A zooming process for specifications with an application to exceptions

D. Duval, C. Lair, C. Oriat, J.-C. Reynaud

Rapport de Recherche IMAG-LMC
n° 1055
Mars 2003

A zooming process for specifications with an application to exceptions

D. Duval [a,*], C. Lair [b], C. Oriat [c], J.-C. Reynaud [c]

[a] *LMC-IMAG, Université Joseph Fourier, Grenoble*
[b] *Université Denis Diderot, Paris*
[c] *LSR-IMAG, Institut National Polytechnique, Grenoble*

Abstract

The zooming method which is described in this paper performs a zoom-in on specifications: it begins with a far specification, which takes care of the main features of the structure to be specified, and it ends up with a near specification, which includes all the details of this structure. This method is formalized in the framework of diagrammatic specifications, as introduced by Duval and Lair. As an example, the zooming process is applied to a treatment of exceptions, and it is compared to the method of monads.

Key words: Sketches and generalizations, algebraic specifications, categorical logic, monads.
MSC: 18C30, 68Q65, 03G30

---

1 Introduction

The zooming method which is described in this paper proceeds from a *far* specification S to a *near* specification U. The basic idea is that the far view is rather simple, at the cost of omitting some details, which are recovered in the near view.

On the one hand, the far specification S is a simplified description of the main syntactic features of the near specification U. On the other hand, it can

---

* Corresponding author. Address: B.P. 53, F 38041 Grenoble cedex 9 (France)
Email addresses: `Dominique.Duval@imag.fr` (D. Duval), `lairchrist@aol.com` (C. Lair), `Catherine.Oriat@imag.fr` (C. Oriat), `Jean-Claude.Reynaud@imag.fr` (J.-C. Reynaud).

Preprint *March 5, 2003*

happen that, as in implementation or refinement techniques, the properties of the models are preserved by the zoom; but this is not compulsory, and it can be very convenient to use zooms which do not preserve the properties of the models, as will appear from our example.

In this paper we focus on a restricted family of zooms, such that the semantics of the whole process is given by the interpretations of U. The zooming process builds up an *intermediate* kind of specification T, which is both simple, like S, and with relevant interpretations, like U. However, the issue is that T is not a specification in the usual sense.

In the framework of (Duval and Lair , 2002), T as well as S and U are *diagrammatic specifications*. Basically, the diagrammatic specifications stem from Lair's *trames* (Lair , 1987). They can be considered as a generalization of the *algebraic specifications* from (Goguen, Thatcher and Wagner , 1978), and some of them can also be considered as a concretization of the *institutions* from (Goguen and Burstall , 1984). In the context of diagrammatic specifications, the description of the zooming process is quite simple. It proceeds in two steps. First, the construction of T from S is called the *decoration* step. Then, the construction of U from T is called the *expansion* step. Whenever T is equal to S, the zooming process is made of the expansion step only, and it may correspond to the construction associated to a *monad*, as introduced in (Moggi , 1991) and used in Haskell (Peyton Jones et al. , 1999).

The running example in this paper deals with a treatment of exceptions. The zooming process is described, in an informal way, from the example of natural numbers with a predecessor operation. The specifications U_1 and U_2 below correspond to two different ways to deal with the fact that the predecessor operation is not defined everywhere: either by raising an exception, or by partiality.

Let us first raise an *exception*: the set $\mathbb{N}$ of natural numbers gives rise to the set $\mathbb{N}' = \mathbb{N} \uplus \{\epsilon\}$ (where $\uplus$ denotes the disjoint union), such that the predecessor of 0 is ϵ. This corresponds to the following *near* specification U_1. The points (or *sorts*) of U_1 are meant to be interpreted as sets, and the arrows (or *operations*) of U_1 as maps. The constraint:

$$V = 1$$

means that V has to be interpreted as a singleton, so that constants can be identified to arrows with source V. The constraint:

$$N' = N_{(i)} +_{(e)} V \text{ , or } \begin{tikzcd} N \arrow[r, dotted, tail, "i"] & N' & V \arrow[l, dotted, tail, "e"'] \end{tikzcd} \text{ ,}$$

means that N' has to be interpreted, up to isomorphism, as the disjoint union of the interpretations of N and V, and i and e as the inclusions.

`Specification` U_1:
`points`: V, N, N';
`arrows`: $z : V \longrightarrow N$, $s : N \longrightarrow N$, $p : N \longrightarrow N'$,
$i : N \longrightarrow N'$, $e : V \longrightarrow N'$;
`constraints`: $V = 1$, $N' = N_{(i)}+_{(e)}V$.

This can be illustrated as follows.

$$\begin{tikzcd}
V=1 \arrow[d, "z"'] \arrow[dr, dotted, "e"] & \\
N \arrow[loop left, "s"] \arrow[r, shift left, "p"] \arrow[r, dotted, tail, shift right, "i"'] & N'
\end{tikzcd}
\qquad
\begin{aligned} p \circ s &\equiv i \\ p \circ z &\equiv e \end{aligned}$$

Let us consider a model of U_1 where N is interpreted as $\mathbb{N}$ and V as the singleton $\{v\}$. Then, because of the constraints, N' is interpreted as $\mathbb{N} \uplus \{\epsilon\}$, where ϵ is considered as an *exception*, while i is interpreted as the inclusion of $\mathbb{N}$ in $\mathbb{N} \uplus \{\epsilon\}$ and e as the map $v \mapsto \epsilon$, from $\{v\}$ to $\mathbb{N} \uplus \{\epsilon\}$. There is such a model where z is interpreted as the constant map $v \mapsto 0$, s as the map $n \mapsto n+1$, and p as the map such that $n \mapsto n-1$ for each positive integer n and $0 \mapsto \epsilon$. As required, in this model the predecessor of 0 raises an exception. The way such an exception can be handled is addressed in section 3.5.

In this example, the *far* specification S is the following specification of natural numbers, which omits the "detail" that the predecessor operation is not defined everywhere.

`Specification` S:
`points`: V, N;
`arrows`: $z : V \longrightarrow N$, $s : N \longrightarrow N$, $p : N \longrightarrow N$;
`constraint`: $V = 1$.

This can be illustrated as follows.

$$\begin{tikzcd}
V=1 \arrow[d, "z"'] \\
N \arrow[loop left, "s"] \arrow[loop right, "p"]
\end{tikzcd}$$

There are models of S where N is interpreted as $\mathbb{N}$, V as $\{v\}$, z as the constant map $v \mapsto 0$, s as the map $n \mapsto n+1$, and p as a map such that $n \mapsto n-1$ for each positive integer n. But the predecessor operation must be total on such a model: it has to map 0 to some value in $\mathbb{N}$. This does not match the usual semantics of the predecessor operation. However, the terms of S correspond to programs, like $p \circ s \circ z$ which is valid, or $s \circ p \circ z$ which is not valid.

Now, let us describe the *zooming* process which, in two steps, builds U_1 from S.

The *decoration step* starts from the far specification S and builds up some intermediate specification T. It will be seen in the paper that T is a specification in the diagrammatic sense. The basic idea for the decoration step is that T is built from S by adding some *keywords*. The keywords are meant to say that the interpretations of the arrows z and s never raise an exception, whereas the interpretation of p may raise an exception for some values of its argument. So, the keywords for the arrows are:

- "*not-erroneous*", or "$*$",
- "*maybe-erroneous*", or "?".

By associating the right keyword to each arrow in S, we get the following *intermediate* specification T.

`Specification` T:
  `points`: V, N;
  `arrows`: $(z : *) : V \longrightarrow N$, $(s : *) : N \longrightarrow N$, $(p :?) : N \longrightarrow N$;
  `constraint`: $V = 1$.

$$\begin{tikzcd}
V = 1 \arrow[d, "z:*"'] \\
N \arrow[loop left, "s:*"] \arrow[loop right, "p:?"]
\end{tikzcd}$$

The *expansion step* starts from the intermediate specification T and builds up the near specification U_1. The basic idea for the expansion step is to state explicitly what is the *meaning* of the keywords:

- an arrow $f : X \longrightarrow Y$ with keyword "$*$" is indeed an arrow $f_* : X \longrightarrow Y$,
- an arrow $f : X \longrightarrow Y$ with keyword "?" stands for an arrow $f_? : X \longrightarrow Y'$, where the new point Y' is submitted to the constraint $Y' = Y + E$ with $E = 1$.

This means that $f : X \longrightarrow Y$ with keyword k stands for a model of the specification $\mathcal{W}_1(k)$, which is described now.

`Specification` $\mathcal{W}_1(*)$:
  `points`: X, Y;

`arrow`: $f_* : X \longrightarrow Y$.

$$\begin{tikzcd}
X \arrow[d, "f_*"] \\
Y
\end{tikzcd}$$

`Specification` $\mathcal{W}_1(?)$:
`points`: X, Y, Y', E;
`arrows`: $f_? : X \longrightarrow Y'$, $i_Y : Y \longrightarrow Y'$, $e_Y : E \longrightarrow Y'$;
`constraints`: $E = 1$, $Y' = Y + E$.

$$\begin{tikzcd}
X \arrow[dr, "f_?"] & & \\
Y \arrow[r, tail, dotted, "i_Y"'] & Y' & E = 1 \arrow[l, tail, dotted, "e_Y"]
\end{tikzcd}$$

Finally, a new specification U_1' is obtained by replacing each arrow with keyword k in T by a copy of the specification $\mathcal{W}_1(k)$. This has to be done in the "most natural" way, as explained now.

First, let us come back to the three arrows of S:

$$\begin{tikzcd}
V \arrow[d, "z"] & N \arrow[d, "s"] & N \arrow[d, "p"] \\
N & N & N
\end{tikzcd}$$

The graphical part of S is obtained by merging these three arrows:

$$\begin{tikzcd}
V \arrow[d, "z"'] \\
N \arrow[loop left, "s"] \arrow[loop right, "p"]
\end{tikzcd}$$

Now, let us replace each arrow by a copy of the specification $\mathcal{W}_1(k)$ for the right keyword k:

$$\begin{tikzcd}
V \arrow[d, "z_*"] & N \arrow[d, "s_*"] & N \arrow[dr, "p_?"] & & \\
N & N & N \arrow[r, tail, dotted, "i_N"'] & N' & E = 1 \arrow[l, tail, dotted, "e_N"]
\end{tikzcd}$$

In a similar way, part of U_1' is obtained by merging these three specifications:

$$\begin{tikzcd}
V \arrow[d, "z_*"'] & & \\
N \arrow[loop left, "s_*"] \arrow[r, bend left, "p_?"] \arrow[r, dotted, tail, "i_N"'] & N' & E = 1 \arrow[l, dotted, tail, "e_N"']
\end{tikzcd}$$

This construction has to be completed by dealing with the constraint $V = 1$ of S, so as to get the specification U_1'.

`Specification` U_1':
`points`: V, N, N', E;
`arrows`: $z_* : V \longrightarrow N$, $s_* : N \longrightarrow N$, $p_? : N \longrightarrow N'$,
$i_N : N \longrightarrow N'$, $e_N : \ \longrightarrow N'$;
`constraints`: $V = 1$, $E = 1$, $N' = N_{(i_N)}+_{(e_N)}E$.

$$\begin{tikzcd}
V = 1 \arrow[d, "z_*"'] & & \\
N \arrow[loop left, "s_*"] \arrow[r, bend left, "p_?"] \arrow[r, dotted, tail, "i_N"'] & N' & E = 1 \arrow[l, dotted, tail, "e_N"']
\end{tikzcd}$$

Clearly, both specifications U_1' and U_1 are equivalent. This is an example of the way the zooming process builds the near specification U_1 from the far specification S, via some intermediate specification T.

Now, let us forget about U_1, and let us use *partiality*: the predecessor operation is defined only on the subset $\mathbb{N}'' = \mathbb{N} \setminus \{0\}$ of $\mathbb{N}$. This corresponds to a second *near* specification U_2. The constraint:

$$j \text{ mono} \quad \text{, or} \quad N'' \overset{j}{\rightarrowtail} N \ ,$$

means that j has to be interpreted as an injective map.

`Specification` U_2:
`points`: N, N'';
`arrows`: $z : V \longrightarrow N$, $s : N \longrightarrow N$, $p : N'' \longrightarrow N$, $j : N'' \longrightarrow N$;
`constraints`: $V = 1$, j mono.

This can be illustrated as follows.

$$\begin{tikzcd}
 & V = 1 \arrow[d, "z"'] \\
N'' \arrow[r, bend left, "p"] \arrow[r, tail, "j"'] & N \arrow[loop right, "s"]
\end{tikzcd}$$

There is a model of U_2 where N is interpreted as $\mathbb{N}$, N'' as $\mathbb{N}'' = \mathbb{N} \setminus \{0\}$, j as the inclusion of $\mathbb{N}''$ in $\mathbb{N}$, z as the constant map $v \mapsto 0$, s as the map $n \mapsto n+1$, and p as the map $n \mapsto n-1$ from $\mathbb{N}''$ to $\mathbb{N}$. As required, in this model the predecessor of 0 is not defined.

Let us now consider the *same* near specification S and the *same* intermediate specification T as above, so that the decoration step is unchanged. For the expansion step, let us modify the meaning of the keyword "?":

- an arrow $f : X \longrightarrow Y$ with keyword "$*$" remains an arrow $f_* : X \longrightarrow Y$,
- but now an arrow $f : X \longrightarrow Y$ with keyword "?" stands for an arrow $f_? : X'' \longrightarrow Y$, with some mono $j_X : X'' \rightarrowtail X$.

This means that $f : X \longrightarrow Y$ with keyword k stands for a model of $\mathcal{W}_2(k)$, where $\mathcal{W}_2(*) = \mathcal{W}_1(*)$ and $\mathcal{W}_2(?)$ is described below.

`Specification` $\mathcal{W}_2(?)$:
`points`: X, X'', Y;
`arrows`: $f_? : X \longrightarrow Y'$, $j_X : X'' \longrightarrow X$;
`constraint`: j_X mono.

$$\begin{tikzcd}
X'' \arrow[r, tail, "j_X"] \arrow[dr, "f_?"'] & X \\
& Y
\end{tikzcd}$$

As above, each arrow of S gives rise to a copy of the specification $\mathcal{W}_2(k)$ for the right keyword k:

$$\begin{tikzcd}
V \arrow[d, "z_*"] & N \arrow[d, "s_*"] & N'' \arrow[r, tail, "j_N"] \arrow[dr, "p_?"'] & N \\
N & N & & N
\end{tikzcd}$$

Then U_2' is obtained by merging these three specifications and adding the constraint $V = 1$:

`Specification` U_2':
`points`: V, N, N'';
`arrows`: $z_* : V \longrightarrow N$, $s_* : N \longrightarrow N$, $p_? : N'' \longrightarrow N$, $j_N : N'' \longrightarrow N$;
`constraints`: $V = 1$, j_X mono.

$$\begin{tikzcd}
 & V = 1 \arrow[d, "z_*"'] \\
N'' \arrow[r, tail, "j_N"'] \arrow[r, bend left, "p_?"] & N \arrow[loop right, "s"]
\end{tikzcd}$$

Clearly, both specifications U'_2 and U_2 are equivalent. This second example shows another property of the zooming process: one decoration step can give rise to several expansion steps. In this way, it is possible to reason at the intermediate level, with the keywords, before deciding which will be exactly the interpretation of the keywords. For instance, it can be said at the intermediate level that a term which is composed from arrows with keyword "$*$" cannot be erroneous. As a consequence, the term $s \circ z$ of S must be non-erroneous in U_1 as well as in U_2.

The aim of this paper is to give a formal status to the intermediate specification T and to the decoration and expansion processes. This is quite easy in the framework of *diagrammatic specifications*, as introduced in (Duval and Lair , 2002): indeed, T is a diagrammatic specification, as well as S, U, and the $\mathcal{W}(k)$'s.

Some basic facts about diagrammatic specifications are stated in section 2, so that this paper can be read without any knowledge of (Duval and Lair , 2002). The decoration step is studied in section 3, then the expansion step in section 4. Finally, the entire zooming process is described in section 5. The application to exceptions is built progressively in subsections 2.5, 3.5, 4.4 and 5.4.

Size issues are not addressed in this paper, although they do occur, for instance, when we speak about a category of sets at the meta-level and another one at the specification level, which are respectively denoted $\mathcal{S}et$ and set. *Naturality issues* are not addressed either, since everything in this paper which can be called natural is indeed natural.

2 Diagrammatic specifications

The paper (Duval and Lair , 2002) introduces a framework for dealing with *diagrammatic specifications*; its main features are summarized here, together with some additional notions.

Some knowledge of category theory is assumed, which can be found in (Mac Lane , 1971). Projective *sketches* are used at the meta-level in order to define diagrammatic specifications. Sketches appear in (Ehresmann , 1966), and an introduction to sketch theory can be found in (Coppey and Lair , 1984) and (Coppey and Lair , 1988), or in (Barr and Wells , 1990). On the other hand, the notion of diagrammatic specification can also be considered as a generalization of sketches.

The basic definitions about diagrammatic specifications are reviewed in section 2.1, with an example in section 2.2. The category of propagators, which stands at the meta-level, is defined in section 2.3. The properties of the Yoneda functor are given in section 2.4. Some propagators for dealing with exceptions are introduced in section 2.5.

As in (Duval and Lair , 2002), the meta-specification level is based on *projective sketches* while the specification level is based on *diagrammatic specifications*, which are much more general than projective sketches but of the same nature, and which stem from Lair's *trames* (Lair , 1987). In order to keep distinct these two levels, at the meta-specification level we speak about the *realizations* of a projective sketch $\mathcal{C}$, which map the distinguished cones of $\mathcal{C}$ to limit cones, and at the specification level we speak about the *models* of a diagrammatic specification S, which map the potential products of S to actual products, and the potential sums of S to actual sums, among many other possible potential properties.

2.1 Specifications and their models

A *compositive graph* can be defined as a directed graph where some points X have an *identity arrow* $\mathrm{id}_X : X \longrightarrow X$ and some consecutive pairs of arrows $(f : X \longrightarrow Y, g : Y \longrightarrow Z)$ have a *composed arrow* $g \circ f : X \longrightarrow Z$. In a compositive graph, there is no assumption about associativity and unitary of composition and identities. A morphism of compositive graphs is a morphism of directed graphs which preserves the identity arrows and the composed arrows. In a compositive graph $\mathcal{G}$, a *cone* is made of a *vertex* point G, a *base* morphism $b : \mathcal{I} \longrightarrow \mathcal{G}$, where the compositive graph $\mathcal{I}$ is called the *index*, and *projection* arrows $p_I : G \longrightarrow b(I)$ for each point I in $\mathcal{I}$, such that $b(i) \circ p_I = p_{I'}$ for each arrow $i : I \longrightarrow I'$ in $\mathcal{I}$.

A *projective sketch* is a compositive graph where some cones are called *distinguished cones.* A *propagator* $P : \mathcal{C} \longrightarrow \mathcal{C}'$ is a morphism of projective sketches, which means that it is a morphism of compositive graphs which preserves the distinguished cones. Some basic examples are described at sections 2.2 and 2.5. A *(set-valued) realization* S of a projective sketch $\mathcal{C}$ interprets each point C of $\mathcal{C}$ as a set $S(C)$ and each arrow $c : C_1 \longrightarrow C_2$ of $\mathcal{C}$ as a map

$S(c) : S(C_1) \longrightarrow S(C_2)$, in such a way that each identity loop becomes an identity map, each composed arrow becomes a composed map, and each distinguished cone of $\mathcal{C}$ becomes a limit cone in the category of sets. For instance a distinguished binary cone $C_1 \longleftarrow C \longrightarrow C_2$ becomes, up to isomorphism, a binary cartesian product $S(C_1) \longleftarrow S(C_1) \times S(C_2) \longrightarrow S(C_2)$. It is easy to define morphisms of realizations of $\mathcal{C}$ as natural transformations, which yields the category $\mathcal{R}eal(\mathcal{C})$ of realizations of $\mathcal{C}$.

Diagrammatic specifications are defined with respect to a propagator:

$$P : \mathcal{C} \longrightarrow \overline{\mathcal{C}} \ .$$

A *P-specification* S is a realization of $\mathcal{C}$, and a *P-domain* $\overline{S}$ is a realization of $\overline{\mathcal{C}}$. More precisely, the category of P-specifications and the category of P-domains are defined as:

$$\mathcal{S}pec(P) = \mathcal{R}eal(\mathcal{C}) \quad , \quad \mathcal{D}om(P) = \mathcal{R}eal(\overline{\mathcal{C}}) \ .$$

So, in this context, the projective sketches and propagators are used at the *meta level*, for the specification of the diagrammatic specifications.

Let us consider a propagator $P : \mathcal{C} \longrightarrow \overline{\mathcal{C}}$, a P-specification S and a P-domain $\overline{S}$:

$$\begin{tikzcd}
\mathcal{C} \arrow[rr, "P"] \arrow[dr, "S"'] & & \overline{\mathcal{C}} \arrow[dl, "\overline{S}"] \\
& \mathcal{S}et &
\end{tikzcd}$$

The *omitting functor* $G_P : \mathcal{D}om(P) \longrightarrow \mathcal{S}pec(P)$ is such that $G_P(\overline{S}) = \overline{S} \circ P$ for all domain $\overline{S}$. This functor has a left adjoint $F_P : \mathcal{S}pec(P) \longrightarrow \mathcal{D}om(P)$, called the *freely generating functor*.

$$\begin{tikzcd}
\mathcal{S}pec(P) \arrow[r, shift left, "F_P"] & \mathcal{D}om(P) \arrow[l, shift left, "G_P"]
\end{tikzcd}$$

The *set of P-models* of S with values in $\overline{S}$ is:

$$\boxed{\mathrm{Mod}_P(S, \overline{S}) = \mathrm{Hom}_{\mathcal{D}om(P)}(F_P(S), \overline{S}) \ ,}$$

so that the property of the adjunction yields:

$$\boxed{\mathrm{Mod}_P(S, \overline{S}) \cong \mathrm{Hom}_{\mathcal{S}pec(P)}(S, G_P(\overline{S})) \ .}$$

For each morphism $\sigma : S \longrightarrow S'$, it is easy to define a map $\mathrm{Mod}_P(\sigma, \overline{S})$:

$\mathrm{Mod}_P(S', \overline{S}) \longrightarrow \mathrm{Mod}_P(S, \overline{S})$, so that $\mathrm{Mod}_P(-, \overline{S})$ is a contravariant functor from the category of P-specifications to the category of sets.

In addition, it can happen, and it does happen in the most usual cases, that there is a natural notion of morphisms of P-models, so that the P-models form a category.

A propagator P is called *conservative* if both functors F_P and G_P are full and faithful. Then, the category of P-specifications and the category of P-domains are equivalent.

A propagator P is *filling* if the functor F_P is full and faithful; this corresponds to a *persistence* property: for each P-specification S, the *saturation* of S, i.e. the P-specification $G_P(F_P(S))$, is isomorphic to S. Then, it will be seen at section 4 that the functor F_P is quite easy to determine.

A propagator P is *fractioning* if the functor G_P is full and faithful; this means that, up to some equivalence, P consists in adding inverses to arrows; such an inverse can be seen as a *property* which has to be satisfied by the P-domains, or as a *rule* for generating P-domains. Then, the functor F_P is usually difficult to determine: typically, this functor describes the way all the theorems in a given logical theory are derived from the axioms.

The *decomposition theorem* in (Duval and Lair , 2002) proves that any propagator P can be decomposed, up to equivalence, as a filling propagator J followed by a fractioning one K. This decomposition is not unique. The proof which is given in (Duval and Lair , 2002) is an effective one: it builds explicitely the required propagators J and K in such a way that, in addition, the functor F_J is very easy to compute. It follows that, for dealing with $F_P(S)$, it can be assumed that P is fractioning, i.e. it can be assumed that P consists in adding inverses to arrows. Let $c : C \longrightarrow H$ be an arrow in $\mathcal{C}$ which gets invertible in $\overline{\mathcal{C}}$. Then in $\mathcal{C}$:

$$\begin{tikzcd} H & C \arrow[l, "c"'] \end{tikzcd}$$

and in $\overline{\mathcal{C}}$:

$$\begin{tikzcd} H \arrow[r, bend left, "c^{-1}"] & C \arrow[l, "c"'] \end{tikzcd}$$

So that the propagator P can be described as the projective sketch $\mathcal{C}$ together with a dashed arrow for each inverse which is added in $\overline{\mathcal{C}}$:

$$\begin{tikzcd} H \arrow[r, bend left, dashed, "c^{-1}"] & C \arrow[l, "c"'] \end{tikzcd}$$

The points H and C stand respectively for "hypotheses" and "conclusion", while the arrow c^{-1} stands for the rule:

$$(c^{-1}) \qquad \frac{H}{C} \, .$$

Moreover, when dealing with specifications, the notion of morphism can be generalized, in the Kleisli way, as follows (the Kleisli category of a monad is described in (Mac Lane , 1971), for instance). First, a morphism of P-specifications $\sigma_2 : S_2 \longrightarrow S'_2$ is called an *entailment* if the freely generated morphism of P-domains $F_P(\sigma_2) : F_P(S_2) \longrightarrow F_P(S'_2)$ is an isomorphism, so that $\mathrm{Mod}_P(\sigma_2, \overline{S}) : \mathrm{Mod}_P(S'_2, \overline{S}) \longrightarrow \mathrm{Mod}_P(S_2, \overline{S})$ is a bijection. Then, essentially, a *generalized morphism* of specifications $\sigma : S_1 \longrightarrow S_2$ can be defined as a morphism $\sigma' : S_1 \longrightarrow S'_2$ of realizations of $\mathcal{C}$, together with an entailment $S_2 \longrightarrow S'_2$. This means that, in order to define a morphism from S_1 to S_2, it is allowed to add to S_2 various ingredients, as soon as they can be deduced from S_2. Typically, in equational specifications, if s and t are consecutive arrows in S_2, it is allowed to build S'_2 by adding the composite $t \circ s$.

2.2 A basic example

In this section, "graph" stands for *directed graph*, and "sketch" for *projective sketch.* Moreover, an arrow $c : C \longrightarrow C'$ in a sketch $\mathcal{C}$ is called a *mono*, and is represented as:

$$\begin{tikzcd} C \arrow[r, tail, "c"] & C' \end{tikzcd}$$

when there is in $\mathcal{C}$ a distinguished cone of the following form:

$$\begin{tikzcd} \Gamma_c \, : & & C \arrow[dl, "\mathrm{id}_C"'] \arrow[dr, "\mathrm{id}_C"] & \\ & C \arrow[dr, "c"'] & & C \arrow[dl, "c"] \\ & & C' & \end{tikzcd}$$

which means that the arrow c becomes an injective map in all realization of $\mathcal{C}$.

Let $\mathcal{C}_0$ be the following graph, where "Pt", "Ar", "sce" and "tgt" stand respectively for *points*, *arrows*, *source* and *target.*

$$\begin{tikzcd} \mathrm{Pt} & \mathrm{Ar} \arrow[l, bend right, "\mathrm{sce}"'] \arrow[l, bend left, "\mathrm{tgt}"] \end{tikzcd}$$

As any graph, $\mathcal{C}_0$ is a sketch of a very simple kind. It is a *sketch of graphs*, in the sense that its category of realizations is isomorphic to the category of graphs. For instance, the graph S_N:

$$\begin{tikzcd}
N_0 \arrow[r, "z"] & N_1 \arrow[loop above, "s"] & N_2 \arrow[l, "a"']
\end{tikzcd}$$

can be identified with the following realization of $\mathcal{C}_0$:

$$\begin{tikzcd}[column sep=huge]
S_N(\mathrm{Pt}) = \{N_0, N_1, N_2\} & S_N(\mathrm{Ar}) = \{z, s, a\} \arrow[l, shift right, "S_N(\mathrm{sce}) = \begin{cases} z \mapsto N_0 \\ s \mapsto N_1 \\ a \mapsto N_2 \end{cases}"'] \arrow[l, shift left, "S_N(\mathrm{tgt}) = \begin{cases} z \mapsto N_1 \\ s \mapsto N_1 \\ a \mapsto N_1 \end{cases}"]
\end{tikzcd}$$

The sketch $\mathcal{C}_0$ can be enriched in a conservative way, in order to deal with consecutive arrows. The resulting sketch $\mathcal{C}_1$ has a new point "Cons", which stands for *consecutive pair*. It is the vertex of a distinguished cone Γ_{Cons}, which formalizes the definition of consecutive pairs.

$$\begin{tikzcd}
\mathrm{Pt} & \mathrm{Ar} \arrow[l, shift right, "\mathrm{sce}"'] \arrow[l, shift left, "\mathrm{tgt}"] & \mathrm{Cons} \arrow[l, shift right, "p_1"'] \arrow[l, shift left, "p_2"]
\end{tikzcd}
\qquad
\Gamma_{\mathrm{Cons}} :
\begin{tikzcd}
 & \mathrm{Cons} \arrow[dl, "p_1"'] \arrow[dr, "p_2"] & \\
\mathrm{Ar} \arrow[dr, "\mathrm{tgt}"'] & & \mathrm{Ar} \arrow[dl, "\mathrm{sce}"] \\
 & \mathrm{Pt} &
\end{tikzcd}$$

In order to get a *sketch of compositive graphs* $\mathcal{C}$, we enrich $\mathcal{C}_1$ in a filling, but non-conservative, way. In $\mathcal{C}$ there is a new point "Comp", which stands for *composable pair*, a mono $j : \mathrm{Comp} \rightarrowtail \mathrm{Cons}$ and an arrow $\mathrm{comp} : \mathrm{Comp} \longrightarrow \mathrm{Ar}$, which stands for the *composition* of composable arrows, such that $\mathrm{sce} \circ \mathrm{comp} = \mathrm{sce} \circ p_1 \circ j$ and $\mathrm{tgt} \circ \mathrm{comp} = \mathrm{tgt} \circ p_2 \circ j$. Additional features should be added in order to deal with the identity arrows.

$$\begin{tikzcd}
\mathrm{Pt} & \mathrm{Ar} \arrow[l, shift right, "\mathrm{sce}"'] \arrow[l, shift left, "\mathrm{tgt}"] & \mathrm{Cons} \arrow[l, shift right, "p_1"'] \arrow[l, shift left, "p_2"] & \mathrm{Comp} \arrow[l, tail, "j"'] \arrow[ll, bend right=40, "\mathrm{comp}"']
\end{tikzcd}$$

A *category* is a compositive graph where each consecutive pair has a composed arrow and each point has an identity, and which satisfies the usual associativity and unitary axioms. A sketch $\overline{\mathcal{C}}$ of categories can be obtained as a fractioning, but non-conservative, enrichment of $\mathcal{C}$. In this sketch $\overline{\mathcal{C}}$, the property that each consecutive pair has a composed arrow is formalized by the fact that the arrow $j : \mathrm{Comp} \longrightarrow \mathrm{Cons}$ becomes invertible. Then the arrow $\mathrm{comp} \circ j^{-1}$:

Cons $\longrightarrow$ Ar can be added to $\overline{\mathcal{C}}$, it stands for the *composition* of consecutive arrows in categories. Additional features should be added in order to deal with the identity arrows, and with the associativity and unitarity axioms.

$$\begin{tikzcd}
\mathrm{Pt} & \mathrm{Ar} \arrow[l, shift right, "\mathrm{sce}"'] \arrow[l, shift left, "\mathrm{tgt}"'] & \mathrm{Cons} \arrow[l, shift right, "p_1"'] \arrow[l, shift left, "p_2"'] \arrow[l, bend left=50, "\mathrm{comp}\circ j^{-1}"] \arrow[r, dashed, shift left, "j^{-1}"] & \mathrm{Comp} \arrow[l, shift left, "j"] \arrow[ll, bend right=40, "\mathrm{comp}"']
\end{tikzcd}$$

This construction gives rise to an instance of the decomposition theorem of (Duval and Lair , 2002). Indeed, the propagator from the sketch of graphs $\mathcal{C}_0$ to the sketch of categories $\overline{\mathcal{C}}$ is decomposed, thanks to the sketch of compositive graphs $\mathcal{C}$, as a filling propagator, from $\mathcal{C}_0$ to $\mathcal{C}$, followed by a fractioning one, from $\mathcal{C}$ to $\overline{\mathcal{C}}$.

The graph S_N freely generates a category, i.e. a realization of $\overline{\mathcal{C}}$, which contains:

$$\begin{tikzcd}
\{N_0, N_1, N_2\} \arrow[d, phantom, "="] & \{z, s, a, s \circ z, s \circ s, \ldots\} \arrow[l, shift right, "S_N(\mathrm{sce})"'] \arrow[l, shift left, "S_N(\mathrm{tgt})"'] \arrow[d, phantom, "="] & \{(z, s), (s, s), \ldots\} \arrow[l, shift right, "S_N(p_1)"'] \arrow[l, shift left, "S_N(p_2)"'] \arrow[l, bend right=30, "S_N(\mathrm{comp}\circ j^{-1})"'] \arrow[d, phantom, "="] \\
S_N(\mathrm{Pt}) & S_N(\mathrm{Ar}) & S_N(\mathrm{Cons})
\end{tikzcd}$$

where $S_N(\mathrm{comp} \circ j^{-1}) : S_N(\mathrm{Cons}) \longrightarrow S_N(\mathrm{Ar})$ maps the pair (z, s) to $s \circ z$, the pair (s, s) to $s \circ s$, and so on.

In addition, up to some care about the size of the sets which are involved, the category of sets is a realization of $\overline{\mathcal{C}}$ where Pt, Ar, and comp, are interpreted as the sets, the maps, and the composition of maps. This P-domain is denoted *set*.

2.3 *The category of propagators*

The definition of the morphisms of propagators is straightforward: for two given propagators $P : \mathcal{C} \longrightarrow \overline{\mathcal{C}}$ and $R : \mathcal{E} \longrightarrow \overline{\mathcal{E}}$, a *morphism* $\alpha : P \longrightarrow R$ is made of two morphisms of projective sketches, both denoted α, namely $\alpha : \mathcal{C} \longrightarrow \mathcal{E}$ and $\alpha : \overline{\mathcal{C}} \longrightarrow \overline{\mathcal{E}}$, such that $\alpha \circ P = R \circ \alpha$.

$$\begin{tikzcd}
\mathcal{C} \arrow[r, "P"] \arrow[d, "\alpha"'] & \overline{\mathcal{C}} \arrow[d, "\alpha"] \\
\mathcal{E} \arrow[r, "R"] & \overline{\mathcal{E}}
\end{tikzcd}
\qquad \text{or} \qquad
\begin{tikzcd}
P \arrow[d, "\alpha"'] \\
R
\end{tikzcd}$$

Whenever needed, the notation α will be precised as either $\alpha : P \longrightarrow R$ or $\alpha : \mathcal{C} \longrightarrow \mathcal{E}$ or $\alpha : \overline{\mathcal{C}} \longrightarrow \overline{\mathcal{E}}$.

Propagators and their morphisms form a category in the obvious way. In the rest of this paper, as above, we use the expression "morphism of projective sketches", rather than "propagator", for dealing with the two components of a morphism of propagators. Some examples of morphisms of propagators are described in section 2.5.

The following result is an easy consequence of the adjunction property.

Proposition 1 *Let $\alpha : P \longrightarrow R$ be a morphism of propagators. Then for all P-specification S and all R-domain $\overline{U}$:*

$$\boxed{\mathrm{Mod}_P(S, G_\alpha(\overline{U})) \cong \mathrm{Mod}_R(F_\alpha(S), \overline{U}) \ .}$$

Proof. We have to prove that $\mathrm{Hom}_{\mathcal{D}om(P)}(F_P(S), G_\alpha(\overline{U}))$ is in one-to-one correspondence with $\mathrm{Hom}_{\mathcal{D}om(R)}(F_R(F_\alpha(S)), \overline{U})$. But $F_R \circ F_\alpha \cong F_\alpha \circ F_P$ since $R \circ \alpha = \alpha \circ P$. So, the result derives from the adjunction which is associated to $\alpha : \overline{\mathcal{C}} \longrightarrow \overline{\mathcal{E}}$. $\Box$

The next result is an easy consequence of the fact that, essentially, a fractioning propagator only adds inverses to arrows.

Proposition 2 *Let us assume that P is fractioning. Then a morphism of projective sketches $\alpha : \mathcal{C} \longrightarrow \mathcal{E}$ can be extended to a morphism of propagators $\alpha : P \longrightarrow R$ if and only if, for each arrow c in $\mathcal{C}$ which becomes invertible in $\overline{\mathcal{C}}$, the arrow $\alpha(c)$ becomes invertible in $\overline{\mathcal{E}}$; if so, then $\alpha : P \longrightarrow R$ is uniquely determined.*

We now introduce the notion of *reliability*, which will be used at section 3. Let $\alpha : P \longrightarrow R$ be a morphism of propagators. Then, it is easy to see, as in (Duval and Lair , 2002), that this morphism gives rise to a natural transformation:

$$\Phi_\alpha : F_P \circ G_\alpha \Rightarrow G_\alpha \circ F_R : \mathcal{S}pec(R) \longrightarrow \mathcal{D}om(P) \ .$$

Definition 3 *A morphism of propagators $\alpha : P \longrightarrow R$ is* reliable *if the natural transformation Φ_α is a natural isomorphism:*

$$F_P \circ G_\alpha \cong G_\alpha \circ F_R \ .$$

$$\begin{tikzcd}
\mathcal{C} \arrow[r, "P"] \arrow[d, "\alpha"'] & \overline{\mathcal{C}} \arrow[d, "\alpha"] \\
\mathcal{E} \arrow[r, "R"] & \overline{\mathcal{E}}
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
\mathcal{S}pec(P) \arrow[r, "F_P"] & \mathcal{D}om(P) \\
\mathcal{S}pec(R) \arrow[r, "F_R"] \arrow[u, "G_\alpha"] & \mathcal{D}om(R) \arrow[u, "G_\alpha"']
\end{tikzcd}$$

2.4 The Yoneda functor

The Yoneda functor plays a basic role in category theory, as a link between a syntax and its set-valued semantics. It is a contravariant functor.

First, let us assume that the projective sketch $\mathcal{C}$ is a *projective prototype*, which means that its underlying compositive graph is a category and that all its distinguished cones are limit cones. Then, the realizations of $\mathcal{C}$ are the limit-preserving functors from $\mathcal{C}$ to $\mathcal{S}et$. The *Yoneda functor* associated to $\mathcal{C}$:

$$\mathcal{Y}_\mathcal{C} : \mathcal{C} \to\!\!\!\!\!\times\; \mathcal{R}eal(\mathcal{C})$$

is the contravariant functor such that, for all point C of $\mathcal{C}$, the realization $\mathcal{Y}_\mathcal{C}(C)$ of $\mathcal{C}$ is:

$$\mathcal{Y}_\mathcal{C}(C) = \mathrm{Hom}_\mathcal{C}(C, -)$$

and for all arrow $c : C \longrightarrow C'$ of $\mathcal{C}$, the morphism of realizations $\mathcal{Y}_\mathcal{C}(c)$ is:

$$\mathcal{Y}_\mathcal{C}(c) = \mathrm{Hom}_\mathcal{C}(c, -) : \mathcal{Y}_\mathcal{C}(C') \longrightarrow \mathcal{Y}_\mathcal{C}(C)$$

which maps $f : C' \longrightarrow C''$ to $f \circ c : C \longrightarrow C''$.

Now, let $\mathcal{C}$ be any projective sketch, then it can be proven that $\mathcal{C}$ freely generates a projective prototype, which is denoted $\mathrm{Proto}(\mathcal{C})$. There is a canonical morphism $\mathrm{proto}(\mathcal{C}) : \mathcal{C} \longrightarrow \mathrm{Proto}(\mathcal{C})$, and the categories of realizations of $\mathcal{C}$ and $\mathrm{Proto}(\mathcal{C})$ can be identified. The *Yoneda functor* associated to $\mathcal{C}$ is composed of $\mathrm{proto}(\mathcal{C})$ followed by $\mathcal{Y}_{\mathrm{Proto}(\mathcal{C})}$:

$$\mathcal{Y}_\mathcal{C} : \mathcal{C} \to\!\!\!\!\!\times\; \mathcal{R}eal(\mathcal{C})\ .$$

As an example, let us consider the projective sketch of graphs $\mathcal{C}_0$, as described in section 2.2. Let $\mathcal{Y}_0$ denote the Yoneda functor $\mathcal{Y}_{\mathcal{C}_0} : \mathcal{C}_0 \to\!\!\!\!\!\times\; \mathcal{G}r$. The projective sketch $\mathcal{C}_0$ is simply the graph:

$$\begin{tikzcd}
\mathrm{Pt} & & \mathrm{Ar} \arrow[ll, bend right=15, "\mathrm{sce}"'] \arrow[ll, bend left=15, "\mathrm{tgt}"]
\end{tikzcd}$$

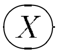
Then clearly, the prototype $\mathrm{Proto}(\mathcal{C}_0)$ is simply the category:

$$\begin{tikzcd}
\mathrm{Pt} \arrow[loop left, "\mathrm{id}_{\mathrm{Pt}}"] & \mathrm{Ar} \arrow[l, bend right, "\mathrm{sce}"'] \arrow[l, bend left, "\mathrm{tgt}"] \arrow[loop right, "\mathrm{id}_{\mathrm{Ar}}"]
\end{tikzcd}$$

In $\mathrm{Proto}(\mathcal{C}_0)$, the identity arrow $\mathrm{id}_{\mathrm{Pt}}$ is the unique arrow from Pt to Pt, and there is no arrow from Pt to Ar. It follows that $\mathcal{Y}_0(\mathrm{Pt})(\mathrm{Pt})$ has a unique element $\mathrm{id}_{\mathrm{Pt}}$, and that $\mathcal{Y}_0(\mathrm{Pt})(\mathrm{Ar})$ is empty. So, the realization $\mathcal{Y}_0(\mathrm{Pt})$ of $\mathcal{C}_0$ is:

$$\begin{tikzcd}
\mathcal{Y}_0(\mathrm{Pt})(\mathrm{Pt}) = \{\mathrm{id}_{\mathrm{Pt}}\} & \mathcal{Y}_0(\mathrm{Pt})(\mathrm{Ar}) = \emptyset \arrow[l, shift right] \arrow[l, shift left]
\end{tikzcd}$$

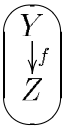
In this context, the element $\mathrm{id}_{\mathrm{Pt}}$ of $\mathcal{Y}_0(\mathrm{Pt})(\mathrm{Pt})$ is a point X of the graph $\mathcal{Y}_0(\mathrm{Pt})$. So, the graph $\mathcal{Y}_0(\mathrm{Pt})$ is made of this point X and no arrow:



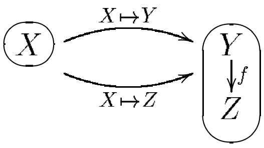
Similarly, in $\mathrm{Proto}(\mathcal{C}_0)$, the identity arrow $\mathrm{id}_{\mathrm{Ar}}$ is the unique arrow from Ar to Ar, that there are only both arrows sce and tgt from Ar to Pt. So, the realization $\mathcal{Y}_0(\mathrm{Ar})$ of $\mathcal{C}_0$ is:

$$\begin{tikzcd}
\mathcal{Y}_0(\mathrm{Ar})(\mathrm{Pt}) = \{\mathrm{sce}, \mathrm{tgt}\} & \mathcal{Y}_0(\mathrm{Ar})(\mathrm{Ar}) = \{\mathrm{id}_{\mathrm{Ar}}\} \arrow[l, shift right, "\mathrm{id}_{\mathrm{Ar}} \mapsto \mathrm{sce}"'] \arrow[l, shift left, "\mathrm{id}_{\mathrm{Ar}} \mapsto \mathrm{tgt}"]
\end{tikzcd}$$

In this context, the element $\mathrm{id}_{\mathrm{Ar}}$ of $\mathcal{Y}_0(\mathrm{Ar})(\mathrm{Ar})$ is an arrow f of the graph $\mathcal{Y}_0(\mathrm{Ar})$. Similarly, the elements sce and tgt of $\mathcal{Y}_0(\mathrm{Ar})(\mathrm{Pt})$ are points Y and Z of the graph $\mathcal{Y}_0(\mathrm{Ar})$. So, the graph $\mathcal{Y}_0(\mathrm{Ar})$ is made of one arrow f together with its source and target Y and Z:



Then, the morphism $\mathcal{Y}_0(\mathrm{sce}) : \mathcal{Y}_0(\mathrm{Ar}) \longrightarrow \mathcal{Y}_0(\mathrm{Pt})$ is determined by the map $\mathcal{Y}_0(\mathrm{sce})(\mathrm{Pt}) : \{\mathrm{id}_{\mathrm{Pt}}\} \longrightarrow \{\mathrm{sce}, \mathrm{tgt}\}$, which maps $\mathrm{id}_{\mathrm{Pt}}$ to $\mathrm{id}_{\mathrm{Pt}} \circ \mathrm{sce} = \mathrm{sce}$; as a morphism of graphs, it maps the point X to Y. Similarly, the morphism $\mathcal{Y}_0(\mathrm{tgt}) : \mathcal{Y}_0(\mathrm{Ar}) \longrightarrow \mathcal{Y}_0(\mathrm{Pt})$ is determined by the map $\mathcal{Y}_0(\mathrm{tgt})(\mathrm{Pt}) : \{\mathrm{id}_{\mathrm{Pt}}\} \longrightarrow \{\mathrm{sce}, \mathrm{tgt}\}$, which maps $\mathrm{id}_{\mathrm{Pt}}$ to $\mathrm{id}_{\mathrm{Pt}} \circ \mathrm{tgt} = \mathrm{tgt}$; as a morphism of graphs, it maps the point X to Z. So, the projective sketch $\mathcal{C}_0$ is mapped, via the Yoneda functor $\mathcal{Y}_0$, to the following diagram in the category of graphs:



Now, let us consider the projective sketch of compositive graphs $\mathcal{C}$, as described in section 2.2. Let $\mathcal{Y}$ denote the Yoneda functor $\mathcal{Y}_{\mathcal{C}} : \mathcal{C} \mathrel{-\!\!\times\!\!\!\rightarrow} \mathit{Comp}$. The projective sketch $\mathcal{C}$ contains $\mathcal{C}_0$, the point Cons and the arrows $p_1, p_2 : \mathrm{Cons} \longrightarrow \mathrm{Ar}$:

$$\begin{tikzcd}
\mathrm{Pt} & \mathrm{Ar} \arrow[l, shift right, "\mathrm{sce}"'] \arrow[l, shift left, "\mathrm{tgt}"'] & \mathrm{Cons} \arrow[l, shift right, "p_1"'] \arrow[l, shift left, "p_2"']
\end{tikzcd}$$

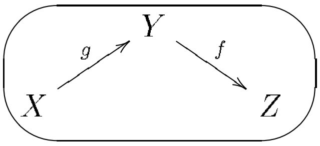
It also contains the distinguished cone Γ_{Cons}:

$$\begin{tikzcd}
& \mathrm{Cons} \arrow[dl, "p_1"'] \arrow[dr, "p_2"] & \\
\mathrm{Ar} \arrow[dr, "\mathrm{tgt}"'] & & \mathrm{Ar} \arrow[dl, "\mathrm{sce}"] \\
& \mathrm{Pt} &
\end{tikzcd}$$

It can be checked that the restriction of $\mathcal{Y}$ to $\mathcal{C}_0$ coincide with $\mathcal{Y}_0$. In addition, the distinguished cone Γ_{Cons} becomes a colimit cone $\mathcal{Y}(\Gamma_{\mathrm{Cons}})$ in $\mathcal{R}eal(\mathcal{C})$:

$$\begin{tikzcd}
& \mathcal{Y}(\mathrm{Cons}) \arrow[dr, phantom] & \\
\mathcal{Y}(\mathrm{Ar}) \arrow[ur, "\mathcal{Y}(p_1)"] & & \mathcal{Y}(\mathrm{Ar}) \arrow[ul, "\mathcal{Y}(p_2)"'] \\
& \mathcal{Y}(\mathrm{Pt}) \arrow[ul, "\mathcal{Y}(\mathrm{tgt})"] \arrow[ur, "\mathcal{Y}(\mathrm{sce})"'] &
\end{tikzcd}$$

It follows that $\mathcal{Y}(\mathrm{Cons})$ is the following compositive graph:



Theorem 4 states some of the properties of the Yoneda functor. The density property, in this theorem, deals with a colimit indexed by a compositive graph which is denoted $\mathcal{C}\backslash S$, where S is a realization of $\mathcal{C}$. The compositive graph $\mathcal{C}\backslash S$ is made of a point $[C, x]$ for all point C of $\mathcal{C}$ and all $x \in S(C)$, an arrow $[c, x] : [C_1, x] \longrightarrow [C_2, S(c)(x)]$ for all arrow $c : C_1 \longrightarrow C_2$ of $\mathcal{C}$ and all $x \in S(C_1)$, together with the identities $\mathrm{id}_{[C,x]} = [\mathrm{id}_C, x]$ and the composites $[c_2 \circ c_1, x] = [c_2, S(c_1)(x)] \circ [c_1, x]$, whenever id_C and $c_2 \circ c_1$ exist in $\mathcal{C}$. In addition, $(\mathcal{C}\backslash S)^{op}$ is the *opposite* compositive graph, with all the arrows in the opposite direction.

Theorem 4 *The Yoneda functor $\mathcal{Y}_{\mathcal{C}}$ is a contravariant realization of $\mathcal{C}$, which means that it maps each distinguished cone in $\mathcal{C}$ to a colimit cone in $\mathcal{R}eal(\mathcal{C})$. In addition:*

- ***Compatibity property:*** *Let $\alpha : \mathcal{C} \longrightarrow \mathcal{E}$ be a morphism of projective sketches, then:*
$$F_\alpha \circ \mathcal{Y}_{\mathcal{C}} \cong \mathcal{Y}_{\mathcal{E}} \circ \alpha\,.$$

- ***Yoneda property:*** *For all realization S of $\mathcal{C}$:*

$$S \cong \mathrm{Hom}_{\mathcal{R}eal(\mathcal{C})}(\mathcal{Y}_\mathcal{C}(-), S)\,.$$

- ***Density property:*** *For all realization S of $\mathcal{C}$:*

$$S \cong \mathrm{colim}_{(\mathcal{C}\backslash S)^{op}}(\mathcal{Y}_\mathcal{C}(C))\,.$$

The Yoneda property means that for all point C of $\mathcal{C}$ there are "$S(C)$" ways to map $\mathcal{Y}_\mathcal{C}(C)$ to S. The density property means that, up to isomorphism, S can be recovered by gluing together, in the right way, one copy of $\mathcal{Y}_\mathcal{C}(C)$ for each element of $S(C)$.

The Yoneda functor can be used in order to illustrate the description of a propagator. Indeed, let $P : \mathcal{C} \longrightarrow \overline{\mathcal{C}}$ be a propagator, and let us assume that P consists in the inversion of arrows: indeed, we know from the decomposition theorem that such propagators play a major role. Then, as in section 2.1, P can be described by adding a dashed arrow in $\mathcal{C}$ for each inverse arrow which is added in $\overline{\mathcal{C}}$. On the other hand, the Yoneda functor $\mathcal{Y}_\mathcal{C} : \mathcal{C} \mathrel{-\!\!\!\times\!\!\!\twoheadrightarrow} \mathcal{R}eal(\mathcal{C})$ yields an illustration of $\mathcal{C}$. Indeed, each arrow of $\mathcal{C}$:

$$\begin{tikzcd} H & C \arrow[l, "c"'] \end{tikzcd}$$

gives rise, in a contravariant way, to a morphism in $\mathcal{R}eal(\mathcal{C})$:

$$\begin{tikzcd} \mathcal{Y}_\mathcal{C}(H) \arrow[r, "\mathcal{Y}_\mathcal{C}(c)"] & \mathcal{Y}_\mathcal{C}(C) \end{tikzcd}$$

When c gets invertible in $\overline{\mathcal{C}}$, according to the compatibility property of the Yoneda functor, the functor F_P maps $\mathcal{Y}_\mathcal{C}(c)$ to $\mathcal{Y}_{\overline{\mathcal{C}}}(c)$ in $\mathcal{R}eal(\overline{\mathcal{C}})$, and there is a morphism $\mathcal{Y}_{\overline{\mathcal{C}}}(c)^{-1} = \mathcal{Y}_{\overline{\mathcal{C}}}(c^{-1})$:

$$\begin{tikzcd} \mathcal{Y}_{\overline{\mathcal{C}}}(H) \arrow[r, "\mathcal{Y}_{\overline{\mathcal{C}}}(c)"] & \mathcal{Y}_{\overline{\mathcal{C}}}(C) \arrow[l, bend right, "\mathcal{Y}_{\overline{\mathcal{C}}}(c)^{-1}"'] \end{tikzcd}$$

In the spirit of *D-algebras* (Coppey , 1972), this rule (c^{-1}) is represented by a dashed arrow $\mathcal{Y}_\mathcal{C}(c)^{(-1)}$ which is added to $\mathcal{R}eal(\mathcal{C})$, although it does not correspond to any actual morphism in $\mathcal{R}eal(\mathcal{C})$:

$$\begin{tikzcd} \mathcal{Y}_\mathcal{C}(H) \arrow[r, "\mathcal{Y}_\mathcal{C}(c)"] & \mathcal{Y}_\mathcal{C}(C) \arrow[l, bend right, dashed, "\mathcal{Y}_\mathcal{C}(c)^{(-1)}"'] \end{tikzcd}$$

This kind of illustration will be used in the examples.

Let us describe three propagators P, R and Q, which will be used in sections 3.5, 4.4 and 5.4 in order to formalize a treatment of exceptions in equational specifications, as outlined in the introduction. It would be quite easy to generalize this treatment to other logics, e.g. to first-order logic.

A propagator P for the far specifications.

Let $P : \mathcal{C} \longrightarrow \overline{\mathcal{C}}$ be the following propagator, which corresponds to equational logic.

In a compositive graph, as defined in section 2.1, two arrows are *parallel* if they share the same source and target. A *(finite discrete) cone* $(Z, (f_i : Z \longrightarrow X_i)_{1 \le i \le n})$ is made of a point Z, called the *vertex* of the cone, together with a finite family of arrows from the vertex. It is simply denoted $(f_i : Z \longrightarrow X_i)_{1 \le i \le n}$ when $n > 0$.

A *P-specification* S is a compositive graph together with a set of pairs of parallel arrows, which are called the *equations* of S, and with a set of cones, which are called the *potential products* of S. When $n = 0$, the vertex of a potential product is called a *potential terminal point* of S. An equation (f, g) is written $f \equiv g$. The vertex of a potential product with base $(X_1, \ldots, X_n)$ can be denoted $X_1 \times \ldots \times X_n$, and a potential terminal point can be denoted 1.

In a category, a *(n-ary) product* is a cone $(p_i : X \longrightarrow X_i)_{1 \le i \le n}$ such that for each cone $(f_i : Z \longrightarrow X_i)_{1 \le i \le n}$ there is a unique *factorization arrow*, i.e. an arrow $\mathrm{fact}(f_1, \ldots, f_n) : Z \longrightarrow X$ such that $p_i \circ \mathrm{fact}(f_1, \ldots, f_n) = f_i$ for all i. When $n = 0$, a product is called a *terminal point*, and its property is that for each point Z there is a unique arrow $\mathrm{fact}(Z) : Z \longrightarrow X$. In the category of sets, the products are (up to isomorphism) the cartesian products, and the terminal points are the singletons.

A *P-domain* is a P-specification such that its underlying compositive graph is a category, its equations are equalities, it has a chosen potential terminal point 1, which is an actual terminal point, and it has a chosen potential product $(p_i : X_1 \times \ldots \times X_n \longrightarrow X_i)_{1 \le i \le n}$ for each n and each n-uple of points $(X_1, X_2, \ldots, X_n)$, which is an actual product.

So, at the meta-level, $\mathcal{C}$ contains a sketch of graphs and $\overline{\mathcal{C}}$ contains a sketch of categories. Since each consecutive pair of arrows in a category has a composed arrow, the arrow:

$$\begin{tikzcd}\mathrm{Cons} & \mathrm{Comp} \arrow[l, "j"']\end{tikzcd}$$

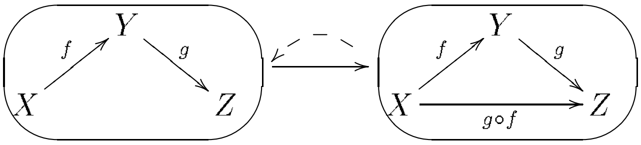
of $\mathcal{C}$ becomes invertible in $\overline{\mathcal{C}}$. This property corresponds to the rule:

$$\frac{f : X \longrightarrow Y \quad g : Y \longrightarrow Z}{g \circ f : X \longrightarrow Z}$$

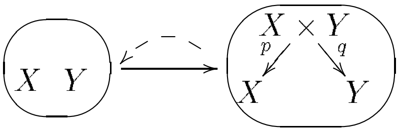
which is illustrated, as explained above, with the help of the Yoneda functor:

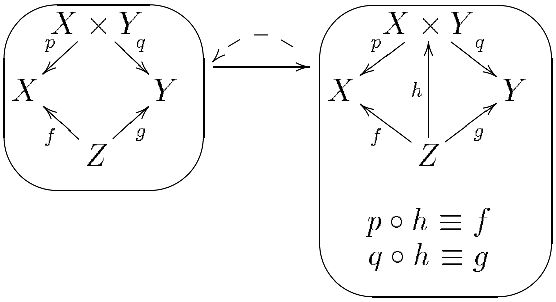
$$\begin{tikzcd}
\mathcal{Y}_{\mathcal{C}}(\text{Cons}) \arrow[r, "\mathcal{Y}_{\mathcal{C}}(j)"] & \mathcal{Y}_{\mathcal{C}}(\text{Comp}) \arrow[l, bend right, dashed, "\mathcal{Y}_{\mathcal{C}}(j)^{(-1)}"']
\end{tikzcd}$$

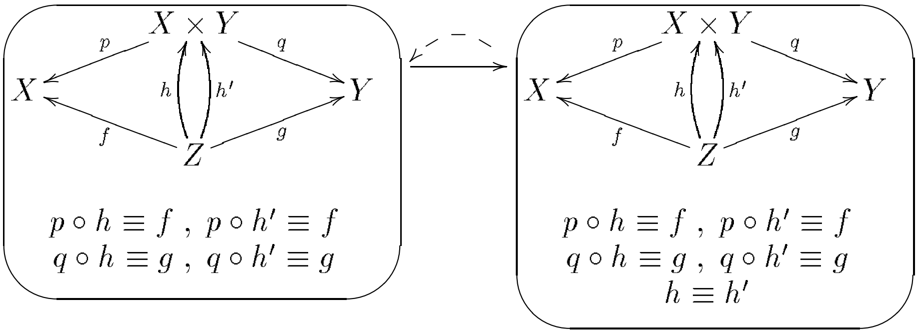
The compositive graph $\mathcal{Y}_{\mathcal{C}}(\text{Cons})$ has been computed in section 2.4. The compositive graph $\mathcal{Y}_{\mathcal{C}}(\text{Comp})$ can been computed in a similar way, it is made of composable pair (f, g) together with its composed arrow $g \circ f$. So, the rule of composition is illustrated as follows:



The existence of chosen products in a P-domain corresponds to the inversion of three arrows in $\overline{\mathcal{C}}$, which is illustrated as follows, in the binary case. The three morphisms below are inclusions.







In our examples, S will denote the following P-specification:

`Specification` S:
  `points:` V, N;
  `arrows:` $z : V \longrightarrow N$, $s : N \longrightarrow N$, $p : N \longrightarrow N$, $q : V \longrightarrow N$;
  `equations:` $p \circ s \equiv \mathrm{id}_N$, $p \circ z \equiv q$;
  `constraint:` $V = 1$.

$$\begin{tikzcd}
V = 1 \arrow[d, shift right, "z"'] \arrow[d, shift left, bend left, "q"] & & \begin{array}{c} p \circ s \equiv \mathrm{id}_N \\ p \circ z \equiv q \end{array} \\
N \arrow[loop left, "s"] \arrow[loop right, "p"] & &
\end{tikzcd}$$

And S_0 will denote the underlying graph of S, which can be considered as a P-specification of a simple form:

`Specification` S_0:
  `points:` V, N;
  `arrows:` $z : V \longrightarrow N$, $s : N \longrightarrow N$, $p : N \longrightarrow N$, $q : V \longrightarrow N$.

$$\begin{tikzcd}
V \arrow[d, shift right, "z"'] \arrow[d, shift left, bend left, "q"] \\
N \arrow[loop left, "s"] \arrow[loop right, "p"]
\end{tikzcd}$$

The P-specification S_0 is such that $S_0(\mathrm{Pt}) = \{V, N\}$ and $S_0(\mathrm{Ar}) = \{z, s, p, q\}$.

Let *set* denote the P-domain of sets, where the chosen products are the cartesian products.

A propagator R for the near specifications.

Let $R : \mathcal{E} \longrightarrow \overline{\mathcal{E}}$ be the following propagator, which corresponds to "equational logic with *xor*'s".

In a compositive graph, a *(finite discrete) cocone* $(Z, (f_i : X_i \longrightarrow Z)_{1 \le i \le n})$ is made of a point Z, called the *vertex* of the cone, together with a finite family of arrows to the vertex. It is simply denoted $(f_i : X_i \longrightarrow Z)_{1 \le i \le n}$ when $n > 0$.

A *R-specification* U is a compositive graph together with a set of pairs of parallel arrows, which are called the *equations* of U, with a set of cones, which are called the *potential products* of U, and with a set of cocones, which are called the *potential mono-sums* of U. A potential binary mono-sum $(j_1 :$

$X_1 \longrightarrow X, j_2 : X_2 \longrightarrow X)$ can be denoted:

$$X_1 = X_{(j_1)}+_{(j_2)}X_2 \text{ , or } \begin{tikzcd} X_1 \arrow[r, tail, dotted, "j_1"] & X & X_2 \arrow[l, tail, dotted, "j_2"'] \end{tikzcd}$$

In a category, a *monomorphism* is an arrow $j : X' \longrightarrow X$ such that for all $g, h : Y \longrightarrow X'$, if $j \circ g = j \circ h$ then $g = h$. A *(n-ary) sum* is a cocone $(j_i : X_i \longrightarrow X)_{1 \leq i \leq n}$ such that for each cocone $(f_i : X_i \longrightarrow Z)_{1 \leq i \leq n}$ there is a unique *cofactorization arrow*, i.e. an arrow $\mathrm{cofact}(f_1, \ldots, f_n) : X \longrightarrow Z$ such that $\mathrm{cofact}(f_1, \ldots, f_n) \circ j_i = f_i$ for all i. Here, a *mono-sum* is a sum $(j_i : X_i \longrightarrow X)_{1 \leq i \leq n}$ such that each coprojection j_i is a monomorphism. In the category of sets, the monomorphisms are the injections, and the sums are (up to isomorphism) the disjoint unions, so that each sum is a mono-sum.

A *R-domain* is a R-specification such that its underlying compositive graph is a category, its equations are equalities, each potential terminal point is an actual terminal point, each potential product is an actual product, and each potential mono-sum is an actual mono-sum. In contrast with P-domains, it is not assumed in a R-domain that each n-uple of points is the base of any potential product or potential mono-sum; the reason for this choice is explained in section 4.4.

In our examples, U will denote the following R-specification:

`Specification` U:
`points`: V, N, N';
`arrows`: $z : V \longrightarrow N$, $s : N \longrightarrow N$, $p : N \longrightarrow N'$,
$i : N \longrightarrow N'$, $e : V \longrightarrow N'$;
`equations`: $p \circ s \equiv i$, $p \circ z \equiv e$;
`constraints`: $V = 1$, $N' = N_{(i)}+_{(e)}V$.

$$\begin{tikzcd} V = 1 \arrow[d, "z"'] \arrow[dr, dotted, tail, "e"] & \\ N \arrow[loop left, "s"] \arrow[r, bend left, "p"] \arrow[r, dotted, tail, "i"'] & N' \end{tikzcd}$$

$$p \circ s \equiv i$$
$$p \circ z \equiv e$$

Let *set* also denote the R-domain of sets.

A propagator Q for the intermediate specifications.

Let us now define a third propagator $Q : \mathcal{D} \longrightarrow \overline{\mathcal{D}}$. Roughly speaking, a Q-specification is a kind of equational specification where each ingredient can be endowed with some keywords: this will get a precise meaning in section 3.4.

Only arrows (and points) are mentioned here, while equations and constraints will be considered in section 3.5.

The basic idea is that there are three families of arrows in the Q-specifications, which are labelled by the three *keywords* "$*$", "?" and "!":

- "$*$" means "*non-erroneous*",
- "?" means "*maybe-erroneous*",
- "!" means "*surely-erroneous*".

So, roughly speaking, Q is obtained from P by taking three copies Ar_*, $\mathrm{Ar}_?$ and $\mathrm{Ar}_!$ of Ar instead of one. In addition, it can be noted that an arrow which is either never erroneous or always erroneous is also sometimes erroneous: this means that "no" and "yes" are special cases of "maybe". So, $\mathcal{D}$ contains the following graph:

$$\begin{tikzcd}
 & \mathrm{Ar}_* \arrow[dl, shift right, "\mathrm{tgt}_*"'] \arrow[dl, shift left, "\mathrm{sce}_*"] \arrow[d, bend left, "\mathrm{ar}_{*,?}"] \\
\mathrm{Pt} & \mathrm{Ar}_? \arrow[l, shift right, "\mathrm{tgt}_?"'] \arrow[l, shift left, "\mathrm{sce}_?"] \\
 & \mathrm{Ar}_! \arrow[ul, shift right, "\mathrm{tgt}_!"'] \arrow[ul, shift left, "\mathrm{sce}_!"] \arrow[u, bend right, "\mathrm{ar}_{!,?}"']
\end{tikzcd}$$

It follows that, in a *Q-specification* T, there are sets $T(\mathrm{Ar}_*)$, $T(\mathrm{Ar}_?)$ and $T(\mathrm{Ar}_!)$, and maps $T(\mathrm{ar}_{*,?}) : T(\mathrm{Ar}_*) \longrightarrow T(\mathrm{Ar}_?)$ and $T(\mathrm{ar}_{!,?}) : T(\mathrm{Ar}_!) \longrightarrow T(\mathrm{Ar}_?)$.

In a *Q-domain* $\overline{T}$, in addition, the sets $\overline{T}(\mathrm{Ar}_*)$ and $\overline{T}(\mathrm{Ar}_!)$ can be identified to subsets of $\overline{T}(\mathrm{Ar}_?)$. This means that in $\overline{\mathcal{D}}$ the arrows $\mathrm{ar}_{*,?}$ and $\mathrm{ar}_{!,?}$ are required to be monos:

$$\begin{tikzcd}
 & \mathrm{Ar}_* \arrow[dl, shift right, "\mathrm{tgt}_*"'] \arrow[dl, shift left, "\mathrm{sce}_*"] \arrow[d, tail, bend left, "\mathrm{ar}_{*,?}"] \\
\mathrm{Pt} & \mathrm{Ar}_? \arrow[l, shift right, "\mathrm{tgt}_?"'] \arrow[l, shift left, "\mathrm{sce}_?"] \\
 & \mathrm{Ar}_! \arrow[ul, shift right, "\mathrm{tgt}_!"'] \arrow[ul, shift left, "\mathrm{sce}_!"] \arrow[u, tail, bend right, "\mathrm{ar}_{!,?}"']
\end{tikzcd}$$

In $\mathcal{D}$, the arrows $\mathrm{ar}_{*,?}$ and $\mathrm{ar}_{!,?}$ are not required to be monos. This property could be required, however it would be useless: since $\mathrm{ar}_{*,?}$ and $\mathrm{ar}_{!,?}$ become monos in $\overline{\mathcal{D}}$, if it happens that the corresponding maps are not injections in a Q-specification T, then they do become injections in the Q-domain $F_Q(T)$.

Now, let us look at the composition of arrows in a Q-specification.

In the projective sketch $\mathcal{C}$, the point Cons stands for the set of consecutive pairs of arrows, since it is the vertex of the distinguished cone Γ_{Cons}:

$$\begin{tikzcd}
\Gamma_{\mathrm{Cons}} : & & \mathrm{Cons} \arrow[dl, "p_1"'] \arrow[dr, "p_2"] & \\
& \mathrm{Ar} \arrow[dr, "\mathrm{tgt}"'] & & \mathrm{Ar} \arrow[dl, "\mathrm{sce}"] \\
& & \mathrm{Pt} &
\end{tikzcd}$$

In $\mathcal{D}$, there is a point $\mathrm{Cons}_{k,l}$ for each pair of keywords (k,l), which stands for the set of consecutive pairs of arrows (f,g) where f has keyword k and g has keyword l. So, $\mathrm{Cons}_{k,l}$ is the vertex of the distinguished cone $\Gamma_{\mathrm{Cons}_{k,l}}$:

$$\begin{tikzcd}
\Gamma_{\mathrm{Cons}_{k,l}} : & & \mathrm{Cons}_{k,l} \arrow[dl, "p_1"'] \arrow[dr, "p_2"] & \\
& \mathrm{Ar}_k \arrow[dr, "\mathrm{tgt}_k"'] & & \mathrm{Ar}_l \arrow[dl, "\mathrm{sce}_l"] \\
& & \mathrm{Pt} &
\end{tikzcd}$$

In $\mathcal{C}$, there is a mono $j : \mathrm{Comp} \rightarrowtail \mathrm{Cons}$ which stands for the inclusion of the set of composable pairs into the set of consecutive pairs, and an arrow $\mathrm{comp} : \mathrm{Comp} \longrightarrow \mathrm{Ar}$ for the composition of composable pairs:

$$\begin{tikzcd}
\mathrm{Ar} & \mathrm{Cons} & \mathrm{Comp} \arrow[l, tail, "j"'] \arrow[ll, bend right, "\mathrm{comp}"']
\end{tikzcd}$$

In $\mathcal{D}$, there is a mono $j_{k,l} : \mathrm{Comp}_{k,l} \rightarrowtail \mathrm{Cons}_{k,l}$ for each pair of keywords (k,l), and an arrow $\mathrm{comp}_{k,l,m} : \mathrm{Comp}_{k,l} \longrightarrow \mathrm{Ar}_m$ for some triples of keywords (k,l,m). For instance, there are arrows $\mathrm{comp}_{*,*,*}$, $\mathrm{comp}_{?,?,?}$, $\mathrm{comp}_{?,!,!}$ and $\mathrm{comp}_{!,?,!}$ in $\mathcal{D}$. The first one means that the composed of two non-erroneous arrows is a non-erroneous arrow, and the last one means that for each composable pair of arrows (f,g), where f is surely erroneous and g may be erroneous, the composed arrow $g \circ f$ is surely erroneous.

$$\begin{tikzcd}
\mathrm{Ar}_* & \mathrm{Cons}_{*,*} & \mathrm{Comp}_{*,*} \arrow[l, tail, "j"'] \arrow[ll, bend right, "\mathrm{comp}_{*,*,*}"']
\end{tikzcd}$$

$$\begin{tikzcd}
\mathrm{Ar}_? & \mathrm{Cons}_{?,?} & \mathrm{Comp}_{?,?} \arrow[l, tail, "j"'] \arrow[ll, bend right, "\mathrm{comp}_{?,?,?}"']
\end{tikzcd}$$

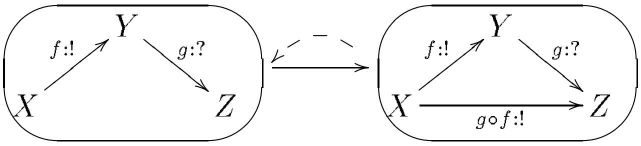
$$\begin{tikzcd}
\mathrm{Ar}_! & \mathrm{Cons}_{?,!} & \mathrm{Comp}_{?,!} \arrow[l, tail, "j"'] \arrow[ll, bend right, "\mathrm{comp}_{?,!,!}"']
\end{tikzcd}$$

$$\begin{tikzcd}
\mathrm{Ar}_! & \mathrm{Cons}_{!,?} & \mathrm{Comp}_{!,?} \arrow[l, tail, "j"'] \arrow[ll, bend right, "\mathrm{comp}_{!,?,!}"']
\end{tikzcd}$$

The arrow j becomes invertible in $\overline{\mathcal{C}}$, and similarly the arrows $j_{k,l,m}$ become invertible in $\overline{\mathcal{D}}$. For instance, the fact that $j_{!,?,!}$ becomes invertible corresponds to the rule:

$$\frac{(f :!) : X \longrightarrow Y \quad (g :?) : Y \longrightarrow Z}{(g \circ f :!) : X \longrightarrow Z}$$

which is illustrated as follows:



Moreover, all the identity arrows have keyword "$*$".

The decoration of the equations and the constraints will be considered in section 3.5.

In our examples, T_0 will denote the Q-specification such that:

- $T_0(\mathrm{Pt}) = \{V, N\}$,
- $T_0(\mathrm{Ar}_*) = \{z : *, s : *\}$,
- $T_0(\mathrm{Ar}_?) = \{z :?, s :?, p :?, q :?\}$,
- $T_0(\mathrm{Ar}_!) = \{q :!\}$,
- $T_0(\mathrm{ar}_{*,?})$ maps $z : *$ to $z :?$ and $s : *$ to $s :?$,
- $T_0(\mathrm{ar}_{!,?})$ maps $q :!$ to $q :?$,
- $T_0(\mathrm{sce}_*)$ maps $z : *$ to V and $s : *$ to N,
- $T_0(\mathrm{sce}_?)$ maps $z :?$ and $q :?$ to V and the others to N,
- $T_0(\mathrm{sce}_!)$ maps $q :!$ to V,
- $T_0(\mathrm{tgt}_*)$ maps $z : *$ and $s : *$ to N,
- $T_0(\mathrm{tgt}_?)$ maps all of $T_0(\mathrm{Ar}_?)$ to N.
- $T_0(\mathrm{tgt}_!)$ maps $q :!$ to N,

This can be described, with obvious conventions, as a "decorated" version of the graph S_0 underlying S. One of the conventions is that, for each arrow, only the strongest keyword is mentioned.

`Specification` T_0:
`points`: V, N;
`arrows`: $(z : *) : V \longrightarrow N$, $(s : *) : N \longrightarrow N$, $(p :?) : N \longrightarrow N$,
$(q :!) : V \longrightarrow N$.

$$\begin{tikzcd}
V \arrow[d, shift right, "z:*"'] \arrow[d, shift left, "q:!"] \\
N \arrow[loop left, "s:*"] \arrow[loop right, "p:?"]
\end{tikzcd}$$

A Q-specification T will be built in section 3.5 by adding equations and constraints to T_0.

There is no straighforward Q-domain of sets. However, for each set $\mathbb{E}$, a Q-domain $set_{\mathbb{E}}$ is described now. This Q-domain will be useful for dealing with exceptions: then, the set $\mathbb{E}$ will stand for the set of exceptions.

- A *point* of $set_{\mathbb{E}}$, i.e. an element of $set_{\mathbb{E}}(\mathrm{Pt})$, is a disjoint union $A \uplus \mathbb{E}$.
- A *non-erroneous map* of $set_{\mathbb{E}}$, i.e. an element of $set_{\mathbb{E}}(\mathrm{Ar}_*)$, is a map $\varphi : A \uplus \mathbb{E} \longrightarrow B \uplus \mathbb{E}$ which is the identity on $\mathbb{E}$ and such that $\varphi(a)$ is in B for all a in A.
- A *maybe-erroneous map* of $set_{\mathbb{E}}$, i.e. an element of $set_{\mathbb{E}}(\mathrm{Ar}_?)$, is a map $\varphi : A \uplus \mathbb{E} \longrightarrow B \uplus \mathbb{E}$ which is the identity on $\mathbb{E}$.
- A *surely-erroneous map* of $set_{\mathbb{E}}$, i.e. an element of $set_{\mathbb{E}}(\mathrm{Ar}_!)$, is a map $\varphi : A \uplus \mathbb{E} \longrightarrow B \uplus \mathbb{E}$ which is the identity on $\mathbb{E}$ and such that $\varphi(a)$ is in $\mathbb{E}$ for all a in A.

We are interested in the model M of T_0 with values in $set_{\mathbb{E}}$ such that:

- $M(V)$ is the disjoint union $\{v\} \uplus \mathbb{E}$,
- $M(N)$ is the disjoint union $\mathbb{N} \uplus \mathbb{E}$,
- $M(z : *)$ is the map such that $\epsilon \mapsto \epsilon$ and $v \mapsto 0$,
- $M(s : *)$ is the map such that $\epsilon \mapsto \epsilon$ and $n \mapsto n + 1$,
- $M(p :?)$ is the map such that $\epsilon \mapsto \epsilon$, $0 \mapsto \epsilon$ and $n \mapsto n - 1$ if $n \neq 0$,
- $M(q :!)$is the map such that $\epsilon \mapsto \epsilon$ and $v \mapsto \epsilon$.

Some morphisms of propagators.

We are interested by two morphisms between these propagators P, Q and R.

The morphism $\delta : Q \longrightarrow P$ maps Pt to Pt, maps Ar_*, $\mathrm{Ar}_?$ and $\mathrm{Ar}_!$ to Ar, and maps $\mathrm{ar}_{*,?}$ and $\mathrm{ar}_{!,?}$ to $\mathrm{id}_{\mathrm{Ar}}$. For instance, the Q-specification $G_\delta(S_0)$ is such that $G_\delta(S_0)(\mathrm{Pt}) = S_0(\mathrm{Pt}) = \{V, N\}$, and:

$$G_\delta(S_0)(\mathrm{Ar}_*) = G_\delta(S_0)(\mathrm{Ar}_?) = G_\delta(S_0)(\mathrm{Ar}_!) = S_0(\mathrm{Ar}) = \{z, s, p, q\} \ ,$$

which means that all the arrows of S_0 are associated to all the keywords. It follows that $T_0(\mathrm{Ar}_k)$ is a subset of $S_0(\mathrm{Ar}_k)$ for each keyword k; this determine a morphism of Q-specifications $\zeta_0 : T_0 \longrightarrow G_\delta(S_0)$.

The morphism $\chi : Q \longrightarrow R$ depends on the choice of a set $\mathbb{E}$. First, let us consider the following R-specification *of exceptions*, with respect to $\mathbb{E}$.

`Specification` $U_{\mathbb{E}}$:
`points`: E, E_ϵ for each $\epsilon \in \mathbb{E}$;
`arrows`: $i_\epsilon : E_\epsilon \longrightarrow E$ for each $\epsilon \in \mathbb{E}$;
`constraints`: $E_\epsilon = 1$ for each $\epsilon \in \mathbb{E}$,
and the cocone $(i_\epsilon)_\epsilon$ is a potential mono-sum.

$$\begin{tikzcd}
 & {} \arrow[dl, dotted, tail] & \vdots \\
E & E_\epsilon = 1 \arrow[l, dotted, tail, "i_\epsilon"'] & \\
 & {} \arrow[ul, dotted, tail] & \vdots
\end{tikzcd}$$

This is a specification of $\mathbb{E}$, in the sense that the unique model of $U_{\mathbb{E}}$ with values in *set* (up to isomorphism) interprets the point E as the set $\mathbb{E}$. Then, in order to make the illustrations of R-specifications easier to read, the point E will represent the whole specification $U_{\mathbb{E}}$.

Now, the morphism $\chi : Q \longrightarrow R$ is easily described via its image by the Yoneda functor $\mathcal{Y}_{\mathcal{E}}$, as follows. Let:

$$\mathcal{W}_\chi = \mathcal{Y}_{\mathcal{E}} \circ \chi : Q \twoheadrightarrow \mathcal{R}eal(R) \ .$$

For instance, while the image of the point Pt of $\mathcal{D}$ via the Yoneda functor $\mathcal{Y}_{\mathcal{D}}$ is the Q-specification made of one point:

$$\mathcal{Y}_{\mathcal{D}}(\mathrm{Pt}) \ : \qquad X$$

the image of the point $\chi(\mathrm{Pt})$ of $\mathcal{E}$ via the Yoneda functor $\mathcal{Y}_{\mathcal{E}}$ is the following R-specification, which is made of a potential mono-sum $X' = X + E$:

$$\mathcal{W}_\chi(\mathrm{Pt}) = \mathcal{Y}_{\mathcal{E}}(\chi(\mathrm{Pt})) \ : \qquad \begin{tikzcd} X \arrow[r, dotted, tail, "i_X"] & X' & E \arrow[l, dotted, tail, "e_X"'] \end{tikzcd}$$

In the same way, the images of the points $\chi(\mathrm{Ar}_k)$ of $\mathcal{E}$ via the Yoneda functor $\mathcal{Y}_{\mathcal{E}}$ are the following R-specifications:

$$\mathcal{W}_\chi(\mathrm{Ar}_*) \ : \qquad \begin{tikzcd} X \arrow[r, dotted, tail, "i_X"] \arrow[d, "f_*"] & X' & E \arrow[l, dotted, tail, "e_X"'] \\ Y \arrow[r, dotted, tail, "i_Y"'] & Y' & E \arrow[l, dotted, tail, "e_Y"] \end{tikzcd}$$

$$\mathcal{W}_\chi(\mathrm{Ar}_?) : \quad \begin{tikzcd} X \arrow[r, tail, dotted, "i_X"] \arrow[dr, "f_?"] & X' & E \arrow[l, tail, dotted, "e_X"'] \\ Y \arrow[r, tail, dotted, "i_Y"'] & Y' & E \arrow[l, tail, dotted, "e_Y"] \end{tikzcd}$$

$$\mathcal{W}_\chi(\mathrm{Ar}_!) : \quad \begin{tikzcd} X \arrow[r, tail, dotted, "i_X"] \arrow[drr, "f_!"] & X' & E \arrow[l, tail, dotted, "e_X"'] \\ Y \arrow[r, tail, dotted, "i_Y"'] & Y' & E \arrow[l, tail, dotted, "e_Y"] \end{tikzcd}$$

In order to describe the image, via χ, of the arrows of $\mathcal{D}$ in $\mathcal{E}$, we can also look at their image via $\mathcal{W}_\chi$ in $\mathcal{S}pec(R)$, keeping in mind that the functor $\mathcal{W}_\chi$ is contravariant. For each keyword k:

- the morphism $\mathcal{W}_\chi(\mathrm{sce}_k)$ maps the point X in $\mathcal{W}_\chi(\mathrm{Pt})$ to the point X in $\mathcal{W}_\chi(\mathrm{Ar}_k)$,
- the morphism $\mathcal{W}_\chi(\mathrm{tgt}_k)$ maps the point X in $\mathcal{W}_\chi(\mathrm{Pt})$ to the point Y in $\mathcal{W}_\chi(\mathrm{Ar}_k)$.

In addition:

- the morphism $\mathcal{W}_\chi(\mathrm{ar}_{*,?})$ maps $f_?$ to $i_Y \circ f_*$,
- the morphism $\mathcal{W}_\chi(\mathrm{ar}_{!,?})$ maps $f_?$ to $e_Y \circ f_!$.

Actually, up to an entailment, an arrow $f_? : X \longrightarrow Y'$, together with the equation $f_? \equiv i_Y \circ f_*$, can be added to $\mathcal{W}_\chi(\mathrm{Ar}_*)$. Similarly, up to an entailment, an arrow $f_? : X \longrightarrow Y'$, together with the equation $f_? \equiv e_Y \circ f_!$, can be added to $\mathcal{W}_\chi(\mathrm{Ar}_?)$. Then the morphisms $\mathcal{W}_\chi(\mathrm{ar}_{*,?})$ and $\mathcal{W}_\chi(\mathrm{ar}_{!,?})$ are the inclusions.

Since the arrow $\mathrm{ar}_{*,?}$ becomes a mono in $\overline{\mathcal{D}}$, it follows from the compatibility property of the Yoneda functor that the morphism $\mathcal{W}_\chi(\mathrm{ar}_{*,?})$ becomes an epimorphism, via F_R, in $\mathcal{D}om(R)$. Indeed, if two morphisms of R-domains $\varphi, \psi : F_R(\mathcal{W}_\chi(\mathrm{Ar}_*)) \longrightarrow \overline{U}$ are such that $\varphi \circ F_R(\mathcal{W}_\chi(\mathrm{ar}_{*,?})) = \psi \circ F_R(\mathcal{W}_\chi(\mathrm{ar}_{*,?}))$, so that $\varphi(i_Y) = \psi(i_Y)$ and $\varphi(f_?) = \psi(f_?)$, then $\varphi(i_Y) \circ \varphi(f_*) = \varphi(i_Y) \circ \psi(f_*)$, and since $\varphi(i_Y)$ is a monomorphism in any R-domain, it follows that $\varphi(f_*) = \psi(f_*)$, as required. In a similar way, the morphism $F_R(\mathcal{W}_\chi(\mathrm{ar}_{!,?}))$ is an epimorphism in $\mathcal{D}om(R)$, because e_Y becomes a monomorphism in any R-domain.

An enrichment of the propagator Q.

Whenever needed, various keywords can be added to Q, together with their interpretation by χ. For instance, it will be useful in section 3.5, in order to handle exceptions, to have a fourth keyword "+" for the arrows in Q-specifications. Until now, all the arrows are assumed to preserve the exceptions. A *loose* arrow is of a more general kind: it does not have to preserve the exceptions.

- "+" means "*loose*".

Since any arrow can be considered as a loose arrow, for each keyword k there is some $\mathrm{ar}_{k,+} : \mathrm{Ar}_k \longrightarrow \mathrm{Ar}_+$ in $\mathcal{D}$, which becomes a mono in $\overline{\mathcal{D}}$.

The Q-domain $set_{\mathbb{E}}$ is completed as follows:

- A *loose map* of $set_{\mathbb{E}}$, i.e. an element of $set_{\mathbb{E}}(\mathrm{Ar}_+)$, is any map $\varphi : A \uplus \mathbb{E} \longrightarrow B \uplus \mathbb{E}$.

The propagator $\chi : Q \longrightarrow R$ is completed as follows:

$$\mathcal{W}_\chi(\mathrm{Ar}_+) \ : \quad \begin{tikzcd}
X \arrow[r, tail, dotted, "i_X"] & X' \arrow[d, "f_+"] & E \arrow[l, tail, dotted, "e_X"'] \\
Y \arrow[r, tail, dotted, "i_Y"'] & Y' & E \arrow[l, tail, dotted, "e_Y"]
\end{tikzcd}$$

For all keyword k among "$*$", "?" and "!", it has been noticed that, up to some entailment, there is an arrow $f_? : X \longrightarrow Y'$ in $\mathcal{W}_\chi(\mathrm{Ar}_k)$. Similarly, up to an entailment, there is an arrow $f_+ : X' \longrightarrow Y'$ in $\mathcal{W}_\chi(\mathrm{Ar}_k)$, together with the equations $f_+ \circ i_X \equiv f_?$ and $f_+ \circ e_X \equiv e_Y$, so that $f_+ = \mathrm{cofact}(f_?, e_Y)$ in any R-domain. Then, the morphism $\mathcal{W}_\chi(\mathrm{ar}_{k,+})$ is the inclusion of $\mathcal{W}_\chi(\mathrm{Ar}_+)$ in $\mathcal{W}_\chi(\mathrm{Ar}_k)$.

Altogether. To sum up, we get the following diagram, usually called a *span*, in the category of propagators:

$$\begin{tikzcd}
 & Q \arrow[dl, "\delta"'] \arrow[dr, "\chi"] & \\
P & & R
\end{tikzcd}$$

3 The decoration step

3.1 The framework for the decoration step

From the example in the introduction, the decoration step in the zooming method proceeds from a far specification S to an intermediate specification T by adding some information to S. The framework for the definition and study of the decoration step is made of a morphism of propagators:

$$\begin{tikzcd} & Q \arrow[ld, "\delta"'] \\ P & \end{tikzcd}$$

In this section, we consider a P-specification S and a Q-domain $\overline{T}$:

$$\begin{tikzcd} & \mathcal{D} \arrow[rr, "Q"] \arrow[ld, "\delta"'] & & \overline{\mathcal{D}} \arrow[ld, "\delta"'] \arrow[ldd, "\overline{T}"] \\ \mathcal{C} \arrow[rr, "P"] \arrow[rrd, bend right, "S"'] & & \overline{\mathcal{C}} & \\ & & \mathcal{S}et & \end{tikzcd}$$

The decorations of S with respect to δ are defined in section 3.2, as well as the corresponding models with value in $\overline{T}$. In section 3.3, the decorations are identified to realizations of a projective sketch. Then decorations with properties are identified to realizations of other projective sketches in section 3.4, and the notion of models is refined. The decoration step for dealing with exceptions is studied in section 3.5.

3.2 The various decorations of a P-specification

Definition 5 A decoration *of a P-specification S with respect to δ is made of a Q-specification T, called the* source *of the decoration, together with a morphism of Q-specifications $\zeta : T \longrightarrow G_\delta(S)$.*

Let (T, ζ) be a decoration of S. In some applications, including the treatment of exceptions, we are interested in the set of Q-models of T with values in $\overline{T}$:

$$\boxed{\mathrm{Mod}_Q(T, \overline{T})\ .}$$

However, it will be seen in section 3.4 that the relevant set of models, in general, is somewhat different from $\mathrm{Mod}_Q(T, \overline{T})$: it does depend on ζ, not only on T.

Let us now consider the triples (S, T, ζ) with $S \in \mathcal{S}pec(P)$, $T \in \mathcal{S}pec(Q)$ and $\zeta : T \longrightarrow G_\delta(S)$, so that (T, ζ) is a decoration of S with respect to δ. A *morphism* from (S_1, T_1, ζ_1) to (S_2, T_2, ζ_2) is defined as a pair $(\sigma : S_1 \longrightarrow S_2, \tau : T_1 \longrightarrow T_2)$ where σ is a morphism of P-specifications and τ is a morphism of

Q-specifications, such that $G_\delta(\sigma) \circ \zeta_1 = \zeta_2 \circ \tau$.

$$\begin{tikzcd}
 & T_1 \arrow[r, "\tau"] \arrow[dl, "\zeta_1"'] & T_2 \arrow[dl, "\zeta_2"] \\
G_\delta(S_1) \arrow[r, "G_\delta(\sigma)"] & G_\delta(S_2) &
\end{tikzcd}$$

Then it is straightforward to define the identities and the composition of morphisms in order to get a category $\mathcal{R}(\delta)$.

In the following sections, we identify decorations of S with respect to δ and specifications with respect to some propagator.

3.3 Decorations as specifications

In order to identify the decorations of S to specifications with respect to some propagator, we introduce the notion of *lax-colimit* of a morphism. This notion can be defined in the general context of 2-categories. Here, we simply define the lax-colimit of a morphism of projective sketches, then the lax-colimit of a morphism of propagators.

Definition 6 *The* lax-colimit *of a morphism of projective sketches $\delta : \mathcal{D} \longrightarrow \mathcal{C}$ is the projective sketch $\mathcal{L}\mathrm{ax}(\delta)$ which is made of:*

- *a copy of $\mathcal{C}$,*
- *a copy of $\mathcal{D}$,*
- *for each point D of $\mathcal{D}$, an arrow $tr_D : D \longrightarrow C$ where $C = \delta(D)$, which is called the* transition arrow *associated to D,*
- *for each arrow $d : D_1 \longrightarrow D_2$ of $\mathcal{D}$, a commutative square $c \circ tr_{D_1} = tr_{D_2} \circ d$ where $c = \delta(d)$.*

$$\begin{tikzcd}
D_1 \arrow[r, "d"] \arrow[d, "tr_{D_1}"'] & D_2 \arrow[d, "tr_{D_2}"] \\
C_1 = \delta(D_1) \arrow[r, "c = \delta(d)"'] & C_2 = \delta(D_2)
\end{tikzcd}$$

Let $\mathcal{L}_0 = \mathcal{L}\mathrm{ax}(\delta)$. From the definition of the lax-colimit, both $\mathcal{C}$ and $\mathcal{D}$ are parts of $\mathcal{L}_0$. The inclusions are denoted $\mu_0 : \mathcal{C} \longrightarrow \mathcal{L}_0$ and $\nu_0 : \mathcal{D} \longrightarrow \mathcal{L}_0$. In addition, the transition arrows determine a natural transformation $tr_0 : \nu_0 \Rightarrow \mu_0 \circ \delta : \mathcal{D} \longrightarrow \mathcal{L}_0$. It should be noted that tr_0 is not a natural transformation between functors, as usual in category theory, but a natural transformation between morphisms of compositive graphs. The composition of such natural transformations can be defined, although not as canonically as in categories. In this paper we only need to compose a natural transformation with a morphism, in either direction, which is indeed similar to the usual composition of a natural transformation with a functor.

$$\begin{tikzcd}
& \mathcal{L}_0 & \\
& & \mathcal{D} \arrow[ul, "\nu_0"', ""{name=N}] \arrow[dll, "\delta"] \\
\mathcal{C} \arrow[uur, bend left, "\mu_0", ""'{name=M}] & &
\arrow[Rightarrow, from=N, to=M, "tr_0"']
\end{tikzcd}$$

The following result is easy to check.

Proposition 7 *The category $\mathcal{R}eal(\mathcal{L}_0)$ of realizations of the lax-colimit of δ is isomorphic to the category $\mathcal{R}(\delta)$.*

In a similar way, there is a lax-colimit $\overline{\mathcal{L}}_0 = \mathcal{L}\mathrm{ax}(\delta : \overline{\mathcal{D}} \longrightarrow \overline{\mathcal{C}})$ with the inclusions $\mu_0 : \overline{\mathcal{C}} \longrightarrow \overline{\mathcal{L}}_0$ and $\nu_0 : \overline{\mathcal{D}} \longrightarrow \overline{\mathcal{L}}_0$, and with the natural transformation $tr_0 : \nu_0 \Rightarrow \mu_0 \circ \delta : \overline{\mathcal{D}} \longrightarrow \overline{\mathcal{L}}_0$. Then it is easy to check that there is a unique propagator:

$$L_0 : \mathcal{L}_0 \longrightarrow \overline{\mathcal{L}}_0$$

such that the pairs $(\mu_0 : \mathcal{C} \longrightarrow \mathcal{L}_0, \mu_0 : \overline{\mathcal{C}} \longrightarrow \overline{\mathcal{L}}_0)$ and $(\nu_0 : \mathcal{D} \longrightarrow \mathcal{L}_0, \nu_0 : \overline{\mathcal{D}} \longrightarrow \overline{\mathcal{L}}_0)$ are morphisms of propagators $\mu_0 : P \longrightarrow L_0$ and $\nu_0 : Q \longrightarrow L_0$.

Both natural transformations called tr_0 are such that $L_0 \circ tr_0 = tr_0 \circ Q$. This yields a *natural transformation* between the corresponding morphisms of propagators:

$$tr_0 : \nu_0 \Rightarrow \mu_0 \circ \delta : Q \longrightarrow L_0 \; .$$

$$\begin{tikzcd}
& \mathcal{L}_0 \arrow[rrr, "L_0"] & & & \overline{\mathcal{L}}_0 & \\
& & \mathcal{D} \arrow[ul, "\nu_0"', ""{name=N1}] \arrow[dll, "\delta"] \arrow[rrr, "Q"] & & & \overline{\mathcal{D}} \arrow[ul, "\nu_0"', ""{name=N2}] \arrow[dll, "\delta"] \\
\mathcal{C} \arrow[uur, bend left, "\mu_0", ""'{name=M1}] \arrow[rrr, "P"] & & & \overline{\mathcal{C}} \arrow[uur, bend left, "\mu_0", ""'{name=M2}] & &
\arrow[Rightarrow, from=N1, to=M1, "tr_0"']
\arrow[Rightarrow, from=N2, to=M2, "tr_0"']
\end{tikzcd}
\qquad
\begin{tikzcd}
& L_0 & \\
& & Q \arrow[ul, "\nu_0"', ""{name=N}] \arrow[dll, "\delta"] \\
P \arrow[uur, bend left, "\mu_0", ""'{name=M}] & &
\arrow[Rightarrow, from=N, to=M, "tr_0"']
\end{tikzcd}$$

Definition 8 *The* lax-colimit *of a morphism of propagators $\delta : Q \longrightarrow P$ is the unique propagator:*

$$L_0 : \mathcal{L}_0 \longrightarrow \overline{\mathcal{L}}_0 \;=\; \mathrm{Lax}(\delta : Q \longrightarrow P) : \mathcal{L}\mathrm{ax}(\delta : \mathcal{D} \longrightarrow \mathcal{C}) \longrightarrow \mathcal{L}\mathrm{ax}(\delta : \overline{\mathcal{D}} \longrightarrow \overline{\mathcal{C}})$$

such that $\mu_0 : P \longrightarrow L_0$ and $\nu_0 : Q \longrightarrow L_0$ are morphisms of propagators.

Let $Z = (S, T, \zeta)$ be a L_0-specification, with $\zeta : T \longrightarrow G_\delta(S)$. Then ζ freely generates a morphism of Q-domains:

$$F_Q(\zeta) : F_Q(T) \longrightarrow F_Q(G_\delta(S)) \; .$$

In addition, we know from section 2.3 that there is a morphism of Q-domains:

$$\Phi_\delta(S) : F_Q(G_\delta(S)) \longrightarrow G_\delta(F_P(S)) \ .$$

By composition, we get a decoration of $F_P(S)$ with respect to δ:

$$\Phi_\delta(S) \circ F_Q(\zeta) : F_Q(T) \longrightarrow G_\delta(F_P(S)) \ .$$

Proposition 9 *The morphisms of propagators* $\mu_0 : P \longrightarrow L_0$ *and* $\nu_0 : Q \longrightarrow L_0$ *are reliable.*

Proof. From the definition of reliability in section 2.3, this means that for each L_0-specification Z there are two isomorphisms:

$$F_P(G_{\mu_0}(Z)) \cong G_{\mu_0}(F_{L_0}(Z)) \text{ and } F_Q(G_{\nu_0}(Z)) \cong G_{\nu_0}(F_{L_0}(Z)) \ .$$

Let $Z = (S, T, \zeta)$, then the triple $(F_P(S), F_Q(T), \Phi_\delta(S) \circ F_Q(\zeta))$ is a L_0-domain, and it is easy to check that it is isomorphic to $F_{L_0}(Z)$, which concludes the proof. □

It follows that:

$$F_P(S) = F_P(G_{\mu_0}(Z)) \cong G_{\mu_0}(F_L(Z)) \text{ and } F_Q(T) = F_P(G_{\nu_0}(Z)) \cong G_{\nu_0}(F_L(Z)) \ .$$

To sum up, a L_0-specification Z can be identified to a decoration of some S with some source T. Then, the freely generated L_0-domain $F_{L_0}(Z)$ can be identified to a decoration of $F_P(S)$ with source $F_Q(T)$, and:

$$\boxed{\mathrm{Mod}_Q(T, \overline{T}) \cong \mathrm{Hom}_{\mathcal{D}om(Q)}(G_{\nu_0}(F_{L_0}(Z)), \overline{T}) \ .}$$

3.4 Decorations with properties

In order to focus on the decorations of S which satisfy some given property, we define the *lax-cocones* over δ.

Definition 10 *A* lax-cocone Λ *over a morphism of propagators* $\delta : Q \longrightarrow P$ *is a propagator:*

$$L : \mathcal{L} \longrightarrow \overline{\mathcal{L}}$$

together with two morphisms $\mu : P \longrightarrow L$ *and* $\nu : Q \longrightarrow L$ *and a natural transformation:*

$$tr : \nu \Rightarrow \mu \circ \delta : Q \longrightarrow L \ .$$

$$\begin{tikzcd}
\mathcal{L} \arrow[rr, "L"] & & \overline{\mathcal{L}} & \\
 & \mathcal{D} \arrow[ul, "\nu"'] \arrow[dl, "\delta"] \arrow[rr, "Q"] \arrow[l, phantom, "\overset{tr}{\Longleftarrow}"] & & \overline{\mathcal{D}} \arrow[ul, "\nu"'] \arrow[dl, "\delta"] \arrow[l, phantom, "\overset{tr}{\Longleftarrow}"] \\
\mathcal{C} \arrow[uu, bend left, "\mu"] \arrow[rr, "P"] & & \overline{\mathcal{C}} \arrow[uu, bend left, "\mu"] &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & L & \\
 & \overset{tr}{\Longleftarrow} & Q \arrow[ul, "\nu"'] \arrow[dll, "\delta"] \\
P \arrow[uur, bend left, "\mu"] & &
\end{tikzcd}$$

So, the lax-colimit L_0 of δ defines a lax-cocone Λ_0 over δ. In addition, for each lax-cocone $\Lambda = (L, \mu, \nu, tr)$ over δ, there is a unique morphism $\kappa : L_0 \longrightarrow L$ such that $\mu = \kappa \circ \mu_0$, $\nu = \kappa \circ \nu_0$ and $tr = \kappa \circ tr_0$. So, a L-specification Z gives rise to a L_0-specification $Z_0 = G_\kappa(Z)$, which can be identified to the triple (S, T, ζ) where $S = G_{\mu_0}(Z_0) = G_\mu(Z)$, $T = G_{\nu_0}(Z_0) = G_\nu(Z)$ and $\zeta : T \longrightarrow G_\delta(S)$ is $Z \circ tr : Z \circ \nu \longrightarrow Z \circ \mu \circ \delta$.

Definition 11 *A decoration (T, ζ) of S with respect to δ* satisfies *Λ if the L_0-specification (S, T, ζ) is the image, via G_κ, of a L-specification.*

Let us assume that $\kappa : L_0 \longrightarrow L$ is fractioning. This means that both functors G_κ are full and faithful, so that $\mathcal{S}pec(L)$ can be identified to a full subcategory of $\mathcal{S}pec(L_0)$ and $\mathcal{D}om(L)$ to a full subcategory of $\mathcal{D}om(L_0)$. It follows that the L-specifications can be identified to the decorations of S with respect to δ which satisfy Λ.

Let us assume that κ is fractioning and μ is reliable. Then the freely generated L-domain $F_L(Z)$ can be identified to a decoration of $F_P(S)$ with source $G_\nu(F_L(Z))$.

Let us assume that κ is fractioning and both μ and ν are reliable. Then $F_L(Z)$ can be identified to a decoration of $F_P(S)$ with source $F_Q(T)$.

To sum up, if $\kappa : L_0 \longrightarrow L$ is fractioning, a L-specification Z can be identified to a decoration of some S with some source T. Then, the freely generated L-domain $F_L(Z)$ can be identified to a decoration of some $\overline{S}$ with some source $\overline{T}$. If μ is reliable then $\overline{S} \cong F_P(S)$, and if ν is reliable then $\overline{T} \cong F_Q(T)$. In this paper, we focus on situations where all these assumptions are satisfied, so that:

$$\boxed{\mathrm{Mod}_Q(T, \overline{T}) \cong \mathrm{Hom}_{\mathcal{D}om(Q)}(G_\nu(F_L(Z)), \overline{T}) \ .}$$

Definition 12 *The* monomorphic lax-colimit *of δ is the lax-cocone Λ_1 over δ which is obtained by adding to both sketches $\mathcal{L}\mathrm{ax}(\delta : \mathcal{D} \longrightarrow \mathcal{C})$ and $\mathcal{L}\mathrm{ax}(\delta : \overline{\mathcal{D}} \longrightarrow \overline{\mathcal{C}})$ a distinguished cone Γ_D for each point D in $\mathcal{D}$, which ensures that* *the transition arrow* $tr_D : D \longrightarrow \delta(D)$ *is a mono.*

$$\begin{tikzcd}
\Gamma_D : & & D \arrow[dl, "\mathrm{id}_D"'] \arrow[dr, "\mathrm{id}_D"] & \\
& D \arrow[dr, "tr_D"'] & & D \arrow[dl, "tr_D"] \\
& & \delta(D) &
\end{tikzcd}$$

Then, clearly, a decoration (T, ζ) of S satisfies Λ_1 if and only if it is *monomorphic*, which means that for all point D in $\mathcal{D}$ the map $\zeta_D : T(D) \longrightarrow S(C)$, where $C = \delta(D)$, is injective. The monomorphic decorations of S can be easily described, up to isomorphism, in the following way. Let (T, ζ) be a monomorphic decoration of S. Then for each point D of $\mathcal{D}$ there is an injective map $\zeta_D : T(D) \longrightarrow S(C)$, where $C = \delta(D)$. Up to isomorphism, $T(D)$ can be replaced by its image in $S(C)$ and ζ_D by the inclusion. Then, for each arrow d of $\mathcal{D}$, the map $T(d)$ is the restriction of the map $S(c)$, where $c = \delta(d)$. For each point C of $\mathcal{C}$ and each element x of $S(C)$, let us define the *keywords* of x as the points D such that $\delta(D) = C$ and x is in $T(D)$. Then clearly the monomorphic decoration (T, ζ) of S is determined, up to isomorphism, by this set of keywords, for each element x. In this situation, it is routine to check that κ is fractioning and that μ is reliable. The morphism ν is not reliable, in general, however in this paper we focus on situations where ν is reliable.

3.5 A decoration for dealing with exceptions

The propagator P and the P-specification S are as in section 2.5. In section 2.5, there is a partial description of the propagator Q, the morphism $\delta : Q \longrightarrow P$, the Q-specification T, the morphism $\zeta : T \longrightarrow G_\delta(S)$, the Q-domain $set_{\mathbb{E}}$ and the model M of T with values in $set_{\mathbb{E}}$. There, only the decoration of the points and arrows have been defined. Now, let us look more closely at the decoration of the equations and of the arity constraints, which allow to state that an operation is either a constant or a n-ary operation.

Equations. In $\mathcal{C}$, there are two points Par and Eq, which stand respectively for *parallel pair* and for *equation*, and a mono $j' : \mathrm{Eq} \rightarrowtail \mathrm{Par}$:

$$\begin{tikzcd}
\mathrm{Pt} & \mathrm{Ar} \arrow[l, bend right, "\mathrm{sce}"'] \arrow[l, bend left, "\mathrm{tgt}"] & \mathrm{Par} \arrow[l, bend right, "p_3"'] \arrow[l, bend left, "p_4"] & \mathrm{Eq} \arrow[l, tail, "j'"']
\end{tikzcd}$$

The point Par is the vertex of a distinguished cone Γ_{Par} which formalizes the definition of parallel pairs:

$$\Gamma_{\mathrm{Par}}\ : \quad \begin{tikzcd} & \mathrm{Par} \arrow[dl, "p_3"'] \arrow[dr, "p_4"] & \\ \mathrm{Ar} \arrow[d, "\mathrm{sce}"'] \arrow[drr, "\mathrm{tgt}" near start] & & \mathrm{Ar} \arrow[d, "\mathrm{tgt}"] \arrow[dll, "\mathrm{sce}"' near start] \\ \mathrm{Pt} & & \mathrm{Pt} \end{tikzcd}$$

In $\mathcal{D}$, for each pair (k, l) of keywords for arrows, there is a point $\mathrm{Par}_{k,l}$ above Par (with respect to δ) and a distinguished cone $\Gamma_{\mathrm{Par}_{k,l}}$ above Γ_{Par}:

$$\Gamma_{\mathrm{Par}_{k,l}}\ : \quad \begin{tikzcd} & \mathrm{Par}_{k,l} \arrow[dl] \arrow[dr] & \\ \mathrm{Ar}_k \arrow[d, "\mathrm{sce}_k"'] \arrow[drr, "\mathrm{tgt}_k" near start] & & \mathrm{Ar}_l \arrow[d, "\mathrm{tgt}_l"] \arrow[dll, "\mathrm{sce}_l"' near start] \\ \mathrm{Pt} & & \mathrm{Pt} \end{tikzcd}$$

In addition, there is at least one point Eq_e above Eq and a mono j'_e above j':

$$\begin{tikzcd} \mathrm{Pt} & \mathrm{Ar}_+ \arrow[l, shift right, "\mathrm{sce}_+"'] \arrow[l, shift left, "\mathrm{tgt}_+"'] & \mathrm{Par}_{+,+} \arrow[l, shift right] \arrow[l, shift left] & \mathrm{Eq}_e \arrow[l, tail, "j'_e"'] \end{tikzcd}$$

The point Eq_e in $\mathcal{D}$ corresponds to a keyword for equations:

- "e" means "everywhere".

An equation with keyword "e" is written as:

$$(f : +) \equiv_e (g : +) \ .$$

The description of the keyword "e" is given by the R-specification:

$$\mathcal{W}_X(\mathrm{Eq}_e)\ : \quad \begin{tikzcd} X \arrow[r, tail, dotted, "i_X"] & X' \arrow[d, shift right, "g_+"'] \arrow[d, shift left, "f_+"] & E \arrow[l, tail, dotted, "e_X"'] \\ Y \arrow[r, tail, dotted, "i_Y"'] & Y' & E \arrow[l, tail, dotted, "e_Y"] \end{tikzcd} \qquad f_+ \equiv g_+$$

In addition, the keywords for the arrows and the keywords for the equations can be combined, by means of distinguished cones in $\mathcal{D}$, as follows. Let $f : k$ and $g : l$ for any keywords k and l. From the fact that the keyword "+" is more general than any other keyword it follows that $f : +$ and $g : +$. Then:

$$(f : k) \equiv_e (g : l)$$

means that $f : k$ and $g : l$ and $(f : +) \equiv_e (g : +)$. In $\mathcal{D}$, this conjunction of decorations corresponds to a point $\mathrm{Eq}_{e,k,l}$ which is the vertex of a distinguished cone:

$$\begin{tikzcd}
& \mathrm{Eq}_{e,k,l} \arrow[rrr] \arrow[d] & & & \mathrm{Eq}_e \arrow[d] & \\
& \mathrm{Par}_{k,l} \arrow[dl] \arrow[dr] & & & \mathrm{Par}_{+,+} \arrow[dl] \arrow[dr] & \\
\mathrm{Ar}_k \arrow[rrr, bend right, "\mathrm{ar}_{k,+}"'] & & \mathrm{Ar}_l \arrow[rrr, bend right, "\mathrm{ar}_{l,+}"'] & \mathrm{Ar}_+ & & \mathrm{Ar}_+
\end{tikzcd}$$

Then, for instance, it is easy to formalize the following property: "if two operations are interpreted in the same way, and if the second one is known to be non-erroneous, then the first one also is non-erroneous":

$$\frac{(f :?) \equiv_e (g : *)}{(f : *) \equiv_e (g : *)}$$

Indeed, the arrow $\mathrm{ar}_{*,?} : \mathrm{Ar}_* \longrightarrow \mathrm{Ar}_?$ in $\mathcal{D}$ gives rise to an arrow:

$$\mathrm{Eq}_{e,*,*} \longrightarrow \mathrm{Eq}_{e,?,*}$$

and the inversion of this arrow in $\overline{\mathcal{D}}$ corresponds to this property.

In $set_{\mathbb{E}}$, let $\varphi : A \uplus \mathbb{E} \longrightarrow B \uplus \mathbb{E}$ and $\psi : A \uplus \mathbb{E} \longrightarrow B \uplus \mathbb{E}$, so that (φ, ψ) is a parallel pair.

- An *everywhere equation* $\varphi \equiv_e \psi$ of $set_{\mathbb{E}}$, i.e. an element of $set_{\mathbb{E}}(\mathrm{Eq}_e)$, is simply the equality of maps $\varphi = \psi$.

It follows that, for any a in A, $\varphi(a)$ raises an exception (i.e., $\varphi(a) \in \mathbb{E}$) if and only if $\psi(a)$ raises an exception.

In our example, the equations of T are:

$$(p \circ s :?) \equiv_e (\mathrm{id}_N : *) \ , \ (p \circ z :?) \equiv_e (q :!) \ ,$$

hence it can be deduced that $p \circ s : *$ and that $p \circ z :!$.

More equations. Other keywords for equations can be added, for instance:

- "*nea*" means "only-on-non-erroneous-arguments",
- "*nev*" means "only-when-non-erroneous-values",

which correspond to two other points Eq_{nea} and Eq_{nev} in $\mathcal{D}$ above Eq. The description of the keyword "*nea*" is given by the R-specification:

$$\mathcal{W}_{\mathcal{X}}(\mathrm{Eq}_{nea}) \ : \qquad \begin{tikzcd} X \arrow[r, tail, dotted, "i_X"] & X' \arrow[d, bend right, "g_+"'] \arrow[d, bend left, "f_+"] & E \arrow[l, tail, dotted, "e_X"'] \\ Y \arrow[r, tail, dotted, "i_Y"'] & Y' & E \arrow[l, tail, dotted, "e_Y"] \end{tikzcd} \qquad f_+ \circ i_X \equiv g_+ \circ i_X$$

For the description of the keyword "*nev*", the propagator R has to be enriched, in order to allow potential limits in R-specifications. Then:

$$\mathcal{W}_{\mathcal{X}}(\mathrm{Eq}_{nev}) \ : \qquad \begin{tikzcd} H \arrow[dd, bend right=60, "f'"'] \arrow[dd, bend left, "g'"] \arrow[dr, "h"] & & \\ X \arrow[r, tail, dotted] & X' \arrow[d, bend right, "f_+"'] \arrow[d, bend left, "g_+"] & E \arrow[l, tail, dotted] \\ Y \arrow[r, tail, dotted, "i_Y"'] & Y' & E \arrow[l, tail, dotted, "e_Y"] \end{tikzcd} \qquad \begin{array}{l} f_+ \circ h \equiv g_+ \circ h \\ i_Y \circ f' \equiv f_+ \circ h \\ i_Y \circ g' \equiv g_+ \circ h \end{array}$$

with the following potential limit, which ensures that H is interpreted as the part of X' where both f_+ and g_+ are non-erroneous, and h as the inclusion:

$$\begin{tikzcd} & & H \arrow[d, "h"] \arrow[ddll, "f'"'] \arrow[ddrr, "g'"] & & \\ & & X' \arrow[dl, "f_+"'] \arrow[dr, "g_+"] & & \\ Y \arrow[r, tail, dotted, "i_Y"'] & Y' & & Y' & Y \arrow[l, tail, dotted, "i_Y"'] \end{tikzcd} \qquad \begin{array}{l} i_Y \circ f' \equiv f_+ \circ h \\ i_Y \circ g' \equiv g_+ \circ h \end{array}$$

Then in $set_{\mathbb{E}}$, let $\varphi, \psi : A \uplus \mathbb{E} \longrightarrow B \uplus \mathbb{E}$.

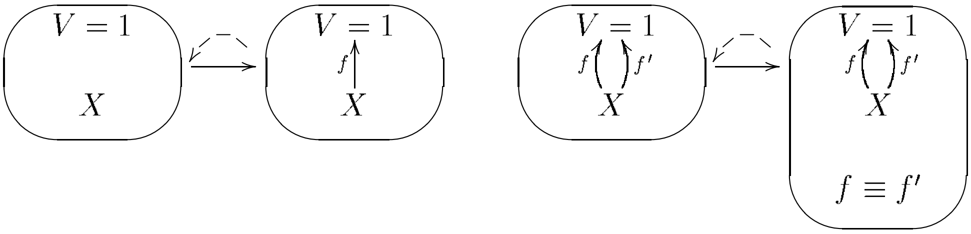
- An *only-on-non-erroneous-arguments equation* $\varphi \equiv_{nea} \psi$ i.e. an element of $set_{\mathbb{E}}(\mathrm{Eq}_{nea})$, is an equality only when the arguments of both maps are non-erroneous: $\varphi(a) = \psi(a)$ for all a in A.
- An *only-when-non-erroneous-values equation* $\varphi \equiv_{nev} \psi$ of $set_{\mathbb{E}}$, i.e. an element of $set_{\mathbb{E}}(\mathrm{Eq}_{nev})$, is an equality only when the values of both maps are non-erroneous: $\varphi(a) = \psi(a)$ for all a in A such that $\varphi(a) \notin \mathbb{E}$ and $\varphi(b) \notin \mathbb{E}$. It follows that there can be some a in $A \uplus \mathbb{E}$ such that $\varphi(a) \in \mathbb{E}$ and $\psi(a) \notin \mathbb{E}$.

As an example of a "only-when-non-erroneous-values" equation, the equation $s \circ p \equiv \mathrm{id}_N$ could be added to S, and the equation $(s \circ p :?) \equiv_{nev} (\mathrm{id}_N : *)$ to T. Indeed, in the model M, the interpretation of $(s \circ p) :?$ is the map such that $\epsilon \mapsto \epsilon$, $n \mapsto n$ for all $n \neq 0$ in $\mathbb{N}$, and $0 \mapsto \epsilon$: it does coincide with the identity map on every $n \in \mathbb{N}$ such that the successor of the predecessor of n is non-erroneous.

Constants. The propagator P has to deal with constants. A constant is a nullary operation, which means that its source is a potential terminal point: indeed, in the category of sets, a terminal point is a singleton $\{v\}$, so that any element x in any set X can be identified with the map $v \mapsto x$ from $\{v\}$ to X.

A terminal point V in a category satisfies the following property: for each point X there is a unique arrow $\mathrm{fact}(X)$ from X to V. So, in the sketch $\mathcal{C}$ there is a point Term which stands for *terminal points*, and a mono $i : \mathrm{Term} \rightarrowtail \mathrm{Pt}$.

In $\overline{\mathcal{C}}$, the universal property of a terminal point V is described by the inversion of two arrows of $\mathcal{C}$: the first one for the existence of an arrow $f : X \longrightarrow V$, the second one for its unicity. This is illustrated as follows.



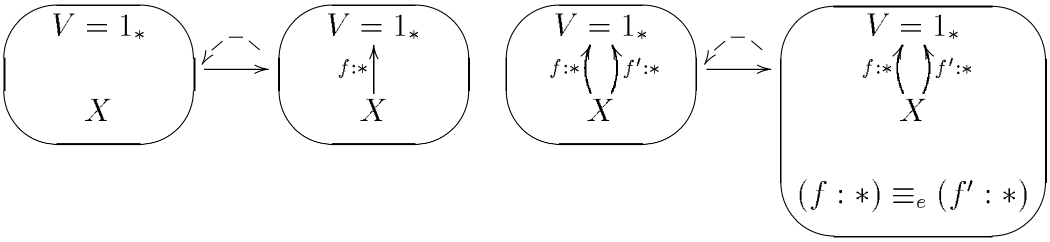
In $\overline{\mathcal{D}}$, these arrows f and f' have to be decorated, as well as the equation $f \equiv f'$. Let us consider three cases, where f and f' have the same keyword, either "$*$" or "?" or "!", and the equation is "everywhere".

- If f and f' are decorated with the keyword "$*$", this property means that V is interpreted as $M(V) = B \uplus \mathbb{E}$ such that, for each $A \uplus \mathbb{E}$, there is a unique map $\varphi : A \uplus \mathbb{E} \longrightarrow B \uplus \mathbb{E}$ which is the identity on $\mathbb{E}$ and such that $\varphi(a)$ is in B for all a in A; it is easy to check that $M(V)$ is $\{v\} \uplus \mathbb{E}$ for any singleton $\{v\}$.
- If f and f' are decorated with the keyword "?", this property means that V is interpreted as $M(V) = B \uplus \mathbb{E}$ such that, for each $A \uplus \mathbb{E}$, there is a unique map $\varphi : A \uplus \mathbb{E} \longrightarrow B \uplus \mathbb{E}$ which is the identity on $\mathbb{E}$; it is easy to check that $M(V)$ does not exist, except when $\mathbb{E}$ is a singleton, in which case $M(V)$ is $\mathbb{E}$.
- If f and f' are decorated with the keyword "!", this property means that V is interpreted as $M(V) = B \uplus \mathbb{E}$ such that, for each $A \uplus \mathbb{E}$, there is a unique map $\varphi : A \uplus \mathbb{E} \longrightarrow B \uplus \mathbb{E}$ which is the identity on $\mathbb{E}$ and such that $\varphi(a)$ is in $\mathbb{E}$ for all a in A; it is easy to check that $M(V)$ does not exist, except when $\mathbb{E}$ is a singleton, in which case $M(V)$ can be $B \uplus \mathbb{E}$ for any set B.

Actually, we are interested in a model M of T with values in $set_{\mathbb{E}}$ such that $M(V)$ is $\{v\} \uplus \mathbb{E}$, so that $M(z : *)$ can be the map such that $v \mapsto 0$. For this purpose, we choose the keyword "$*$" for f and f' and "e" for the equation. So, in $\mathcal{D}$ there is a point Term_* which stands for *terminal points with respect to the non-erroneous arrows.* In $\overline{\mathcal{D}}$, the corresponding property is illustrated as follows, with the convention that:

$$V = 1_*$$

means that V is a potential terminal point with respect to the non-erroneous arrows.



The description of the Q-specification T can now be completed. The constraint $V = 1$ in S is decorated as $V = 1_*$ in T. So, altogether:

`Specification` T:
`points`: V, N;
`arrows`: $(z : *) : V \longrightarrow N$, $(s : *) : N \longrightarrow N$, $(p :?) : N \longrightarrow N$, $(q :!) : V \longrightarrow N$.
`equations`: $(p \circ s :?) \equiv_e (\mathrm{id}_N : *)$, $(p \circ z :?) \equiv_e (q :!)$, ;
`constraint`: $V = 1_*$.

$$\begin{tikzcd}
V = 1_* \arrow[d, shift right, "z{:}*"'] \arrow[d, shift left, "q{:}!"] \\
N \arrow[loop left, "s{:}*"] \arrow[loop right, "p{:}?"]
\end{tikzcd}
\qquad
\begin{array}{c}
(p \circ s :?) \equiv_e (\mathrm{id}_N : *) \\
(p \circ z :?) \equiv_e (q :!)
\end{array}$$

Then the morphism $\zeta : T \longrightarrow G_\delta(S)$ is easily completed, in such a way that (T, ζ) is a monomorphic decoration of S. So, (S, T, ζ) can be identified to a L_1-specification Z, where L_1 is the monomorphic lax-colimit of δ as defined in section 3.4. As with any monomorphic lax-colimit, the morphism κ is fractioning and the morphism μ is reliable. In addition, here, it happens that the morphism ν is also reliable, so that indeed:

$$\mathrm{Mod}_Q(T, set_{\mathbb{E}}) \cong \mathrm{Hom}_{\mathcal{D}om(Q)}(G_\nu(F_L(Z)), set_{\mathbb{E}}) \ .$$

Binary operations. An operation is n-ary when its source is the vertex of a potential product of n sorts. Let us consider binary operations: $n = 2$.

The way binary operations are dealt with in P is illustrated in section 2.5. The existence and the unicity of the factorization arrow correspond to the inversion of two arrows of $\mathcal{C}$ in $\overline{\mathcal{C}}$:

$$\begin{tikzcd}
& X\times Y \arrow[dl, "p"'] \arrow[dr, "q"] & & & & X\times Y \arrow[dl, "p"'] \arrow[dr, "q"] & \\
X & & Y \arrow[r, phantom, "\overset{\curvearrowleft}{\longrightarrow}"] & \; & X & & Y \\
& Z \arrow[ul, "f"] \arrow[ur, "g"'] & & & & Z \arrow[ul, "f"] \arrow[ur, "g"'] \arrow[uu, "h"] &
\end{tikzcd}$$

$$\begin{array}{c} p\circ h\equiv f \\ q\circ h\equiv g \end{array}$$

$$\begin{tikzcd}
& X\times Y \arrow[dl, "p"'] \arrow[dr, "q"] & & & & X\times Y \arrow[dl, "p"'] \arrow[dr, "q"] & \\
X & & Y \arrow[r, phantom, "\overset{\curvearrowleft}{\longrightarrow}"] & \; & X & & Y \\
& Z \arrow[ul, "f"] \arrow[ur, "g"'] \arrow[uu, bend left, "h"] \arrow[uu, bend right, "h'"'] & & & & Z \arrow[ul, "f"] \arrow[ur, "g"'] \arrow[uu, bend left, "h"] \arrow[uu, bend right, "h'"'] &
\end{tikzcd}$$

$$\begin{array}{c} p\circ h\equiv f \\ p\circ h'\equiv f \\ q\circ h\equiv g \\ q\circ h'\equiv g \end{array} \qquad\qquad \begin{array}{c} p\circ h\equiv f \\ p\circ h'\equiv f \\ q\circ h\equiv g \\ q\circ h'\equiv g \\ h\equiv h' \end{array}$$

In Q, all this has to be decorated.

In any P-domain, the expression "$a(f(z),g(z))$" stands for the composed arrow $a\circ \mathrm{fact}(f,g)\circ z$. The issue is to determine, in Q-domains, with respect to the decorations $(f:k)$ and $(g:l)$ of f and g, whether some kind of factorization arrow of $(f:k)$ and $(g:l)$ does exist, and what are precisely its existence and unicity properties.

When both f and g are non-erroneous, then clearly a factorization arrow should exist: it is non-erroneous, and its properties are obtained by decorating all the arrows with "$*$" and all the equations with "e", as follows.

$$\begin{tikzcd}
& X\times Y \arrow[dl, "p:*"'] \arrow[dr, "q:*"] & & & & X\times Y \arrow[dl, "p:*"'] \arrow[dr, "q:*"] & \\
X & & Y \arrow[r, phantom, "\overset{\curvearrowleft}{\longrightarrow}"] & \; & X & & Y \\
& Z \arrow[ul, "f:*"] \arrow[ur, "g:*"'] & & & & Z \arrow[ul, "f:*"] \arrow[ur, "g:*"'] \arrow[uu, "h:*"] &
\end{tikzcd}$$

$$\begin{array}{c} (p\circ h:*)\equiv_e (f:*) \\ (q\circ h:*)\equiv_e (g:*) \end{array}$$

$$\begin{tikzcd}
& X\times Y \arrow[ld, "p:*"'] \arrow[rd, "q:*"] & \\
X & & Y \\
& Z \arrow[lu, "f:*"] \arrow[ru, "g:*"'] \arrow[uu, bend left, "h:*"] \arrow[uu, bend right, "h':*"'] &
\end{tikzcd}$$

$$\begin{gathered}
(p\circ h:*)\equiv_e(f:*)\\
(p\circ h':*)\equiv_e(f:*)\\
(q\circ h:*)\equiv_e(g:*)\\
(q\circ h':*)\equiv_e(g:*)
\end{gathered}$$

$$\longrightarrow$$

$$\begin{tikzcd}
& X\times Y \arrow[ld, "p:*"'] \arrow[rd, "q:*"] & \\
X & & Y \\
& Z \arrow[lu, "f:*"] \arrow[ru, "g:*"'] \arrow[uu, bend left, "h:*"] \arrow[uu, bend right, "h':*"'] &
\end{tikzcd}$$

$$\begin{gathered}
(p\circ h:*)\equiv_e(f:*)\\
(p\circ h':*)\equiv_e(f:*)\\
(q\circ h:*)\equiv_e(g:*)\\
(q\circ h':*)\equiv_e(g:*)\\
(h:*)\equiv_e(h':*)
\end{gathered}$$

In the Q-domain $set_{\mathbb{E}}$, it is easy to check that:

$$A\uplus\mathbb{E}\times B\uplus\mathbb{E}=(A\times B)\uplus\mathbb{E}\ ,$$

with the obvious projections. Let $\varphi:D\uplus\mathbb{E}\longrightarrow A\uplus\mathbb{E}$ and $\psi:D\uplus\mathbb{E}\longrightarrow B\uplus\mathbb{E}$ be non-erroneous maps in $set_{\mathbb{E}}$. Then the factorization arrow of φ and ψ is the non-erroneous map $\theta:D\uplus\mathbb{E}\longrightarrow(A\times B)\uplus\mathbb{E}$, which is the identity on $\mathbb{E}$ and such that $\theta(d)=(\varphi(d),\psi(d))$ for all d in D.

More about arities. It is well known that there is no hope to get any kind of canonical factorization arrow in general: indeed, whenever f raises an exception, and g raises another exception, there is no canonical way to decide what should happen with any kind of "fact(f,g)" arrow. However, it can be required that the same cone as above, with vertex $X\times Y$ and projections $p:*$ and $q:*$, gives rise to some kind of factorization arrow when one among f and g may be erroneous. Let us assume that f is non-erroneous and g may be erroneous. Then, let us decide that this factorization arrow should raise an exception if and only if g does, and it should be the same exception as g. This corresponds to adding the following properties.

$$\begin{tikzcd}
& X\times Y \arrow[ld, "p:*"'] \arrow[rd, "q:*"] & \\
X & & Y \\
& Z \arrow[lu, "f:*"] \arrow[ru, "g:?"'] &
\end{tikzcd}$$

$$\longrightarrow$$

$$\begin{tikzcd}
& X\times Y \arrow[ld, "p:*"'] \arrow[rd, "q:*"] & \\
X & & Y \\
& Z \arrow[lu, "f:*"] \arrow[ru, "g:?"'] \arrow[uu, "h:?"] &
\end{tikzcd}$$

$$\begin{gathered}
(p\circ h:?)\equiv_{nev}(f:*)\\
(q\circ h:?)\equiv_e(g:?)
\end{gathered}$$

$$\begin{tikzcd}
& X \times Y \arrow[ld, "p:*"'] \arrow[rd, "q:*"] & \\
X & & Y \\
& Z \arrow[lu, "f:*"] \arrow[ru, "g:?"'] \arrow[uu, bend left, "h:?"] \arrow[uu, bend right, "h':?"'] &
\end{tikzcd}$$

$$\begin{gathered}
(p \circ h :?) \equiv_{nev} (f : *) \\
(p \circ h' :?) \equiv_{nev} (f : *) \\
(q \circ h :?) \equiv_{e} (g :?) \\
(q \circ h' :?) \equiv_{e} (g :?)
\end{gathered}$$

$$\begin{tikzcd}
{} \arrow[r] & {} \arrow[l, dashed, bend right]
\end{tikzcd}$$

$$\begin{tikzcd}
& X \times Y \arrow[ld, "p:*"'] \arrow[rd, "q:*"] & \\
X & & Y \\
& Z \arrow[lu, "f:*"] \arrow[ru, "g:?"'] \arrow[uu, bend left, "h:?"] \arrow[uu, bend right, "h':?"'] &
\end{tikzcd}$$

$$\begin{gathered}
(p \circ h :?) \equiv_{nev} (f : *) \\
(p \circ h' :?) \equiv_{nev} (f : *) \\
(q \circ h :?) \equiv_{e} (g :?) \\
(q \circ h' :?) \equiv_{e} (g :?) \\
(h :?) \equiv_{e} (h' :?)
\end{gathered}$$

As required, f is decorated with the keyword "$*$" and g with the keyword "?", as well as h. The equation $(q \circ h :?) \equiv_e (g :?)$ is an everywhere equation, so that h is erroneous if and only if g is erroneous, and both raise the same exception. On the other hand, the equation $(p \circ h :?) \equiv_{nev} (f : *)$ is an only-when-non-erroneous-values equation, since h can raise an exception whereas f cannot. The decorations are the same when h' occurs instead of h, and finally the equation $(h :?) \equiv_e (h' :?)$ is an everywhere equation, so that the factorization arrow is unique in any Q-domain.

In the Q-domain $set_{\mathbb{E}}$, it is easy to check that the product $(A \times B) \uplus \mathbb{E}$ satisfies these properties. Indeed, let $\varphi : D \uplus \mathbb{E} \longrightarrow A \uplus \mathbb{E}$ and $\psi : D \uplus \mathbb{E} \longrightarrow B \uplus \mathbb{E}$ be maps which are identities on $\mathbb{E}$ and such that $\varphi(d)$ is in A for all d in D. Then the factorization arrow of φ and ψ is $\theta : D \uplus \mathbb{E} \longrightarrow (A \times B) \uplus \mathbb{E}$, which is the identity on $\mathbb{E}$ and such that $\theta(d) = \psi(d)$ whenever $\psi(d)$ is in $\mathbb{E}$, and $\theta(d) = (\varphi(d), \psi(d))$ otherwise.

The symmetric situation is easy to describe. Whenever both f and g are non-erroneous, the three factorization arrows coincide.

Exception handling.

Basically, the keyword "+" for arrows allows exception handling.

In order to be more precise, we can define a map for exception handling in $set_{\mathbb{E}}$ as a map $\theta : A \uplus \mathbb{E} \longrightarrow A \uplus \mathbb{E}$ such that $\theta(a) = a$ for all a in A. This corresponds to the notion which is used in (Benton, Hughes and Moggi , 2002). Indeed, if $t : \mathbb{E} \longrightarrow A \uplus \mathbb{E}$ denotes the restriction of θ to $\mathbb{E}$, which can be any map, then θ is entirely determined by t, as follows. Let $a' \in A \uplus \mathbb{E}$:

$$\begin{cases} \text{if } a' \in A \text{ then } \theta(a') = a' \\ \text{if } a' \in \mathbb{E} \text{ then } \theta(a') = t(a') \end{cases}$$

This can be formalized with the help of the keyword "*nea*" for equations, i.e. "only-on-non-erroneous-arguments". Then, in a Q-specification T, an exception handling arrow in T, with respect to a point X, can be defined as an arrow $(h:+):X \longrightarrow X$ such that:

$$(h:+) \equiv_{nea} (\mathrm{id}_X : *) \ .$$

4 The expansion step

4.1 The framework for the expansion step

From the example in the introduction, the expansion step in the zooming method proceeds from an intermediate specification T to a near specification U by expliciting some features in T. The framework for the definition and study of the expansion step is made of a morphism of propagators:

$$\begin{tikzcd}
Q \arrow[dr, "\chi"] & \\
& R
\end{tikzcd}$$

In this section, we consider a Q-specification T and a R-domain $\overline{U}$:

$$\begin{tikzcd}
\mathcal{D} \arrow[rr, "Q"] \arrow[dr, "\chi"] \arrow[ddr, "T"'] & & \overline{\mathcal{D}} \arrow[dr, "\chi"] & \\
& \mathcal{E} \arrow[rr, "R"] & & \overline{\mathcal{E}} \arrow[dll, bend left, "\overline{U}"] \\
& \mathcal{S}et & &
\end{tikzcd}$$

The expansion of T with respect to χ is defined in section 4.2, as well as the corresponding models with value in $\overline{U}$. In section 4.3 it is proven that $F_\chi(T)$ is easy to determine when $\chi : \mathcal{D} \longrightarrow \mathcal{E}$ is filling. The expansion step for dealing with exceptions is studied in section 4.4. Basically, this section is made of variations on the theme of adjunction.

4.2 The unique expansion of a Q-specification

Definition 13 The expansion *of a Q-specification T with respect to χ is the R-specification $F_\chi(T)$.*

We are interested in the set of Q-models of T with values in $G_\chi(\overline{U})$:

$$\boxed{\mathrm{Mod}_Q(T, G_\chi(\overline{U}))\ .}$$

Proposition 14 *The set of Q-models of T with values in $G_\chi(\overline{U})$ is such that:*

$$\boxed{\mathrm{Mod}_Q(T, G_\chi(\overline{U})) \cong \mathrm{Mod}_R(F_\chi(T), \overline{U})\ .}$$

and also:

$$\boxed{\mathrm{Mod}_Q(T, G_\chi(\overline{U})) \cong \mathrm{Hom}_{\mathcal{D}om(R)}(F_\chi(F_Q(T)), \overline{U})\ .}$$

Proof. The first bijection follows immediately from proposition 1. Then, the second bijection comes from the definition of models $\mathrm{Mod}_R(F_\chi(T), \overline{U}) = \mathrm{Hom}_{\mathcal{D}om(R)}(F_R(F_\chi(T)), \overline{U})$, and from the fact that $F_R \circ F_\chi \cong F_\chi \circ F_Q$ because $R \circ \chi = \chi \circ Q$. $\square$

As usual, a category is *cocomplete* if there is at least one colimit cone for each base. If there are several colimit cones for one base, then all of them are isomorphic, so that there is no danger in using the "colim" notation.

Definition 15 *Let $\mathcal{A}$ be a cocomplete category. A* contravariant realization $\mathcal{W} : \mathcal{D} \twoheadrightarrow\!\!\!\!\!\times \mathcal{A}$ of $\mathcal{D}$ with values in $\mathcal{A}$ *is a contravariant functor which maps each distinguished cone in $\mathcal{D}$ to a colimit cone in $\mathcal{A}$.*

For instance, $\mathcal{W} = \mathcal{Y}_\mathcal{D} : \mathcal{D} \twoheadrightarrow\!\!\!\!\!\times \mathcal{R}eal(\mathcal{D})$ is a contravariant realization of $\mathcal{D}$ with values in $\mathcal{R}eal(\mathcal{D})$.

Let us consider the contravariant realization of $\mathcal{D}$ with values in $\mathcal{R}eal(\mathcal{E})$:

$$\mathcal{W}_\chi = \mathcal{Y}_\mathcal{E} \circ \chi : \mathcal{D} \twoheadrightarrow\!\!\!\!\!\times \mathcal{R}eal(\mathcal{E})\ .$$

Some properties of the contravariant realization $\mathcal{W}_\chi$ are easily derived from the properties of the Yoneda functor, as stated in theorem 4.

Theorem 16 *The contravariant realization $\mathcal{W}_\chi$ satisfies the following properties:*

- ***Compatibity property:***

$$\mathcal{W}_\chi \cong F_\chi \circ \mathcal{Y}_\mathcal{D}\ .$$

- ***Yoneda property:*** *For all realization U of $\mathcal{E}$:*

$$G_\chi(U) \cong \mathrm{Hom}_{\mathcal{R}eal(\mathcal{E})}(\mathcal{W}_\chi(-), U)\ .$$

- ***Density property:*** *For all realization T of $\mathcal{D}$:*

$$F_\chi(T) \cong \operatorname{colim}_{(\mathcal{D}\backslash T)^{op}}(\mathcal{W}_\chi(D)) \;.$$

So, when $\mathcal{Y}_\mathcal{E}$ is replaced by $\mathcal{W}_\chi$ in the right handsides of the Yoneda and density isomorphisms, one gets a description of the functors G_χ and F_χ in terms of $\mathcal{W}_\chi$.

Proof. The compatibility property of $\mathcal{W}_\chi$ follows immediately from the compatibility property of the Yoneda functor.
The Yoneda property of $\mathcal{Y}_\mathcal{E}$ implies that $U \circ \chi \cong \operatorname{Hom}_{\mathcal{R}eal(\mathcal{E})}(\mathcal{Y}_\mathcal{E}(\chi(-)), U)$, where $U \circ \chi = G_\chi(U)$, so that the Yoneda property of $\mathcal{W}_\chi$ follows.
The density property of $\mathcal{Y}_\mathcal{C}$ implies that $F_\chi(S) \cong F_\chi(\operatorname{colim}_{(\mathcal{C}\backslash S)^{op}}(\mathcal{Y}_\mathcal{C}(C)))$. It is a well-known fact that left-adjoint functors, like F_χ, preserve the colimits (Mac Lane , 1971). So, $F_\chi(S) \cong \operatorname{colim}_{(\mathcal{C}\backslash S)^{op}}(F_\chi(\mathcal{Y}_\mathcal{C}(C)))$, and finally the result is derived from the compatibity property. □

The following result corresponds to the way our example has been presented in the introduction: it states that an ingredient of T with keyword k should be interpreted as a model of a specification $\mathcal{W}(k)$.

Proposition 17 *The set of Q-models of T with values in $G_\chi(\overline{U})$ is such that:*

$$\boxed{\operatorname{Mod}_Q(T, G_\chi(\overline{U})) \cong \operatorname{Hom}_{\mathcal{S}pec(Q)}(T, \operatorname{Mod}_R(\mathcal{W}_\chi(-), \overline{U}))}$$

Proof. Let $\overline{T} = G_Q(G_\chi(\overline{U})))$, so that $\operatorname{Mod}_Q(T, G_\chi(\overline{U})) \cong \operatorname{Hom}_{\mathcal{S}pec(Q)}(T, \overline{T})$. Since $\chi \circ Q = R \circ \chi$, it follows that $\overline{T} = G_\chi(G_R(\overline{U}))$. According to the Yoneda property of $\mathcal{W}_\chi$, this implies that $\overline{T} \cong \operatorname{Hom}_{\mathcal{R}eal(\mathcal{E})}(\mathcal{W}_\chi(-), G_R(\overline{U}))$, which means that $\overline{T} \cong \operatorname{Mod}_R(\mathcal{W}_\chi(-), \overline{U})$, as required. □

4.3 A construction of the expansion

Proposition 18 *For all realization T of $\mathcal{D}$ there is an isomorphism:*

$$T \cong \operatorname{Hom}_{\mathcal{R}eal(\mathcal{E})}(\mathcal{W}_\chi(-), F_\chi(T))$$

if and only if the morphism of projective sketches $\chi : \mathcal{D} \longrightarrow \mathcal{E}$ is filling. Then:

$$\operatorname{Hom}_{\mathcal{R}eal(\mathcal{E})}(\mathcal{W}_\chi(-), F_\chi(T)) \cong \operatorname{Hom}_{\mathcal{R}eal(\mathcal{D})}(\mathcal{Y}_\mathcal{D}(-), T) \;.$$

Proof. From the Yoneda property of $\mathcal{W}_\chi$:

$$G_\chi(F_\chi(T)) \cong \mathrm{Hom}_{\mathcal{R}eal(\mathcal{E})}(\mathcal{W}_\chi(-), F_\chi(T)) \; .$$

On the other hand, $T \cong G_\chi(F_\chi(T))$ for all T if and only if $\chi : \mathcal{D} \longrightarrow \mathcal{E}$ is filling, which proves the first isomorphism. Then, the second isomorphism is an immediate consequence of the Yoneda property of $\mathcal{Y}_\mathcal{D}$. $\square$

This result means that, for all point D of $\mathcal{D}$, there are "as many" copies of $\mathcal{W}_\chi(D)$ in $F_\chi(T)$ as there are copies of $\mathcal{Y}_\mathcal{D}(D)$ in T. So, when χ is filling, $F_\chi(T)$ can be computed from T *pointwise*, where the *points* must be understood as the points $\mathcal{W}_\chi(D)$ in the category $\mathcal{R}eal(\mathcal{E})$.

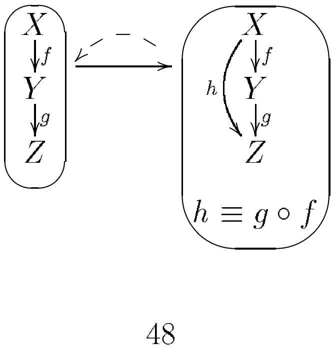
Here is an example of a non-filling morphism χ. Let $\mathcal{D}$ be a sketch of directed graphs and $\mathcal{E}$ a sketch of categories, as described in section 2.2, and let $\chi : \mathcal{D} \longrightarrow \mathcal{E}$ be the inclusion. Then the category $\mathcal{W}_\chi(\mathrm{Ar}) = \mathcal{Y}_\mathcal{E}(\mathrm{Ar})$ contains an arrow $f : X \longrightarrow Y$, both points X and Y and both identity arrows id_X and id_Y. Let T be the directed graph made of one point, so that $F_\chi(T)$ is the category made of one point and one identity arrow. Then $T(\mathrm{Ar})$ is empty, whereas $\mathrm{Hom}_{\mathcal{C}at}(\mathcal{W}_\chi(\mathrm{Ar}), F_\chi(T))$ has one element.

4.4 An expansion for dealing with exceptions

The propagators Q and R, the morphism of propagators $\chi : Q \longrightarrow R$, the Q-specification T and the R-specification U are as in sections 2.5 and 3.5.

The contravariant realization $\mathcal{W}_\chi = \mathcal{Y}_\mathcal{E} \circ \chi : \mathcal{D} \mathrel{-\!\!\!\times\!\!\!\twoheadrightarrow} \mathcal{R}eal(\mathcal{E})$ is described in section 2.5, on points and arrows. Let us now describe how the composition of operations in a Q-domain is described via $\mathcal{W}_\chi$. As explained in section 2.5, in order to deal with the composition of operations in the Q-domains, there are four arrows $j_{\dots} : \mathrm{Comp}_{\dots} \longrightarrow \mathrm{Cons}_{\dots}$ in $\mathcal{D}$ which become invertible in $\overline{\mathcal{D}}$.

First, let us come back to the illustration of the composition for P-domains, which corresponds to the inversion, in $\overline{\mathcal{C}}$, of the arrow $j : \mathrm{Comp} \longrightarrow \mathrm{Cons}$ of $\mathcal{C}$, as explained in section 2.5.



Now, here is the illustration of the four kinds of composition of arrows in Q-domains from section 2.5, when interpreted via χ. The second one is a composition "à la Kleisli".

$$\begin{tikzcd}
X \arrow[r, dotted, tail] \arrow[d, "f_*"] & X' & E \arrow[l, dotted, tail] \\
Y \arrow[r, dotted, tail] \arrow[d, "g_*"] & Y' & E \arrow[l, dotted, tail] \\
Z \arrow[r, dotted, tail] & Z' & E \arrow[l, dotted, tail]
\end{tikzcd}$$

$$\longrightarrow$$

$$\begin{tikzcd}
X \arrow[r, dotted, tail] \arrow[d, "f_*"] \arrow[dd, bend left, "h_*"] & X' & E \arrow[l, dotted, tail] \\
Y \arrow[r, dotted, tail] \arrow[d, "g_*"] & Y' & E \arrow[l, dotted, tail] \\
Z \arrow[r, dotted, tail] & Z' & E \arrow[l, dotted, tail]
\end{tikzcd}$$

$$h_* \equiv g_* \circ f_*$$

$$\begin{tikzcd}
X \arrow[r, dotted, tail] \arrow[dr, "f_?"] & X' & E \arrow[l, dotted, tail] \\
Y \arrow[r, dotted, tail] \arrow[dr, "g_?"'] & Y' & E \arrow[l, dotted, tail] \\
Z \arrow[r, dotted, tail] & Z' & E \arrow[l, dotted, tail]
\end{tikzcd}$$

$$\longrightarrow$$

$$\begin{tikzcd}
X \arrow[r, dotted, tail] \arrow[dr, "f_?"] \arrow[ddr, "h_?"'] & X' & E \arrow[l, dotted, tail] \\
Y \arrow[r, dotted, tail] \arrow[dr, "g_?"'] & Y' \arrow[d, "g_+"] & E \arrow[l, dotted, tail] \\
Z \arrow[r, dotted, tail] & Z' & E \arrow[l, dotted, tail]
\end{tikzcd}$$

$$h_? \equiv g_+ \circ f?$$
$$g_+ \circ i_Y \equiv g_?$$
$$g_+ \circ e_Y \equiv e_Z$$

$$\begin{tikzcd}
X \arrow[r, dotted, tail] \arrow[dr, "f_?"'] & X' & E \arrow[l, dotted, tail] \\
Y \arrow[r, dotted, tail] \arrow[drr, "g_!"] & Y' & E \arrow[l, dotted, tail] \\
Z \arrow[r, dotted, tail] & Z' & E \arrow[l, dotted, tail]
\end{tikzcd}$$

$$\longrightarrow$$

$$\begin{tikzcd}
X \arrow[r, dotted, tail] \arrow[dr, "f_?"'] \arrow[ddrr, bend left, "h_!"] & X' & E \arrow[l, dotted, tail] \\
Y \arrow[r, dotted, tail] \arrow[drr, "g_!"] & Y' \arrow[dr, "g'"] & E \arrow[l, dotted, tail] \\
Z \arrow[r, dotted, tail] & Z' & E \arrow[l, dotted, tail]
\end{tikzcd}$$

$$h_! \equiv g' \circ f?$$
$$g' \circ i_Y \equiv g_!$$
$$g' \circ e_Y \equiv \mathrm{id}_E$$

$$\begin{tikzcd}
X \arrow[r, dotted, tail] \arrow[drr, "f_!"] & X' & E \arrow[l, dotted, tail] \\
Y \arrow[r, dotted, tail] \arrow[dr, "g_?"] & Y' & E \arrow[l, dotted, tail] \\
Z \arrow[r, dotted, tail] & Z' & E \arrow[l, dotted, tail]
\end{tikzcd}$$

$$\longrightarrow$$

$$\begin{tikzcd}
X \arrow[r, dotted, tail] \arrow[drr, "f_!"] \arrow[ddrr, bend right, "h_!"] & X' & E \arrow[l, dotted, tail] \\
Y \arrow[r, dotted, tail] \arrow[dr, "g_?"] & Y' & E \arrow[l, dotted, tail] \\
Z \arrow[r, dotted, tail] & Z' & E \arrow[l, dotted, tail]
\end{tikzcd}$$

$$h_! \equiv f!$$

Since the morphism $\chi : \mathcal{D} \longrightarrow \mathcal{E}$ is filling, the expansion $F_\chi(T)$ can be computed pointwise. Actually, $F_\chi(T)$ is equivalent to U.

The fact that V remains a potential terminal point in U, as in S, comes from the way it has been decorated in section 3.5. It is easy to check that the constraint $V = 1_*$ is expanded as:

$$\begin{tikzcd}
V = 1 \arrow[r, tail, dotted, "i_V"] & V' & E \arrow[l, tail, dotted, "e_V"']
\end{tikzcd}$$

Similarly, it is easy to check that the binary product of X and Y in a Q-specification T, as defined in section 3.5, is expanded as:

$$\begin{tikzcd}
X \times Y \arrow[r, tail, dotted] & (X \times Y)' & E \arrow[l, tail, dotted]
\end{tikzcd}$$

So, the expansion of the properties of the product associates to each pair of arrows $(f_* : Z \longrightarrow X, g_? : Z \longrightarrow Y + E)$, an arrow $h_? : Z \longrightarrow (X \times Y) + E$. This means that it associates to each arrow from Z to $X \times (Y + E)$ an arrow from Z to $(X \times Y) + E$. When applied to the identity of $X \times (Y + E)$, this determines an arrow:

$$X \times (Y + E) \longrightarrow (X \times Y) + E \ .$$

The expansion of an exception handling arrow, as defined in section 3.5, is an arrow $h_+ : (X + E) \longrightarrow (X + E)$ such that h_+ is equivalent to the identity on X, and is any arrow on E:

$$\begin{tikzcd}
X \arrow[r, tail, dotted, "i_X"] \arrow[d, "\mathrm{id}_X"'] & X' \arrow[d, "h_+"] & E \arrow[l, tail, dotted, "e_X"'] \\
X \arrow[r, tail, dotted, "i_X"'] & X' & E \arrow[l, tail, dotted, "e_X"]
\end{tikzcd}
\qquad\qquad h_+ \circ i_X \equiv i_X$$

Remark. According to the definitions given in section 2.5, in a R-domain there are potential mono-sums for some pairs of points, but usually not for all of them. Let us look at the kind of potential mono-sums which can occur in $F_R(U)$, where U is the expansion $F_{\chi}(T)$ of a Q-specification T.

If X denotes a point in T, then from the description of $\mathcal{W}(\mathrm{Pt})$ in section 2.5 there is a potential mono-sum $X' = X + E$ in U. However, if Y denotes a point in T, there is no potential mono-sum $X + Y$ in U, and because of our definition of R-domains there is no potential mono-sum $X+Y$ in $F_R(U)$ either. Similarly, there is no potential mono-sum $E+E$ in U. So, some pairs of points have a potential mono-sum in $F_R(F_{\chi}(T))$, but not all of them.

5 Zooms

5.1 The framework for the zooming process

The zooming process is now easily defined from the decoration process of section 3 and the expansion process of section 4: a zoom is a decoration followed by an expansion. The framework for the zooms is made of a *span* Σ of propagators, i.e. three propagators P, Q, R and two morphisms $\delta : Q \longrightarrow P$ and $\chi : Q \longrightarrow R$, and a *lax-cocone* $\Lambda = (L, \mu, \nu, tr)$ over $\delta : Q \longrightarrow P$, as defined in section 3.4.

$$\begin{tikzcd}
 & L & & \\
 & & Q \arrow[ul, "\nu"', ""{name=N}] \arrow[dll, "\delta"] \arrow[dr, "\chi"] & \\
P \arrow[uur, bend left, "\mu", ""{name=M}] & & & R
\arrow[Rightarrow, from=N, to=M, "tr"']
\end{tikzcd}$$

In this section, we consider a L-specification Z and a R-domain $\overline{U}$.

Zooms and their models are defined in section 5.2, then some rigidity issues are considered in section 5.3. In section 5.4, the previous examples are merged in order to describe the zoom which corresponds to a treatment of exceptions. Finally, in section 5.5, our method is compared with the method of monads.

5.2 The zooms and their models

Definition 19 *The* zoom *with respect to Λ and Σ which is associated to the L-specification Z is made of:*

- *the P-specification $S = G_\mu(Z)$, called the* far specification *of the zoom,*
- *the Q-specification $T = G_\nu(Z)$, called the* intermediate specification *of the zoom,*
- *the R-specification $U = F_\chi(G_\nu(Z))$, called the* near specification *of the zoom,*
- *and the span in the category of Q-specifications:*

$$\begin{tikzcd}
 & T \arrow[dl, "\zeta"'] \arrow[dr, "\eta"] & \\
G_\delta(S) & & G_\chi(U)
\end{tikzcd}$$

where (T, ζ) is the decoration of S determined by Z, and $\eta = \eta_{\chi,T} : T \longrightarrow G_\chi(F_\chi(T))$ is the canonical morphism.

So, (T, ζ) is a *decoration* of S with respect to δ which satisfies Λ, in the sense of section 3.

And $U = F_\chi(T)$ is the *expansion* of T with respect to χ, in the sense of section 4.

The zoom which is associated to the specification Z will be also denoted Z. However, Z as a L-specification does not depend on Σ, whereas Z as a zoom depends on both Λ and Σ.

Definition 20 *The set of* models *of the zoom Z with values in $\overline{U}$ is:*

$$\boxed{\mathrm{Mod}_{\Lambda,\Sigma}(Z, \overline{U}) = \mathrm{Hom}_{\mathcal{D}om(Q)}(G_\nu(F_L(Z)), G_\chi(\overline{U}))\ .}$$

Let T denote the intermediate specification of Z. As observed in section 3.4, the natural transformation:

$$\Phi_\nu : F_Q \circ G_\nu \Rightarrow G_\nu \circ F_L : \mathcal{S}pec(L) \longrightarrow \mathcal{D}om(Q)$$

is a natural isomorphism if and only if ν is reliable. So, in general, the morphism:

$$\Phi_\nu(Z) : F_Q(T) \longrightarrow G_\nu(F_L(Z))$$

is not an isomorphism, and the map $\mathrm{Hom}_{\mathcal{D}om(Q)}(\Phi_\nu(Z), G_\chi(\overline{U}))$:

$$\mathrm{Hom}_{\mathcal{D}om(Q)}(G_\nu(F_L(Z)), G_\chi(\overline{U})) \longrightarrow \mathrm{Mod}_Q(T, G_\chi(\overline{U}))$$

is not a bijection, so that the sets

$$\mathrm{Mod}_{\Lambda,\Sigma}(Z, \overline{U}) \text{ and } \mathrm{Mod}_Q(T, G_\chi(\overline{U}))$$

are not in one-to-one correspondence.

Definition 21 *The morphism ν is* reliable with respect to *the morphism χ when the natural transformation:*

$$F_\chi \circ \Phi_\nu : F_\chi \circ F_Q \circ G_\nu \Rightarrow F_\chi \circ G_\nu \circ F_L : \mathcal{S}pec(L) \longrightarrow \mathcal{D}om(R)$$

is a natural isomorphism.

Clearly, if ν is reliable, then it is reliable with respect to χ.

Proposition 22 *Let Z be a zoom with respect to Λ and Σ, with intermediate specification T. If ν is reliable with respect to χ, then:*

$$\boxed{\mathrm{Mod}_{\Lambda,\Sigma}(Z, \overline{U}) \cong \mathrm{Mod}_Q(T, G_\chi(\overline{U})).}$$

Proof. From the reliability assumption :

$$\mathrm{Hom}_{\mathcal{D}om(R)}(F_\chi(F_Q(T)),\overline{U}) \cong \mathrm{Hom}_{\mathcal{D}om(R)}(F_\chi(G_\nu(F_L(Z))),\overline{U}) \ .$$

So, the result follows by adjunction with respect to χ. □

Then, various expressions for $\mathrm{Mod}_{\Lambda,\Sigma}(Z,\overline{U})$ can be derived from propositions 14 and 17.

Proposition 23 *Let us assume that ν is reliable with respect to χ. Let T be the intermediate specification and U the near specification of the zoom Z. Then the set of models of Z with values in $\overline{U}$ is such that:*

$$\boxed{\mathrm{Mod}_{\Lambda,\Sigma}(Z,\overline{U}) \cong \mathrm{Mod}_R(U,\overline{U}) \cong \mathrm{Hom}_{\mathcal{S}pec(Q)}(T,\mathrm{Mod}_R(\mathcal{W}_\chi(-),\overline{U})) \ .}$$

The expression $\mathrm{Mod}_R(U,\overline{U})$ means that the models of Z with values in $\overline{U}$ can be obtained from the near specification U in a straightforward way. However, U is quite complex and unstructured.

The expression $\mathrm{Hom}_{\mathcal{S}pec(Q)}(T,\mathrm{Mod}_R(\mathcal{W}_\chi(-),\overline{U}))$ means that the models of Z with values in $\overline{U}$ can be obtained from the intermediate specification T in a non-canonical way, via the morphism χ.

On the contrary, when ν is not reliable with respect to χ, the set $\mathrm{Mod}_{\Lambda,\Sigma}(Z,\overline{U})$ cannot be defined without knowing the L-domain $F_L(Z)$, whereas the set $\mathrm{Mod}_Q(T,G_\chi(\overline{U}))$ can always be defined directly from the Q-specification T, without knowing the Q-domain $F_Q(T)$. Then, the models of T are some kind of approximation for the models of Z. This situation is not considered any further in this paper.

Remark. Since a zoom is determined from a L-specification, it is endowed with an interesting genericity property: it is possible to change χ, thus the near specification U and the models of the zoom, without changing the L-specification. An example is given in the introduction, where two different near specifications U_1 and U_2 are computed from a unique intermediate specification T. The deductions which are made in $F_Q(T)$ remain valid in $F_R(U)$, via χ, for any choice of χ.

5.3 Rigidity

Let us assume that ν is reliable with respect to χ. It has been seen that the models of a zoom Z with values in $\overline{U}$ can be defined either directly from the near specification U or indirectly from the intermediate specification T. But usually, they cannot be recovered from the far specification S, which means that the semantics of S is irrelevant. However, as explained below, it can happen that some part of S gives rise to relevant interpretations.

The proposition below proves that the semantics of S can be adequate, under some powerful assumption.

Assumption.

(A) There is a morphism $\iota : P \longrightarrow R$ such that $\iota \circ \delta = \chi$.

Roughly speaking, the morphism $\iota : P \longrightarrow R$ means that the logic which corresponds to R is more powerful than the logic which corresponds to P. The equality $\iota \circ \delta = \chi$ means that the following triangle is commutative:

$$\begin{tikzcd} & Q \arrow[dl, "\delta"'] \arrow[dr, "\chi"] & \\ P \arrow[rr, "\iota"'] & & R \end{tikzcd}$$

In section 5.4, it will be seen that in our example there is "nearly" a morphism ι, but the assumption that $\iota \circ \delta = \chi$ is false.

Proposition 24 *Under the assumption (A), if* $S = F_\delta(T)$ *then:*

$$\boxed{\mathrm{Mod}_{\Lambda,\Sigma}(Z,\overline{U}) \cong \mathrm{Mod}_P(S, G_\iota(\overline{U}))\ .}$$

Proof. Since $\mathrm{Mod}_{\Lambda,\Sigma}(Z,\overline{U}) \cong \mathrm{Mod}_R(U,\overline{U})$, from $U = F_\chi(T) \cong F_\iota(F_\delta(T))$ we get:

$$\mathrm{Mod}_R(U,\overline{U}) \cong \mathrm{Mod}_R(F_\iota(F_\delta(T)),\overline{U})$$

so that, from proposition 1:

$$\mathrm{Mod}_R(U,\overline{U}) \cong \mathrm{Mod}_P(F_\delta(T), G_\iota(\overline{U}))\ .$$

The result follows from the assumption $S = F_\delta(T)$. $\Box$

Most often, the assumptions of this proposition are not satisfied, but it happens that they become true on a restricted part of the propagators and specifications, which is called *rigid*.

5.4 A zoom for dealing with exceptions

In order to deal with exceptions, one can use the span of propagators Σ from section 2.5, and the lax-cocone Λ_1 which is associated to the monomorphic lax-colimit, as in section 3.5. Then ν is reliable. So, the models of a zoom Z with values in *set* can be defined as the set-valued models of the far specification U, or as the models of the intermediate specification T with values in $G_\chi(set)$. It is easy to check that the Q-domain $G_\chi(set)$ is $set_{\mathbb{E}}$, as defined in section 2.5. Our running example corresponds to such a zoom. Its description is obtained by merging sections 3.5 and 4.4.

As an example of a proof which can be done in $F_L(Z)$, or in $F_Q(T)$, let us prove that the term $s \circ p \circ s$ of S is non-erroneous.

In $F_Q(T)$, every arrow with keyword "$*$", like s, is also an arrow with keyword "?", thanks to $\mathrm{ar}_{*,?}$. And every arrow which is composed from arrows with keyword "?" is an arrow with keyword "?". So, in $F_Q(T)$ there is an arrow $s \circ p \circ s$:?, as well as arrows $p \circ s$:? and $\mathrm{id}_N : *$, and there is an equation $(p \circ s :?) \equiv_e (\mathrm{id}_N : *)$, which leads to the equation:

$$(s \circ p \circ s :?) \equiv_e (s : *) \ .$$

In section 3.5, it has been formalized in the propagator Q, by the inversion of an arrow in $\overline{\mathcal{D}}$, that whenever $(f :?) \equiv_e (g : *)$ it can be deduced that $f : *$. So, it can be deduced, as required, that:

$$s \circ p \circ s : * \ .$$

Of course, this result could also be obtained by a classical deduction in first-order logic in $F_R(U)$. However, by reasoning at the intermediate level, first-order logic is replaced by some kind of *decorated equational logic*, somewhat similar to the one in (Hintermeier et al , 1998). Moreover, if the treatment of exceptions is modified by changing R and χ, the proof which is performed at the intermediate level remains valid.

If R is enriched in such a way that there are chosen products in the R-domains, then there is an inclusion morphism $\iota : P \longrightarrow R$, and there is a rigid part in the zooms: it is made of all the points and potential products of S, together with the arrows of S which get the keyword "$*$". Indeed, this part of S remains the same in U, because of the definition of χ.

More precisely, in order to restrict to the rigid part, P and R are unchanged, but Q_r is the part of Q which keeps only the keyword "$*$" for arrows and the keyword "e" for equations. The morphisms of propagators δ and χ are restricted to Q_r, so that they become isomorphisms.

Then T_r is the Q_r-specification underlying T, and S_r is the image of T_r by ζ, so that $S_r = F_\delta(T_r)$. Finally $U_r = F_\chi(T_r)$, so that the three of them are isomorphic.

In our example, the rigid part of the zoom is the following one, from the three points of view:

$$S_r : \begin{tikzcd} V = 1 \arrow[d, "z"'] \\ N \arrow[loop left, "s"'] \end{tikzcd} \qquad T_r : \begin{tikzcd} V = 1_* \arrow[d, "z{:}*"'] \\ N \arrow[loop left, "s{:}*"'] \end{tikzcd} \qquad U_r : \begin{tikzcd} V = 1 \arrow[d, "z_*"'] \\ N \arrow[loop left, "s_*"'] \end{tikzcd}$$

5.5 *About monads*

Let us compare our example to the treatment of exceptions with the help of the monad $A \mapsto A \uplus \mathbb{E}$ in the category of sets (Moggi , 1991). There are three parts in this comparison: first we look only at the graphical part of S, then we deal with its arity features, and finally with exception handling.

When dealing with the graphical part of S, in the zooming process for dealing with exceptions, let us use the unique keyword "?" for arrows and the unique keyword "e" for equations. This means that P and R remain the same, while Q is simplified. Then δ is an isomorphism between Q and P, so that the decoration step is trivial. The zooming process can be summarized as follows. An arrow:

$$f : X \longrightarrow Y \ \text{ in } S$$

is decorated as:

$$(f :?) : X \longrightarrow Y \ \text{ in } T$$

then it is expanded as:

$$f : X \longrightarrow Y + E \ \text{ in } U$$

which in turn is interpreted, in any set-valued model M of U, as a map:

$$M(f) : M(X) \longrightarrow M(Y) \uplus \mathbb{E} \ \text{ in } set\ .$$

This is indeed the kind of maps which are obtained when using the monad $A \mapsto A \uplus \mathbb{E}$. So, when dealing with the graphical part of the far specification, for such a treatment of exceptions, the zooming method can be considered as a generalization of the monad approach. Indeed, with the unique keyword

"?" in the zoom, both points of view are quite similar, and in addition in the zooming method it is possible to introduce other keywords, like "$*$" and "!".

Let us now consider the arities of S, typically the binary product constraints. As explained in section 3.5, the zooming method requires a second keyword "$*$" for arrows in order to define the notion of product which is needed for dealing with ordinary algebraic binary operations. The main point here is that this product is defined by means of its universal property, i.e. by the existence and unicity properties of the decorated factorization arrows. A consequence of this definition, as explained in section 4.4, is that the expansion step yields an arrow:

$$X \times (Y + E) \longrightarrow (X \times Y) + E \ ,$$

which in *set* becomes a map (natural in A and B):

$$A \times (B \uplus \mathbb{E}) \longrightarrow (A \times B) \uplus \mathbb{E} \ .$$

Actually, from the point of view of monads, according to (Moggi , 1991), this means that the monad $A \mapsto A \uplus \mathbb{E}$ in *set* is *strong.* Moreover, this can be expressed in terms of *monoidal categories*, as in (Plotkin and Power , 2001). So, in both points of view, the monad has to be strong. However, this is the starting point for the monads method, while for the zooming method this is a consequence of the universal property of a decorated product.

Finally, the framework for exception handling is quite simple in the zooming method, with a keyword "+" for arrows and a keyword "*nea*" for equations, as explained in section 3.5. Indeed, an arrow with keyword "+" in a T-specification is interpreted as a map which does not have to preserve the exceptions. With monads, according to (Plotkin and Power , 2001), exception handling looks somewhat mysterious.

A detailed comparison of the zooming process with other notions related to monads and effects, like (Plotkin and Power , 2001) or (Benton, Hughes and Moggi , 2002), remains to be done.

6 Conclusion

This paper presents a zooming-in process, which leads from a far specification S to a near specification U. An intermediate specification T is needed, which actually plays a major role in the zooming process. The zooming process is an example of the way propagators can cooperate for dealing easily and rigourously with several levels of logic.

Such a zooming process has many applications, among which the treatment of exceptions which is presented in this paper, the explicitization of the denotational semantics of some computational effects such as the state, and various issues related to overloading.

In many applications, the far and the near specifications are described with respect to classical logics, whereas the intermediate specification is described with respect to some kind of non-classical logic. Most deductions in S are irrelevant, and deductions in U may be rather complex, while deductions in T are both relevant and quite simple. In spite of the non-classical aspect of the corresponding logic, the deductions in T can easily be handled in the framework of diagrammatic specifications. Moreover, there is a lot of flexibility in the way the intermediate specifications can be defined, which allows to express easily and in a coherent way many kinds of properties.

Aknowledgments. We are grateful to the participants of the working group on *Sketches and Computer Algebra* in the *Categories and Structures* seminar, to P. Berlioux, and to many others.

References

M. Barr and C. Wells. *Category Theory for Computing Science*, Prentice Hall (1990).

N. Benton, J. Hughes and E. Moggi. Monads and Effects, *Lecture Notes in Computer Science* **2395**, 42–122 (2002).

L. Coppey. Théories algébriques et extensions de pré-faisceaux, *Cahiers de Topologie et Géométrie Différentielle* **13** (1972).

L. Coppey and C. Lair. Leçons de Théorie des esquisses (I), *Diagrammes* **12** (1984).

L. Coppey and C. Lair. Leçons de Théorie des esquisses (II), *Diagrammes* **19** (1988).

D. Duval. Diagrammatic specifications. To appear in *Mathematical Structures in Computer Science.* A preliminary version is available as: D. Duval. and C. Lair. Diagrammatic specifications. *Rapport de recherche IMAG-LMC* **1043** (2002). `http://www-lmc.imag.fr/lmc-cf/Dominique.Duval/`

C. Ehresmann. Introduction to the theory of structured categories, Report **10**, University of Kansas, Lawrence (1966).

J. Goguen and R. Burstall. Introducing institutions. *Lecture Notes in Computer Science* **164**, 221–256 (1984).

J. Goguen, J. Thatcher and E. Wagner. An initial algebra approach to the specification, correctness, and implementation of abstract data types. In R.

T. Yeh, editor, *Current Trends in Programming Methodology*, **IV**, 80–149, Prentice-Hall (1978).

C. Hintermeier, C. Kirchner and H. Kirchner. Dynamically-Typed Computations for Order-Sorted Equational Presentations, *Journal of Symbolic Computation* **25**, 455–526 (1998).

C. Lair. Trames et sémantiques catégoriques des systèmes de trames, *Diagrammes* **18**, Paris, CL1–CL47 (1987).

S. Mac Lane. *Categories for the working mathematician*, Springer-Verlag (1971).

E. Moggi. Notions of computation and monads, *Information and Computation* **93**, 55–92 (1991).

S. Peyton Jones, J. Hughes, et al. (1 February 1999) Report of the programming language Haskell 98; a non-strict, purely functional language (1999). `http://www.haskell.org/onlinereport/index.html`

G. Plotkin and J. Power. Algebraic Operations and Generic Effects. This is the journal version of "Semantics for Algebraic Operations", Electronic Notes in Theoretical Computer Science **45** Elsevier (2001).