon \sigma \vdash N \colon \tau}{\Gamma \vdash N[M/x] =_\tau N[M'/x]} \text{(compat)}$$

$$\frac{\Gamma \vdash M \colon \sigma \qquad \Gamma, x \colon \sigma \vdash N =_\tau N'}{\Gamma \vdash N[M/x] =_\tau N'[M/x]} \text{(subst)}$$

The first of these rules tells that convertible terms ($M = M' \colon \sigma$) are equal ($M =_\sigma M'$) in this equational logic. As a consequence, $=_\sigma$ is reflexive.

An equation which is derivable (from some set $\mathcal{E}$ of axioms) using these five rules plus the equations associated with finite product and coproduct types, will be called a **theorem**. For example, for the specification of groups in Example 2.8 (ii) above, one has an obvious theorem

$$x \colon \Omega, y \colon \Omega \vdash i(m(x, y)) =_\Omega m(i(y), i(x)).$$

2.10. Definition. (i) A specification $(\Sigma, \mathcal{E})$ is a **theory** if the collection $\mathcal{E}$ of equations is closed under derivability, i.e. if an equation E is derivable using the above five rules from $E_1, \ldots E_n \in \mathcal{E}$, then already $E \in \mathcal{E}$. Equivalently, if $\mathcal{E}$ contains all the associated theorems.

Every specification $(\Sigma, \mathcal{E})$ determines a theory $(\Sigma, \overline{\mathcal{E}})$ by closing $\mathcal{E}$ under derivability.

(ii) A **morphism of specifications** from $(\Sigma, \mathcal{E})$ to $(\Sigma', \mathcal{E}')$ consists of a morphism of signatures $\phi \colon \Sigma \to \Sigma'$ such that ϕ maps axioms to theorems, i.e.

$$E \in \mathcal{E} \;\Rightarrow\; \phi E \in \overline{\mathcal{E}'}$$

where ϕE is obtained from E by replacing all types and function symbols in E by their image under ϕ. The resulting category of specifications will be written as **Spec**.

2.11. Examples. In the previous series of examples one finds obvious morphisms of specifications $\{\alpha\} \longrightarrow QUEUE(\alpha)$ and $MONOID \longrightarrow GROUP$. A rather trivial but important example is the inclusion $S \longrightarrow (\sigma(X) \to X)$ or $S \longrightarrow (X \to \sigma(X))$ of a set S of atomic types in a Hagino specification.

2.12. Definition. A **parametrized specification** is a morphism of specifications $\phi \colon (\Sigma_0, \mathcal{E}_0) \to (\Sigma, \mathcal{E})$. The domain $(\Sigma_0, \mathcal{E}_0)$ of ϕ will be called the parameter specification.

In the examples above, it is clear that α forms a parameter in the specification of queues. Similarly in the specification of a group, the subspecification of a monoid can be seen as a parameter, on top of which a group is specified.

Mostly in a parametrized specification $\phi \colon (\Sigma_0, \mathcal{E}_0) \to (\Sigma, \mathcal{E})$ the parameter specification $(\Sigma_0, \mathcal{E}_0)$ will be a subspecification of $(\Sigma, \mathcal{E})$ in the sense that ϕ is injective both on types and on function symbols and maps Σ_0-equations in $\mathcal{E}_0$ to Σ-equations in $\mathcal{E}$.

[4]Of course, if we know on a syntactic (type theoretic) level that σ is not empty—i.e. $\Gamma \vdash N \colon \sigma$, for some N—then we may simply substitute N for x, which yields the conclusion $\Gamma \vdash M =_\tau M'$.

3 Semantics

The set theoretic semantics of a signature Σ is given in the following way. A Σ-**model** (or **algebra**) consists of a collection of 'carrier' sets $\{A_s\}_{s\in S}$, one for each s in the set S of atomic types in Σ. Then one extends this assignment $s \mapsto A_s$ to $\sigma \mapsto \widehat{A}_\sigma$ for $\sigma \in \overline{S}$ by putting

$$\widehat{A}_s = A_s, \quad \text{for } s \in S \qquad \widehat{A}_1 = 1 = \{\emptyset\} \qquad \widehat{A}_0 = \emptyset$$
$$\widehat{A}_{\sigma\times\tau} = \widehat{A}_\sigma \times \widehat{A}_\tau \qquad \widehat{A}_{\sigma+\tau} = \widehat{A}_\sigma + \widehat{A}_\tau$$

where $+$ between sets means disjoint union. In this way one obtains a carrier set for each type. The collection $\{A_s\}_{s\in S}$ can be seen as a functor $A: S \to \mathbf{Sets}$ from the discrete category S to **Sets**. It is extended to a functor $\widehat{A}: \overline{S} \to \mathbf{Sets}$.

Besides these carrier sets $\{A_s\}$, a Σ-model consists of an interpretation of each function symbol,

$$F: \sigma \longrightarrow \tau$$

as an actual function,

$$[\![F]\!]: \widehat{A}_\sigma \longrightarrow \widehat{A}_\tau$$

The notion of **morphism** of Σ-models is as follows. A morphism $h: (\{A_s\}, [\![\_]\!]) \to (\{A'_s\}, [\![\_]\!]')$ consists of a collection of functions between carrier sets,

$$h_s: A_s \longrightarrow A'_s, \qquad \text{for atomic types } s \in S$$

such that for each function symbol $F: \sigma \longrightarrow \tau$ one has that the diagram

$$\begin{tikzcd}
\widehat{A}_\sigma \arrow[r, "\widehat{h}_\sigma"] \arrow[d, "{[\![F]\!]}"'] & \widehat{A'}_\sigma \arrow[d, "{[\![F]\!]'}"] \\
\widehat{A}_\tau \arrow[r, "\widehat{h}_\tau"] & \widehat{A'}_\tau
\end{tikzcd}$$

commutes—where the collection $\{\widehat{h}_\sigma\}_{\sigma\in\overline{S}}$ is obtained by extending $\{h_s\}_{s\in S}$ in the obvious way.

Assuming one has such a Σ-model $(\{A_s\}, [\![\_]\!])$, then one can extend the interpretation $[\![\_]\!]$ of function symbols to all terms: for a context $\Gamma = x_1: \sigma_1, \ldots, x_n: \sigma_n$, write

$$\widehat{A}_\Gamma = \widehat{A}_{\sigma_1} \times \cdots \times \widehat{A}_{\sigma_n}$$

Then one defines by induction on derivations for each term $\Gamma \vdash M: \sigma$ a function

$$[\![\Gamma \vdash M: \sigma]\!]: \widehat{A}_\Gamma \longrightarrow \widehat{A}_\tau$$

in the following way.

$$\begin{array}{rcl}
[\![x_1: \sigma_1, \ldots, x_n: \sigma_n \vdash x_i: \sigma_i]\!] & = & \pi_i : \widehat{A}_{\sigma_1} \times \cdots \times \widehat{A}_{\sigma_n} \to \widehat{A}_{\sigma_i} \\
[\![x: \sigma \vdash F(x): \tau]\!] & = & [\![F]\!] : \widehat{A}_\sigma \to \widehat{A}_\tau \qquad \text{(for } F \text{ a function symbol)} \\
[\![\Gamma \vdash \langle M, N\rangle: \tau_1 \times \tau_2]\!] & = & \langle [\![\Gamma \vdash M: \tau_1]\!], [\![\Gamma \vdash N: \tau_2]\!] \rangle : \widehat{A}_\Gamma \to \widehat{A}_{\tau_1} \times \widehat{A}_{\tau_2} \\
[\![\Gamma \vdash \langle\rangle: 1]\!] & = & \lambda\vec{a}.\, \emptyset : \widehat{A}_\Gamma \to \{\emptyset\} = 1 \\
[\![\Gamma \vdash \pi P: \tau_1]\!] & = & \pi \circ [\![\Gamma \vdash P: \tau_1 \times \tau_2]\!] : \widehat{A}_\Gamma \to \widehat{A}_{\tau_1} \times \widehat{A}_{\tau_2} \to \widehat{A}_{\tau_1} \\
[\![\Gamma \vdash \pi' P: \tau_2]\!] & = & \pi' \circ [\![\Gamma \vdash P: \tau_1 \times \tau_2]\!] : \widehat{A}_\Gamma \to \widehat{A}_{\tau_1} \times \widehat{A}_{\tau_2} \to \widehat{A}_{\tau_2} \\
[\![\Gamma, z: 0 \vdash \{\}: \tau]\!] & = & \widehat{A}_\Gamma \times \emptyset \cong \emptyset \to \widehat{A}_\tau \qquad \text{(the empty map)} \\
[\![\Gamma \vdash \kappa M: \tau_1 + \tau_2]\!] & = & \kappa \circ [\![\Gamma \vdash M: \tau_1]\!] : \widehat{A}_\Gamma \to \widehat{A}_{\tau_1} \to \widehat{A}_{\tau_1} + \widehat{A}_{\tau_2} \\
[\![\Gamma \vdash \kappa' N: \tau_1 + \tau_2]\!] & = & \kappa' \circ [\![\Gamma \vdash N: \tau_2]\!] : \widehat{A}_\Gamma \to \widehat{A}_{\tau_2} \to \widehat{A}_{\tau_1} + \widehat{A}_{\tau_2}
\end{array}$$

where the functions κ and κ' are the obvious injections into the disjoint union. For the interpretation of the case construction we observe that for sets A, B and C one has that the obvious map

$$(A \times B) + (A \times C) \longrightarrow A \times (B + C) \qquad \text{by} \qquad \begin{cases} (0, (a, b)) & \mapsto & (a, (0, b)) \\ (1, (a, c)) & \mapsto & (a, (1, c)) \end{cases}$$

is an isomorphism. Thus for terms $\Gamma \vdash P: \tau_1 + \tau_2$, $\Gamma, x: \tau_1 \vdash Q: \rho$ and $\Gamma, y: \tau_2 \vdash R: \rho$ we can interpret the case term $\Gamma \vdash \left\{\begin{array}{lcl} x: \tau_1 & \mapsto & Q \\ y: \tau_2 & \mapsto & R \end{array}\right\}(P): \rho$ as the composite

$$\widehat{A}_\Gamma \xrightarrow{\langle id, [\![\Gamma \vdash P: \tau_1 + \tau_2]\!] \rangle} \widehat{A}_\Gamma \times (\widehat{A}_{\tau_1} + \widehat{A}_{\tau_2})$$
$$\cong (\widehat{A}_\Gamma \times \widehat{A}_{\tau_1}) + (\widehat{A}_\Gamma \times \widehat{A}_{\tau_2}) \xrightarrow{[[\![\Gamma, x: \tau_1 \vdash Q: \rho]\!], [\![\Gamma, y: \tau_2 \vdash R: \rho]\!]]} A_\rho$$

where the square braces $[\_, \_]$ describe the 'co-tupling' or 'source-tupling' of two functions $f: A \to C$, $g: B \to C$ to a single function $[f, g]: A + B \to C$.

A **model of a specification** $(\Sigma, \mathcal{E})$ is then a model of Σ in which one has an equation of functions,

$$[\![\Gamma \vdash M: \tau]\!] = [\![\Gamma \vdash M': \tau]\!]: [\![\Gamma]\!] \longrightarrow [\![\tau]\!]$$

for each equation $\Gamma \vdash M =_\tau M'$ in $\mathcal{E}$. We then say that this equation **holds** or **is valid**. One easily verifies that if an equation is derivable from $\mathcal{E}$, then it holds in every model of $(\Sigma, \mathcal{E})$.

We leave it to the reader to check that all of the conversions in the previous section are valid under the above interpretation. Further if h is a morphism of models $(\{A_s\}, [\![\_]\!]) \to (\{A'_s\}, [\![\_]\!]')$, then one has for a term $x_1: \sigma_1, \ldots, x_n: \sigma_n \vdash M: \tau$ that the following diagram commutes.

$$\begin{tikzcd}
\widehat{A}_{\sigma_1} \times \cdots \times \widehat{A}_{\sigma_n} \arrow[r, "\widehat{h}_{\sigma_1} \times \cdots \times \widehat{h}_{\sigma_n}"] \arrow[d, "{[\![\Gamma \vdash M: \tau]\!]}"'] & \widehat{A'}_{\sigma_1} \times \cdots \times \widehat{A'}_{\sigma_n} \arrow[d, "{[\![\Gamma \vdash M: \tau]\!]'}"] \\
\widehat{A}_\tau \arrow[r, "\widehat{h}_\tau"] & \widehat{A'}_\tau
\end{tikzcd}$$

Thus h also preserves the interpretation of terms.

The essential (categorical) aspect of the category **Sets** that is used for the above interpretation, is the presence of finite products and finite coproducts (sums), in such a way that binary products distribute over finite coproducts. We recall here that finite products $(1, \times)$ and coproducts $(0, +)$ are described categorically by

- for each object X there are unique maps $0 \to X$ from the initial object 0 to X and $X \to 1$ from X to the terminal object 1; we usually write $!_X$ for both these maps. This hardly ever leads to confusion.
- for objects X, Y, Z there are (natural) bijective correspondences

$$\frac{Z \longrightarrow X \times Y}{\overline{Z \longrightarrow X \qquad Z \longrightarrow Y}} \qquad\qquad \frac{X + Y \longrightarrow Z}{\overline{X \longrightarrow Z \qquad Y \longrightarrow Z}}$$

We shall write the projections as $X \xleftarrow{\pi} X \times Y \xrightarrow{\pi'} Y$ and the coprojections (sometimes called injections) as $X \xrightarrow{\kappa} X + Y \xleftarrow{\kappa'} Y$. The tupling of $f: Z \to X$ and $g: Z \to Y$ is written as $\langle f, g\rangle: Z \to X \times Y$ and the cotupling (or source-tupling) of $f: X \to Z$ and $g: Y \to Z$ as $[f, g]: X + Y \to Z$. There are the familiar equations

$$\pi \circ \langle f, g\rangle = f \qquad \pi' \circ \langle f, g\rangle = g \qquad \langle f, g\rangle \circ h = \langle f \circ h, g \circ h\rangle \qquad \langle \pi, \pi'\rangle = id$$

and

$$[f, g] \circ \kappa = f \qquad [f, g] \circ \kappa' = g \qquad h \circ [f, g] = [h \circ f, h \circ g] \qquad [\kappa, \kappa'] = id$$

One writes $f \times g = \langle f \circ \pi, g \circ \pi'\rangle$ and $f + g = [\kappa \circ f, \kappa' \circ g]$.

The following definition captures the essential structure needed for the interpretation of distributive type theory.

3.1. Definition. (i) A category $\mathbb{B}$ is called **distributive** if it has finite products and coproducts, such that for each object $I \in \mathbb{B}$, the functor $I \times (\_): \mathbb{B} \to \mathbb{B}$ preserves finite coproducts, i.e.

(a) the (unique) canonical map $0 \to I \times 0$ is an isomorphism;

(b) for each pair $X, Y \in \mathbb{B}$, the canonical map

$$[id \times \kappa, id \times \kappa']: (I \times X) + (I \times Y) \longrightarrow I \times (X + Y)$$

is an isomorphism.

(ii) A functor $F: \mathbb{B} \to \mathbb{C}$ between distributive categories is called **distributive** if it preserves finite products and coproducts. Using these we get a category **Distr** of distributive categories and functors.

The above functor $F: \mathbb{B} \to \mathbb{C}$ preserves finite products if for the initial / terminal object 0 / 1 in $\mathbb{B}$ one has that $F0$ / $F1$ is initial / terminal in $\mathbb{C}$; further if the canonical maps

$$F(X \times Y) \longrightarrow FX \times FY \qquad\qquad FX + FY \longrightarrow F(X + Y)$$

are isomorphisms (for each pair X, Y).

It can be shown (as noted by Robin Cockett) that condition (a) in the above definition follows already from (b). Also that the coprojections are monomorphisms and that every map $X \to 0$ is an isomorphism. Further, the distributivity in (b) can alternatively be expressed by (natural) bijective correspondences

$$\frac{I \times X \longrightarrow Z \qquad I \times Y \longrightarrow Z}{\overline{I \times (X + Y) \longrightarrow Z}}$$

involving binary coproducts with parameters. We should warn that the opposite $\mathbb{B}^{op}$ of a distributive category need not be distributive again: for example in **Sets** the above two isomorphisms do not exist with products and coproducts interchanged. For distributive lattices however (i.e. for poset distributive categories) one does have that the opposite is a distributive lattice again. More information on these distributive categories can be found in [3] and in [2].

3.2. Examples. (i) The category **Sets** of sets and functions is a distributive category. Also the subcategory of finite sets is distributive. More generally, every topos is a distributive category.

(ii) The categories of posets with monotone functions and of directed complete posets (dcpo's) with continuous functions are distributive. In both cases, the empty poset is initial and the coproduct is given by disjoint union, ordered by $\langle i, x\rangle \leq \langle j, y\rangle \Leftrightarrow i = j \;\&\; x \leq y$—where $i, j \in \{0, 1\}$.[5]

(iii) Every cartesian closed category with finite coproducts is automatically distributive: each functor $I \times (\_)$ has a right adjoint $I \Rightarrow (\_)$ and thus preserves all colimits. The examples in (i) and (ii) are instances of this phenomenon.

(iv) Let **MS** be the category of metric spaces (X, d_X) with non-expansive functions: a morphism $f: (X, d_X) \to (Y, d_Y)$ is a function $f: X \to Y$ between the underlying sets with $d_Y(f(x), f(x')) \leq d_X(x, x')$ for all $x, x' \in X$. This gives an example of a category which is distributive but not cartesian closed.

(v) A non-example is the category $\mathbf{Sets}_\bullet$ of pointed sets. Objects are sets with a distinguished base point and morphisms are functions preserving these base points. Alternatively, one can think of $\mathbf{Sets}_\bullet$ as the category of sets and partial functions. $\mathbf{Sets}_\bullet$ has finite products and coproducts, but is not distributive; the one-element set $\{\bullet\}$ is both initial and terminal. Therefore $I \times \{\bullet\} \cong I$, which is not $\{\bullet\}$ in general (as required in (a) in Definition 3.1 (i)).

One can also syntactically construct distributive categories from specifications.

3.3. Definition. For a specification $(\Sigma, \mathcal{E})$ one forms the **classifying category**, written as $\mathcal{C}\ell(\Sigma, \mathcal{E})$, with

objects	types σ
morphisms	$\sigma \to \tau$ are equivalence classes $[M(x)]$ of terms $x{:}\sigma \vdash M{:}\tau$. The equivalence relation $M \sim M'$ is given by derivability from $\mathcal{E}$ of the equation $x{:}\sigma \vdash M =_\tau M'$. That is, M and M' are equivalent if $x{:}\sigma \vdash M =_\tau M'$ is a theorem.

Identities are given by variables $x{:}\sigma \vdash x{:}\sigma$ and composition of $x{:}\sigma \vdash M{:}\tau$ and $y{:}\tau \vdash N{:}\rho$ by substitution $x{:}\sigma \vdash N[M/y]{:}\rho$.

3.4. Proposition. *Classifying categories $\mathcal{C}\ell(\Sigma, \mathcal{E})$ are distributive.*

Proof. It is easy to see that the type 0 is initial and that 1 is terminal. Further, there are obvious projections $\sigma \xleftarrow{\pi} \sigma \times \tau \xrightarrow{\pi'} \tau$ and coprojections $\sigma \xrightarrow{\kappa} \sigma + \tau \xleftarrow{\kappa'} \tau$ forming product and coproduct diagrams. The required distributivity follows from Lemma 2.5. □

[5] In this category of dcpo's and continuous functions one does not have the result that every endomorphism has a fixed point, but one does have a fixpoint object for the lift monad $1 + (\_)$, as described in [7]; this is enough to interpret recursively defined functions (as in a language like PCF).

3.5. Remarks. (i) The above classifying category forms a categorical version of what is called in universal algebra the **closed** (or **ground**) **term algebra** $(\{T_\sigma\}, [\![\_]\!])$ of $(\Sigma, \mathcal{E})$. This term algebra lives inside the classifying category: the elements of T_σ are the terms $1 \to \sigma$ in $\mathcal{C}\ell(\Sigma, \mathcal{E})$; these can indeed be identified with the closed terms.

The essential difference between such term algebras and classifying categories is that the latter deal with all terms (and not just with the closed ones) in a natural way.

(ii) For a specification given by a signature Σ (without equations) we write

$$\mathcal{C}\ell(\Sigma) \stackrel{\text{def}}{=} \mathcal{C}\ell(\Sigma, \emptyset)$$

and call $\mathcal{C}\ell(\Sigma)$ the classifying category of Σ.

(iii) The restriction to terms $x{:}\,\sigma \vdash M{:}\,\tau$ containing a single term variable $x{:}\,\sigma$ only, is not really a restriction, because using the finite product types, there is a bijective correspondence between terms M and N in

$$\frac{x_1{:}\,\sigma_1, \ldots, x_n{:}\,\sigma_n \vdash M{:}\,\tau}{\overline{y{:}\,\sigma_1 \times \cdots \times \sigma_n \vdash N{:}\,\tau}}$$

where $\sigma_1 \times \cdots \times \sigma_n$ is 1 if $n = 0$.

(iv) As will be shown in Section 5, classifying categories $\mathcal{C}\ell(\Sigma, \mathcal{E})$ are *free* distributive categories.

(v) According to Walters [30], an **imperative program** consists of an alphabet A of input symbols together with a functor $A^\star \to \mathcal{C}\ell(\Sigma, \mathcal{E})$ from the free monoid $A^\star$ of finite sequences (words) of A to a classifying category $\mathcal{C}\ell(\Sigma, \mathcal{E})$. It is thus given by a type σ (representing the state) together with an A-indexed collection of terms (programs) $\{M_a{:}\,\sigma \to \sigma\}_{a \in A}$. The functor maps an input string $\langle a_1, \ldots, a_n\rangle$ to the composite program $M_{a_1} \circ \cdots \circ M_{a_n}{:}\,\sigma \to \sigma$.

(vi) For algebraic specifications $(\Sigma, \mathcal{E})$—which do not involve finite coproducts, see Remark 2.9—one can form a simpler classifying category $\mathcal{C}\ell_a(\Sigma, \mathcal{E})$ which has types σ built with finite products from atomic types, as objects. Morphisms $\sigma \to \tau$ are equivalence classes of terms which do not involve finite coproducts. Then one has that these classifying categories $\mathcal{C}\ell_a(\Sigma, \mathcal{E})$ have finite products.

3.6. Example. We are now in a position to give a diagrammatic construction of the overloaded plus, as promised in Example 2.6. It can be performed in any distributive category with maps $\mathsf{plus}_\mathsf{N}{:}\,\mathsf{N} \times \mathsf{N} \to \mathsf{N}$ and $\mathsf{plus}_\mathsf{R}{:}\,\mathsf{R} \times \mathsf{R} \to \mathsf{R}$. In particular in classifying categories (in which one has such maps). We make essential use of distributivity. We form $\mathsf{plus}{:}\,(\mathsf{N}+\mathsf{R}) \times (\mathsf{N}+\mathsf{R}) \to \bot(\mathsf{N}+\mathsf{R})$ as composite,

$$\begin{array}{rcl}
(\mathsf{N}+\mathsf{R}) \times (\mathsf{N}+\mathsf{R}) & \cong & ((\mathsf{N}+\mathsf{R}) \times \mathsf{N}) + ((\mathsf{N}+\mathsf{R}) \times \mathsf{R}) \\
& \cong & (\mathsf{N} \times \mathsf{N}) + (\mathsf{R} \times \mathsf{N}) + (\mathsf{N} \times \mathsf{R}) + (\mathsf{R} \times \mathsf{R}) \\
& & \quad\quad \downarrow id + {!} + {!} + id \\
& & (\mathsf{N} \times \mathsf{N}) + 1 + 1 + (\mathsf{R} \times \mathsf{R}) \\
& & \quad\quad \downarrow id + \nabla + id \\
& & (\mathsf{N} \times \mathsf{N}) + 1 + (\mathsf{R} \times \mathsf{R}) \\
& & \quad\quad \downarrow \mathsf{plus}_\mathsf{N} + id + \mathsf{plus}_\mathsf{R} \\
& & \mathsf{N} + 1 + \mathsf{R} \;\cong\; \bot(\mathsf{N}+\mathsf{R})
\end{array}$$

where $\nabla = [id, id]$ is the codiagonal $1 + 1 \to 1$. One sees how the combinations $\mathsf{R} \times \mathsf{N}$ and $\mathsf{N} \times \mathsf{R}$ of inputs of different types are mapped to error values.

This gives an example of 'programming in distributive categories', as in [29].

3.7. Example. We shall describe two very simple classifying categories concretely.

If the specification is the empty one, the classifying category $\mathcal{C}\ell(\emptyset)$ consists of two types 0 and 1 with only one arrow $0 \to 1$. It is thus the 2-element partial order.

If the specification consists of one type $\{\alpha\}$, no function symbols and no equations, then one obtains a classifying category $\mathcal{C}\ell(\{\alpha\})$, the objects of which can be identified with finite polynomials $\sum_i n_i \alpha^i$ with 0 and 1 as initial and terminal object and coproduct $+$ and product $\times$ given by addition and multiplication of polynomials. Morphisms

$$\Big(\sum_i n_i \alpha^i\Big) \longrightarrow \Big(\sum_j m_j \alpha^j\Big)$$

in $\mathcal{C}\ell(\{\alpha\})$ are built from (co)tuples and (co)projections.

Next we describe models in arbitrary distributive categories. The definitions and results are obvious generalizations of the earlier set theoretic ones.

3.8. Definition. (i) A **model of a signature** Σ in a distributive category $\mathbb{B}$ is given by

- a 'carrier' object $A_s \in \mathbb{B}$ for each s in the set S of atomic types in Σ. Then one extends this assignment $A{:}\,S \to \mathbb{B}$ to an assignment $\widehat{A}{:}\,\overline{S} \to \mathbb{B}$ as in the beginning of this section.
- a morphism $[\![F]\!]{:}\,\widehat{A}_\sigma \to \widehat{A}_\tau$ for each function symbol $F{:}\,\sigma \longrightarrow \tau$.

Again one extends this assignment of morphisms to function symbols to an assignment of morphisms to terms as done earlier for **Sets**, in such a way that for a term $\Gamma = x_1{:}\,\sigma_1, \ldots, x_n{:}\,\sigma_n \vdash M{:}\,\tau$ one obtains a morphism

$$[\![\Gamma \vdash M{:}\,\tau]\!]{:}\,\widehat{A}_{\sigma_1} \times \cdots \times \widehat{A}_{\sigma_n} \longrightarrow \widehat{A}_\tau$$

(ii) A **model of a specification** $(\Sigma, \mathcal{E})$ **in** $\mathbb{B}$ consists of a model of Σ that validates all equations in $\mathcal{E}$; that is for an equation $\Gamma \vdash M = M'{:}\,\tau$ in $\mathcal{E}$ one has an equality of morphisms,

$$[\![\Gamma \vdash M{:}\,\tau]\!] = [\![\Gamma \vdash M'{:}\,\tau]\!]$$

An interpretation $\widehat{A}_\sigma$ of a type σ together with the morphisms resulting from interpretations of terms going in and out of this carrier object, may be called a **data structure**.

We leave it to the reader to verify an obvious soundness result: if an equation E is derivable in a specification $(\Sigma, \mathcal{E})$, then it holds in every model of $(\Sigma, \mathcal{E})$ in a distributive category. In particular the associated theory is validated. The notion of morphism between models in **Sets** is formulated in such a way that it generalizes in an obvious way to a notion of morphism between models in an arbitrary (but fixed) distributive category. In this way we get a category

$$\mathbf{Mod}\big((\Sigma, \mathcal{E}),\ \mathbb{B}\big)$$

of models of $(\Sigma, \mathcal{E})$ in a distributive category $\mathbb{B}$. A model of $(\Sigma, \mathcal{E})$ in $\mathbb{B}$ is then called **initial** or **terminal** if it is initial or terminal in this category.

Notice that we may have total functions $\{A_s \to B_s\}$ as morphisms of models, but partial functions $[\![F]\!]$ as interpretations of function symbols; the reason being that partiality can occur *within* a distributive specification via $\sigma \to \bot\tau$.

3.9. Examples. (i) It is immediate that a model in **Sets** as defined above is the same as described in the beginning of this section.

(ii) A model of the first (Hagino) specification of natural numbers in Example 2.8 (iii) in a distributive category $\mathbb{B}$ consist of an object A_{N} together with maps

$$[\![\, 0 \,]\!]: 1 \longrightarrow A_{\mathsf{N}} \qquad [\![\, \mathsf{S} \,]\!]: A_{\mathsf{N}} \longrightarrow A_{\mathsf{N}}$$

From now on, the we often omit the $A_{\_}$ and $[\![\, \_ \,]\!]$ notation. This is an initial model if for each other model

$$1 \xrightarrow{a} Y \xrightarrow{g} Y$$

there is a unique morphism $h: \mathsf{N} \to Y$ making the following diagram commute

$$\begin{tikzcd}
1 \arrow[r, "0"] \arrow[d, equal] & \mathsf{N} \arrow[r, "\mathsf{S}"] \arrow[d, "h"] & \mathsf{N} \arrow[d, "h"] \\
1 \arrow[r, "a"'] & Y \arrow[r, "g"'] & Y
\end{tikzcd}$$

A diagram $1 \xrightarrow{0} \mathsf{N} \xrightarrow{\mathsf{S}} \mathsf{N}$ which is initial in this sense is what Lawvere defines to be a **natural numbers object** (NNO) in a category.

(iii) A model in $\mathbb{B}$ of the second specification of natural numbers in Example 2.8 (iii) involves an extra morphism

$$\mathsf{P}: \mathsf{N} \longrightarrow 1 + \mathsf{N}$$

The validity of the equations means that the maps

$$1 + \mathsf{N} \underset{\mathsf{P}}{\overset{[0,\mathsf{S}]}{\rightleftarrows}} \mathsf{N}$$

are each other's inverses. Initiality of this model means that for every other model

$$1 + Y \underset{g}{\overset{[a,f]}{\rightleftarrows}} Y$$

There is a unique morphism $h: \mathsf{N} \to Y$ with

$$h \circ 0 = a, \qquad h \circ \mathsf{S} = f, \qquad id + h \circ \mathsf{P} = g \circ h.$$

In fact, this last equation follows from the first two.

(iv) In a similar but more complicated way, one can check that a model of the specification of queues in Example 2.8 (iv) is given by an object A (interpreting α) and an object $\mathcal{Q}(A)$ (for $\mathcal{Q}(\alpha)$ together with maps

$$\mathsf{nil}: 1 \longrightarrow \mathcal{Q}(A) \qquad \mathsf{push}: A \times \mathcal{Q}(A) \longrightarrow \mathcal{Q}(A) \qquad \mathsf{pop}: \mathcal{Q}(A) \longrightarrow 1 + \mathcal{Q}(A) \times A$$

such that the following diagram from [28] commutes.

$$\begin{tikzcd}
1 + A \times \mathcal{Q}(A) \arrow[rr, "{[\mathsf{nil}, \mathsf{push}]}"] \arrow[d, "id + id \times \mathsf{pop}"'] & & \mathcal{Q}(A) \arrow[ddd, "\mathsf{pop}"] \\
1 + A \times (1 + \mathcal{Q}(A) \times A) \arrow[d, phantom, "\wr|"] & & \\
1 + A + A \times \mathcal{Q}(A) \times A \arrow[d, phantom, "\wr|"] & & \\
1 + (1 + A \times \mathcal{Q}(A)) \times A \arrow[rr, "id + {[\mathsf{nil}, \mathsf{push}]} \times id"] & & 1 + \mathcal{Q}(A) \times A
\end{tikzcd}$$

B. Jacobs/Parameters and parametrization

(v) There is an obvious way to get a model of a specification $(\Sigma, \mathcal{E})$ in its own classifying category $\mathcal{C}\ell(\Sigma, \mathcal{E})$. This is called the **generic model** of $(\Sigma, \mathcal{E})$. The equations which hold in this model are precisely the theorems, i.e. the equations which are derivable from $\mathcal{E}$. In this way one obtains a completeness result: an equation E is a theorem if and only if it holds in all models.

(vi) Models of algebraic specifications can be described in categories with finite products—since there are no finite coproducts involved.

4 Semantics of Hagino specifications

Hagino specifications $\sigma \to X$ or $X \to \sigma$ define a type σ (co)-inductively—since the type variable X may occur in σ—using finite products and coproducts of atomic types and X, see Example 2.2 (ii). The inductive case, say of the form $(\sigma_1 + \cdots + \sigma_n) \to X$ occurs in the functional programming language ML with syntax

$$\mathsf{datatype}\ X \quad = \quad C_1\ \mathsf{of}\ \sigma_1 \mid \cdots \mid C_n\ \mathsf{of}\ \sigma_n$$

where $C_1, \ldots, C_n$ are constructors. We would simply write $[C_1, \ldots, C_n]: (\sigma_1 + \cdots + \sigma_n) \to X$ in this case. Hagino [10, 9] writes $\mu X. \sigma$ for the initial solution of $\sigma \to X$ and $\nu X. \sigma$ for the terminal solution of $X \to \sigma$. In the experimental programming language CHARITY, see [4], one can define both these initial and terminal types. Thus one can define for example a type of trees of finite depth with nodes having infinitely many branches.

These recursively defined types with initial or terminal characterizations occur already in [1], but are first investigated systematically from a type theoretic perspective by Hagino.

The semantics described in the previous section for arbitrary specifications specializes to Hagino specifications. This special case is often described in terms of algebras and coalgebras for an endofunctor $T: \mathbb{B} \to \mathbb{B}$ (i.e. for a functor from a category $\mathbb{B}$ to itself). For such a functor we often spare on parentheses and write TY for $T(Y)$.

Recall that for an arbitrary endofunctor $T: \mathbb{B} \to \mathbb{B}$ an **algebra** (or T**-algebra**) is an object $Y \in \mathbb{B}$ together with a morphism $\varphi: TY \to Y$. Dually, a **coalgebra** is a pair (Z, ψ) where $\psi: Z \to TZ$. One forms a category T-**Alg** with T-algebras as objects and as morphisms

$$\Big(TY \xrightarrow{\varphi} Y\Big) \xrightarrow{h} \Big(TZ \xrightarrow{\psi} Z\Big)$$

maps $h: Y \to Z$ in $\mathbb{B}$ for which the following diagram commutes

$$\begin{tikzcd}
TY \arrow[r, "Th"] \arrow[d, "\varphi"'] & TZ \arrow[d, "\psi"] \\
Y \arrow[r, "h"] & Z
\end{tikzcd}$$

Dually, there is a category T-**CoAlg** of coalgebras and similar morphisms. In these categories of algebras and coalgebras one can again study initial and terminal objects. Notice that an initial algebra for $T: \mathbb{B} \to \mathbb{B}$ is a terminal coalgebra for $T^{\mathrm{op}}: \mathbb{B}^{\mathrm{op}} \to \mathbb{B}^{\mathrm{op}}$.

4.1. Fact (Lambek). An initial T-algebra $\varphi: TY \to Y$ is an isomorphism.

By duality one has a similar result for terminal coalgebras. The fact is proved by considering the T-algebra $T\varphi: T^2Y \to TY$; one obtains by initiality a map $\overline{\varphi}: Y \to TY$ serving as inverse of φ. Thus initial algebras are fixed points $TY \cong Y$ of functors.

A model of the Hagino signature $[0, \mathsf{S}]: 1 + \mathsf{N} \longrightarrow \mathsf{N}$ of natural numbers in a distributive category $\mathbb{B}$ (see Example 2.8 (iii)) consists of an algebra for the endofunctor $T(X) = 1 + X$. Requiring this algebra to be initial means that for each object $Y \in \mathbb{B}$, together with maps $a: 1 \to Y$ and $g: Y \to Y$, there is a unique $h: \mathsf{N} \to Y$ with

$$\begin{tikzcd}
1 + \mathsf{N} \arrow[r, "1+h"] \arrow[d, "{[0,\mathsf{S}]}"'] & 1 + Y \arrow[d, "{[a,g]}"] \\
\mathsf{N} \arrow[r, "h"] & Y
\end{tikzcd}$$

This initial algebra approach yields the same notion of initial model as the one resulting from the general description in the previous section (see Example 3.9 (i)). Our aim is to establish such a correspondence for all Hagino signatures.

Recall that a Hagino signature involves a set S of atomic types, a type variable X and either a function symbol $\mathsf{constr}: \sigma \to X$ (in the inductive case) or a function symbol $\mathsf{destr}: X \to \sigma$, where σ is a type in $\overline{S \cup \{X\}}$. A model of the set (or subsignature) S in a category $\mathbb{B}$ consists of a functor $A: S \to \mathbb{B}$ (i.e. of a collection $\{A_s\}_{s \in S}$ of objects $A_s \in \mathbb{B}$). The category of models of S in $\mathbb{B}$ is the functor category $\mathbb{B}^S$, in which a morphism $f: \{A_s\}_{s\in S} \to \{B_s\}_{s \in S}$ consists of a collection of morphisms $f = \{f_s: A_s \to B_s\}_{s\in S}$ in $\mathbb{B}$. We have written $\widehat{(\_)}$ for the functor $\mathbb{B}^S \to \mathbb{B}^{\overline{S}}$ which extends the assignment $s \mapsto A_s$, $s \mapsto f_s$ for $s \in S$ to $\sigma \mapsto \widehat{A}_\sigma$ and $\sigma \mapsto \widehat{f}_\sigma$ for $\sigma \in \overline{S}$, as described in the beginning of the previous section.

4.2. Definition. Each model $A: S \to \mathbb{B}$ in a distributive category $\mathbb{B}$ together with a type $\sigma \in \overline{S \cup \{X\}}$ determines a **polynomial** functor $T(A)_\sigma: \mathbb{B} \to \mathbb{B}$ which follows the structure of σ:

$$T(A)_\sigma \stackrel{\text{def}}{=} \begin{cases} \text{the constant functor } A_s & \text{if } \sigma \equiv s \in S \\ \text{the identity functor} & \text{if } \sigma \equiv X \\ \text{the constant functor } 0 & \text{if } \sigma \equiv 0 \\ \text{the constant functor } 1 & \text{if } \sigma \equiv 1 \\ Y \mapsto T(A)_{\sigma_1}(Y) + T(A)_{\sigma_2}(Y) & \text{if } \sigma \equiv \sigma_1 + \sigma_2 \\ Y \mapsto T(A)_{\sigma_1}(Y) \times T(A)_{\sigma_2}(Y) & \text{if } \sigma \equiv \sigma_1 \times \sigma_2 \end{cases}$$

4.3. Lemma. *The assignment $A \mapsto T(A)_\sigma$ extends to a functor $\mathbb{B}^S \to \mathbb{B}^{\mathbb{B}}$.*

Proof. Given $f = \{f_s: A_s \to B_s\}_{s \in S}$ in $\mathbb{B}^S$, one defines a natural transformation

$$T(f)_\sigma: T(A)_\sigma \longrightarrow T(B)_\sigma$$

by induction on σ. The component at $Y \in \mathbb{B}$ is

$$T(f)_\sigma(Y) \stackrel{\text{def}}{=} \begin{cases} f_s & \text{if } \sigma \equiv s \in S \\ id_Y & \text{if } \sigma \equiv X \\ id_0 & \text{if } \sigma \equiv 0 \\ id_1 & \text{if } \sigma \equiv 1 \\ T(f)_{\sigma_1}(Y) + T(f)_{\sigma_2}(Y) & \text{if } \sigma \equiv \sigma_1 + \sigma_2 \\ T(f)_{\sigma_1}(Y) \times T(f)_{\sigma_2}(Y) & \text{if } \sigma \equiv \sigma_1 \times \sigma_2 \end{cases}$$

Then one checks (again by induction on σ) that for $h: Y \to Z$ in $\mathbb{B}$,

$$\begin{tikzcd}
T(A)_\sigma(Y) \arrow[r, "T(f)_\sigma(Y)"] \arrow[d, "T(A)_\sigma(h)"'] & T(B)_\sigma(Y) \arrow[d, "T(B)_\sigma(h)"] \\
T(A)_\sigma(Z) \arrow[r, "T(f)_\sigma(Z)"] & T(B)_\sigma(Z)
\end{tikzcd}$$

□

We see how the syntactic structure of the type σ determines the shape of the associated polynomial functor $T(A)_\sigma$. It turns out that models of the Hagino specifications $\sigma(X) \to X$ and $X \to \sigma(X)$ can be described as algebras $T(A)_\sigma(X) \to X$ and coalgebras $X \to T(A)_\sigma(X)$, see Proposition 4.6 below. The initial algebras and terminal coalgebras herein play a special role.

But one can also go a step further as in [11]. If one has a suitable fibred category $\begin{smallmatrix}\mathbb{E}\\\downarrow\\\mathbb{B}\end{smallmatrix}$ where the category $\mathbb{E}$ provides some logic to reason about what happens in $\mathbb{B}$, then one can get a 'lifted' endofunctor $\mathbb{E} \to \mathbb{E}$ which captures the logic of induction and coinduction associated with the Hagino specification. This, however, will not be further pursued in this paper.

4.4. Discussion on parametrization. A priori there are three ways in which a polynomial functor $T(A)_\sigma: \mathbb{B} \to \mathbb{B}$ as defined above can have an initial algebra—and dually a terminal coalgebra

(a) **Unparametrized initiality.** This is ordinary initiality of a $T(A)_\sigma$-algebra $\varphi: T(A)_\sigma(U) \to U$; for each $\psi: T(A)_\sigma(V) \to V$ there is a unique $h: U \to V$ with $h \circ \varphi = \psi \circ T(A)_\sigma(h)$.

(b) **Parametrized initiality.** In this case one also allows variation in the model of S; $\varphi: T(A)_\sigma(U) \to U$ is initial in this parametrized sense if for each other model $B: S \to \mathbb{B}$ together with a morphism $f: A \to B$ of models and a $T(B)_\sigma$-algebra $\psi: T(B)_\sigma(V) \to V$, there is a unique $h: U \to V$ making the following diagram commute.

$$\begin{tikzcd}
& T(A)_\sigma(U) \arrow[rr, "\varphi"] \arrow[dl, "T(A)_\sigma(h)"'] \arrow[dr, "T(f)_\sigma(U)"] & & U \arrow[dd, "h"] \\
T(A)_\sigma(V) \arrow[dr, "T(f)_\sigma(V)"'] & & T(B)_\sigma(U) \arrow[dl, "T(B)_\sigma(h)"] & \\
& T(B)_\sigma(V) \arrow[rr, "\psi"'] & & V
\end{tikzcd} \qquad (*)$$

where the diamond on the left hand side commutes by naturality.

(c) **Absolute initiality.** For each other model $B\colon S \to \mathbb{B}$ and algebra $\psi\colon T(B)_\sigma(V) \to V$ there is a unique pair $f\colon A \to B$, $h\colon U \to V$ making the above diagram $(*)$ commute. This third form of initiality is initiality in the following category $T(\_)_\sigma$-**Alg**, with

objects pairs consisting of an S-model $A\colon S \to \mathbb{B}$ and an algebra $\varphi\colon T(A)_\sigma(U) \to U$

morphisms $(A, T(A)_\sigma(U) \xrightarrow{\varphi} V) \to (B, T(B)_\sigma(V) \xrightarrow{\psi} V)$ are morphisms $f\colon A \to B$ of S-models together with a map $h\colon U \to V$ in $\mathbb{B}$ making the above diagram $(*)$ commute.

Notice that we have left the dependence on S implicitly in the notation $T(\_)_\sigma$-**Alg**.

The next two results are about these three forms of initiality (a), (b) and (c). Firstly it will be shown that (a) and (b) are the same; and secondly that initiality as in (c) is the same as initiality in a category of models described in the previous section. We will argue below that (c) is too strong.

4.5. Lemma. *An algebra $T(A)_\sigma(U) \xrightarrow{\varphi} U$ is initial in the above unparametrized sense (a) if and only if it is initial in the parametrized sense (b).*

Proof. The if-part is obvious. As to the only-if-part, assume that $\varphi\colon T(A)_\sigma(U) \to U$ is initial as in (a) and that another S-model $B\colon S \to \mathbb{B}$ is given with $f\colon A \to B$ and a $T(B)_\sigma$-algebra $\psi\colon T(B)_\sigma(V) \to V$. Then one has a $T(A)_\sigma$-algebra $\psi' = \psi \circ T(f)_\sigma(V)\colon T(A)_\sigma(V) \to V$ and thus a unique map $h\colon U \to V$ with $h \circ \varphi = \psi' \circ T(A)_\sigma(h) = \psi \circ T(f)_\sigma(V) \circ T(A)_\sigma(h)$. It is clearly the unique one with this property. $\square$

4.6. Proposition. *The category* $\mathrm{Mod}(\Sigma, \mathbb{B})$ *of models (in a fixed distributive category $\mathbb{B}$) of an inductive Hagino specification $\Sigma = (S \cup \{X\}, \mathsf{constr}\colon \sigma \to X)$ as described in the previous section can be identified with the category of algebras $T(\_)_\sigma$-**Alg** described in 4.4.*

Proof. A model of Σ in $\mathbb{B}$ consists of a model $A\colon S \to \mathbb{B}$ together with an object $U \in \mathbb{B}$ and a morphism

$$\varphi = [\![\,\mathsf{constr}\,]\!]\colon \widehat{A(U)}_\sigma \longrightarrow U$$

where $A(U)\colon S \cup \{X\} \to \mathbb{B}$ is

$$A(U)_t = \mathsf{if}\ t \equiv X\ \mathsf{then}\ U\ \mathsf{else}\ A_t$$

By induction on σ, one checks that

$$\widehat{A(U)}_\sigma = T(A)_\sigma(U)$$

see the proof of Lemma 4.3 for the description of the value of the functor $T(A)_\sigma$ at U. Thus φ is a $T(A)_\sigma$-algebra and (A, φ) is an object of the category $T(\_)_\sigma$-**Alg**.

Similarly one verifies that morphisms of Σ-models correspond to morphisms in $T(\_)_\sigma$-**Alg**: a morphism between two Σ-models

$$(A, U, \varphi\colon \widehat{A(U)}_\sigma \to U) \longrightarrow (B, V, \psi\colon \widehat{B(V)}_\sigma \to V)$$

consists of a morphism $f\colon A \to B$ of S-models, together with a map $h\colon U \to V$ in $\mathbb{B}$ such that the following diagram commutes.

$$\begin{tikzcd}
\widehat{A(U)}_\sigma \arrow[r, "\varphi"] \arrow[d, "\widehat{f(h)}_\sigma"'] & U \arrow[d, "h"] \\
\widehat{B(V)}_\sigma \arrow[r, "\psi"] & V
\end{tikzcd}$$

where $f(h)$ is a morphism of $S \cup \{X\}$-models, given by the collection $f(h)_t$ with

$$f(h)_t = \mathsf{if}\ t \equiv X\ \mathsf{then}\ h\ \mathsf{else}\ f_t$$

By induction on σ one verifies that

$$\widehat{f(h)}_\sigma = T(f)_\sigma \circ T(A)_\sigma(h)$$

and thus that the above diagram corresponds to a morphism of algebras in $T(\_)_\sigma$-**Alg**. $\square$

One can define a similar category $T(\_)_\sigma$-**CoAlg** of coalgebras and get a result like above for coinductive Hagino specifications.

The point that we would like to make here is that absolute initiality as in (c) is really too much to require: it does not make sense to require that any model A of the atomic types can be mapped via $f\colon A \to B$ onto another model B. For example, one can check that for the Hagino specification $[\mathsf{nil}, \mathsf{cons}]\colon 1 + (\alpha \times X) \longrightarrow X$ of lists of type α one gets the following. Since the set S of atomic types is $\{\alpha\}$, a model $A\colon S \to \mathbf{Sets}$ can be identified with a set A. And the set $A^\star$ of finite sequences of elements of A is initial in the sense of (a) = (b), but not as in (c). The latter would require a map into any other set B. This is non-sensical.

Thus the above Proposition 4.6 tells us that models of Hagino specifications are algebras, but what is lacking is the right notion of morphism of models, such that a correspondence with algebra maps is obtained. In order to get such a correspondence we have to use models of parametrized specifications. (Indeed, Hagino specifications are parametrized by the set of atomic types, see Example 2.11.) But to give a smooth description of models of parametrized specifications, we first need a more categorical approach to models using functors. This will be in the next section.

In the remainder of this section we have a brief look at the existence of initial / terminal models for Hagino specifications, see [1, 22] or [23, Chapter 11]. These models can be constructed as colimits or limits of ω-**chains**. The latter are functors $\omega \to \mathbb{B}$, where ω is the poset category of natural numbers

$$0 \longrightarrow 1 \longrightarrow 2 \longrightarrow \cdots$$

A functor $F\colon \mathbb{B} \to \mathbb{C}$ will then be called **continuous** if it preserves colimits of such ω-chains. It is easy to see that polynomial functors $T(A)_\sigma$—see Lemma 4.3—are (at least on **Sets**) always continuous, using that finite limits and filtered colimits commute, see [19], IX 2.

The initial algebra of a continuous endofunctor $T\colon \mathbb{B} \to \mathbb{B}$ can be constructed from the colimit U of the ω-chain

$$0 \xrightarrow{!} T0 \xrightarrow{T!} T^2 0 \xrightarrow{T^2!} T^3 0 \longrightarrow \cdots \longrightarrow U$$

where $0 \in \mathbb{B}$ is initial object. This is as in [26]; it generalizes the construction of a fixed point as join of $\bot \leq f(\bot) \leq f^2(\bot) \leq \cdots$ for an endofunction f on a dcpo with bottom $\bot$. Since T preserves the colimit of this ω-chain we get a map $TU \xrightarrow{\sim} U$ which is initial algebra. For example in **Sets**, the endofunctor associated with the Hagino specification of the natural numbers is $T(X) = 1 + X$. The resulting colimit is

$$0 \longrightarrow 1 \longrightarrow 2 \longrightarrow \cdots \longrightarrow \mathsf{N}$$

And for finite lists one has $T(X) = 1 + A \times X$ with initial algebra given by

$$0 \longrightarrow 1 \longrightarrow 1 + A \longrightarrow 1 + A + A^2 \longrightarrow 1 + A + A^2 + A^3 \longrightarrow \cdots \longrightarrow A^\star$$

where $A^\star = \bigcup_{n \in \mathsf{N}} A^n$ is the set of finite sequences of elements of A.

Dually a terminal coalgebra for $T: \mathbb{B} \to \mathbb{B}$ can be constructed as an initial algebra for $T^{\mathrm{op}}: \mathbb{B}^{\mathrm{op}} \to \mathbb{B}^{\mathrm{op}}$[6]. That is, as a limit in $\mathbb{B}$ of the ω^{op}-chain

$$1 \xleftarrow{!} T0 \xleftarrow{T!} T^2 0 \xleftarrow{T^2!} T^3 0 \longleftarrow \cdots \longleftarrow V$$

where $T: \mathbb{B} \to \mathbb{B}$ is required to preserve limits of ω^{op}-chains. Again one has that polynomial functors $T(A)_\sigma$ are 'co-continuous' in this sense. As an example, infinite lists (of type α) are described by a desctructor $X \to \alpha \times X$; the resulting endofunctor $T(X) = A \times X$ on **Sets** leads to the limit

$$1 \longleftarrow A \longleftarrow A^2 \longleftarrow A^3 \longleftarrow \cdots \longleftarrow A^{\mathbb{N}}$$

Finite and infinite lists together are described by a destructor $X \to 1 + \alpha \times X$ and thus by the limit

$$1 \longleftarrow 1 + A \longleftarrow 1 + A + A^2 \longleftarrow 1 + A + A^2 + A^3 \longleftarrow \cdots \longleftarrow (A^\star + A^{\mathbb{N}})$$

The reader may wish to check that the resulting isomorphism

$$1 + A \times (A^\star + A^{\mathbb{N}}) \;\cong\; 1 + A \times A^\star + A \times A^{\mathbb{N}} \;\cong\; 1 + A^+ + A^{\mathbb{N}} \;\cong\; A^\star + A^{\mathbb{N}}$$

is a terminal coalgebra indeed—where $A^+ = \bigcup_{n \geq 1} A^n$ is the set of non-empty finite sequences of elements of A.

We conclude that in a distributive category with colimits of ω-chains / limits of ω^{op}-chains, every inductive / coinductive Hagino specification has an initial algebra / terminal coalgebra. Especially these initial and terminal models exist in **Sets**. This yields solutions to domain equations involving finite products and coproducts. If one additionally wishes higher types via exponents $\to$, then one cannot find set theoretic solutions anymore (when X occurs negatively). For example, there are no non-trivial sets U with $U \cong U \to U$. For solutions of such domain equations with higher types one works in categories of dcpo's, see [26] using embedding-projection pairs as morphisms. Alternatively, one can work in synthetic domain theory, where domains are just sets, see [13].

[6] Recall however that $\mathbb{B}^{\mathrm{op}}$ need not be a distributive category.

5 Functorial semantics

In universal algebra there is a basic result that an algebra $\mathcal{A}$ for a signature Σ is nothing but a morphism of algebras from the free algebra $\mathcal{F}(\Sigma)$ on Σ to $\mathcal{A}$. There is a similar result in the present setting: a model of a specification $(\Sigma, \mathcal{E})$ in a distributive category $\mathbb{B}$ is nothing but a distributive functor $\mathcal{C}\ell(\Sigma, \mathcal{E}) \to \mathbb{B}$ from the classifying category $\mathcal{C}\ell(\Sigma, \mathcal{E})$ of $(\Sigma, \mathcal{E})$ (see Definition 3.3) to $\mathbb{B}$. This classifying category can be understood as the free distributive category on $(\Sigma, \mathcal{E})$.

The idea of describing models as functors preserving appropriate structure goes back to Lawvere [21]. It has the clear advantage that it enables one to use well-established categorical tools for the semantics of specifications. This will be crucial in the rest of this paper.

We start with some basic observations.

5.1. Lemma. *The assignment $(\Sigma, \mathcal{E}) \mapsto \mathcal{C}\ell(\Sigma, \mathcal{E})$ extends to a functor $\mathcal{C}\ell(\_): \mathbf{Spec} \to \mathbf{Distr}$ from the category of specifications to the category of distributive categories.*

Proof. For a morphism $\phi: (\Sigma, \mathcal{E}) \to (\Sigma', \mathcal{E}')$ of specifications one gets a functor $\mathcal{C}\ell(\phi): \mathcal{C}\ell(\Sigma, \mathcal{E}) \to \mathcal{C}\ell(\Sigma', \mathcal{E}')$ by

$$\sigma \mapsto \overline{\phi}(\sigma) \qquad \text{and} \qquad [M] \mapsto [\phi M]$$

where ϕM is obtained from M by replacing each function symbol F in M by its image ϕF. This is well-defined on equivalence classes since ϕ maps theorems in $(\Sigma, \mathcal{E})$ to theorems in $(\Sigma', \mathcal{E}')$. $\square$

5.2. Theorem. *Let $(\Sigma, \mathcal{E})$ be a specification and $\mathbb{B}$ a distributive category. There is an equivalence*

$$\mathbf{Mod}\big((\Sigma, \mathcal{E}),\ \mathbb{B}\big) \;\simeq\; \mathbf{Distr}\big(\mathcal{C}\ell(\Sigma, \mathcal{E}),\ \mathbb{B}\big)$$

between the category of models of $(\Sigma, \mathcal{E})$ in $\mathbb{B}$ and morphisms of these, and the category of distributive functors $\mathcal{C}\ell(\Sigma, \mathcal{E}) \to \mathbb{B}$ and natural transformations.

Proof. Given a model $(\{A_s\}, [\![\_]\!])$ of $(\Sigma, \mathcal{E})$ in $\mathbb{B}$, one gets a distributive functor $\mathcal{A}: \mathcal{C}\ell(\Sigma, \mathcal{E}) \to \mathbb{B}$ by

$$\sigma \mapsto \widehat{A}_\sigma \qquad \text{and} \qquad \Big(\sigma \xrightarrow{[M]} \tau\Big) \mapsto [\![\, x{:}\sigma \vdash M(x){:}\tau \,]\!]$$

A morphism $\{h_s\}: (\{A_s\}, [\![\_]\!]) \to (\{B_s\}, [\![\_]\!])$ of models yields a natural transformation $\mathcal{A} \to \mathcal{B}$ with components $\widehat{h}_\sigma: \widehat{A}_\sigma \to \widehat{B}_\sigma$; these are natural in σ, because they commute with the interpretations of terms (see Section 3).

In the reverse direction, given a distributive functor $\mathcal{M}: \mathcal{C}\ell(\Sigma, \mathcal{E}) \to \mathbb{B}$ one obtains a model with carrier objects

$$A_s = \mathcal{M}(s) \in \mathbb{B}, \qquad \text{for atomic types } s$$

and interpretations of functions symbols $F: \sigma \longrightarrow \tau$ as morphisms

$$\widehat{A}_\sigma \cong \mathcal{M}(\sigma) \xrightarrow{\mathcal{M}([F])} \mathcal{M}(\tau) \cong \widehat{A}_\tau$$

A natural transformation $\alpha: \mathcal{M} \to \mathcal{N}$ yields a collection of morphisms $\alpha(s): \mathcal{M}(s) \to \mathcal{N}(s)$ which commute by naturality with the interpretations of function symbols. $\square$

The above result gives us the possibility to describe a model of a specification $(\Sigma, \mathcal{E})$ in a distributive category $\mathbb{B}$, either as a collection of objects and arrows (interpreting atomic types and function symbols in such a way that equations in $\mathcal{E}$ are validated) or as a distributive functor $\mathcal{C}\ell(\Sigma, \mathcal{E}) \to \mathbb{B}$. We shall freely switch between these two perspectives according to what is most convenient. See for example the next lemma below.

The previous theorem says in particular that a set theoretic model of a specification $(\Sigma, \mathcal{E})$ is (or, corresponds to, if you like) a distributive functor $\mathcal{C}\ell(\Sigma, \mathcal{E}) \to \mathbf{Sets}$.

If $\mathbb{B}$ is a distributive category associated with some programming language—so that morphisms in $\mathbb{B}$ are (equivalence classes of) programs—then a model $\mathcal{M}: \mathcal{C}\ell(\Sigma, \mathcal{E}) \to \mathbb{B}$ may be understood as an implementation of $(\Sigma, \mathcal{E})$ in this language. It then makes sense to write $\mathcal{M}: (\Sigma, \mathcal{E})$ and read this as an inhabitation statement telling that '$\mathcal{M}$ meets specification $(\Sigma, \mathcal{E})$'. In this way one can try to set up a calculus of specifications and implementations.

5.3. Lemma. *Let $(\Sigma, \mathcal{E})$ be a specification and $\mathbb{B}$ a distributive category.*

(i) *A distributive functor $F: \mathbb{B} \to \mathbb{C}$ induces a functor from the category of models of $(\Sigma, \mathcal{E})$ in $\mathbb{B}$ to the category of models in $\mathbb{C}$.*

(ii) *A morphism $\phi: (\Sigma, \mathcal{E}) \to (\Sigma', \mathcal{E}')$ of specifications induces a functor from the category of models of $(\Sigma', \mathcal{E}')$ in $\mathbb{B}$ to the category of models of $(\Sigma, \mathcal{E})$.*

Proof. (i) Post-composition with F yields a functor

$$\mathbf{Distr}\Big(\mathcal{C}\ell(\Sigma, \mathcal{E}),\ \mathbb{B}\Big) \ \longrightarrow\ \mathbf{Distr}\Big(\mathcal{C}\ell(\Sigma, \mathcal{E}),\ \mathbb{C}\Big).$$

(ii) Pre-compostion with $\mathcal{C}\ell(\phi): \mathcal{C}\ell(\Sigma, \mathcal{E}) \to \mathcal{C}\ell(\Sigma', \mathcal{E}')$—see Lemma 5.1—yields a functor

$$\mathbf{Distr}\Big(\mathcal{C}\ell(\Sigma', \mathcal{E}'),\ \mathbb{B}\Big) \ \longrightarrow\ \mathbf{Distr}\Big(\mathcal{C}\ell(\Sigma, \mathcal{E}),\ \mathbb{B}\Big). \qquad \square$$

As another example one has that models $\mathcal{M}_1: \mathcal{C}\ell(\Sigma, \mathcal{E}) \to \mathbb{B}_1$ and $\mathcal{M}_2: \mathcal{C}\ell(\Sigma, \mathcal{E}) \to \mathbb{B}_2$ give rise to a model $\langle \mathcal{M}_1, \mathcal{M}_2 \rangle: \mathcal{C}\ell(\Sigma, \mathcal{E}) \to \mathbb{B}_1 \times \mathbb{B}_2$ in the (distributive) product category $\mathbb{B}_1 \times \mathbb{B}_2$.

There is some extra mileage we can get from Theorem 5.2; therefore we first need the following way to go from distributive categories to specifications.

5.4. Definition. Every distributive category $\mathbb{B}$ determines a signature $\mathrm{Sign}(\mathbb{B})$ with

- objects $X \in \mathbb{B}$ as atomic types
- morphisms $f: X \to Y$ in $\mathbb{B}$ as function symbols $f: X \to Y$.

The category $\mathbb{B}$ then forms an obvious model for this signature $\mathrm{Sign}(\mathbb{B})$—and for the associated term calculus. We let $\mathcal{T}h(\mathbb{B})$ be the set of equations

$$x_1: X_1, \ldots, x_n: X_n \vdash M =_Y M'$$

which hold in this model (when these terms M, M' receive their obvious interpretations in $\mathbb{B}$, as morphisms $X_1 \times \cdots \times X_n \rightrightarrows Y$). Then $\mathrm{Spec}(\mathbb{B}) = (\mathrm{Sign}(\mathbb{B}), \mathcal{T}h(\mathbb{B}))$ will be the specification associated with $\mathbb{B}$.

5.5. Lemma. *Let $\mathbb{B}$ be a distributive category.*

(i) *The collection of equations $\mathcal{T}h(\mathbb{B})$ is closed under derivability—and thus forms a theory.*

(ii) *The assignment $\mathbb{B} \mapsto \mathrm{Spec}(\mathbb{B})$ extends to a functor* $\mathbf{Distr} \to \mathbf{Spec}$.

Proof. (i) By definition, every equation in $\mathcal{T}h(\mathbb{B})$ holds in $\mathbb{B}$. Hence, by soundness, every equation E derivable from (equations in) $\mathcal{T}h(\mathbb{B})$ must hold in $\mathbb{B}$. But then E is already in $\mathcal{T}h(\mathbb{B})$.

(ii) A distributive functor $F: \mathbb{B} \to \mathbb{C}$ induces an obvious morphism of signatures $\mathrm{Sign}(\mathbb{B}) \to \mathrm{Sign}(\mathbb{C})$ by

$$X \mapsto FX \qquad \text{and} \qquad (f: X \to Y) \mapsto (Ff: FX \to FY)$$

One then shows that for a term $x_1: X_1, \ldots, x_n: X_n \vdash M: Y$ in the calculus on $\mathrm{Sign}(\mathbb{B})$ the following diagram commutes

$$\begin{tikzcd}
F(X_1 \times \cdots \times X_n) \arrow[rr, "F[\![\vec{x}: \vec{X} \,\vdash\, M: Y]\!]"] \arrow[d, equal, "\wr"] & & FY \arrow[d, equal, "\wr"] \\
\overline{F}X_1 \times \cdots \times \overline{F}X_n \arrow[rr, "{[\![\vec{x}: \overline{F}\vec{X} \,\vdash\, FM: \overline{F}Y]\!]}"'] & & \overline{F}Y
\end{tikzcd}$$

Here we use that F preserves finite products and coproducts; first of all to establish that $\overline{F}X \cong FX$ by induction on the structure of the type X. Commutation of the diagram is then obtained by induction on the derivation of M. We conclude that the image under F of an equation in $\mathbb{B}$ is an equation in $\mathbb{C}$. Hence we get a morphism of specifications $\mathrm{Spec}(\mathbb{B}) \to \mathrm{Spec}(\mathbb{C})$. $\square$

Next we relate these functors $\mathbf{Spec} \to \mathbf{Distr}$ and $\mathbf{Distr} \to \mathbf{Spec}$.

5.6. Theorem. *Let $(\Sigma, \mathcal{E})$ be a specification and $\mathbb{B}$ a distributive category.*

(i) *There is a bijective correspondence (up-to-isomorphism)*

$$\frac{(\Sigma, \mathcal{E}) \ \longrightarrow\ \mathrm{Spec}(\mathbb{B}) \ =\ (\mathrm{Sign}(\mathbb{B}), \mathcal{T}h(\mathbb{B})) \qquad \textit{in } \mathbf{Spec}}{\mathcal{C}\ell(\Sigma, \mathcal{E}) \ \longrightarrow\ \mathbb{B} \qquad \textit{in } \mathbf{Distr}}$$

(ii) *The induced model $\varepsilon: \mathcal{C}\ell(\mathrm{Sign}(\mathbb{B}), \mathcal{T}h(\mathbb{B})) \longrightarrow \mathbb{B}$ of the specification of $\mathbb{B}$ in $\mathbb{B}$ itself, is an isomorphism.*

The theorem states that taking specifications yields a full and faithful functor $\mathbf{Distr} \to \mathbf{Spec}$, which—in a suitable 2-categorical sense—has a left adjoint.

Proof. (i) Just observe that a morphism of specifications $(\Sigma, \mathcal{E}) \to \mathrm{Spec}(\mathbb{B})$ is nothing but a model of $(\Sigma, \mathcal{E})$ in $\mathbb{B}$. The latter corresponds by the previous theorem to a distributive functor $\mathcal{C}\ell(\Sigma, \mathcal{E}) \to \mathbb{B}$.

(ii) The inverse $\mathbb{B} \longrightarrow \mathcal{C}\ell(\mathrm{Sign}(\mathbb{B}), \mathcal{T}h(\mathbb{B}))$ is given by $X \mapsto X$ and $(f: X \to Y) \mapsto$ (the equivalence class $[f(x)]$ of $x: X \vdash f(x): Y$). $\square$

5.7. Remarks. (i) In particular we may conclude from (ii) that every distributive category can be described as a classifying category. Thus a distributive functor $F: \mathbb{B} \to \mathbb{C}$ can be understood as a model of (the specification $\mathrm{Spec}(\mathbb{B})$ of) $\mathbb{B}$ in $\mathbb{C}$.

(ii) From this theorem we get an (idempotent) monad $\mathrm{Spec}(\mathcal{C}\ell(\_)): \mathbf{Spec} \to \mathbf{Spec}$. The resulting Kleisli category may be called the category of specifications and **translations**, where a translation $\phi: (\Sigma, \mathcal{E}) \to (\Sigma', \mathcal{E}')$ corresponds to an assignment from atomic types in Σ to arbitrary types and from function symbols to arbitrary terms. Thus translations are not so restrictive as the signature morphisms described in Definition 2.1 (which map atomic types to atomic types and function symbols to function symbols).

Alternatively, by Theorem 5.6, a translation $(\Sigma, \mathcal{E}) \to (\Sigma', \mathcal{E}')$ can be understood as a distributive functor $\mathcal{C}\ell(\Sigma, \mathcal{E}) \to \mathcal{C}\ell(\Sigma', \mathcal{E}')$.

(iii) For algebraic specifications—see Remark 2.9 and 3.5 (iv)—one has in a similar way that models can be described as finite product preserving functors $\mathcal{C}\ell_a(\Sigma, \mathcal{E}) \to \mathbb{B}$ and that $\mathcal{C}\ell_a(\Sigma, \mathcal{E})$ is the free category with finite products on $(\Sigma, \mathcal{E})$. Lawvere's original investigations in [21] are about these algebraic (single-typed) specifications.

(iv) Goguen and Burstall [14] have a notion of **institution** which they use for abstract model theory of (arbitrary) specifications. We briefly indicate how the above functorial semantics of distributive specifications gives a particularly straightforward example of such an institution. One has a functor **Sign** $\to$ **Sets** by

$$\begin{cases} \Sigma & \mapsto & \{E \mid E \text{ is a } \Sigma\text{-equation}\} \\ \phi & \mapsto & \text{the function } E \mapsto \phi E \end{cases}$$

and for each distributive category $\mathbb{B}$ a functor $\mathbf{Sign}^{\mathrm{op}} \to \mathbf{Cat}$ by

$$\begin{cases} \Sigma & \mapsto & \mathbf{Distr}\big(\mathcal{C}\ell(\Sigma),\ \mathbb{B}\big) \\ \phi & \mapsto & \phi^* = {}_- \circ \mathcal{C}\ell(\phi) \end{cases}$$

For each Σ-equation $E = (\Gamma \vdash N =_\sigma N')$ and Σ-model $\mathcal{M}: \mathcal{C}\ell(\Sigma) \to \mathbb{B}$ there is a satisfaction relation $\models$ given by

$$\mathcal{M} \models E \quad \stackrel{\text{def}}{\Leftrightarrow} \quad \mathcal{M}(N) = \mathcal{M}(N'): \mathcal{M}(\Gamma) \longrightarrow \mathcal{M}(\sigma)$$

The notion of institution then requires that one has

$$\mathcal{N} \models \phi E \quad \Leftrightarrow \quad \phi^*(\mathcal{N}) \models E$$

But this correspondence between reindexing equations and reindexing models exists by definition.

6 Parametrized specifications

Recall from Definition 2.12 that parametrized specifications are morphisms of specifications $\phi: (\Sigma_0, \mathcal{E}_0) \to (\Sigma, \mathcal{E})$. These are important in a modular approach to specification: if one already has a specification *MONOID* of monoids (see Example 2.8 (ii)), then one specifies a group simply as a monoid plus an extra function symbol for inverse and the associated equations. In this way one obtains a specification *GROUP*, which extends *MONOID*. That is, there is an inclusion specification morphism *MONOID* $\to$ *GROUP*. This is of course a very elementary case, but it is exemplaric. The possibility to built such a new specification on top of a given one, is a basic feature of specification languages—like Clear, OBJ, Act One, Act Two (see [14] for extensive references).

The general form one recognizes in such situations is described by a morphism of specifications $\phi: (\Sigma_0, \mathcal{E}_0) \to (\Sigma, \mathcal{E})$, where $(\Sigma_0, \mathcal{E}_0)$ is the **parameter specification**. *MONOID* is thus a parameter in the above specification of *GROUP*. In this section we concentrate on the semantics of such parametrized specifications $\phi: (\Sigma_0, \mathcal{E}_0) \to (\Sigma, \mathcal{E})$. Semantically we shall understand ϕ as an operation which, given a model $\mathcal{M}$ of the parameter specification $(\Sigma_0, \mathcal{E}_0)$ in a fixed category $\mathbb{B}$, produces the free model of $(\Sigma, \mathcal{E})$ on $\mathcal{M}$ incorporating the $(\Sigma_0, \mathcal{E}_0)$-structure. Concretely, in the above example *MONOID* $\to$ *GROUP*, we understand the parametrized specification to denote the free group construction on an arbitrary monoid (incorporating the monoid structure). Categorically, this will be understood in terms of **Kan extensions**. Using Kan extensions in this way is not new (it occurs already in rudimentary form in [17]), but is little known. Therefore we provide details.

Let $H: \mathbb{C} \to \mathbb{D}$ be an arbitrary functor. For a given category $\mathbb{B}$, one obtains a functor

$$H^*: \mathit{Fun}(\mathbb{D},\ \mathbb{B}) \longrightarrow \mathit{Fun}(\mathbb{C},\ \mathbb{B}) \qquad \text{by} \qquad \mathcal{M} \mapsto \mathcal{M} \circ H$$

where $\mathit{Fun}(\mathbb{D},\ \mathbb{B})$ is the category of functors $\mathbb{D} \to \mathbb{B}$ and natural transformations between them. In a diagram,

$$\begin{tikzcd} \mathbb{C} \arrow[rr, "H"] \arrow[dr, "H^*(\mathcal{M}) = \mathcal{M} \circ H"'] & \arrow[d, phantom, "\leftarrow\!\shortmid"] & \mathbb{D} \arrow[dl, "\mathcal{M}"] \\ & \mathbb{B} & \end{tikzcd}$$

In such a setting, $\mathbb{B}$ is often called the **receiving** category.

Kan extensions involve an inverse $\mathit{Fun}(\mathbb{C},\ \mathbb{B}) \to \mathit{Fun}(\mathbb{D},\ \mathbb{B})$ to this operation H^*, which satisfies a universal property. The latter is called 'left' c.q. 'right' dependening on whether the extension is initial c.q. terminal (in a suitable sense). For the moment, let's concentrate on left. For a functor $\mathcal{N}: \mathbb{C} \to \mathbb{B}$, a **left Kan extension** along H is a functor $\mathbb{D} \to \mathbb{B}$, written as $\mathrm{Lan}_H(\mathcal{N})$, which comes equipped with bijective correspondences between natural transformations

$$\frac{\mathrm{Lan}_H(\mathcal{N}) \ \longrightarrow\ \mathcal{M} \qquad \text{in } \ \mathit{Fun}(\mathbb{D},\ \mathbb{B})}{\mathcal{N} \ \longrightarrow\ H^*(\mathcal{M}) = \mathcal{M} \circ H \qquad \text{in } \ \mathit{Fun}(\mathbb{C},\ \mathbb{B})}$$

which are natural in $\mathcal{M}$.

In the triangle we get a bijective correspondence between natural transformations $\Rightarrow$ in

$$\begin{tikzcd} \mathbb{C} \arrow[rr, "H"] \arrow[dr, bend right, "\mathcal{N}"'] \arrow[dr, "H^*(\mathcal{M})"] & & \mathbb{D} \arrow[dl, "\mathcal{L}"'] \arrow[dl, bend left, "\mathcal{M}"] \\ & \mathbb{B} & \end{tikzcd}$$

where we have written $\mathcal{L}$ for $\mathrm{Lan}_H(\mathcal{N})$. Alternatively, one can describe the universal property as follows. There is a unit $\eta: \mathcal{N} \to H^*\mathrm{Lan}_H(\mathcal{N})$ such that for each $\alpha: \mathcal{N} \to H^*(\mathcal{M})$ there is a unique $\overline{\alpha}: \mathrm{Lan}_H(\mathcal{N}) \to \mathcal{M}$ with $\alpha = H^*(\overline{\alpha}) \circ \eta$. Using the universal property one easily establishes that Kan extensions are unique up-to-isomorphism—if they exist.

A *right* Kan extension $\mathrm{Ran}_H(\mathcal{N}): \mathbb{D} \to \mathbb{B}$ along H involves (natural) bijective correspondences

$$\frac{\mathcal{M} \ \longrightarrow\ \mathrm{Ran}_H(\mathcal{N}) \qquad \text{in } \ \mathit{Fun}(\mathbb{D},\ \mathbb{B})}{H^*(\mathcal{M}) \ \longrightarrow\ \mathcal{N} \qquad \text{in } \ \mathit{Fun}(\mathbb{C},\ \mathbb{B})}$$

Below we are intereseted in *distributive* Kan extensions, where all categories and functors involved are distributive—including $\mathrm{Lan}_H(\mathcal{N})$ and $\mathrm{Ran}_H(\mathcal{N})$. These need not exist, see the discussion 6.4 later.

First we spell out an example. A specification of **stacks** can be given with two atomic types α, for the parameter, and $\mathcal{S}(\alpha)$, for the stack on α. These come together with function symbols

$$\mathsf{nil}: 1 \longrightarrow \mathcal{S}(\alpha) \qquad \mathsf{push}: \alpha \times \mathcal{S}(\alpha) \longrightarrow \mathcal{S}(\alpha) \qquad \mathsf{pop}: \mathcal{S}(\alpha) \longrightarrow 1 + \alpha \times \mathcal{S}(\alpha)$$

which satisfy the equations

$$\begin{array}{rcl}
 & \vdash & \mathsf{pop}(\mathsf{nil}) =_{1+\alpha\times\mathcal{S}(\alpha)} \perp \\
a{:}\,\alpha, s{:}\,\mathcal{S}(\alpha) & \vdash & \mathsf{pop}(\mathsf{push}(a,s)) \;=\; \kappa'(\langle a,s\rangle) : 1+\alpha\times\mathcal{S}(\alpha) \\
z{:}\,\mathcal{S}(\alpha) & \vdash & \left\{\!\!\left| \begin{array}{rcl} x{:}\,1 & \mapsto & \mathsf{nil} \\ \langle a,s\rangle : \alpha\times\mathcal{S}(\alpha) & \mapsto & \mathsf{push}(a,s) \end{array} \right|\!\!\right\}(\mathsf{pop}(z)) \;=_{\mathcal{S}(\alpha)}\; z.
\end{array}$$

Let's write $STACK(\alpha)$ for this specification. It is not a Hagino specification.

A model of such a stack in a distributive category consists of a diagram

$$1 + A\times\mathcal{S}(A) \underset{\mathsf{pop}}{\overset{[\mathsf{nil},\mathsf{push}]}{\rightleftarrows}} \mathcal{S}(A)$$

in which the two arrows are each other's inverse. Suppose we wish to describe a suitable form of initiality of such a model. The following two forms come to mind.

(a) **Unparametrized initiality.** For each object $\mathcal{S}'(A)$ together with a commuting diagram

$$1 + A\times\mathcal{S}'(A) \underset{\mathsf{pop}'}{\overset{[\mathsf{nil}',\mathsf{push}']}{\rightleftarrows}} \mathcal{S}'(A)$$

there is a unique $h{:}\,\mathcal{S}(A)\to\mathcal{S}'(A)$ making the two squares below commute.

$$\begin{tikzcd}
1 + A\times\mathcal{S}(A) \arrow[r, shift left, "{[\mathsf{nil},\mathsf{push}]}"] \arrow[d, "id + id\times h"'] & \mathcal{S}(A) \arrow[l, shift left, "\mathsf{pop}"] \arrow[d, "h"] \\
1 + A\times\mathcal{S}'(A) \arrow[r, shift left, "{[\mathsf{nil}',\mathsf{push}']}"] & \mathcal{S}'(A) \arrow[l, shift left, "\mathsf{pop}'"]
\end{tikzcd}$$

In this formulation we keep the parameter object A fixed. It says that for any other stack on A, there is a unique morphism to that stack.

(b) **Parametrized initiality.** For each object B together with a morphism $f{:}\,A\to B$ and a stack on B,

$$1 + B\times\mathcal{S}'(B) \underset{\mathsf{pop}'}{\overset{[\mathsf{nil}',\mathsf{push}']}{\rightleftarrows}} \mathcal{S}'(B)$$

there is a unique $h{:}\,\mathcal{S}(A)\to\mathcal{S}'(B)$ such that the following diagram commutes.

$$\begin{tikzcd}
1 + A\times\mathcal{S}(A) \arrow[r, shift left, "{[\mathsf{nil},\mathsf{push}]}"] \arrow[d, "id + f\times h"'] & \mathcal{S}(A) \arrow[l, shift left, "\mathsf{pop}"] \arrow[d, "h"] \\
1 + B\times\mathcal{S}'(B) \arrow[r, shift left, "{[\mathsf{nil}',\mathsf{push}']}"] & \mathcal{S}'(B) \arrow[l, shift left, "\mathsf{pop}'"]
\end{tikzcd}$$

In this formulation we also allow variation in the object on which one has a stack.

It is obvious that the parametrized formulation in (b) is more general than the unparametrized one in (a). After a moment's thought one sees that the version in (b) is what one wants: only (b) enables us to define such an elementary operation as length

$$\mathsf{len}{:}\,\mathcal{S}(A) \longrightarrow \mathsf{N}$$

by taking in (b), $B = 1$, $f = !{:}\,A\to 1$ and as stack on 1 the isomorphism

$$1 + (1\times\mathsf{N}) \cong \mathsf{N}$$

which follows from the isomorphism $1+\mathsf{N} \underset{\mathsf{P}}{\overset{[0,\mathsf{S}]}{\rightleftarrows}} \mathsf{N}$. This yields a unique map $\mathsf{len}{:}\,\mathcal{S}(A)\to\mathsf{N}$ which satisfies, in functional notation,

$$\begin{array}{rcl}
\mathsf{len}(\mathsf{nil}) & = & 0 \\
\mathsf{len}(\mathsf{push}(a,s)) & = & \mathsf{S}(\mathsf{len}(s))
\end{array}
\qquad\qquad
\mathsf{P}(\mathsf{len}(s)) \;=\; \begin{cases} \perp & \text{if } \mathsf{pop}(s) = \perp \\ \mathsf{len}(\pi'\mathsf{pop}(s)) & \text{else} \end{cases}$$

where the latter equation follows from the first two. Such a length map does not arise from the unparametrized initiality in (a), since N is not a stack on an arbitrary object A.

The reader may wish to verify that in the category of sets, the definition

$$\mathcal{S}(A) = A^\star = \textstyle\bigcup_{n\in\mathsf{N}} A^n$$

yields an obvious isomorphism

$$1 + A\times\mathcal{S}(A) \cong \mathcal{S}(A)$$

which is initial in the parametrized sense (b). Of course there are other stacks on A in **Sets**. For example one can take finite plus infinite lists: $A^\star + A^{\mathsf{N}}$, or also $A^\star + A^{\mathsf{N}} + A^{\mathsf{N}}$.

We now try and capture this parametrized initiality more abstractly. We have an obvious parametrized specification $\phi{:}\,\{\alpha\}\to STACK(\alpha)$ which sends α to α. It induces a functor $\mathcal{C}\ell(\phi){:}\,\mathcal{C}\ell(\{\alpha\})\to\mathcal{C}\ell(STACK(\alpha))$ between the classifying categories, see Lemma 5.1. Thus we get a forgetful functor $\phi^* = \_\circ\mathcal{C}\ell(\phi)$ from the category of $STACK(\alpha)$-models to the category of $\{\alpha\}$-models (in a fixed category $\mathbb{B}$), see Lemma 5.3 (ii). A bit more concretely, ϕ^* sends a $STACK(\alpha)$-model

$$STACK(B) = [1 + B\times\mathcal{S}(B) \rightleftarrows \mathcal{S}(B)] \qquad \text{to} \qquad B.$$

Now let A be an object in $\mathbb{B}$; that is, a $\{\alpha\}$-model or a distributive functor $A{:}\,\mathcal{C}\ell(\{\alpha\})\to\mathbb{B}$, see Theorem 5.6. The initial stack $ISTACK(A)$ on A as described in (b) above satisfies the following property: for each other stack $STACK(B)$ and each morphism $f{:}\,A\to B = \phi^*(STACK(B))$, there is a unique morphism of stacks $ISTACK(A)\to STACK(B)$. Thus, parametrized initiality says that there is a bijective correspondence

$$\begin{array}{c}
A \longrightarrow \phi^*(STACK(B)) \qquad \text{in } \mathbf{Distr}\big(\mathcal{C}\ell(\{\alpha\}),\ \mathbb{B}\big) \\
\hline\hline
ISTACK(A) \longrightarrow STACK(B) \qquad \text{in } \mathbf{Distr}\big(\mathcal{C}\ell(STACK(\alpha)),\ \mathbb{B}\big)
\end{array}$$

which describes the initial stack $ISTACK(A)$ on A as the distributive left Kan extension along $\mathcal{C}\ell(\phi)$. We can therefore write $\mathrm{Lan}_\phi(A) = ISTACK(A)$.

This description is at the right level of abstraction to be generalized.

6.1. Definition. Let $\phi{:}\,(\Sigma_0,\mathcal{E}_0)\to(\Sigma,\mathcal{E})$ be a parametrized specification and $\mathbb{B}$ a distributive category; following the above description, ϕ induces a functor $\phi^* = \_\circ\mathcal{C}\ell(\phi)$ from the category $\mathbf{Distr}\big(\mathcal{C}\ell(\Sigma,\mathcal{E}),\ \mathbb{B}\big)$ of models of $(\Sigma,\mathcal{E})$ in $\mathbb{B}$ to the category $\mathbf{Distr}\big(\mathcal{C}\ell(\Sigma_0,\mathcal{E}_0),\ \mathbb{B}\big)$.

(i) Let $\mathcal{N}: \mathcal{C}\ell(\Sigma_0, \mathcal{E}_0) \to \mathbb{B}$ be a $(\Sigma_0, \mathcal{E}_0)$ model. The **ϕ-initial model on $\mathcal{N}$** is the distributive left Kan extension $\mathrm{Lan}_\phi(\mathcal{N}): \mathcal{C}\ell(\Sigma, \mathcal{E}) \to \mathbb{B}$—if it exists. It comes equipped with (natural) bijective correspondences

$$\frac{\mathrm{Lan}_\phi(\mathcal{N}) \longrightarrow \mathcal{M} \quad \text{in } \mathbf{Distr}\big(\mathcal{C}\ell(\Sigma, \mathcal{E}),\ \mathbb{B}\big)}{\mathcal{N} \longrightarrow \phi^*(\mathcal{M}) \quad \text{in } \mathbf{Distr}\big(\mathcal{C}\ell(\Sigma_0, \mathcal{E}_0),\ \mathbb{B}\big)}$$

The **ϕ-terminal model on $\mathcal{N}$** is the distributive right Kan extension $\mathrm{Ran}_\phi(\mathcal{N}): \mathcal{C}\ell(\Sigma, \mathcal{E}) \to \mathbb{B}$.

(ii) A model $\mathcal{K}: \mathcal{C}\ell(\Sigma, \mathcal{E}) \to \mathbb{B}$ is called **ϕ-initial** if it is ϕ-initial on $\phi^*(\mathcal{K})$. Equivalently, if the induced counit $\varepsilon: \mathrm{Lan}_\phi(\phi^*\mathcal{K}) \to \mathcal{K}$ is an isomorphism.

Similarly one defines **ϕ-terminal** $(\Sigma, \mathcal{E})$-models.

(iii) We say that the specification morphism $\phi: (\Sigma_0, \mathcal{E}_0) \to (\Sigma, \mathcal{E})$ has **initial extensions** if for each $(\Sigma_0, \mathcal{E}_0)$ model $\mathcal{N}$, the distributive left Kan extension $\mathrm{Lan}_\phi(\mathcal{N})$ exists. Dually, **terminal extensions** involves existence of all right Kan extensions.

The above functor ϕ^* is often called a forgetful functor since usually ϕ is an inclusion, so that $\phi^*(\mathcal{M})$ has less structure than $\mathcal{M}$. The terminology 'ϕ-initial / terminal on $\mathcal{N}$' is often used in a sloppy way. In case ϕ is understood from the context, then it is often omitted. One also finds 'free' for 'initial' and 'cofree' for 'terminal'. Further, $\mathcal{N}$ sometimes serves as a specific model in the notation $\mathrm{Lan}_\phi(\mathcal{N})$ and sometimes as a parameter. In the latter case ϕ has initial extensions and the assignment $\mathcal{N} \mapsto \mathrm{Lan}_\phi(\mathcal{N})$ extends to a left adjoint to ϕ^*.

6.2. Examples. (i) The specification of groups described in Example 2.8 (ii) can be seen as $GROUP(\alpha)$ parametrized by an atomic type α used for the underlying set; it involves function symbols $m: \alpha \times \alpha \to \alpha$, $e: 1 \to \alpha$ and $i: \alpha \to \alpha$. There is then an obvious morphism $\phi: \{\alpha\} \to GROUP(\alpha)$ of specifications. A model of $\{\alpha\}$ in **Sets** is just a set, say S. A model of $GROUP(\alpha)$ in **Sets** is a group G and $\phi^*(G)$ is the underlying set of G. The specification morphism ϕ has initial extensions: the left Kan extension $\mathrm{Lan}_\phi(S)$ always exists and is the **free group on** S. It is determined by the correspondence between functions $S \to \phi^*(G)$ and group homorphisms $\mathrm{Lan}_\phi(S) \to G$.

The above Definition 6.1 captures free groups on objects in arbitrary categories—and not just in **Sets**.

(ii) Earlier in Example 2.8 (iii) we saw a morphism of specifications $MONOID \to GROUP$. It restricts to a morphism of specifications $\phi: COMMONOID \to ABGROUP$ between commutative monoids and abelian groups; these both involve an additional commutativity equation

$$x: \Omega, y: \Omega \vdash m(x, y) = m(y, x): \Omega$$

The functor ϕ^* sends an abelian group G to its underlying commutative monoid $\phi^*(G)$. In the reverse direction, a left Kan extension exists. It sends a commutative monoid M to the free abelian group $\mathrm{Lan}_\phi(M)$ on M, which incorporates the monoid structure of M. Usually, $\mathrm{Lan}_\phi(M)$ is called the **Grothendieck group** of M, and written as $\mathcal{K}(M)$, see e.g. [20]. There is the bijective correspondence between homorphisms $M \to \phi^*(G)$ and $\mathcal{K}(M) \to G$.

Finally we come to our first theorem. It relates initial algebras of Hagino specifications and left Kan extensions along the associated morphism of specification.

6.3. Theorem. *Consider an inductive Hagino specification $\Sigma = (S \cup \{X\}, \mathsf{constr}: \sigma \to X)$ with a model $A: S \to \mathbb{B}$ of the atomic types S, in a distributive category $\mathbb{B}$. Then*

a morphism $\varphi = [\![\,\mathsf{constr}\,]\!]: T(A)_\sigma(U) \to U$ is initial $T(A)_\sigma$-algebra

if and only if

φ, as a model $\mathcal{C}\ell(\Sigma) \to \mathbb{B}$, is the left Kan extension on $A: \mathcal{C}\ell(S) \to \mathbb{B}$ along $S \to \Sigma$ in

$$\begin{tikzcd} \mathcal{C}\ell(S) \arrow[rr] \arrow[dr, "A"'] & & \mathcal{C}\ell(\Sigma) \arrow[dl, "\varphi"] \\ & \mathbb{B} & \end{tikzcd}$$

Proof. Let's write the morphism of specification as $\phi: S \to \Sigma$. Another model $\psi: \mathcal{C}\ell(\Sigma) \to \mathbb{B}$ can by Proposition 4.6 be identified with an algebra $\psi: T(B)_\sigma(V) \to V$ with $B = \phi^*(\varphi)$ as the model $\mathcal{C}\ell(S) \to \mathbb{B}$ of the atomic types. One has that φ together with $\eta = id: A \to A = \phi^*(\varphi)$ is left Kan extension if and only if for every $f: A \to B = \phi^*(\psi)$ there is a unique $\overline{f}: \varphi \to \psi$ with $f = \phi^*(\overline{f})$. This means, again by Proposition 4.6, that $\overline{f}$ is of the form $(f: A \to B, h: U \to V)$ forming a commuting diagram $(*)$ as in 4.4. Thus we have shown that φ is left Kan extension if and only if φ is initial in a parametrized sense (b) as in 4.4. By Lemma 4.5 the latter says that φ is initial $T(A)_\sigma$-algebra. □

6.4. Discussion on distributive Kan extensions. In receiving categories $\mathbb{B}$ with colimits (like **Sets**) one can compute left Kan extensions along $H: \mathbb{C} \to \mathbb{D}$ via the so-called pointwise colimit

$$\mathrm{Lan}_\phi(\mathcal{N})(X) \;=\; \varinjlim\Big((H \downarrow X) \xrightarrow{\mathsf{proj}} \mathbb{C} \xrightarrow{\mathcal{N}} \mathbb{B}\Big) \tag{$*$}$$

see e.g. [19]. One has that the associated unit $\eta: \mathcal{N} \to H^*\mathrm{Lan}_H(\mathcal{N})$ is an isomorphism in case H is full and faithful. A dual formula exists for right Kan extensions.

If we assume the above functors H and $\mathcal{N}$ are distributive, then there is no reason why the above pointwise colimit should yield the *distributive* Kan extension. Even worse, the distributive Kan extension need not exist at all, as the following simple example (due to Dominic Verity) shows. Consider the free distributive category $\mathbb{I} = \mathcal{C}\ell(\{\alpha\})$ on one object (see Example 3.7) with finite polynomials $\sum_i n_i\alpha^i$ as objects. A distributive functor $H: \mathbb{I} \to \mathbb{I}$ is determined by its action on $\alpha \in \mathbb{I}$. Thus we may identify such an H with a particular polynomial $H = \sum_i n_i\alpha^i$. A distributive functor $\mathbb{I} \to \mathbf{Sets}$ corresponds to a model $A \in \mathbf{Sets}$ of $\{\alpha\}$. The distributive left Kan extension of A along H should be a set A_0 together with a bijective correspondence

$$\frac{A_0 \longrightarrow B}{A \longrightarrow \sum_i n_i B^i \;=\; H^*(B)}$$

for every $B \in \mathbf{Sets}$. In particular, for $B = \emptyset$, there is at most one function $A \to \emptyset$, whereas there may be many functions $A \to n_0 = \sum_i n_i\emptyset^i$ (namely if $n_0 > 0$). We conclude that such a distributive Kan extension A_0 does not exist in general.

It is open under which circumstances distributive Kan extensions do exist.

In some cases however, the above pointwise formula $(*)$ does give the right answer[7]. Consider for example the Hagino specification $LIST(\alpha)$ with constructor $[\mathsf{nil}, \mathsf{cons}]: 1 + \alpha \times list(\alpha) \to list(\alpha)$. There is an obvious morphism of specifications $\phi: \{\alpha\} \to LIST(\alpha)$ which sends α to α. It induces a functor $\mathcal{C}\ell(\phi): \mathcal{C}\ell(\{\alpha\}) = \mathbb{I} \to \mathcal{C}\ell(LIST(\alpha))$. The left Kan extension of an $\{\alpha\}$-model $A \in \mathbf{Sets}$ along $\mathcal{C}\ell(\phi)$ at $list(\alpha) \in \mathcal{C}\ell(LIST(\alpha))$ is according to the formula $(*)$ the colimit of the diagram in $\mathbf{Sets}$ resulting from all possible

$$\begin{tikzcd}
\left(\sum_i n_i\alpha^i\right) \arrow[r] \arrow[dr] & \left(\sum_j m_j\alpha^j\right) \arrow[d] \\
& list(\alpha)
\end{tikzcd}$$

By the requirement that all these triangle commute, this can be cut down to

$$\begin{tikzcd}
0 \arrow[r] \arrow[drr] & 1 \arrow[r] \arrow[dr] & 1+\alpha \arrow[r] \arrow[d] & 1+\alpha+\alpha^2 \arrow[r] \arrow[dl] & \cdots \arrow[dll] \\
& & list(\alpha) & &
\end{tikzcd}$$

Thus in $\mathbf{Sets}$ we have to take the colimit $A^\star$ in

$$0 \longrightarrow 1 \longrightarrow 1 + A \longrightarrow 1 + A + A^2 \longrightarrow \cdots \longrightarrow A^\star$$

Hence the pointwise construction $(*)$ yields the right result in this case.

In the last chapter of [29] one finds the Generalized Todd-Coxeter Procedure (due to Carmody and Walters) which computes left Kan extensions in **Sets**. It takes finite descriptions (in terms of graphs and relations) of categories and functors as input and produces the left Kan extension L together with the universal natural transformation. The procedure terminates if and only if each LX is a finite set. There is an implementation. It remains to be investigated to what extend this procedure can be adapted meaningfully to the above situation and compute extensions in the case of parametrized specifications.

7 Models with parameters

The universal property of the natural numbers (as described in 3.9 (ii)) says in informal notation, that for each $a: Y$ and $g: Y \to Y$, there is a unique $h: \mathsf{N} \to Y$ with

$$h\,\mathsf{0} = a \qquad \text{and} \qquad h(\mathsf{S}\,n) = g(h\,n)$$

More useful is a version with parameters: for each parameter set (or object) I with $f: I \to Y$ and $g: I \times Y \to Y$, there is a unique $h: I \times \mathsf{N} \to Y$ with

$$h(i, \mathsf{0}) = f\,i \qquad \text{and} \qquad h(i, \mathsf{S}\,n) = g(i, h(i, n))$$

For example, using this version with parameters, one can define addition by taking

[7] This happens for example when the specifications involved are algebraic and the receiving category has all colimits which are preserved by functors $I \times (\_)$.

$$I = \mathsf{N}, \qquad Y = \mathsf{N}, \qquad f(m) = m, \qquad g(m, n) = \mathsf{S}\,n$$

Categorically, these 'natural numbers with parameters' are given in a (distributive) category $\mathbb{B}$ by a diagram

$$1 \xrightarrow{\mathsf{0}} \mathsf{N} \xrightarrow{\mathsf{S}} \mathsf{N}$$

such that for each parameter object $I \in \mathbb{B}$ and for each $f: I \times 1 \to Y$ and $g: I \times Y \to Y$, there is a unique $h: I \times \mathsf{N} \to Y$ making the following diagram commute.

$$\begin{tikzcd}
I \times 1 \arrow[r, "id \times \mathsf{0}"] \arrow[d, equal] & I \times \mathsf{N} \arrow[r, "id \times \mathsf{S}"] \arrow[d, "{\langle \pi, h \rangle}"'] & I \times \mathsf{N} \arrow[d, "{\langle \pi, h \rangle}"] \\
I \times 1 \arrow[r, "{\langle \pi, f \rangle}"'] & I \times Y & I \times Y \arrow[l, "{\langle \pi, g \rangle}"]
\end{tikzcd}$$

We have written $f: I \times 1 \to Y$ instead of $f: I \to Y$ to make things come out nicer.

This universal property can best be described in what we call the 'simple slice category' $\mathbb{B}/\!\!/I$ associated with $\mathbb{B}$.

7.1. Definition. (i) For a category $\mathbb{B}$ with binary products $\times$ and an object $I \in \mathbb{B}$, the **simple slice category** $\mathbb{B}/\!\!/I$ has

objects $\quad X \in \mathbb{B}$
morphisms $\quad X \to Y$ in $\mathbb{B}/\!\!/I$ are morphisms $I \times X \to Y$ in $\mathbb{B}$

The identity $X \to X$ in $\mathbb{B}/\!\!/I$ is the second projection $\pi': I \times X \to X$ and composition of $f: X \to Y$ and $g: Y \to Z$ in $\mathbb{B}/\!\!/I$ is

$$g \bullet f = g \circ \langle \pi, f \rangle : I \times X \longrightarrow I \times Y \longrightarrow Z$$

We shall often write this fat dot $\bullet$ for composition in simple slices, to distinguish it from composition $\circ$ in $\mathbb{B}$.

(ii) Each object $I \in \mathbb{B}$ determines a functor $I^*: \mathbb{B} \to \mathbb{B}/\!\!/I$ by

$$X \mapsto X \qquad \text{and} \qquad f \mapsto f \circ \pi'$$

An important observation is that initiality with parameters of $1 \xrightarrow{\mathsf{0}} \mathsf{N} \xrightarrow{\mathsf{S}} \mathsf{N}$ is precisely initiality of the diagram $1 \xrightarrow{I^*(\mathsf{0})} \mathsf{N} \xrightarrow{I^*(\mathsf{S})} \mathsf{N}$ in each simple slice $\mathbb{B}/\!\!/I$. This will be generalized below. But first we need to take a closer look at these simple slices.

7.2. Lemma. *Let $\mathbb{B}$ be a category with finite products.*

(i) *The simple slice $\mathbb{B}/\!\!/1$ over the terminal object $1 \in \mathbb{B}$ is isomorphic to $\mathbb{B}$.*

(ii) *Each simple slice $\mathbb{B}/\!\!/I$ has finite products and $I^*: \mathbb{B} \to \mathbb{B}/\!\!/I$ preserves them.*

(iii) *If $\mathbb{B}$ is a distributive category, then so is every simple slice $\mathbb{B}/\!\!/I$. Moreover $I^*: \mathbb{B} \to \mathbb{B}/\!\!/I$ is then a distributive functor.*

(iv) *$\mathbb{B}$ is cartesian closed if and only if each $I^*: \mathbb{B} \to \mathbb{B}/\!\!/I$ has a right adjoint $I \Rightarrow (\_)$. Such a right adjoint necessarily preserves finite products, but need not preserve finite coproducts.*

(v) *There is a full and faithful functor $\mathbb{B}/\!\!/I \to \mathbb{B}/I$ from the simple to the ordinary slice, given by $X \mapsto [\pi: I \times X \to I]$.*

Proof. (i) Since $\mathbb{B}/\!\!/1(X,Y) \cong \mathbb{B}(1 \times X, Y) \cong \mathbb{B}(X,Y)$.

(ii) The object $1 \in \mathbb{B}/\!\!/I$ is terminal, because $\mathbb{B}/\!\!/I(X,1) \cong \mathbb{B}(I \times X, 1)$, which is a singleton.

Similarly, the product of $X, Y \in \mathbb{B}/\!\!/I$ is $X \times Y$, since

$$\begin{array}{rcl} \mathbb{B}/\!\!/I(Z,\ X \times Y) & = & \mathbb{B}(I \times Z,\ X \times Y) \\ & \cong & \mathbb{B}(I \times Z,\ X)\ \times\ \mathbb{B}(I \times Z,\ Y) \\ & = & \mathbb{B}/\!\!/I(Z,\ X)\ \times\ \mathbb{B}/\!\!/I(Z,\ Y) \end{array}$$

(iii) The initial object $0 \in \mathbb{B}$ is also initial in $\mathbb{B}/\!\!/I$, since $\mathbb{B}/\!\!/I(0,X) \cong \mathbb{B}(I \times 0, X) \cong \mathbb{B}(0,X)$, which is a singleton. Here we use the isomorphism $0 \cong I \times 0$ from (a) in Definition 3.1 (i).

Also the coproduct is inherited from $\mathbb{B}$ using the distributivity in (b):

$$\begin{array}{rcl} \mathbb{B}/\!\!/I(X+Y,\ Z) & = & \mathbb{B}(I \times (X+Y),\ Z) \\ & \cong & \mathbb{B}((I \times X) + (I \times Y),\ Z) \\ & \cong & \mathbb{B}(I \times X,\ Z)\ \times\ \mathbb{B}(I \times Y,\ Z) \\ & = & \mathbb{B}/\!\!/I(X,\ Z)\ \times\ \mathbb{B}/\!\!/I(Y,\ Z). \end{array}$$

(iv) A right adjoint $I \Rightarrow (\_)$ to I^*—if it exists—must satisfy

$$\mathbb{B}(X,\ I \Rightarrow Y)\ \cong\ \mathbb{B}/\!\!/I(I^*(X),\ Y)\ =\ \mathbb{B}(I \times X,\ Y)$$

which makes it satisfy the requirements of an exponent functor.

(v) Easy. □

7.3. Remarks on simple and ordinary slice categories. There is a great similarity between simple slice categories $\mathbb{B}/\!\!/I$ and ordinary slice categories $\mathbb{B}/I$. In fact this is our motivation for calling these categories $\mathbb{B}/\!\!/I$ slices as well.

In general, one can describe what it means for an object I in a category $\mathbb{B}$ to 'adjoin an indeterminate' $x{:}\,I$ in the form of an arrow $x{:}\,1 \to I$ to $\mathbb{B}$. Such a description involves a univeral property of a functor from $\mathbb{B}$ to the thus extended category $\mathbb{B}[x{:}\,I]$, see [18]. For particular categories, this functor $\mathbb{B} \to \mathbb{B}[x{:}\,I]$ can be identified concretely. It is

- $I^*{:}\,\mathbb{B} \to \mathbb{B}/\!\!/I$ for categories $\mathbb{B}$ with finite products;
- $I^*{:}\,\mathbb{B} \to \mathbb{B}/I$ for categories $\mathbb{B}$ with finite limits.

We conclude that simple and ordinary slice categories arise in similar ways. Further, the simple slice $\mathbb{B}/\!\!/I$ is the Kleisli category of the comonad $I \times (\_){:}\,\mathbb{B} \to \mathbb{B}$, wheras the ordinary slice $\mathbb{B}/I$ is the Eilenberg-Moore category of this comonad. Thus for a specific (lax) diagram, $\mathbb{B}/\!\!/I$ is the colimit and $\mathbb{B}/I$ is the limit.

In general, the assignment $I \mapsto \mathbb{B}[x{:}\,I]$ extends to an indexed category $\mathbb{B}^{\mathrm{op}} \to \mathbf{Cat}$, using the universal property of $\mathbb{B} \to \mathbb{B}[x{:}\,I]$. By applying the Grothendieck construction one obtains a fibration with $\mathbb{B}$ as base category. In the particular case of $\mathbb{B}$ being a category with finite limits, one obtains the **codomain fibration** $\begin{smallmatrix}\mathbb{B}^{\to}\\ \downarrow\\ \mathbb{B}\end{smallmatrix}$. And for $\mathbb{B}$ with finite products, this leads to what we call the **simple fibration** $\begin{smallmatrix}s(\mathbb{B})\\ \downarrow\\ \mathbb{B}\end{smallmatrix}$ on $\mathbb{B}$. Codomain fibrations are essential for the categorical description of dependent type theories (see [25]), whereas simple fibrations are used for simple type theories (see [16]). More information can be found in [12].

The role which ordinary slices play for categories with finite limits is played by simple slices for categories with finite products. Completely analogous to the above result (iv) that a category $\mathbb{B}$ with finite products is cartesian closed if and only if each $I^*{:}\,\mathbb{B} \to \mathbb{B}/\!\!/I$ has a right adjoint $I \Rightarrow (\_)$, one has that a category $\mathbb{B}$ with finite limits is cartesian closed if and only if each functor $I^*{:}\,\mathbb{B} \to \mathbb{B}/I$ has a right adjoint $\prod_I$.

The approach below deals with 'simple' parameters in simple slices $\mathbb{B}/\!\!/I$. It extends in an obvious way to 'dependent' parameters in ordinary slices $\mathbb{B}/I$. But note that $\mathbb{B}/I$ need not be distributive if $\mathbb{B}$ is. The appropriate coproducts one needs here are called universal.

Perhaps as a warning we should add that the notation $\mathbb{B}/\!\!/I$ is sometimes used for the 'lax comma category' of a 2-category $\mathbb{B}$. Its objects are morphisms $\varphi{:}\,X \to I$ in $\mathbb{B}$ and its morphisms are diagrams

$$\begin{tikzcd} X \arrow[rr] \arrow[dr, "\varphi"'] & \arrow[d, phantom, "\Rightarrow"] & Y \arrow[dl, "\psi"] \\ & I & \end{tikzcd}$$

We can now say that a model of a specification $(\Sigma, \mathcal{E})$ in $\mathbb{B}$ with parameter $I \in \mathbb{B}$ is a distributive functor $\mathcal{M}{:}\,\mathcal{C}\ell(\Sigma,\mathcal{E}) \to \mathbb{B}/\!\!/I$; that is, a model of $(\Sigma,\mathcal{E})$ in the simple slice $\mathbb{B}/\!\!/I$. But in order to get the right notion of initiality with parameters we extend Kan extensions to 'Kan extensions with parameters'. The definition is like for natural numbers with parameters: upon application of $I^*{:}\,\mathbb{B} \to \mathbb{B}/\!\!/I$ one requires initiality (or extension, as below).

7.4. Definition. Consider functors $H{:}\,\mathbb{C} \to \mathbb{D}$ and $\mathcal{N}{:}\,\mathbb{C} \to \mathbb{B}$. A left Kan extension **with parameter** $I \in \mathbb{B}$ of $\mathcal{N}$ along H is a functor $\mathcal{L}{:}\,\mathbb{D} \to \mathbb{B}$ together with a natural transformation $\eta{:}\,\mathcal{N} \to H^*(\mathcal{L}) = \mathcal{L}H$ such that $I^*\mathcal{L}$ with $I^*\eta{:}\,I^*\mathcal{N} \longrightarrow I^*\mathcal{L}H$ in $\mathbb{B}/\!\!/I$ is left Kan extension of $I^*\mathcal{N}$ along H. This means that for each functor $\mathcal{M}{:}\,\mathbb{D} \to \mathbb{B}/\!\!/I$ with $\alpha{:}\,I^*\mathcal{N} \to H^*(\mathcal{M})$ in $\mathbb{B}/\!\!/I$, there is a unique $\overline{\alpha}{:}\,I^*\mathcal{L} \to \mathcal{M}$ with

$$I^*\mathcal{N} \xrightarrow{I^*\eta} I^*\mathcal{L}H \xrightarrow{\overline{\alpha}H} \mathcal{M}H \qquad \text{is} \qquad I^*\mathcal{N} \xrightarrow{\alpha} \mathcal{M}H$$

In a diagram,

$$\begin{tikzcd} \mathbb{C} \arrow[rr, "H"] \arrow[dr, "\mathcal{N}"] \arrow[ddr, bend right, "H^*(\mathcal{M}) = \mathcal{M}H"'] & & \mathbb{D} \arrow[dl, "\mathcal{L}"'] \arrow[ddl, bend left, "\mathcal{M}"] \\ & \mathbb{B} \arrow[d, "I^*"] \arrow[l, phantom, "\overset{\alpha}{\Leftarrow}"] \arrow[r, phantom, "\overset{\overline{\alpha}}{\Rightarrow}"] & \\ & \mathbb{B}/\!\!/I & \end{tikzcd}$$

Notice that for $I = 1$ we get the ordinary notion of Kan extension.

7.5. Definition. Consider a parametrized specification $\phi{:}\,(\Sigma_0, \mathcal{E}_0) \to (\Sigma, \mathcal{E})$ and a distributive category $\mathbb{B}$. The **ϕ-initial model with parameters** on $\mathcal{N}{:}\,\mathcal{C}\ell(\Sigma_0,\mathcal{E}_0) \to \mathbb{B}$ is a distributive functor $\mathcal{L} = \mathrm{Lan}_\phi(\mathcal{N}){:}\,\mathcal{C}\ell(\Sigma,\mathcal{E}) \to \mathbb{B}$ which is left Kan extension with parameter I along $\mathcal{C}\ell(\phi){:}\,\mathcal{C}\ell(\Sigma_0,\mathcal{E}_0) \longrightarrow \mathcal{C}\ell(\Sigma,\mathcal{E})$, for every object $I \in \mathbb{B}$.

What we'll do below is give an appropriate extension of Theorem 6.3 with parameters. The above definition applies to arbitrary specifications and hence to Hagino specifications in particular. This special case is studied by Cockett and Spencer [5, 6, 27] in terms of so-called strong functors. In the remainder we show that their notion of model with parameters (or 'strong' model as they call it) is a special case of the above one.

7.6. Definition. An endofunctor $T: \mathbb{B} \to \mathbb{B}$ on a category $\mathbb{B}$ with finite products is called **strong** if it comes equipped with a **strength** natural transformation with components

$$st_{I,X}: I \times TX \longrightarrow T(I \times X)$$

making the following two diagrams commute.

$$\begin{tikzcd} I \times TX \arrow[r, "st"] \arrow[dr, "\pi'"'] & T(I \times X) \arrow[d, "T(\pi')"] \\ & TX \end{tikzcd} \qquad \begin{tikzcd} I \times (J \times TX) \arrow[r, "id \times st"] \arrow[d, "\alpha", "\wr"'] & I \times T(J \times X) \arrow[r, "st"] & T(I \times (J \times X)) \arrow[d, "T(\alpha)", "\wr"'] \\ (I \times J) \times TX \arrow[rr, "st"] & & T((I \times J) \times X) \end{tikzcd}$$

where α is the obvious isomorphism $\langle\langle \pi, \pi \circ \pi' \rangle, \pi' \circ \pi' \rangle$.

Here we talk about strength for an endofunctor $\mathbb{B} \to \mathbb{B}$ on a category $\mathbb{B}$ with finite products. More generally, strength can be defined in terms of a monoidal structure on $\mathbb{B}$. The notion of a strong monad is crucial in [24]; it is a monad whose underlying functor is strong in such a way that the strength map is compatible (in a suitable sense) with the unit and multiplication.

The next lemma gives a useful result about strong functors; for more information, see Remark 7.14.

7.7. Lemma. *A strong functor $T: \mathbb{B} \to \mathbb{B}$ extends for every $I \in \mathbb{B}$ to a functor*

$$T/\!\!/I: \mathbb{B}/\!\!/I \longrightarrow \mathbb{B}/\!\!/I$$

which is strong again.

In Remark 7.14 below one finds a sharper version of this result. It shows that one can take the above extension to be the defining property of strong functors.

Proof. One defines $T/\!\!/I$ by

$$\begin{array}{rcl} X & \mapsto & TX \\ I \times X \xrightarrow{f} Y & \mapsto & I \times TX \xrightarrow{st} T(I \times X) \xrightarrow{Tf} TY \end{array}$$

By the first diagram in Definition 7.6 one has that $T/\!\!/I$ preserves identities; by the second diagram it preserves composition. This is a bit subtle:

$$\begin{array}{rcl} T/\!\!/I(g) \bullet T/\!\!/I(f) & = & Tg \circ st \circ \langle \pi, Tf \circ st \rangle \\ & = & Tg \circ st \circ id \times Tf \circ \langle \pi, st \rangle \circ id \times \pi' \circ \langle \pi, id \rangle \\ & = & Tg \circ T(id \times f) \circ st \circ id \times st \circ \langle \pi, id \rangle \\ & & \qquad \text{by naturality of } st \\ & = & Tg \circ T(id \times f \circ \alpha^{-1}) \circ T\alpha \circ st \circ id \times st \circ \langle \pi, id \rangle \\ & = & Tg \circ T(id \times f \circ \alpha^{-1}) \circ st \circ \alpha \circ \langle \pi, id \rangle \\ & = & Tg \circ T(id \times f \circ \alpha^{-1}) \circ st \circ \delta \times T(id) \\ & = & Tg \circ T(id \times f \circ \alpha^{-1} \circ \delta \times id) \circ st \\ & = & Tg \circ T(\langle \pi, f \rangle) \circ st \\ & = & T/\!\!/I(g \bullet f) \end{array}$$

As strength map for $T/\!\!/I$ one has $I^*(st) = st \circ \pi'$. □

It is easy to see that the composite $\mathbb{B} \cong \mathbb{B}/\!\!/1 \xrightarrow{T/\!/1} \mathbb{B}/\!\!/1 \cong \mathbb{B}$ is equal to $T: \mathbb{B} \to \mathbb{B}$.

7.8. Lemma. *On a distributive category $\mathbb{B}$, identity functors and constant functors are strong. Moreover, if $T, S: \mathbb{B} \to \mathbb{B}$ are strong, then so are*

$$Y \mapsto T(Y) \times S(Y) \qquad \textit{and} \qquad Y \mapsto T(Y) + S(Y).$$

Proof. The identity functor has the identity natural transformation as strength. A constant functor has the second projection as strength. As strength for $Y \mapsto T(Y) \times S(Y)$ one has

$$I \times (T(Y) \times S(Y)) \xrightarrow{\langle id \times \pi, id \times \pi' \rangle} (I \times T(Y)) \times (I \times S(Y)) \xrightarrow{st \times st} T(I \times Y) \times S(I \times Y)$$

and for $Y \mapsto T(Y) + S(Y)$ one has

$$I \times (T(Y) + S(Y)) \;\cong\; (I \times T(Y)) + (I \times S(Y)) \xrightarrow{st + st} T(I \times Y) + S(I \times Y).$$

□

7.9. Corollary. *Each polynomial functor $T(A)_\sigma: \mathbb{B} \to \mathbb{B}$ as in Definition 4.2 is strong.*

The following definition contains a compact formulation of a notion used by Cockett and Spencer in their description of Hagino specifications with parameters.

7.10. Definition (See [5, 6, 27]). An algebra $\varphi: TU \to U$ for a strong functor $T: \mathbb{B} \to \mathbb{B}$ is called **strongly initial** if for each $I \in \mathbb{B}$, one has that $I^*(\varphi)$ is an initial algebra for $T/\!\!/I: \mathbb{B}/\!\!/I \to \mathbb{B}/\!\!/I$.

There is of course a dual notion of **strongly terminal** coalgebra.

If we spell out strong initiality of $\varphi: TU \to U$ as described above, then we come to the formulation used by Cockett and Spencer. It means that for each parameter object $I \in \mathbb{B}$ and for each $\psi: I \times TV \to V$ in $\mathbb{B}$, there is a unique $h: I \times U \to V$ making the following diagram commute.

$$\begin{tikzcd} I \times TU \arrow[r, "\langle \pi, st \rangle"] \arrow[d, "id \times \varphi"'] & I \times T(I \times U) \arrow[r, "id \times Th"] & I \times TV \arrow[d, "\psi"] \\ I \times U \arrow[rr, "h"] & & V \end{tikzcd}$$

The fact that 'strong initiality' is a special case of our 'initiality with parameters' crucially depends on the following observation.

7.11. Lemma. *Assume a set S of atomic types, plus a type variable X and a type $\sigma \in \overline{S \cup \{X\}}$. A model $A: S \to \mathbb{B}$ gives rise to a strong functor $T(A)_\sigma: \mathbb{B} \to \mathbb{B}$ and thus by Lemma 7.7 to a functor $T(A)_\sigma/\!\!/I: \mathbb{B}/\!\!/I \longrightarrow \mathbb{B}/\!\!/I$ (for each $I \in \mathbb{B}$).*

But since A extends to a model $I^*A: S \to \mathbb{B}/\!\!/I$ we get by the construction in Definition 4.2 an endofunctor on $\mathbb{B}/\!\!/I$; it is written as $T(I^*A)_\sigma: \mathbb{B}/\!\!/I \to \mathbb{B}/\!\!/I$. One has that these two functors are equal, i.e.

$T(A)_\sigma /\!\!/ I = T(I^*A)_\sigma$.

Proof. By induction on σ. Let's do the case $\sigma \equiv \sigma_1 \times \sigma_2$. It's obvious that the two functors act identically on objects. Assume $f: X \to Y$ in $\mathbb{B}/\!\!/I$ (i.e. $f: I \times X \to Y$ in $\mathbb{B}$). Then

$$\begin{array}{rll}
& T(I^*A)_{\sigma_1\times\sigma_2}(f) & \\
= & T(I^*A)_{\sigma_1}(f) \times T(I^*A)_{\sigma_2}(f) & (\text{in } \mathbb{B}/\!\!/I) \\
= & T(I^*A)_{\sigma_1}(f) \times T(I^*A)_{\sigma_2}(f) \circ \langle id \times \pi, id \times \pi' \rangle & (\text{in } \mathbb{B}) \\
= & T(A)_{\sigma_1}/\!\!/I(f) \times T(A)_{\sigma_2}/\!\!/I(f) \circ \langle id \times \pi, id \times \pi' \rangle & (\text{by IH}) \\
= & T(A)_{\sigma_1}(f) \times T(A)_{\sigma_2}(f) \circ st \times st \circ \langle id \times \pi, id \times \pi' \rangle & \\
= & T(A)_{\sigma_1\times\sigma_2}(f) \circ st & \\
= & T(A)_{\sigma_1\times\sigma_2}/\!\!/I(f). & \square
\end{array}$$

7.12. Theorem. *Let $\Sigma = (S \cup \{X\}, \mathsf{constr}: \sigma \to X)$ be an inductive Hagino specification. Then a model of Σ in a distributive category $\mathbb{B}$ is initial with parameters on S if and only if the interpretation of constr is a strongly initial algebra for the induced polynomial functor on $\mathbb{B}$.*

There is a similar result for strong terminal models of co-inductive Hagino specifications.

Proof. Assume a model of Σ in $\mathbb{B}$, given by a model $A: S \to \mathbb{B}$, $U \in \mathbb{B}$ of $S \cup \{X\}$ and an algebra $\varphi = [\![\mathsf{constr}]\!]: T(A)_\sigma(U) \to U$. Then

$$\begin{array}{ll}
\multicolumn{2}{l}{\varphi \text{ is initial with parameters}} \\
\stackrel{\text{def}}{\Longleftrightarrow} & \text{for each } I \in \mathbb{B},\ \varphi \text{ is left Kan extension with parameter } I \\
\stackrel{7.4}{\Longleftrightarrow} & \text{for each } I \in \mathbb{B},\ I^*(\varphi) \text{ is left Kan extension} \\
\stackrel{6.3}{\Longleftrightarrow} & \text{for each } I \in \mathbb{B},\ I^*(\varphi) \text{ is initial algebra of } T(I^*A)_\sigma \\
\stackrel{7.11}{\Longleftrightarrow} & \text{for each } I \in \mathbb{B},\ I^*(\varphi) \text{ is initial algebra of } T(A)_\sigma/\!\!/I \\
\stackrel{\text{def}}{\Longleftrightarrow} & \varphi \text{ is strongly initial algebra of } T(A)_\sigma. \quad \square
\end{array}$$

The following result can also be found in [5].

7.13. Lemma. *Let $\mathbb{B}$ be a distributive category and $T: \mathbb{B} \to \mathbb{B}$ a strong functor.*

(i) *Any terminal coalgebra $X \to TX$ is automatically strongly terminal.*

(ii) *In case $\mathbb{B}$ is cartesian closed, then every initial algebra $TX \to X$ is strongly initial.*

Proof. (i) Suppose $\varphi: TX \to X$ is a terminal coalgebra. For a map $\psi: I \times Y \to TY$ we seek a unique $h: I \times Y \to X$ with $\varphi \circ h = Th \circ st \circ \langle \pi, \psi \rangle$. Let ψ' be the coalgebra $I \times Y \to T(I \times Y)$ given by $\psi' = st \circ \langle \pi, \psi \rangle$. There is then a unique morphism of coalgebras $\psi' \to \varphi$, which is as required.

(ii) Let $\varphi: TX \to X$ be initial T-algebra. Given $\psi: I \times TY \to Y$, we seek a unique $h: I \times X \to Y$ with $\psi \circ id \times Th \circ \langle \pi, st \rangle = h \circ id \times \varphi$. There is an algebra $\psi': T(I \Rightarrow Y) \to (I \Rightarrow Y)$ obtained by abstraction from the composite

$$I \times T(I \Rightarrow Y) \xrightarrow{\langle \pi, st \rangle} I \times T(I \times (I \Rightarrow Y)) \xrightarrow{id \times T(ev)} I \times TY \xrightarrow{\psi} Y$$

One obtains a unique $k: X \to (I \Rightarrow Y)$ forming a morphism of algebras $\varphi \to \psi'$. Then $h = ev \circ id \times h: I \times X \to Y$ is as required. $\square$

B. Jacobs/Parameters and parametrization

In particular, we have that initial algebras / terminal coalgebras in sets, posets or bottomless dcpo's are always strongly initial / terminal.

A result like in (ii) can also be proved for *algebraic* specifications: a model $\mathcal{M}$ of an algebraic specification $(\Sigma, \mathcal{E})$ in a cartesian closed category $\mathbb{B}$ (i.e. a finite product preserving functor $\mathcal{M}: \mathcal{C}\ell_a(\Sigma, \mathcal{E}) \to \mathbb{B}$) is initial if and only if it is initial with parameters. For the only-if-part, consider a model $\mathcal{N}: \mathcal{C}\ell_a(\Sigma, \mathcal{E}) \to \mathbb{B}/\!\!/I$ in the simple slice over I. It gives rise to a model in $\mathbb{B}$,

$$I \Rightarrow \mathcal{N} \stackrel{\text{def}}{=} (I \Rightarrow (\_)) \circ \mathcal{N}: \mathcal{C}\ell_a(\Sigma, \mathcal{E}) \longrightarrow \mathbb{B}/\!\!/I \longrightarrow \mathbb{B}$$

where $I \Rightarrow (\_)$ is the right adjoint to I^* given by exponentiation, see Lemma 7.2 (iv). Note that $I \Rightarrow \mathcal{N}$ preserves finite products again and is thus a model of $(\Sigma, \mathcal{E})$ in $\mathbb{B}$. There is an isomorphism between natural transformations $I^*(\mathcal{M}) \to \mathcal{N}$ and $\mathcal{M} \to (I \Rightarrow \mathcal{N})$; but $\mathcal{M}$ is initial among models in $\mathbb{B}/\!\!/I$. That is, $\mathcal{M}$ is initial with parameters.

This argument does not work for distributive functors, since $I \Rightarrow \mathcal{N}$ need not be distributive if $\mathcal{N}$ is.

7.14. Remarks on strength. Recall the simple fibration $\begin{smallmatrix} s(\mathbb{B}) \\ \downarrow \\ \mathbb{B} \end{smallmatrix}$ arising from $I \mapsto \mathbb{B}/\!\!/I$ as described in Remark 7.3. One can show that there is a bijective correspondence between strong functors $\mathbb{B} \to \mathbb{B}$ and fibred functors $\begin{smallmatrix} s(\mathbb{B}) \\ \downarrow \\ \mathbb{B} \end{smallmatrix} \to \begin{smallmatrix} s(\mathbb{B}) \\ \downarrow \\ \mathbb{B} \end{smallmatrix}$. The latter are functors $S: s(\mathbb{B}) \to s(\mathbb{B})$ for which one has a commuting triangle

$$\begin{tikzcd}
s(\mathbb{B}) \arrow[rr, "S"] \arrow[dr] & & s(\mathbb{B}) \arrow[dl] \\
& \mathbb{B} &
\end{tikzcd}$$

and which preserves cartesian morphisms[8]. This tells us that strong functors are functors which are appropriately defined in each context—and not functors which are enriched; this is something completely different. A comparable result is that $\mathbb{B}$ has distributive coproducts (i.e. coproducts which are appropriately defined in each context, i.e. satisfying (a)+(b) in Definition 3.1 (i)) if and only if the simple fibration $\begin{smallmatrix} s(\mathbb{B}) \\ \downarrow \\ \mathbb{B} \end{smallmatrix}$ has fibred coproducts. For more information on these fibred concepts, see e.g. [15].

The above Theorem 7.12 says that the concept of a strong functor is not necessary for the description of Hagino specifications with parameters. After all, Definition 7.5 does not involve any strength. What it does involve (implicitly) is an appropriate description in every context: a model $\mathcal{L}: \mathcal{C}\ell(\Sigma, \mathcal{E}) \to \mathbb{B}$ which is Kan extension with parameters corresponds to a 'fibrewise' model consisting of a collection of Kan extensions $\mathcal{L}_I: \mathcal{C}\ell(\Sigma, \mathcal{E}) \to \mathbb{B}/\!\!/I$ in each fibre over I, which are preserved by reindexing functors $u^*: \mathbb{B}/\!\!/J \to \mathbb{B}/\!\!/I$ for $u: I \to J$ (i.e. $u^* \circ \mathcal{L}_J \cong \mathcal{L}_I$).

Acknowledgements. I wish to thank Robin Cockett, Peter Knijnenburg and Dominic Verity for helpful discussions. Special thanks to Horst Reichel for spotting a mistake in an earlier version.

[8] A similar result exists for strong monads (as used in [24]); it is generally attributed to Plotkin.

References

[1] M.A. Arbib and E.G. Manes. Parametrized data types do not need highly constrained parameters. *Inf. & Contr.*, 52:139–158, 1982.

[2] A. Carboni, S. Lack, and R.F.C. Walters. Introduction to extensive and distributive categories. *Journ. Pure Appl. Algebra*, 84(2):145–158, 1993.

[3] J.R.B. Cockett. Introduction to distributive categories. *Math. Struct. Comp. Sci.*, 3:277–307, 1993.

[4] J.R.B. Cockett and T. Fukushima. About charity. Technical Report 92/480/18, Dep. Comp. Sci., Univ. Calgary, 1992.

[5] J.R.B. Cockett and D. Spencer. Strong categorical datatypes I. In R.A.G. Seely, editor, *Category Theory 1991*, number 13 in CMS Conference Proceedings, pages 141–169, 1992.

[6] J.R.B. Cockett and D. Spencer. Strong categorical datatypes II: A term logic for categorical programming. Manuscript, 1992.

[7] R.L. Crole and A.M. Pitts. New foundations for fixpoint computations: Fix-hyperdoctrines and the fix-logic. *Inf. & Comp.*, 98(2):171–210, 1992.

[8] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification I: Equations and Initial Semantics*. Number 6 in EATCS Monographs. Springer, Berlin, 1985.

[9] T. Hagino. *A categorical programming language*. PhD thesis, Univ. Edinburgh, 1987. Techn. Rep. 87/38.

[10] T. Hagino. A typed lambda calculus with categorical type constructors. In D.H. Pitt, A Poigné, and D.E. Rydeheard, editors, *Category and Computer Science*, number 283 in Lect. Notes Comp. Sci., pages 140–157. Springer, Berlin, 1987.

[11] C. Hermida. *Fibrations, Logical Predicates and Indeterminates*. PhD thesis, Univ. Edinburgh, 1993. Techn. rep. LFCS-93-277. Also available as Aarhus Univ. DAIMI Techn. rep. PB-462.

[12] C. Hermida and B. Jacobs. Fibrations with indeterminates: Contextual and functional completeness of polymorphic lambda calculi. *Math. Struct. Comp. Sci.*, 1995.

[13] J.M.E Hyland. First steps in synthetic domain theory. In A. Carboni, M.C. Pedicchio, and G. Rosolini, editors, *Como Conference on Category Theory*, number 1488 in Lect. Notes Math., pages 131–156. Springer, Berlin, 1991.

[14] J.A Goguen and R. Burstall. Institutions: Abstract model theory for specification and programming. *Journ. ACM*, 39(1):95–146, 1992.

[15] B. Jacobs. *Categorical Type Theory*. PhD thesis, Univ. Nijmegen, 1991.

[16] B. Jacobs. Simply typed and untyped lambda calculus revisited. In M.P. Fourman, P.T. Johnstone, and A.M. Pitts, editors, *Applications of Categories in Computer Science*, number 177 in LMS, pages 119–142. Cambridge Univ. Press, 1992.

B. Jacobs/Parameters and parametrization

[17] H. Kaphengst and H. Reichel. Operative Theorien und Kategorien von operativen Systemen. In H.J. Hoehnke, editor, *Studien zur Algebra und ihren Anwendungen*, pages 41–56, Berlin, 1972. Akademie-Verlag.

[18] J. Lambek and P.J. Scott. *Introduction to higher order Categorical Logic*. Number 7 in Studies in Adv. Math. Cambridge Univ. Press, 1986.

[19] S. Mac Lane. *Categories for the Working Mathematician*. Springer, Berlin, 1971.

[20] S. Lang. *Algebra*. Addison Wesley, 2^{nd} rev. edition, 1984.

[21] F.W. Lawvere. Functorial semantics. *Proc. Nat. Acad. Sci. USA*, 50:869–872, 1963.

[22] D.J. Lehmann and M.B. Smyth. Algebraic specification of data types: A synthetic approach. *Math. Systems Theory*, 14:97–139, 1981.

[23] E.G. Manes and M.A. Arbib. *Algebraic Appoaches to Program Semantics*. AKM Theor. Comp. Sci.,. Springer, Berlin, 1986.

[24] E. Moggi. Notions of computation and monads. *Inf. & Comp.*, 93(1):55–92, 1991.

[25] R.A.G. Seely. Locally cartesian closed categories and type theories. *Math. Proc. Cambridge Phil. Soc.*, 95:33–48, 1984.

[26] M.B. Smyth and G.D. Plotkin. The category theoretic solution of recursive domain equations. *SIAM Journ. Comput.*, 11:761–783, 1982.

[27] D.L. Spencer. *Categorical programming with functorial strength*. PhD thesis, Oregon graduate inst. of Sci. & Techn., 1993.

[28] R.F.C. Walters. Data types in distributive categories. *Bull. Austr. Math. Soc.*, 40:79–82, 1989.

[29] R.F.C. Walters. *Categories and Computer Science*. Carslaw Publications, Sydney, 1991. Also available as: Cambridge Computer Science Text 28, 1992.

[30] R.F.C. Walters. An imperative language based on distributive categories. *Math. Struct. Comp. Sci.*, 2:249–256, 1992.