States and exceptions are dual effects

Jean-Guillaume Dumas, Dominique Duval, Laurent Fousse, Jean-Claude Reynaud

States and exceptions are dual effects

Jean-Guillaume Dumas
LJK, Université de Grenoble
BP 53, 38041 GRENOBLE Cedex 9, France
Jean-Guillaume .Dumas@imag.fr

Dominique Duval
LJK, Université de Grenoble
BP 53, 38041 GRENOBLE Cedex 9, France
Dominique.Duval@imag.fr

Laurent Fousse
LJK, Université de Grenoble
BP 53, 38041 GRENOBLE Cedex 9, France
Laurent.Fousse@imag.fr

Jean-Claude Reynaud
Malhivert, 38640 CLAIX, France
Jean-Claude.Reynaud@imag.fr

January 21, 2010

Abstract

Global states and exceptions form two basic computational effects. In this paper it is proved that they can be seen as dual to each other: the lookup and update operations for global states are dual to the raise and handle operations for exceptions, respectively. In order to get this result we use a monad for exceptions and a comonad for global states.

1 Introduction

The denotational semantics of languages with *computational effects* can be expressed in categorical terms, thanks to *monads* [7, 8]: for instance, global states correspond to the monad on $\mathbf{Set}$ with endofuntor $T(X) = (X \times St)^{St}$ where St is the set of states, while exceptions correspond to the monad on $\mathbf{Set}$ with endofuntor $T(X) = X + Exc$ where Exc is the set of exceptions. Each computational effect comes with its associated operations: for instance, the operations lookup and update for the states, the operations raise and handle for the exceptions [10]. Another approach is based on *Lawvere theories*; this point of view is related to monads by an adjunction [6].

A computational effect relies on a kind of zooming process from a *hidden* point of view, where the effect is partially hidden, to an *explicit* point of view, where the effect appears explicitly. For instance, there is no explicit type S for states in an imperative language (the state is hidden), however the set of states St does appear explicitly in the denotational semantics for global states. In this paper, we focus on two computational effects: global states and exceptions, and we prove that they are mutually *dual*, both from the explicit and from the hidden point of view.

First in section 2 these two effects are treated in an *explicit* way. We prove that they can be seen as mutually *dual*, in the sense that their denotational semantics is defined by the models of two dual specifications. We use *sketches* as specifications, since they provide a clean treatement of sums as dual to products. The idea is that $X \times St$ for a fixed St is dual to $X + Exc$ for a fixed Exc,

and that this duality can be extended to the operations: lookup is dual to raise, and update is dual to handle. Then in section 3 we look at the effects from the *hidden* point of view. For this purpose, as in the classical approach we use the monad $T(X) = X + Exc$ for the exceptions [7], but we use the dual *comonad* $T(X) = X \times St$ for the global states. We prove that the duality can also be easily expressed from this point of view. Our notations and results are summarized in appendix A.

To our knowledge, the fact that global states and exceptions are dual computational effects is a new result. It has been necessary to use both monads and comonads for getting this result in its right setting, i.e., when the effects are hidden. We would like to suggest that, while monads are indeed one major tool for expliciting effects, additional tools like comonads or other categorical features may also be helpful.

2 Duality of effects, explicitly

In this section the effects appear explicitly in the specifications. We use sketches as specifications [1]: a sketch with finite products $\mathbf{S}$ for global states and dually a sketch with finite sums $\mathbf{E}$ for exceptions. The category of models of a sketch $\mathbf{Sp}$ with values in a category $\mathbf{C}$ is denoted $\mathrm{Mod}(\mathbf{Sp}, \mathbf{C})$. When $\mathbf{Sp}_0$ is a subsketch of $\mathbf{Sp}$ and M_0 a model of $\mathbf{Sp}_0$, we denote $\mathrm{Mod}(\mathbf{Sp}, \mathbf{C})|_{M_0}$ the subcategory of $\mathrm{Mod}(\mathbf{Sp}, \mathbf{C})$ made of the models M of $\mathbf{Sp}$ with values in $\mathbf{C}$ which coincide with M_0 on $\mathbf{Sp}_0$ and of the morphisms of models which extend the identity of M_0. We often write $M\ldots$ for $M(\ldots)$ when M is a model of a sketch. For set-valued models we denote $\mathrm{Mod}(\mathbf{Sp}) = \mathrm{Mod}(\mathbf{Sp}, \mathbf{Set})$.

2.1 Global states, explicitly

The explicit specification for global states is defined in several steps.

Definition 2.1. Let Loc be a set, called the set of *locations*.

- Let $\mathbf{S}_p$ denote the specification simply made of a point V_i for each $i \in Loc$, called the type of *values* of i.
- Let $\mathbf{S}_q$ denote the specification made of $\mathbf{S}_p$, a point S, called the type of *states* and an arrow $l_i : S \to V_i$ for each $i \in Loc$, called the *lookup* at i. Let $l = (l_j)_{j \in Loc} : S \to \prod_j V_j$ denote the tuple of the l_i's.
- For each location i, let $\varphi_i : V_i \times S \to \prod_j V_j$ be defined as $\varphi_i = (\varphi_{i,j})_{j \in Loc}$ where $\varphi_{i,j} : V_i \times S \to V_j$ is:

  $$\varphi_{i,i} = pr_{V_i} \text{ and } \varphi_{i,j} = l_j \circ pr_S \text{ when } j \neq i$$

  where $pr_{V_i} : V_i \times S \to V_i$ and $pr_S : V_i \times S \to S$ denote the projections.
- The *explicit specification for global states* $\mathbf{S}$ is made of $\mathbf{S}_q$ and an arrow $u_i : V_i \times S \to S$ for each $i \in Loc$, called the *update* at i, together with the equation:

  $$l \circ u_i = \varphi_i \ . \tag{1}$$

Remark 2.2. A model M_p of $\mathbf{S}_p$ in a category $\mathbf{C}$ with finite products is simply made of an object $Val_i = M_p V_i$ for each $i \in Loc$. A model M_q of $\mathbf{S}_q$ in $\mathbf{C}$ is made, in addition, of a set $St = M_q S$ and a morphism $M_q l : St \to \prod_j Val_j$. For getting a model M of $\mathbf{S}$ in $\mathbf{C}$ we add for each $i \in Loc$ a morphism $Mu_i : Val_i \times St \to St$ such that $Ml \circ Mu_i = M\varphi_i$ for each $i \in Loc$. When $\mathbf{C} = \mathbf{Set}$, this means that for each $x_i \in Val_i$ and $s \in St$, $Ml_i(Mu_i(x_i, s)) = x_i$ and $Ml_j(Mu_i(x_i, s)) = Ml_j(s)$ when $j \neq i$.

Proposition 2.3. *Let $\mathbf{C}$ be a category with finite products. Let M_p be a model of $\mathbf{S}_p$ in $\mathbf{C}$, made of an object $Val_i = M_p V_i$ for each $i \in Loc$. There is a terminal model of $\mathbf{S}_q$ in $\mathbf{C}$ above M_p, denoted $[[\,]]$, it is such that $[[S]] = \prod_j Val_j$ and $[[l]] : [[S]] \to \prod_j Val_j$ is the identity. Then $[[\,]]$ can be extended in a unique way as a model of $\mathbf{S}$ in $\mathbf{C}$ above M_p, still denoted $[[\,]]$: for each $i \in Loc$ the morphism $[[u_i]] : Val_i \times [[S]] \to [[S]]$ is defined by $[[u_i]] = [[\varphi_i]]$. In addition, $[[\,]]$ is a terminal model of $\mathbf{S}$ above M_p.*

Proof. Let M_q be a model of $\mathbf{S}_q$ above M_p, then the unique morphism $m : M_q \to [[\,]]$ above M_p is defined by $m_S = M_q(l) : M_q(S) \to \prod_j Val_j$, so that $[[\,]]$ is a terminal model of $\mathbf{S}_q$ above M_p. Since $[[l]]$ is the identity, equations (1) determine $[[u_i]] = [[\varphi_i]]$ for each location i, so that $[[\,]]$ extends in this unique way as a model of $\mathbf{S}$ above M_p. Now let M be a model of $\mathbf{S}$ above M_p, it is easy to

check that $m_S \circ Mu_i = [[\varphi_i]] \circ (id_{Val_i} \times m_S)$ for each location i, so that m is the unique morphism $m : M \to [[\,]]$ above M_p.

$$\begin{array}{ccc} MS & \xrightarrow{Ml} & \prod Val_j \\ {\scriptstyle Ml}\downarrow & = & \downarrow{\scriptstyle id} \\ \prod Val_j & \xrightarrow{id} & \prod Val_j \end{array} \qquad \begin{array}{ccc} Val_i \times MS & \xrightarrow{Mu_i} & MS \\ {\scriptstyle id \times Ml}\downarrow & = & \downarrow{\scriptstyle Ml} \\ Val_i \times \prod Val_j & \xrightarrow{[[\varphi_i]]} & \prod Val_j \end{array}$$

□

Remark 2.4. When $\mathbf{C} = \mathbf{Set}$, in the terminal model the function $[[u_i]] : Val_i \times \prod_j Val_j \to \prod_j Val_j$ maps $(x_i, (x_j)_j)$ to $(z_j)_j$ where $z_i = x_i$ and $z_j = y_j$ when $j \neq i$. When in addition the set Val_j does not depend on j, say $Val_j = Val$ for each j, then:

$$[[S]] = Val^{Loc} .$$

It follows from remark 2.2 and proposition 2.3 that the next definition corresponds to the usual semantics of global states.

Definition 2.5. Let $\mathbf{C}$ be a category with finite products. Let M_p be a model of $\mathbf{S}_p$ with values in $\mathbf{C}$. The category of *loose semantics for global states in* $\mathbf{C}$ *above* M_p is the category $\mathrm{Mod}(\mathbf{S}, \mathbf{C})|_{M_p}$ of models of $\mathbf{S}$ in $\mathbf{C}$ above M_p. The *terminal semantics for global states in* $\mathbf{C}$ *above* M_p is the terminal model of $\mathbf{S}$ in $\mathbf{C}$ above M_p.

2.2 Exceptions, explicitly

Now the *explicit specification for exceptions* is defined as the *dual* of the explicit specification for global states. For readability, the names of the points and arrows are changed.

Let $Etype$ be a set, called the set of *exception types*.

Definition 2.6.

- Let $\mathbf{E}_p$ denote the specification simply made of a point P_i for each $i \in Etype$, called the type of *parameters* for exceptions of type i.
- Let $\mathbf{E}_q$ denote the specification made of $\mathbf{E}_p$, a point E, called the type of *exceptions* and an arrow $r_i : P_i \to E$ for each $i \in Etype$, called the *raising* of exceptions of type i. Let $r = [r_j]_{j \in Etype} : \sum_j P_j \to E$ denote the cotuple of the r_i's.
- For each location i, let $\psi_i : \sum_j P_j \to P_i + E$ be defined as $\psi_i = [\psi_{i,j}]_{j \in Etype}$ where $\psi_{i,j} : P_j \to P_i + E$ is:

  $$\psi_{i,i} = in_{P_i} \text{ and } \psi_{i,j} = in_{E,i} \circ r_j \text{ when } j \neq i$$

  where $in_{P_i} : P_i \to P_i + E$ and $in_{E,i} : E \to P_i + E$ denote the coprojections.
- The *explicit specification for exceptions* $\mathbf{E}$ is made of $\mathbf{E}_q$ and an arrow $h_i : E \to P_i + E$ for each $i \in Etype$, called the *handling* of exceptions of type i, together with the equation:

  $$h_i \circ r = \psi_i \, . \tag{2}$$

Remark 2.7. A model M_p of $\mathbf{E}_p$ in a category $\mathbf{C}$ with finite sums is simply made of an object $Par_i = M_p P_i$ for each $i \in Etype$. A model M_q of $\mathbf{E}_q$ in $\mathbf{C}$ is made, in addition, of an object $Exc = M_q E$ and a morphism $M_q r : \sum_j Par_j \to Exc$. For getting a model M of $\mathbf{E}$ in $\mathbf{C}$ we add for each $i \in Etype$ a morphism $Mh_i : Exc \to Par_i + Exc$ such that $Mh_i \circ Mr = M\psi_i$ for each $i \in Etype$. When $\mathbf{C} = \mathbf{Set}$, this means that (writing explicitly the inclusions as $in_{Par_i} : Par_i \to Par_i + Exc$ and $in_{Exc,i} : Exc \to Par_i + Exc$ in order to avoid ambiguity) $Mh_i(Mr_i(x_i)) = in_{Par_i}(x_i)$ for each $x_i \in Par_i$ and $Mh_i(Mr_j(x_j)) = in_{Exc,i}(Mr_j(x_j))$ for each $j \in Etype$ such that $j \neq i$ and each $x_j \in P_j$. This means that the handling function Mh_i runs as follows: for each $e \in Exc$, the value $Mh_i(e)$ says whether e is an exception of type i, if so then $Mh_i(e)$ returns the parameter $x_i \in Par_i$ such that $e = Mr_i(x_i)$, otherwise $Mh_i(e)$ returns $e \in Exc$ without analyzing it any further.

Proposition 2.8. *Let* $\mathbf{C}$ *be a category with finite sums. Let* M_p *be a model of* $\mathbf{E}_p$ *in* $\mathbf{C}$*, made of an objec* $Par_i = M_p P_i$ *for each* $i \in Etype$*. There is an initial model of* $\mathbf{E}_q$ *in* $\mathbf{C}$ *above* M_p*, denoted* $[[\,]]$*, it is such that* $[[E]] = \sum_j Par_j$ *and* $[[r]] : \sum_j Par_j \to [[E]]$ *is the identity. Then* $[[\,]]$ *can be extended in a unique way as a model of* $\mathbf{E}$ *in* $\mathbf{C}$ *above* M_p*, still denoted* $[[\,]]$*: for each* $i \in Etype$ *the morphism* $[[h_i]] : [[E]] \to Par_i + [[E]]$ *is defined by* $[[h_i]] = [[\psi_i]]$*. In addition,* $[[\,]]$ *is an initial model of* $\mathbf{E}$ *above* M_p*.*

Proof. This proof is dual to the proof of proposition 2.3.

The corresponding diagrams are:

$$\begin{array}{ccc} \sum Par_j & \xrightarrow{Mr} & ME \\ {\scriptstyle id}\uparrow & = & \uparrow{\scriptstyle Mr} \\ \sum Par_j & \xrightarrow{id} & \sum Par_j \end{array} \qquad \begin{array}{ccc} ME & \xrightarrow{Mh_i} & P_i + ME \\ {\scriptstyle Mr}\uparrow & = & \uparrow{\scriptstyle id+Mr} \\ \sum Par_j & \xrightarrow{[[\psi_i]]} & P_i + \sum Par_j \end{array}$$

□

Remark 2.9. When $\mathbf{C} = \mathbf{Set}$, in the initial model the function $[[h_i]]$ maps $x_i \in Par_i$ to $x_i \in Par_i$ (in the first summand) and $x_j \in \sum_j Par_j$ to $x_i j \in \sum_j Par_j$ (in the second summand) when $j \neq i$. When in addition the set Par_j does not depend on j, say $Par_j = Par$ for each j, then:

$$[[E]] = Etype \times Par\ .$$

When in addition $Par = 1$, then $[[E]] = Etype$.

We claim that the next definition corresponds to the usual semantics of exceptions in programming languages. This claim is supported by remark 2.13.

Definition 2.10. Let $\mathbf{C}$ be a category with finite sums. Let M_p be a model of $\mathbf{E}_p$ with values in $\mathbf{C}$. The category of *loose semantics for exceptions in* $\mathbf{C}$ *above* M_p is the category $\mathrm{Mod}(\mathbf{E}, \mathbf{C})|_{M_p}$ of models of $\mathbf{E}$ in $\mathbf{C}$ above M_p. The *initial semantics for exceptions in* $\mathbf{C}$ *above* M_p is the initial model of $\mathbf{E}$ in $\mathbf{C}$ above M_p.

Remark 2.11. Let us come back to the usual meaning of exceptions, in an explicit set-valued context. Let Exc be a set, called the *set of exceptions*, with a function $r_i : Par_i \to Exc$ for *raising an exception of type* i. Let $f : X \to Y + Exc$ be some function; for each $x \in X$, if $f(x) = e \in Exc$ then we say that $f(x)$ *raises the exception* e. Let $i \in Etype$ and let $g : X \times Par_i \to Y + Exc$, then g may be used to handle an exception raised by f if this exception is of type i; it should be noted that g itself may raise an exception. The fact of using g for handling an exception of type i raised by f means that instead of $f : X \to Y + E$ we call a function $H : X \to Y + E$, which may be denoted $H = f$ `handle` $[i \Rightarrow g]$, defined as follows. For each $x \in X$:

- if $f(x)$ does not raise any exception then $H(x) = f(x) \in Y$,
- otherwise if $f(x)$ raises an exception of the form $r_i(x_i)$ for some $x_i \in Par_i$, then $H(x) = g(x, x_i) \in Y + Exc$,
- otherwise (i.e., if $f(x)$ raises an exception e which is not of type i), then $H(x) = f(x) \in Exc$.

The handling of several types of exceptions is easily obtained by iterating the construction f `handle` $[i \Rightarrow g]$.

We are going to generalize this construction to any extensive category. Following [2], we define an *extensive category* as a category with finite sums where the sums are *well-behaved*, in the sense that for each commutative diagram:

$$\begin{array}{ccc} X_1 & \longrightarrow & Y_1 \\ \downarrow & = & \downarrow \\ X & \longrightarrow & Y_1 + Y_2 \\ \uparrow & = & \uparrow \\ X_0 & \longrightarrow & Y_2 \end{array}$$

where the right column is a sum, the two squares are pullbacks if and only if the left column is a sum.

Definition 2.12. Let $\mathbf{C}$ be an extensive category. Let us consider a model M of the specification $\mathbf{E}$ with values in $\mathbf{C}$, and let $Par_i = MP_i$ for each i and $Exc = ME$. Let X, Y be two objects and $f : X \to Y + Exc$, $g : X \times Par_i \to Y + Exc$ two morphisms in $\mathbf{C}$. Let us decompose $X = X_1 + X_i + X_{\overline{i}}$ thanks to the well-behaved property of sums, applied twice as follows (note: Mh_i is used in the second diagram):

$$\begin{array}{ccc} X_1 & \xrightarrow{f_1} & Y \\ {\scriptstyle in_1}\downarrow & = & \downarrow{\scriptstyle in_Y} \\ X & \xrightarrow{f} & Y + Exc \\ {\scriptstyle in_0}\uparrow & = & \uparrow{\scriptstyle in_{Exc}} \\ X_0 & \xrightarrow{f_0} & Exc \end{array}$$

$$\begin{array}{ccccc} X_i & & \xrightarrow{f_i} & & Par_i \\ {\scriptstyle in_i}\downarrow & & = & & \downarrow{\scriptstyle in_{Par_i}} \\ X_0 & \xrightarrow{f_0} & Exc & \xrightarrow{Mh_i} & Par_i + Exc \\ {\scriptstyle in_{\overline{i}}}\uparrow & & = & & \uparrow{\scriptstyle in_{Exc,i}} \\ X_{\overline{i}} & & \xrightarrow{f_{\overline{i}}} & & Exc \end{array}$$

We define three morphisms:

- $H_1 : X_1 \to Y + Exc$ as $H_1 = f \circ in_1 = in_Y \circ f_1$,
- $H_i : X_i \to Y + Exc$ as $H_i = g \circ (in_0 \circ in_i, f_i)$,
- $H_{\overline{\imath}} : X_{\overline{\imath}} \to Y + Exc$ as $H_{\overline{\imath}} = in_{Exc} \circ f_{\overline{\imath}}$.

Then f `handle` $[i \Rightarrow g] : X \to Y + Exc$ is defined as $[H_1|H_i|H_{\overline{\imath}}] : X_1 + X_i + X_{\overline{\imath}} \to Y + Exc$.

Remark 2.13. When $\mathbf{C} = \mathbf{Set}$, both functions denoted f `handle` $[i \Rightarrow g]$ in definition 2.12 and in remark 2.11 coincide.

2.3 Duality, explicitly

Now we can state our main result, from the explicit point of view on effects. Indeed:

- the specification $\mathbf{S}$ for global states (definition 2.1) and the specification $\mathbf{E}$ for exceptions (definition 2.6) are dual,
- categories with finite products (denoted $\mathbf{C}_S$) and categories with finite sums (denoted $\mathbf{C}_E$) are dual,
- the category $\mathrm{Mod}(\mathbf{S}, \mathbf{C}_S)|_{M_p}$ and the category $\mathrm{Mod}(\mathbf{E}, \mathbf{C}_E)|_{M_p}$ are dual,
- terminality and initiality are dual.

The next result follows immediately from these remarks and from the definitions of the semantics (definitions 2.1 and 2.6).

Theorem 2.14.
- *The loose semantics for global states and the loose semantics for exceptions are dual.*
- *The terminal semantics for global states and the initial semantics for exceptions are dual.*

3 Duality of effects

In this section we define global states and exceptions as effects with hidden types S and E, respectively. We show that their semantics, as defined in section 2, can also be defined directly from this hidden point of view, so that theorem 2.14 is also a theorem about the duality of effects from the hidden point of view. In order to safely hide the types S and E from the specifications, we have to distinguish three kinds of arrows in these specifications; then we say that the specifications are decorated. Moreover, we define a decorated category as a category with three kinds of morphisms satisfying some compatibility properties. Then we can define decorated models and prove that we recover, from the hidden point of view, the semantics of global states and the semantics of exceptions.

3.1 Decorated categories and specifications

Given a monad (T, η, μ) (or simply T) on a category $\mathbf{C}$, the canonical functor from $\mathbf{C}$ to the Kleisli category $\mathbf{C}_T$ of T is the identity on objects. If in addition the *mono requirement* is satisfied, i.e., if η_X is a monomorphism in $\mathbf{C}$ for each object X in $\mathbf{C}$, then this canonical functor is faithful, so that up to isomorphism it is an inclusion. In [7], the morphisms of $\mathbf{C}_T$ are called the *computations* and the morphisms in $\mathbf{C}$ the *values* (so that each value is a computation). The values are sometimes also called the *pure* morphisms. This classification of morphisms is now generalized (for a more subtle use of the notion of decoration, see [3]).

Definition 3.1. In this paper, a *decorated specification* $\mathbf{Sp}^{dec}$ is a sketch where each arrow has at least one decoration $d \in \{p, q, r\}$. A *decorated category* $\mathbf{C}^{dec}$ is made of three nested categories with the same objects $\mathbf{C}^p \subseteq \mathbf{C}^q \subseteq \mathbf{C}^r$ (we use "$\subseteq$" for an inclusion which is the identity on objects). A morphism f in $\mathbf{C}^d$, for every $d \in \{p, q, r\}$, is denoted f^d, and the symbol d is called a *decoration* of f. Clearly every f^p is also an f^q and every f^q is also an f^r, and the identities are in $\mathbf{C}^p$. A *decorated model* of a decorated specification $\mathbf{Sp}^{dec}$ with values in a decorated category $\mathbf{C}^{dec}$ is defined like a model of a specification in a category, which in addition preserves the decorations. This gives rise to the category of $\mathrm{Mod}^{dec}(\mathbf{Sp}^{dec}, \mathbf{C}^{dec})$.

Decorated categories can be built from monads and dually from comonads, as follows.

Definition 3.2. Let $\mathbf{C}$ be a category and (T, η, μ) (or simply T) a monad on $\mathbf{C}$ satisfying the mono requirement (each η_X is a mono). Then $D_T(\mathbf{C}) = \mathbf{C}_T^p \subseteq \mathbf{C}_T^q \subseteq \mathbf{C}_T^r$

is the decorated category with the same objects as $\mathbf{C}$ such that:

- $\mathbf{C}^p_T = \mathbf{C}$, so that it has a morphism $f^p : X \to Y$ for each $f : X \to Y$ in $\mathbf{C}$,
- $\mathbf{C}^q_T = \mathbf{C}_T$ (the Kleisli category of T), so that it has a morphism $f^q : X \to Y$ for each $f : X \to TY$ in $\mathbf{C}$,
- $\mathbf{C}^r_T$ has a morphism $f^r : X \to Y$ in $\mathbf{C}^q_T$ for each $f : TX \to TY$ in $\mathbf{C}$, and the composition is as in $\mathbf{C}$.

The inclusion $\mathbf{C}^p_T \subsetneqq \mathbf{C}^q_T$ corresponds to mapping $f : X \to Y$ to $\eta_Y \circ f : X \to TY$ in $\mathbf{C}$. The inclusion $\mathbf{C}^q_T \subsetneqq \mathbf{C}^r_T$ corresponds to mapping $f : X \to TY$ to $\mu_Y \circ Tf : TX \to TY$ in $\mathbf{C}$.
In addition, the composition of a morphism in $\mathbf{C}^q_T$ with a morphism in $\mathbf{C}^r_T$ is in $\mathbf{C}^q_T$: $g^r \circ f^q = (g \circ f)^q$.

Remark 3.3. If there are enough sums in $\mathbf{C}$, then every family of morphisms $f^q_i : X_i \to Y$ gives rise to a cotuple $[f_i]^q_i : \sum_i X_i \to Y$ characterized by $[f_i]^q_i \circ in^p_i = f^q_i$ for each i, where $in^p_i : X_i \to \sum_i X_i$ is the coprojection.

Definition 3.4. Let $\mathbf{C}$ be a category and (T, ε, δ) (or simply T) a comonad on $\mathbf{C}$ satisfying the epi requirement (each ε_X is an epi). Then $D_T(\mathbf{C}) = \mathbf{C}^p_T \subsetneqq \mathbf{C}^q_T \subsetneqq \mathbf{C}^r_T$ is the decorated category with the same objects as $\mathbf{C}$ such that:

- $\mathbf{C}^p_T = \mathbf{C}$, so that it has a morphism $f^p : X \to Y$ for each $f : X \to Y$ in $\mathbf{C}$,
- $\mathbf{C}^q_T = \mathbf{C}_T$ (the coKleisli category of T), so that it has a morphism $f^q : X \to Y$ for each $f : TX \to Y$ in $\mathbf{C}$,
- $\mathbf{C}^r_T$ has a morphism $f^r : X \to Y$ in $\mathbf{C}^q_T$ for each $f : TX \to TY$ in $\mathbf{C}$, and the composition is as in $\mathbf{C}$.

The inclusion $\mathbf{C}^p_T \subsetneqq \mathbf{C}^q_T$ corresponds to mapping $f : X \to Y$ to $f \circ \varepsilon_X : TX \to Y$ in $\mathbf{C}$. The inclusion $\mathbf{C}^q_T \subsetneqq \mathbf{C}^r_T$ corresponds to mapping $f : TX \to Y$ to $Tf \circ \delta_X : TX \to TY$ in $\mathbf{C}$.
In addition, the composition of a morphism in $\mathbf{C}^r_T$ with a morphism in $\mathbf{C}^q_T$ is in $\mathbf{C}^q_T$: $g^q \circ f^r = (g \circ f)^q$.

Remark 3.5. If there are enough products in $\mathbf{C}$, then every family of morphisms $f^q_i : X \to Y_i$ gives rise to a tuple $(f_i)^q_i : X \to \prod_i Y_i$ characterized by $pr^p_i \circ (f_i)^q_i = f^q_i$ for each i, where $pr^p_i : \prod_i Y_i \to Y_i$ is the projection.

3.2 Global states

Let $\mathbf{C}$ be a category with a terminal object 1, with a distinguished object St called the type of *states*, and a product-with-St functor $T(\ldots) = \ldots \times St$. Then T is the endofunctor of a comonad (T, ε, δ) where $\varepsilon_X : X \times St \to X$ is the projection and $\delta_X : X \times St \to X \times St \times St$ duplicates the St-component. Therefore, we get a decorated category $D_T(\mathbf{C})$, denoted $D_{St}(\mathbf{C})$, as in section 3.1.

Definition 3.6. Let $\mathbf{Sp}^{dec}$ be a decorated specification. The *expansion* of $\mathbf{Sp}^{dec}$ for global states is the specification $E_S(\mathbf{Sp}^{dec})$ with the same points as $\mathbf{Sp}^{dec}$ and a new point S and with:

- an arrow $f_p : X \to Y$ for each $f^p : X \to Y$ in $\mathbf{Sp}^{dec}$,
- an arrow $f_q : X \times S \to Y$ for each $f^q : X \to Y$ in $\mathbf{Sp}^{dec}$,
- an arrow $f_r : X \times S \to Y \times S$ for each $f^r : X \to Y$ in $\mathbf{Sp}^{dec}$.

The following result is easy to check directly, it can also be obtained from an adjunction [4].

Proposition 3.7. *Let $\mathbf{Sp}^{dec}$ be a decorated specification and $\mathbf{C}$ a category with a terminal object 1, a distinguished object St and a comonad $\ldots \times St$. Then there is a bijection:*

$$\mathrm{Mod}^{dec}(\mathbf{Sp}^{dec}, D_{St}(\mathbf{C})) \cong \mathrm{Mod}(E_S(\mathbf{Sp}^{dec}), \mathbf{C})|_{St}$$

where $\mathrm{Mod}(E_S(\mathbf{Sp}^{dec}), \mathbf{C})|_{St}$ is the subcategory of $\mathrm{Mod}(E_S(\mathbf{Sp}^{dec}), \mathbf{C})$ made of the models which map S to St and of the morphisms with the identity as S-component.

Now, we define the decorated specification for global states $\mathbf{S}^{dec}$ by hiding S in definition 2.1, then we apply proposition 3.7 to $\mathbf{S}^{dec}$.

Definition 3.8. Let Loc be a set. The *decorated specification for global states* $\mathbf{S}^{dec}$ is made of, for each $i \in Loc$, a point V_i, an arrow $l_i^q : 1 \to V_i$, an arrow $u_i^r : V_i \to 1$ and an equation:

$$l^q \circ u_i^r = \varphi_i^q \ . \tag{3}$$

where $l^q = (l_j^q)_{j \in Loc} : 1 \to \prod_j V_j$ and $\varphi_i^q : V_i \to \prod_j V_j$ is defined as $\varphi_i^q = (\varphi_{i,j}^q)_{j \in Loc}$ where $\varphi_{i,j}^q : V_i \to V_j$ is (with $(\,)_{V_i}^p : V_i \to 1$):

$$\varphi_{i,i}^q = id_{V_i}^p \text{ and } \varphi_{i,j}^q = l_j^q \circ (\,)_{V_i}^p \text{ when } j \neq i$$

It is easy to check that the expansion $E_S(\mathbf{S}^{dec})$ of $\mathbf{S}^{dec}$ is the specification $\mathbf{S}$ from definition 2.1, so that proposition 3.7 has the following consequence.

Corollary 3.9. *Let $\mathbf{C}$ be a category with finite products and with a distinguished object St. Let M_p be a decorated model of $\mathbf{S}^p$ with values in $D_{St}(\mathbf{C})$ (M_p is made of an object $Val_i = M_p(V_i)$ for each $i \in Loc$). Then there is a bijection:*

$$\mathrm{Mod}^{dec}(\mathbf{S}^{dec}, D_{St}(\mathbf{C}))|_{M_p} \cong \mathrm{Mod}(E_S(\mathbf{S}^{dec}), \mathbf{C})|_{M_p, St}$$

This result allows to define the semantics for global states directly from the decorated specification $\mathbf{S}^{dec}$.

Remark 3.10. Usually the global state effect is formalized using the monad with endofunctor $T'(X) = (X \times S)^S$, assuming that there are exponentials of the form $(\ldots \times S)^S$ in $\mathbf{C}$. Up to currification, the Kleisli category of the monad T' can be identified to the category $\mathbf{C}_T^r$. Thus, with the point of view of monads, we get the inclusion $\mathbf{C}_T^p \subsetneqq \mathbf{C}_T^r$: the morphisms in $\mathbf{C}_T^r$ (the computations) may modify the state, the morphisms in $\mathbf{C}_T^p$ (the values) are the pure functions, but the intermediate category $\mathbf{C}_T^q$ for the inspectors, which may observe the state without modifying it, is lacking.

3.3 Exceptions

Following the same lines as in sections 2.1 and 2.2, the treatment of exceptions as hidden effect is dual to the treatment of global states as hidden effect in section 3.2: see the table in appendix A. Thus, the semantics for exceptions can be defined directly from the decorated specification $\mathbf{E}^{dec}$.

Remark 3.11. It is usual to formalize the exceptions thanks to the monad $\ldots + E$ [10]. Then the raising of exceptions is often defined by operations $\mathtt{raise}_{i,X} : P_i \to X$ for each i and for each object X, and it is often assumed that $P_i = 1$ for every i. This point of view is easily recovered from our approach, by defining $\mathtt{raise}_{i,X} = [\,]_X \circ r_i : P_i \to X$ (with $[\,]_X : 0 \to X$), and assuming that $P_i = 1$ if it is required. On the other hand, [5] contains a preliminary version of the treatment of exceptions as in this paper.

4 Conclusion

In this paper we have proved the duality between two fundamental effects: the global states and the exceptions. One key point is the introduction of the morphisms h_i for handling exceptions. Another key point is the use of the comonad $\ldots \times St$ for dealing with the global states. Dealing more generally with multivariate functions would require more sophisticated products, like the sequential products in [3]. Adding exponentials is a challenge, where closed Freyd-categories might prove helpful [11].

References

[1] Michael Barr, Charles Wells. *Category Theory for Computing Science.* 3rd ed., Publications CRM PM023 (1999).

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

[3] Jean-Guillaume Dumas, Dominique Duval, Jean-Claude Reynaud. Cartesian effect categories are Freyd-categories. CoRR abs/0903.3311 (2009)

[4] Dominique Duval. Diagrammatic Specifications. *Mathematical Structures in Computer Science* (13): 857-890 (2003).

[5] Dominique Duval, Jean-Claude Reynaud. Diagrammatic logic and effects : the example of exceptions. Rapport de Recherche (21 décembre 2004) ccsd-00004129.

[6] Martin Hyland, John Power. The Category Theoretic Understanding of Universal Algebra: Lawvere Theories and Monads. *Electr. Notes Theor. Comput. Sci.* 172: 437-458 (2007).

[7] Eugenio Moggi. Computational Lambda-Calculus and Monads. Logic in Computer Science (LICS '89) *IEEE Conference Proceedings*: 14-23 (1989).

[8] Eugenio Moggi. Notions of Computation and Monads. *Inf. Comput.* 93(1): 55-92 (1991).

[9] Gordon D. Plotkin, John Power. Notions of Computation Determine Monads. FoSSaCS'02. *Lecture Notes in Computer Science* 2303: 342-356 (2002).

[10] Gordon D. Plotkin, John Power. Algebraic Operations and Generic Effects. *Applied Categorical Structures* 11(1): 69-94 (2003).

[11] John Power, Hayo Thielecke. Closed Freyd- and kappa-categories. ICALP'99 *Lecture Notes in Computer Science* 1644: 625-634 (1999).

A Table of notations

The following table summarizes most notations used in the paper. When the main columns are subdivided, the left hand-side is from the hidden point of view while the right hand-side is from the explicit point of view.

Global states		**Exceptions**	
Category			
C with 1, with S and $\ldots \times S$		**C** with 0, with E and $\ldots + E$	
(Co)Monad			
CoMonad $T(X) = X \times S$ $\varepsilon_X : X \times S \to X$, $\delta_X : X \times S \to X \times S \times S$		**Monad** $T(X) = X + E$ $\eta_X : X \to X + E$, $\mu_X : X + E + E \to X + E$	
$p \Rightarrow q \Rightarrow r$			
$(p)\quad X \to Y$ $(q)\quad X \times S \to Y$ $(r)\quad X \times S \to Y \times S$		$(p)\quad X \to Y$ $(q)\quad X \to Y + E$ $(r)\quad X + E \to Y + E$	
"q"			
lookup		**raise**	
$l_i^q : 1 \to V_i$ $l^q = (l_i^q)_i : 1 \to \prod_i V_i$	$l_i : S \to V_i$ $l = (l_i)_i : S \to \prod_i V_i$	$r_i^q : P_i \to 0$ $r^q = [r_i^q]_i : \sum_i P_i \to 0$	$r_i : P_i \to E$ $r = [r_i]_i : \sum_i P_i \to E$
"r"			
update		**handle**	
$u_i^r : V_i \to 1$ $\varphi_{i,j}^q : V_i \to V_j$ $\varphi_{i,i}^q = id_{V_i}^p$ $\varphi_{i,j\neq i}^q = l_j^q \circ (\,)_{V_i}$ $\varphi_i^q = (\varphi_{i,j}^q)_j$ $\begin{array}{ccc} V_i & \xrightarrow{\varphi_i^q} & \prod_j V_j \\ {\scriptstyle u_i^r}\downarrow & = & \downarrow{\scriptstyle id^p} \\ 1 & \xrightarrow{l^q} & \prod_j V_j \end{array}$	$u_i : V_i \times S \to S$ $\varphi_{i,j} : V_i \times S \to V_j$ $\varphi_{i,i} = pr_{V_i}$ $\varphi_{i,j\neq i} = l_j \circ pr_S$ $\varphi_i = (\varphi_{i,j})_j$ $\begin{array}{ccc} V_i \times S & \xrightarrow{\varphi_i} & \prod_j V_j \\ {\scriptstyle u_i}\downarrow & = & \downarrow{\scriptstyle id} \\ S & \xrightarrow{l} & \prod_j V_j \end{array}$	$h_i^r : 0 \to P_i$ $\psi_{i,j}^q : P_j \to P_i$ $\psi_{i,i}^q = id_{P_i}^p$ $\psi_{i,j\neq i}^q = [\,]_E^p \circ r_j^q$ $\psi_i^q = [\psi_{i,j}^q]_j$ $\begin{array}{ccc} \sum_j P_j & \xrightarrow{\psi_i^q} & P_i \\ {\scriptstyle id^p}\uparrow & = & \uparrow{\scriptstyle h_i^r} \\ \sum_j P_j & \xrightarrow{r^q} & 0 \end{array}$	$h_i : E \to P_i + E$ $\psi_{i,j} : P_j \to P_i + E$ $\psi_{i,i} = in_{P_i}$ $\psi_{i,j\neq i} = in_{E,i} \circ r_j$ $\psi_i = [\psi_{i,j}]_j$ $\begin{array}{ccc} \sum_j P_j & \xrightarrow{\psi_i} & P_i + E \\ {\scriptstyle id}\uparrow & = & \uparrow{\scriptstyle h_i} \\ \sum_j P_j & \xrightarrow{r} & E \end{array}$
Remarks			
if $l = (l_i)_i : S \to \prod_i V_i$ is **terminal** (for fixed V_i's) then $l : S \xrightarrow{\simeq} \prod_i V_i$ and by **coinduction:** existence and unicity of u		if $r = [r_i]_i : \sum_i P_i \to E$ is **initial** (for fixed P_i's then $r : \sum_i P_i \xrightarrow{\simeq} E$ and by **induction:** existence and unicity of h	