HAL open science

Sequential products in effect categories

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

To cite this version:

Jean-Guillaume Dumas, Dominique Duval, Jean-Claude Reynaud. Sequential products in effect categories. [Research Report] Université Grenoble Alpes (UGA). 2007. hal-00161303

HAL Id: hal-00161303

https://hal.science/hal-00161303v1

Submitted on 10 Jul 2007

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

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

hal-00161303, version 1 - 10 Jul 2007

Sequential products in effect categories

Jean-Guillaume Dumas
LJK, University of Grenoble, France. Jean-Guillaume.Dumas@imag.fr

Dominique Duval
LJK, University of Grenoble, France. Dominique.Duval@imag.fr

Jean-Claude Reynaud
Malhiver, 38640 Claix, France. Jean-Claude.Reynaud@imag.fr

July 4., 2007

Abstract

A new categorical framework is provided for dealing with multiple arguments in a programming language with effects, for example in a language with imperative features. Like related frameworks (Monads, Arrows, Freyd categories), we distinguish two kinds of functions. In addition, we also distinguish two kinds of equations. Then, we are able to define a kind of product, that generalizes the usual categorical product. This yields a powerful tool for deriving many results about languages with effects.

1 Introduction

The aim of this paper is to provide a new categorical framework dealing with multiple arguments in a programming language with effects, for example in a language with imperative features. In our *cartesian effect categories*, as in other related frameworks (Monads, Arrows, Freyd categories), two kinds of functions are distinguished. The new feature here is that two kinds of equations are also distinguished. Then, we define a kind of product, that is mapped to the usual categorical product when the distinctions (between functions and between equations) are forgotten. In addition, we prove that cartesian effect categories determine Arrows.

A well-established framework for dealing with computational effects is the notion of *strong monads*, that is used in Haskell [8, 12]. Monads have been generalized on the categorical side to *Freyd categories* [10] and on the functional programming side to *Arrows* [7]. The claims that Arrows generalize Monads and that Arrows are Freyd categories are made precise in [6]. In all these frameworks, effect-free functions are distinguished among all functions,

generalizing the distinction of *values* among all *computations* in [8]. In this paper, as in [1, 6], effect-free functions are called *pure* functions; however, the symbols **C** and **V**, that are used for the category of all functions and for the subcategory of pure functions, respectively, are reminiscent of Moggi's terminology.

In all these frameworks, one major issue is about the order of evaluation of the arguments of multivariate operations. When there is no effect, the order does not matter, and the notion of product in a cartesian category provides a relevant framework. So, the category **V** is cartesian, and *products of pure funtions* are defined by the usual characteristic property of products. But, when effects do occur, the order of evaluation of the arguments becomes fundamental, which cannot be dealt with the categorical product. So, the category **C** is not cartesian, and products of functions do not make sense, in general. However, some kind of *sequential product of computations* should make sense, in order to evaluate the arguments in a given order. This is usually defined, by composition, from some kinds of products of a computation with an identity. This is performed by the *strength* of the monad [8], by the *symmetric premonoidal category* of the Freyd category [10], and by the *first* operator of Arrows [7].

In this paper, the framework of *cartesian effect categories* is introduced. We still distinguish two kinds of functions: pure functions among arbitrary functions, that form two categories **V** and **C**, with **V** a subcategory of **C**, and **V** cartesian. Let us say that the functions are *decorated*, either as pure or as arbitrary. The new feature that is introduced in this paper is that we also distinguish two kinds of equations: strong equations and *semi-equations*, respectively denoted $\equiv$ and $\lesssim$, so that equations also are *decorated*. Strong equations can be seen, essentially, as equalities between computations, while semi-equations are much weaker, and can be seen as a kind of approximation relation. Moreover, as suggested by the symbols $\equiv$ and $\lesssim$, the strong equations form an equivalence relation, while the semi-equations form a preorder relation. Then, we define the *semi-product* of two functions when at least one is pure, by a characteristic property that is a decorated version of the characteristic property of the usual product. Since all identities are values, we get the semi-product of any function with an identity, that is used for building sequential products of functions.

Cartesian effect categories give rise to Arrows, in the sense of [7], and they provide a deduction system: it is possible to decorate many proofs on cartesian categories in order to get proofs on cartesian effect categories.

As for terminology, our *graphs* are directed multi-graphs, made of *points* (or vertices, or objects) and *functions* (or edges, arrows, morphisms). We use *weak* categories rather than categories, i.e., we use a congruence $\equiv$ rather than the equality, however this "syntactic" choice is not fundamental here. As for notations, we often omit the subscripts in the diagrams and in the proofs.

Cartesian weak categories are reminded in section 2, then cartesian effect categories are defined in section 3; they are compared with Arrows in section 4, and examples are presented in section 5. In appendix A are given the proofs of some properties of cartesian weak categories, that are well-known, followed

by their decorated versions, that yield proofs of properties of cartesian effect categories.

2 Cartesian weak categories

Weak categories are reminded in this section, with their notion of product. Except for the minor fact that equality is weakened as a congruence, all this section is very well known. Some detailed proofs are given in appendix A, with their decorated versions.

2.1 Weak categories

A weak category is like a category, except that the equations (for unitarity and associativity) hold only "up to congruence".

Definition 2.1. A *weak category* is a graph where:

- for each point X there is a loop $\mathrm{id}_X : X \to X$ called the *identity* of X,
- for each consecutive functions $f : X \to Y$, $g : Y \to Z$, there is a function $g \circ f : X \to Z$ called the *composition* of f and g,
- and there is a relation $\equiv$ between parallel functions (each $f_1 \equiv f_2$ is called an *equation*), such that:
  - $\equiv$ is a *congruence*, i.e., it is an equivalence relation and for each $f : X \to Y$, $g_1, g_2 : Y \to Z$, $h : Z \to W$, if $g_1 \equiv g_2$ then $g_1 \circ f \equiv g_2 \circ f$ (*substitution*) and $h \circ g_1 \equiv h \circ g_2$ (*replacement*),
  - for each $f : X \to Y$, the *unitarity equations* hold: $f \circ \mathrm{id}_X \equiv f$ and $\mathrm{id}_Y \circ f \equiv f$,
  - and for each $f : X \to Y$, $g : Y \to Z$, $h : Z \to W$, the *associativity equation* holds: $h \circ (g \circ f) \equiv (h \circ g) \circ f$.

So, a weak category is a special kind of a bicategory, and a category is a weak category where the congruence is the equality.

2.2 Products

In a weak category, a *weak product*, or simply a *product*, is defined as a product "up to congruence". We focus on nullary products (i.e., terminal points) and binary products; it is well-know that products of any arity can be recovered from those.

Definition 2.2. A *(weak) terminal point* is a point U (for "*Unit*") such that for every point X there is a function $\langle\,\rangle_X : X \to U$, unique up to congruence.

Definition 2.3. A *binary cone* is made of two functions with the same source $Y_1 \xleftarrow{f_1} X \xrightarrow{f_2} Y_2$. A *binary (weak) product* is a binary cone $Y_1 \xleftarrow{q_1} Y_1 \times Y_2 \xrightarrow{q_2} Y_2$ such that for every binary cone with the same base $Y_1 \xleftarrow{f_1} X \xrightarrow{f_2} Y_2$ there is a function $\langle f_1, f_2 \rangle : X \to Y_1 \times Y_2$, called the *pair* of f_1 and f_2, unique up to congruence, such that:

$$q_1 \circ \langle f_1, f_2 \rangle \equiv f_1 \ \text{ and } \ q_2 \circ \langle f_1, f_2 \rangle \equiv f_2 \ .$$

As usual, all terminal points are isomorphic, and the fact of using U for denoting a terminal point corresponds to the choice of one terminal point. Similarly, all products on a given base are isomorphic (in a suitable sense), and the notations correspond to the choice of one product for each base.

Definition 2.4. A *cartesian weak category* is a weak category with a chosen terminal point and chosen binary products.

2.3 Products of functions

Definition 2.5. In a cartesian weak category, the *(weak) binary product* of two functions $f_1 : X_1 \to Y_1$ and $f_2 : X_2 \to Y_2$ is the function:

$$f_1 \times f_2 = \langle f_1 \circ p_1, f_2 \circ p_2 \rangle : X_1 \times X_2 \to Y_1 \times Y_2 \ .$$

So, the binary product of functions is characterized, up to congruence, by the equations:

$$q_1 \circ (f_1 \times f_2) \equiv f_1 \circ p_1 \ \text{ and } \ q_2 \circ (f_1 \times f_2) \equiv f_2 \circ p_2 \ .$$

The defining equations of a pair and a product can be illustrated as follows:

$$\begin{tikzcd}
 & Y_1 \\
X \arrow[ur, "f_1"] \arrow[r, "\langle f_1, f_2 \rangle"] \arrow[dr, "f_2"'] \arrow[ur, phantom, shift right=3ex, "\equiv"] \arrow[dr, phantom, shift right=3ex, "\equiv"] & Y_1 \times Y_2 \arrow[u, "q_1"'] \arrow[d, "q_2"] \\
 & Y_2
\end{tikzcd}
\qquad
\begin{tikzcd}
X_1 \arrow[r, "f_1"] \arrow[dr, phantom, "\equiv"] & Y_1 \\
X_1 \times X_2 \arrow[u, "p_1"] \arrow[r, "f_1 \times f_2"] \arrow[d, "p_2"'] \arrow[dr, phantom, "\equiv"] & Y_1 \times Y_2 \arrow[u, "q_1"'] \arrow[d, "q_2"] \\
X_2 \arrow[r, "f_2"] & Y_2
\end{tikzcd}$$

So, the products are defined from the pairs (note that we use the same symbols f_1, f_2 for the general case $f_i : X_i \to Y_i$ and for the special case $f_i : X \to Y_i$). The other way round, the pairs can be recovered from the products and the *diagonals*, i.e., the pairs $\langle \mathrm{id}, \mathrm{id} \rangle$; indeed, it is easy to prove that for each cone $X_1 \xleftarrow{f_1} X \xrightarrow{f_2} X_2$

$$\langle f_1, f_2 \rangle \equiv (f_1 \times f_2) \circ \langle \mathrm{id}_X, \mathrm{id}_X \rangle \ .$$

In the following, we consider products $X_1 \xleftarrow{p_1} X_1 \times X_2 \xrightarrow{p_2} X_2$, $Y_1 \xleftarrow{q_1} Y_1 \times Y_2 \xrightarrow{q_2} Y_2$ and $Z_1 \xleftarrow{r_1} Z_1 \times Z_2 \xrightarrow{r_2} Z_2$.

Proposition 2.6 (congruence). *For each* $f_1 \equiv f_1' : X_1 \to Y_1$ *and* $f_2 \equiv f_2' : X_2 \to Y_2$

1. *if* $X_1 = X_2$
$$\langle f_1, f_2 \rangle \equiv \langle f_1', f_2' \rangle \ ,$$

2. *in all cases*
$$f_1 \times f_2 \equiv f_1' \times f_2' \ .$$

Proposition 2.7 (composition). *For each* $f_1 : X_1 \to Y_1$, $f_2 : X_2 \to Y_2$, $g_1 : Y_1 \to Z_1$, $g_2 : Y_2 \to Z_2$

1. *if* $X_1 = X_2$ *and* $Y_1 = Y_2$ *and* $f_1 = f_2 (= f)$
$$\langle g_1, g_2 \rangle \circ f \equiv \langle g_1 \circ f, g_2 \circ f \rangle \ ,$$

2. *if* $X_1 = X_2$
$$(g_1 \times g_2) \circ \langle f_1, f_2 \rangle \equiv \langle g_1 \circ f_1, g_2 \circ f_2 \rangle \ ,$$

3. *in all cases*
$$(g_1 \times g_2) \circ (f_1 \times f_2) \equiv (g_1 \circ f_1) \times (g_2 \circ f_2) \ .$$

Let us consider the products $X_1 \xleftarrow{p_1} X_1 \times X_2 \xrightarrow{p_2} X_2$ and $X_2 \xleftarrow{p_2'} X_2 \times X_1 \xrightarrow{p_1'} X_1$. The *swap* function is the isomorphism:

$$\gamma_{(X_1,X_2)} = \langle p_1', p_2' \rangle_{p_1,p_2} = \langle p_1', p_2' \rangle : X_2 \times X_1 \to X_1 \times X_2 \ ,$$

characterized by:

$$p_1 \circ \gamma_{(X_1,X_2)} \equiv p_1' \ \text{ and } \ p_2 \circ \gamma_{(X_1,X_2)} \equiv p_2' \ .$$

Proposition 2.8 (swap). *For each* $f_1 : X_1 \to Y_1$ *and* $f_2 : X_2 \to Y_2$, *let* $\gamma_Y = \gamma_{(Y_1,Y_2)}$ *and* $\gamma_X = \gamma_{(X_1,X_2)}$, *then:*

1. *if* $X_1 = X_2$
$$\gamma_Y \circ \langle f_2, f_1 \rangle \equiv \langle f_1, f_2 \rangle \ ,$$

2. *in all cases*
$$\gamma_Y \circ (f_2 \times f_1) \circ \gamma_X^{-1} \equiv f_1 \times f_2 \ .$$

Let us consider the products $X_1 \xleftarrow{p_1} X_1 \times X_2 \xrightarrow{p_2} X_2$, $X_1 \times X_2 \xleftarrow{p_{1,2}} (X_1 \times X_2) \times X_3 \xrightarrow{p_3} X_3$, $X_2 \xleftarrow{p_2'} X_2 \times X_3 \xrightarrow{p_3'} X_3$ and $X_1 \xleftarrow{p_1'} X_1 \times (X_2 \times X_3) \xrightarrow{p_{2,3}'} X_2 \times X_3$. The *associativity* function is the isomorphism:

$$\alpha_{(X_1,X_2,X_3)} = \langle \langle p_1', p_2' \circ p_{2,3}' \rangle_{p_1,p_2}, p_3' \circ p_{2,3}' \rangle_{p_{1,2},p_3} : X_1 \times (X_2 \times X_3) \to (X_1 \times X_2) \times X_3 \ ,$$

characterized by:

$$p_1 \circ p_{1,2} \circ \alpha_{(X_1,X_2,X_3)} \equiv p_1' \ , \ p_2 \circ p_{1,2} \circ \alpha_{(X_1,X_2,X_3)} \equiv p_2' \circ p_{2,3}' \ \text{ and } \ p_3 \circ \alpha_{(X_1,X_2,X_3)} \equiv p_3' \circ p_{2,3}' \ .$$

Proposition 2.9 (associativity). *For each $f_1 : X_1 \to Y_1$, $f_2 : X_2 \to Y_2$ and $f_3 : X_3 \to Y_3$, let $\alpha_Y = \alpha_{(Y_1,Y_2,Y_3)}$ and $\alpha_X = \alpha_{(X_1,X_2,X_3)}$, then:*

1. *if $X_1 = X_2 = X_3$*
$$\alpha_Y \circ \langle f_1, \langle f_2, f_3\rangle\rangle \equiv \langle\langle f_1, f_2\rangle, f_3\rangle \ ,$$

2. *in all cases*
$$\alpha_Y \circ (f_1 \times (f_2 \times f_3)) \equiv ((f_1 \times f_2) \times f_3) \circ \alpha_X \ .$$

In the definition of the binary product $f_1 \times f_2$, both f_1 and f_2 play symmetric rôles. This symmetry can be broken: "first f_1 then f_2" corresponds to $(\mathrm{id}_{Y_1} \times f_2) \circ (f_1 \times \mathrm{id}_{X_2})$, using the intermediate product $Y_1 \times X_2$, while "first f_2 then f_1" corresponds to $(f_1 \times \mathrm{id}_{Y_2}) \circ (\mathrm{id}_{X_1} \times f_2)$, using the intermediate product $X_1 \times Y_2$. These are called the *(left and right) sequential products* of f_1 and f_2. The three versions of the binary product of functions coincide, up to congruence; this is a kind of *parallelism* property, meaning that both f_1 and f_2 can be computed either simultaneously, or one after the other, in any order:

Proposition 2.10 (parallelism). *For each $f_1 : X_1 \to Y_1$ and $f_2 : X_2 \to Y_2$*
$$f_1 \times f_2 \equiv (\mathrm{id}_{Y_1} \times f_2) \circ (f_1 \times \mathrm{id}_{X_2}) \equiv (f_1 \times \mathrm{id}_{X_2}) \circ (\mathrm{id}_{Y_1} \times f_2) \ .$$

3 Cartesian effect categories

Sections 3.1 to 3.3 form a decorated version of section 2. Roughly speaking, a kind of structure is *decorated* when there is some classification of its ingredients. Here, the classification involves two kinds of functions and two kinds of equations. Effect categories are defined in section 3.1 as decorated weak categories. In section 3.2, semi-products are defined as decorated weak products, then cartesian effect category as decorated cartesian weak categories. Decorated propositions are stated here, and the corresponding decorated proofs are given in appendix A. Then, in sections 3.4 and 3.5, the sequential product of functions is defined by composing semi-products, and some of its properties are derived.

3.1 Effect categories

A *(weak) subcategory* **V** of a weak category **C** is a subcategory of **C** such that each equation of **V** is an equation of **C**. It is a *wide (weak) subcategory* when **V** and **C** have the same points, and each equation of **C** between functions in **V** is an equation in **V**. Then only one symbol $\equiv$ can be used, for both **V** and **C**.

Definition 3.1. Let **V** be a weak category. An *effect category extending* **V** is a weak category **C**, such that **V** is a wide subcategory of **C**, together with a relation $\lesssim$ between parallel functions in **C** such that:

- the relation $\lesssim$ is weaker than $\equiv$ for f_1, f_2 in **C**, $f_1 \equiv f_2 \Rightarrow f_1 \lesssim f_2$;
- $\lesssim$ is transitive;

- $\lesssim$ and $\equiv$ coincide on $\mathbf{V}$ for v_1, v_2 in $\mathbf{V}$, $v_1 \equiv v_2 \iff v_1 \lesssim v_2$;
- $\lesssim$ satisfies the substitution property:
  if $f : X \to Y$ and $g_1 \lesssim g_2 : Y \to Z$ then $g_1 \circ f \lesssim g_2 \circ f : X \to Z$;
- $\lesssim$ satisfies the replacement property *with respect to* $\mathbf{V}$:
  if $g_1 \lesssim g_2 : Y \to Z$ and $v : Z \to W$ in $\mathbf{V}$ then $v \circ g_1 \lesssim v \circ g_2 : Y \to W$.

The first property implies that $\lesssim$ is reflexive, and when $\equiv$ is the equality it means precisely that $\lesssim$ is reflexive. Since $\lesssim$ is transitive and weaker than $\equiv$, if either $f_1 \equiv f_2 \lesssim f_3$ or $f_1 \lesssim f_2 \equiv f_3$, then $f_1 \lesssim f_3$; this is called the *compatibility* of $\lesssim$ with $\equiv$. An effect category is *strict* when $\equiv$ is the equality. In this paper, there is no major difference between effect categories and strict effect categories.

A *pure function* is a function in $\mathbf{V}$. The symbol $\rightsquigarrow$ is used for pure functions, and $\to$ for all functions. It follows from definition 3.1 that all the identities of $\mathbf{C}$ are pure, the composition of pure functions is pure, and more precisely a composition of functions is pure if and only if all the composing functions are pure. It should be noted that there can be equations $f \equiv v$ between a non-pure function and a pure one; then the function f is proved effect-free, without being pure. This "syntactic" choice could be argued; note that this situation disappears when the congruence $\equiv$ is the equality. The relation $\lesssim$ is called the *semi-congruence* of the effect category, and each $f_1 \lesssim f_2$ is called a *semi-equation*. The semi-congruence generally is not a congruence, for two reasons: it may not be symmetric, and it may not satisfy the replacement property for all functions.

Examples of strict effect categories are given in section 5. For dealing with partiality in section 5.1, the semi-congruence $\lesssim$ coincides with the usual ordering of partial functions, it is not symmetric but it satisfies the replacement property for all partial functions. On the other hand, in section 5.2, the semi-congruence $\lesssim$ means that two functions in an imperative language have the same result but may act differently on the state, it is an equivalence relation that does not satisfy the replacement property for non-pure functions.

Clearly, if the decorations are forgotten, i.e., if both the distinction between pure functions and arbitrary functions and the distinction between the congruence and the semi-congruence are forgotten, then an effect category is just a weak category.

A cartesian effect category, as defined below, is an effect category where $\mathbf{V}$ is cartesian and where this cartesian structure on $\mathbf{V}$ has some kind of generalization to $\mathbf{C}$, that does *not*, in general, turn $\mathbf{C}$ into a cartesian weak category.

3.2 Semi-products

Now, let us assume that $\mathbf{C}$ is an effect category extending $\mathbf{V}$, and that $\mathbf{V}$ is cartesian. We define nullary and binary *semi-products* in $\mathbf{C}$, for building pairs of functions when at least one of them is pure.

Definition 3.2. A *semi-terminal point* in $\mathbf{C}$ is a terminal point U in $\mathbf{V}$ such that every function $g : X \to U$ satisfies $g \lesssim \langle\,\rangle_X$.

Definition 3.3. A *binary semi-product* in **C** is a binary product $Y_1 \overset{q_1}{\leftsquigarrow} Y_1 \times Y_2 \overset{q_2}{\rightsquigarrow} Y_2$ in **V** such that:

- for every binary cone with the same base $Y_1 \overset{f_1}{\longleftarrow} X \overset{v_2}{\rightsquigarrow} Y_2$ and with v_2 pure, there is a function $\langle f_1, v_2 \rangle_{q_1,q_2} = \langle f_1, v_2 \rangle : X \to Y_1 \times Y_2$, unique up to $\equiv$, such that
$$q_1 \circ \langle f_1, v_2 \rangle \equiv f_1 \ \text{ and } \ q_2 \circ \langle f_1, v_2 \rangle \lesssim v_2 \ ,$$

- and for every binary cone with the same base $Y_1 \overset{v_1}{\leftsquigarrow} X \xrightarrow{f_2} Y_2$ and with v_1 pure, there is a function $\langle v_1, f_2 \rangle_{q_1,q_2} = \langle v_1, f_2 \rangle : X \to Y_1 \times Y_2$, unique up to $\equiv$, such that
$$q_1 \circ \langle v_1, f_2 \rangle \lesssim v_1 \ \text{ and } \ q_2 \circ \langle v_1, f_2 \rangle \equiv f_2 \ .$$

The defining (semi-)equations of a binary semi-product can be illustrated as follows:

$$\begin{tikzcd}
 & Y_1 \\
X \arrow[ur, squiggly, "v_1"] \arrow[r, squiggly, "\langle v_1,v_2 \rangle"] \arrow[dr, squiggly, "v_2"'] \arrow[r, phantom, shift right=4ex, "\equiv"] \arrow[r, phantom, shift left=4ex, "\equiv"] & Y_1 \times Y_2 \arrow[u, squiggly, "q_1"'] \arrow[d, squiggly, "q_2"] \\
 & Y_2
\end{tikzcd}
\qquad
\begin{tikzcd}
 & Y_1 \\
X \arrow[ur, "f_1"] \arrow[r, "\langle f_1,v_2 \rangle"] \arrow[dr, squiggly, "v_2"'] \arrow[r, phantom, shift right=4ex, "\equiv"] \arrow[r, phantom, shift left=4ex, "\gtrsim"] & Y_1 \times Y_2 \arrow[u, squiggly, "q_1"'] \arrow[d, squiggly, "q_2"] \\
 & Y_2
\end{tikzcd}
\qquad
\begin{tikzcd}
 & Y_1 \\
X \arrow[ur, squiggly, "v_1"] \arrow[r, "\langle v_1,f_2 \rangle"] \arrow[dr, "f_2"'] \arrow[r, phantom, shift right=4ex, "\gtrsim"] \arrow[r, phantom, shift left=4ex, "\equiv"] & Y_1 \times Y_2 \arrow[u, squiggly, "q_1"'] \arrow[d, squiggly, "q_2"] \\
 & Y_2
\end{tikzcd}$$

Clearly, if the decorations are forgotten, then semi-products are just products.

The notation is not ambiguous. Indeed, if $Y_1 \overset{v_1}{\leftsquigarrow} X \overset{v_2}{\rightsquigarrow} Y_2$ is a binary cone in **V**, then the three definitions of the pair $\langle v_1, v_2 \rangle$ above coincide, up to congruence: let t denote any one of the three pairs, then t is characterized, up to congruence, by $q_1 \circ t \equiv v_1$ and $q_2 \circ t \equiv v_2$, because $\equiv$ and $\lesssim$ coincide on pure functions.

Definition 3.4. A *cartesian effect category extending* a cartesian weak category **V** is an effect category extending **V** such that each terminal point of **V** is a semi-terminal point of **C** and each binary product of **V** is a binary semi-product of **C**.

3.3 Semi-products of functions

Definition 3.5. In a cartesian effect category, the *binary semi-product* $f_1 \times v_2$ of a function $f_1 : X_1 \to Y_1$ and a pure function $v_2 : X_2 \rightsquigarrow Y_2$ is the function:

$$f_1 \times v_2 = \langle f_1 \circ p_1, v_2 \circ p_2 \rangle : X_1 \times X_2 \to Y_1 \times Y_2$$

It follows that $f_1 \times v_2$ is characterized, up to $\equiv$, by:

$$q_1 \circ (f_1 \times v_2) \equiv f_1 \circ p_1 \ \text{ and } \ q_2 \circ (f_1 \times v_2) \lesssim v_2 \circ p_2$$

$$
\begin{tikzcd}
X_1 \arrow[r, "f_1"] & Y_1 \\
X_1 \times X_2 \arrow[u, squiggly, "p_1"] \arrow[r, "f_1 \times v_2"] \arrow[ur, phantom, "\equiv"] \arrow[d, squiggly, "p_2"'] \arrow[dr, phantom, "\gtrsim"] & Y_1 \times Y_2 \arrow[u, squiggly, "q_1"'] \arrow[d, squiggly, "q_2"] \\
X_2 \arrow[r, squiggly, "v_2"] & Y_2
\end{tikzcd}
$$

The *binary semi-product* $v_1 \times f_2 : X_1 \times X_2 \to Y_1 \times Y_2$ of a pure function $v_1 : X_1 \rightsquigarrow Y_1$ and a function $f_2 : X_2 \to Y_2$ is defined in the symmetric way, and it is characterized, up to $\equiv$, by the symmetric property.

The notation is not ambiguous, because so is the notation for pairs; if v_1 and v_2 are pure functions, then the three definitions of $v_1 \times v_2$ coincide, up to congruence.

Propositions about products in cartesian weak categories are called *basic* propositions. It happens that each basic proposition in section 2 has a *decorated* version, about semi-products of the form $f_1 \times v_2$ in cartesian effect categories, that is stated below. The symmetric decorated version also holds, for semi-products of the form $v_1 \times f_2$. Each function in the basic proposition is replaced either by a function or by a pure function, and each equation is replaced either by an equation ($\equiv$) or by a semi-equation ($\lesssim$ or $\gtrsim$).

In addition, in appendix A, the proofs of the decorated propositions are *decorated* versions of the *basic* proofs. It happens that no semi-equation appears in the decorated propositions below, but they are used in the proofs. Indeed, a major ingredient in the basic proofs is that a function $\langle f_1, f_2 \rangle$ or $f_1 \times f_2$ is characterized, up to $\equiv$, by its projections, both up to $\equiv$. The decorated version of this property is that a function $\langle f_1, f_2 \rangle$ or $f_1 \times f_2$, where f_1 or f_2 is pure, is characterized, up to $\equiv$, by its projections, one up to $\equiv$ *and the other one up to* $\lesssim$. It should be noted that even when some decorated version of a basic proposition is valid, usually not all the basic proofs can be decorated. In addition, when equations are decorated as semi-equations, some care is required when the symmetry and replacement properties are used.

Proposition 3.6 (congruence). *For each congruent functions $f_1 \equiv f_1' : X \to Y_1$ and pure functions $v_2 \equiv v_2' : X \rightsquigarrow Y_2$*

1. *if $X_1 = X_2$*
$$\langle f_1, v_2 \rangle \equiv \langle f_1', v_2' \rangle \ .$$
2. *in all cases*
$$f_1 \times v_2 \equiv f_1' \times v_2' \ .$$

Proposition 3.7 (composition). *For each functions $f_1 : X_1 \to Y_1$, $g_1 : Y_1 \to Z_1$ and pure functions $v_2 : X_2 \rightsquigarrow Y_2$, $w_2 : Y_2 \rightsquigarrow Z_2$*

1. *if $X_1 = X_2$ and $Y_1 = Y_2$ and $f_1 = v_2 (= v)$*
$$\langle g_1, w_2 \rangle \circ f \equiv \langle g_1 \circ v, w_2 \circ v \rangle \ ,$$

2. *if* $X_1 = X_2$
$$(g_1 \times w_2) \circ \langle f_1, v_2 \rangle \equiv \langle g_1 \circ f_1, w_2 \circ v_2 \rangle \ ,$$
3. *in all cases*
$$(g_1 \times w_2) \circ (f_1 \times v_2) \equiv (g_1 \circ f_1) \times (w_2 \circ v_2) \ .$$

The swap and associativity functions are defined in the same way as in section 2; they are products of projections, so that they are pure functions. It follows that the swap and associativity functions are characterized by the same equations as in section 2, and that they are still isomorphisms.

Proposition 3.8 (swap). *For each function* $f_1 : X \to Y_1$ *and pure function* $v_2 : X \rightsquigarrow Y_2$*, let* $\gamma_Y = \gamma_{(Y_1,Y_2)}$ *and* $\gamma_X = \gamma_{(X_1,X_2)}$*, then:*

1. *if* $X_1 = X_2$
$$\gamma_Y \circ \langle v_2, f_1 \rangle \equiv \langle f_1, v_2 \rangle \ ,$$
2. *in all cases*
$$\gamma_Y \circ (v_2 \times f_1) \circ \gamma_X^{-1} \equiv f_1 \times v_2 \ .$$

Proposition 3.9 (associativity). *For each function* $f_1 : X_1 \to Y_1$ *and pure functions* $v_2 : X_2 \rightsquigarrow Y_2$, $v_3 : X_3 \rightsquigarrow Y_3$*, let* $\alpha_Y = \alpha_{(Y_1,Y_2,Y_3)}$ *and* $\alpha_X = \alpha_{(X_1,X_2,X_3)}$*, then:*

1. *if* $X_1 = X_2 = X_3$
$$\alpha_Y \circ \langle f_1, \langle v_2, v_3 \rangle \rangle \equiv \langle \langle f_1, v_2 \rangle, v_3 \rangle \ ,$$
2. *in all cases:*
$$\alpha_Y \circ (f_1 \times (v_2 \times v_3)) \equiv ((f_1 \times v_2) \times v_3) \circ \alpha_X \ .$$

The sequential product of a function $f_1 : X_1 \to Y_1$ and a pure function $v_2 : X_2 \rightsquigarrow Y_2$ can be defined as in section 2, using the intermediate products $Y_1 \overset{s_1}{\leftsquigarrow} Y_1 \times X_2 \overset{s_2}{\rightsquigarrow} X_2$ and $X_1 \overset{t_1}{\leftsquigarrow} X_1 \times Y_2 \overset{t_2}{\rightsquigarrow} Y_2$. It does coincide with the semi-product of f_1 and v_2, up to congruence:

Proposition 3.10 (parallelism). *For each function* $f_1 : X_1 \to Y_1$ *and pure function* $v_2 : X_2 \rightsquigarrow Y_2$
$$f_1 \times v_2 \equiv (\mathrm{id}_{Y_1} \times v_2) \circ (f_1 \times \mathrm{id}_{X_2}) \equiv (f_1 \times \mathrm{id}_{X_2}) \circ (\mathrm{id}_{Y_1} \times v_2) \ .$$

3.4 Sequential products of functions

It has been stated in proposition 2.10 that, in a cartesian weak category, the binary product of functions coincide with both sequential products, up to congruence:
$$f_1 \times f_2 \equiv (\mathrm{id}_{Y_1} \times f_2) \circ (f_1 \times \mathrm{id}_{X_2}) \equiv (f_1 \times \mathrm{id}_{X_2}) \circ (\mathrm{id}_{Y_1} \times f_2) \ .$$
In a cartesian effect category, when f_1 and f_2 are any functions, the product $f_1 \times f_2$ is not defined. But $(\mathrm{id}_{Y_1} \times f_2) \circ (f_1 \times \mathrm{id}_{X_2})$ and $(f_1 \times \mathrm{id}_{X_2}) \circ (\mathrm{id}_{Y_1} \times f_2)$ make sense, thanks to semi-products, because identities are pure. They are called the sequential products of f_1 and f_2, and they do not coincide up to congruence, in general: parallelism is not satisfied.

Definition 3.11. The *left binary sequential product* of two functions $f_1 : X_1 \to Y_1$ and $f_2 : X_2 \to Y_2$ is the function:

$$f_1 \ltimes f_2 = (\mathrm{id}_{Y_1} \times f_2) \circ (f_1 \times \mathrm{id}_{X_2}) : X_1 \times X_2 \to Y_1 \times Y_2 \ .$$

So, the left binary sequential product is obtained from:

$$\begin{tikzcd}
X_1 \arrow[r, "f_1"] \arrow[d, squiggly, "p_1"'] \arrow[dr, phantom, "\equiv"] & Y_1 \arrow[r, squiggly, "\mathrm{id}"] \arrow[d, squiggly, "s_1"'] \arrow[dr, phantom, "\gtrsim"] & Y_1 \arrow[d, squiggly, "q_1"'] \\
X_1 \times X_2 \arrow[r, "f_1 \times \mathrm{id}"] \arrow[d, squiggly, "p_2"'] \arrow[dr, phantom, "\gtrsim"] & Y_1 \times X_2 \arrow[r, "\mathrm{id} \times f_2"] \arrow[d, squiggly, "s_2"'] \arrow[dr, phantom, "\equiv"] & Y_1 \times Y_2 \arrow[d, squiggly, "q_2"'] \\
X_2 \arrow[r, squiggly, "\mathrm{id}"] & X_2 \arrow[r, "f_2"] & Y_2
\end{tikzcd}$$

The left sequential product extends the semi-product:

Proposition 3.12. *For each function f_1 and pure function v_2, $f_1 \ltimes v_2 \equiv f_1 \times v_2$.*

Proof. ¿From proposition 3.7, $f_1 \ltimes v_2 = (\mathrm{id} \times v_2) \circ (f_1 \times \mathrm{id}) \equiv (\mathrm{id} \circ f_1) \times (v_2 \circ \mathrm{id}) \equiv f_1 \times v_2$. □

Note that the diagonal $\langle \mathrm{id}_X, \mathrm{id}_X \rangle$ is a pair of pure functions. So, by analogy with the property $\langle f_1, f_2 \rangle \equiv (f_1 \times f_2) \circ \langle \mathrm{id}_X, \mathrm{id}_X \rangle$ in weak categories:

Definition 3.13. The *left sequential pair* of two functions $f_1 : X \to Y_1$ and $f_2 : X \to Y_2$ is:

$$\langle f_1, f_2 \rangle_l = (f_1 \ltimes f_2) \circ \langle \mathrm{id}_X, \mathrm{id}_X \rangle \ .$$

The left sequential pairs do not satisfy the usual equations for pairs, as in definition 2.3. However, they satisfy some weaker properties, as stated in corollary 3.22.

The *right binary sequential product* of f_1 and f_2 is defined in the symmetric way; it is the function:

$$f_1 \rtimes f_2 = (f_1 \times \mathrm{id}_{Y_2}) \circ (\mathrm{id}_{X_1} \times f_2) : X_1 \times X_2 \to Y_1 \times Y_2 \ .$$

It does also extend the product of a pure function and a function: for each pure function v_1, $v_1 \rtimes f_2 \equiv v_1 \times f_2$. The *right sequential pair* of $f_1 : X \to Y_1$ and $f_2 : X \to Y_2$ is:

$$\langle f_1, f_2 \rangle_r = (f_1 \rtimes f_2) \circ \langle \mathrm{id}_X, \mathrm{id}_X \rangle \ .$$

Here are some properties of the sequential products that are easily deduced from the properties of semi-products in 3.2. The symmetric properties also hold.

Proposition 3.14 (congruence). *For each congruent functions $f_1 \equiv f_1' : X_1 \to Y_1$ and $f_2 \equiv f_2' : X_2 \to Y_2$*

$$f_1 \ltimes f_2 \equiv f_1' \ltimes f_2' \ .$$

Proof. Clear, from 3.6. □

Proposition 3.15 (composition). *For each functions $f_1 : X_1 \to Y_1$, $g_1 : Y_1 \to Z_1$, $g_2 : Y_2 \to Z_2$ and pure function $v_2 : X_2 \rightsquigarrow Y_2$*

$$(g_1 \ltimes g_2) \circ (f_1 \times v_2) \equiv (g_1 \circ f_1) \ltimes (g_2 \circ v_2) \ .$$

$$\begin{tikzcd}
X_1 \arrow[r, "f_1"] & Y_1 \arrow[r, "g_1"] & Z_1 \arrow[r, squiggly, "\mathrm{id}"] & Z_1 \\
X_1 \times X_2 \arrow[u, squiggly, "\equiv"'] \arrow[r, "f_1 \times v_2"] \arrow[d, squiggly, "\equiv"] & Y_1 \times Y_2 \arrow[u, squiggly, "\equiv"'] \arrow[r, "g_1 \times \mathrm{id}"] \arrow[d, squiggly, "\gtrsim"] & Z_1 \times Y_2 \arrow[u, squiggly, "\gtrsim"'] \arrow[r, "\mathrm{id} \times g_2"] \arrow[d, squiggly, "\equiv"] & Z_1 \times Z_2 \arrow[u, squiggly, "\gtrsim"'] \arrow[d, squiggly, "\equiv"] \\
X_2 \arrow[r, squiggly, "v_2"] & Y_2 \arrow[r, squiggly, "\mathrm{id}"] & Y_2 \arrow[r, "g_2"] & Z_2
\end{tikzcd}$$

Proof. ¿From several applications of proposition 3.7 and its symmetric version: $(\mathrm{id} \times g_2) \circ (g_1 \times \mathrm{id}) \circ (f_1 \times v_2) \equiv (\mathrm{id} \times g_2) \circ ((g_1 \circ f_1) \times v_2) \equiv (\mathrm{id} \times g_2) \circ (\mathrm{id} \times v_2) \circ ((g_1 \circ f_1) \times \mathrm{id}) \equiv (\mathrm{id} \times (g_2 \circ v_2)) \circ ((g_1 \circ f_1) \times \mathrm{id})$. □

Proposition 3.16 (swap). *For each functions $f_1 : X_1 \to Y_1$ and $f_2 : X_2 \to Y_2$, the left and right sequential products are related by swaps:*

$$\gamma_Y \circ (f_2 \rtimes f_1) \circ \gamma_X^{-1} \equiv f_1 \ltimes f_2 \ .$$

Proof. ¿From proposition 3.8 and its symmetric version:
$\gamma \circ (\mathrm{id} \times f_2) \circ (f_1 \times \mathrm{id}) \equiv (f_2 \times \mathrm{id}) \circ \gamma \circ (f_1 \times \mathrm{id}) \equiv (f_2 \times \mathrm{id}) \circ (\mathrm{id} \times f_1) \circ \gamma$. □

Proposition 3.17 (associativity). *For each functions $f_1 : X_1 \to Y_1$, $f_2 : X_2 \to Y_2$ and $f_3 : X_3 \to Y_3$, let $\alpha_Y = \alpha_{(Y_1,Y_2,Y_3)}$ and $\alpha_X = \alpha_{(X_1,X_2,X_3)}$, then: :*

$$\alpha_Y \circ (f_1 \ltimes (f_2 \ltimes f_3)) \equiv ((f_1 \ltimes f_2) \ltimes f_3) \circ \alpha_X \ .$$

Proof. ¿From proposition 3.9. □

3.5 Projections of sequential products

Let us come back to a weak category, as in section 2. The binary product of functions is characterized, up to congruence, by the equations:

$$q_1 \circ (f_1 \times f_2) \equiv f_1 \circ p_1 \ \text{ and } \ q_2 \circ (f_1 \times f_2) \equiv f_2 \circ p_2 \ ,$$

so that for all constant functions $x_1 : U \to X_1$ and $x_2 : U \to X_2$

$$q_1 \circ (f_1 \times f_2) \circ \langle x_1, x_2 \rangle \equiv f_1 \circ x_1 \ \text{ and } \ q_2 \circ (f_1 \times f_2) \circ \langle x_1, x_2 \rangle \equiv f_2 \circ x_2 \ .$$

In a cartesian effect category, it is proved in theorem 3.21 that $f_1 \ltimes f_2$, when applied to a pair of constant pure functions $\langle x_1, x_2 \rangle$, returns on the Y_1 side a

function that is semi-congruent to $f_1(x_1)$, and on the Y_2 side a function that is congruent to $f_2 \circ x_2 \circ \langle\,\rangle \circ f_1 \circ x_1$, which means "first $f_1(x_1)$, then forget the result, then $f_2(x_2)$". More precise statements are given in propositions 3.18 and 3.20. Proofs are presented in the same formalized way as in appendix A.

As above, we consider the semi-terminal point U and semi-products $X_1 \overset{p_1}{\leftsquigarrow} X_1 \times X_2 \overset{p_2}{\rightsquigarrow} X_2$, $Y_1 \overset{q_1}{\leftsquigarrow} Y_1 \times Y_2 \overset{q_2}{\rightsquigarrow} Y_2$ and $Y_1 \overset{s_1}{\leftsquigarrow} Y_1 \times X_2 \overset{s_2}{\rightsquigarrow} X_2$.

Proposition 3.18. *For each functions $f_1 : X_1 \to Y_1$ and $f_2 : X_2 \to Y_2$*

$$q_1 \circ (f_1 \ltimes f_2) \lesssim f_1 \circ p_1 : X_1 \times X_2 \to Y_1 \ .$$

$$\begin{tikzcd}
X_1 \arrow[r, "f_1"] & Y_1 \arrow[r, squiggly, "\mathrm{id}"] & Y_1 \\
X_1 \times X_2 \arrow[u, squiggly, "p_1"] \arrow[r, "f_1 \times \mathrm{id}"] \arrow[rr, bend right, "f_1 \ltimes f_2"'] \arrow[ur, phantom, "\equiv"] & Y_1 \times X_2 \arrow[u, squiggly, "s_1"] \arrow[r, "\mathrm{id} \times f_2"] \arrow[ur, phantom, "\gtrsim"] & Y_1 \times Y_2 \arrow[u, squiggly, "q_1"] \arrow[ur, phantom, "="]
\end{tikzcd}$$

Proof.

(a)	$q_1 \circ (\mathrm{id} \times f_2) \lesssim s_1$	
(b)	$q_1 \circ (f_1 \ltimes f_2) \lesssim s_1 \circ (f_1 \times \mathrm{id})$	(a), *subst*$_{\lesssim}$
(c)	$s_1 \circ (f_1 \times \mathrm{id}) \equiv f_1 \circ p_1$	
(d)	$q_1 \circ (f_1 \ltimes f_2) \lesssim f_1 \circ p_1$	(b), (c), *comp*

□

Lemma 3.19. *For each function $f_1 : X_1 \to Y_1$ and pure function $x_2 : U \rightsquigarrow X_2$*

$$\langle \mathrm{id}_{Y_1}, x_2 \circ \langle\,\rangle_{Y_1} \rangle \circ f_1 \equiv \langle f_1, x_2 \circ \langle\,\rangle_{X_1} \rangle : X_1 \to Y_1 \times X_2 \ .$$

Both handsides can be illustrated as follows:

$$\begin{tikzcd}
 & Y_1 \arrow[r, squiggly, "\mathrm{id}"] \arrow[d, equal] \arrow[dr, phantom, "\equiv"] & Y_1 \\
X_1 \arrow[r, "f_1"] & Y_1 \arrow[r, squiggly, "\langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle"] \arrow[d, squiggly, "\langle\,\rangle"'] \arrow[dr, phantom, "\equiv"] & Y_1 \times X_2 \arrow[u, squiggly, "s_1"'] \arrow[d, squiggly, "s_2"] \\
 & U \arrow[r, squiggly, "x_2"] & X_2
\end{tikzcd}
\qquad
\begin{tikzcd}
X_1 \arrow[r, "f_1"] \arrow[d, equal] \arrow[dr, phantom, "\equiv"] & Y_1 \\
X_1 \arrow[r, "\langle f_1, x_2 \circ \langle\,\rangle \rangle"] \arrow[d, squiggly, "\langle\,\rangle"'] \arrow[dr, phantom, "\gtrsim"] & Y_1 \times X_2 \arrow[u, squiggly, "s_1"'] \arrow[d, squiggly, "s_2"] \\
U \arrow[r, squiggly, "x_2"] & X_2
\end{tikzcd}$$

Proof.

(a_1)	$s_1 \circ \langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle \equiv \mathrm{id}$	
(b_1)	$s_1 \circ \langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle \circ f_1 \equiv f_1$	(a_1), *subst*$_{\equiv}$
(a_2)	$s_2 \circ \langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle \equiv x_2 \circ \langle\,\rangle$	
(b_2)	$s_2 \circ \langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle \circ f_1 \equiv x_2 \circ \langle\,\rangle \circ f_1$	(a_2), *subst*$_{\equiv}$
(c_2)	$\langle\,\rangle \circ f_1 \lesssim \langle\,\rangle$	semi-terminality of U
(d_2)	$x_2 \circ \langle\,\rangle \circ f_1 \lesssim x_2 \circ \langle\,\rangle$	(c_2), *repl*$_{\lesssim}$ (x_2 is pure)
(e_2)	$s_2 \circ \langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle \circ f_1 \lesssim x_2 \circ \langle\,\rangle$	(b_2), (d_2), *trans*$_{\lesssim}$
(f)	$\langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle \circ f_1 \equiv \langle f_1, x_2 \circ \langle\,\rangle \rangle$	(b_1), (e_2)

□

Proposition 3.20. *For each functions $f_1 : X_1 \to Y_1$, $f_2 : X_2 \to Y_2$ and pure function $x_2 : U \rightsquigarrow X_2$*

$$q_2 \circ (f_1 \ltimes f_2) \circ \langle \mathrm{id}_{X_1}, x_2 \circ \langle\,\rangle_{X_1} \rangle \equiv f_2 \circ x_2 \circ \langle\,\rangle_{Y_1} \circ f_1 : X_1 \to Y_2 \ .$$

Both handsides can be illustrated as follows:

$$\begin{tikzcd}
X_1 \arrow[r, squiggly, "\mathrm{id}"] \arrow[d, equal] \arrow[dr, phantom, "\equiv"] & X_1 \\
X_1 \arrow[r, squiggly, "\langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle"] \arrow[d, squiggly, "\langle\,\rangle"'] \arrow[dr, phantom, "\equiv"] & X_1 \times X_2 \arrow[u, squiggly, "p_1"'] \arrow[r, "f_1 \times \mathrm{id}"] \arrow[d, squiggly, "p_2"] \arrow[dr, phantom, "\gtrsim"] \arrow[rr, bend left, "f_1 \ltimes f_2", "="'] & Y_1 \times X_2 \arrow[r, "\mathrm{id} \times f_2"] \arrow[d, squiggly, "s_2"] \arrow[dr, phantom, "\equiv"] & Y_1 \times Y_2 \arrow[d, squiggly, "q_2"] \\
U \arrow[r, squiggly, "x_2"] & X_2 \arrow[r, squiggly, "\mathrm{id}"] & X_2 \arrow[r, "f_2"] & Y_2
\end{tikzcd}
\qquad
\begin{tikzcd}
X_1 \arrow[r, "f_1"] & Y_1 \arrow[d, squiggly, "\langle\,\rangle"'] & \\
& U \arrow[r, squiggly, "x_2"] & X_2 \arrow[r, "f_2"] & Y_2
\end{tikzcd}$$

Proof.

(a)	$q_2 \circ (\mathrm{id} \times f_2) \equiv f_2 \circ s_2$	
(b)	$q_2 \circ (f_1 \ltimes f_2) \circ \langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle \equiv f_2 \circ s_2 \circ (f_1 \times \mathrm{id}) \circ \langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle$	(a), *subst*$_\equiv$
(c)	$(f_1 \times \mathrm{id}) \circ \langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle \equiv \langle f_1, x_2 \circ \langle\,\rangle \rangle$	prop. 3.7
(d)	$\langle f_1, x_2 \circ \langle\,\rangle \rangle \equiv \langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle \circ f_1$	lemma 3.19
(e)	$(f_1 \times \mathrm{id}) \circ \langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle \equiv \langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle \circ f_1$	(c), (d), *trans*$_\equiv$
(f)	$f_2 \circ s_2 \circ (f_1 \times \mathrm{id}) \circ \langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle \equiv f_2 \circ s_2 \circ \langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle \circ f_1$	(e), *repl*$_\equiv$
(g)	$q_2 \circ (f_1 \ltimes f_2) \circ \langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle \equiv f_2 \circ s_2 \circ \langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle \circ f_1$	(b), (f), *trans*$_\equiv$
(h)	$p_2 \circ \langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle \equiv x_2 \circ \langle\,\rangle$	
(i)	$f_2 \circ s_2 \circ \langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle \circ f_1 \equiv f_2 \circ x_2 \circ \langle\,\rangle \circ f_1$	(h), *subst*$_\equiv$, *repl*$_\equiv$
(j)	$q_2 \circ (f_1 \ltimes f_2) \circ \langle \mathrm{id}, x_2 \circ \langle\,\rangle \rangle \equiv f_2 \circ x_2 \circ \langle\,\rangle \circ f_1$	(g), (i), *trans*$_\equiv$

□

Theorem 3.21. *For each functions $f_1 : X_1 \to Y_1$, $f_2 : X_2 \to Y_2$ and pure functions $x_1 : U \rightsquigarrow X_1$ and $x_2 : U \rightsquigarrow X_2$, the function $(f_1 \ltimes f_2) \circ \langle x_1, x_2 \rangle$ satisfies:*

$$q_1 \circ (f_1 \ltimes f_2) \circ \langle x_1, x_2 \rangle \lesssim f_1 \circ x_1 \ \ \textit{and} \ \ q_2 \circ (f_1 \ltimes f_2) \circ \langle x_1, x_2 \rangle \equiv f_2 \circ x_2 \circ \langle\,\rangle_{Y_1} \circ f_1 \circ x_1 \ .$$

$$\begin{tikzcd}
U \arrow[r, squiggly, "x_1"] \arrow[d, equal] & X_1 \arrow[rr, "f_1"] \arrow[dr, phantom, "\gtrsim"] & & Y_1 \\
U \arrow[r, squiggly, "\langle x_1, x_2 \rangle"] \arrow[d, equal] & X_1 \times X_2 \arrow[rr, "f_1 \ltimes f_2"] \arrow[dr, phantom, "\equiv"] & & Y_1 \times Y_2 \arrow[u, squiggly, "q_1"'] \arrow[d, squiggly, "q_2"] \\
U \arrow[r, squiggly, "x_1"] & X_1 \arrow[r, "f_1"] & Y_1 \arrow[r, squiggly, "\langle\,\rangle"] & U \arrow[r, squiggly, "x_2"] & X_2 \arrow[r, "f_2"] & Y_2
\end{tikzcd}$$

Proof.

(a_1)	$q_1 \circ (f_1 \ltimes f_2) \lesssim f_1 \circ p_1$	prop. 3.18
(b_1)	$q_1 \circ (f_1 \ltimes f_2) \circ \langle x_1, x_2 \rangle \lesssim f_1 \circ p_1 \circ \langle x_1, x_2 \rangle$	(a_1), $subst_{\lesssim}$
(c_1)	$p_1 \circ \langle x_1, x_2 \rangle \equiv x_1$	(on values)
(d_1)	$f_1 \circ p_1 \circ \langle x_1, x_2 \rangle \equiv f_1 \circ x_1$	(c_1), $repl_{\equiv}$
(e_1)	$q_1 \circ (f_1 \ltimes f_2) \circ \langle x_1, x_2 \rangle \lesssim f_1 \circ x_1$	(b_1), (d_1), *comp*
(a_2)	$\langle x_1, x_2 \rangle \equiv \langle \mathrm{id}_{X_1}, x_2 \circ \langle\,\rangle_{X_1} \rangle \circ x_1$	(on values)
(b_2)	$q_2 \circ (f_1 \ltimes f_2) \circ \langle x_1, x_2 \rangle \equiv q_2 \circ (f_1 \ltimes f_2) \circ \langle \mathrm{id}_{X_1}, x_2 \circ \langle\,\rangle_{X_1} \rangle \circ x_1$	(a_2), $repl_{\equiv}$
(c_2)	$q_2 \circ (f_1 \ltimes f_2) \circ \langle \mathrm{id}_{X_1}, x_2 \circ \langle\,\rangle_{X_1} \rangle \equiv f_2 \circ x_2 \circ \langle\,\rangle_{Y_1} \circ f_1$	prop. 3.20
(d_2)	$q_2 \circ (f_1 \ltimes f_2) \circ \langle \mathrm{id}_{X_1}, x_2 \circ \langle\,\rangle_{X_1} \rangle \circ x_1 \equiv f_2 \circ x_2 \circ \langle\,\rangle_{Y_1} \circ f_1 \circ x_1$	(c_2), $subst_{\equiv}$
(e_2)	$q_2 \circ (f_1 \ltimes f_2) \circ \langle x_1, x_2 \rangle \equiv f_2 \circ x_2 \circ \langle\,\rangle_{Y_1} \circ f_1 \circ x_1$	(b_2), (d_2), $trans_{\equiv}$

□

The corresponding properties of left sequential pairs easily follow.

Corollary 3.22. *For each functions $f_1 : X \to Y_1$, $f_2 : X \to Y_2$ and pure function $x : U \rightsquigarrow X$*

$q_1 \circ \langle f_1, f_2 \rangle_l \lesssim f_1$ *, hence* $q_1 \circ \langle f_1, f_2 \rangle_l \circ x \lesssim f_1 \circ x$ *, and* $q_2 \circ \langle f_1, f_2 \rangle_l \circ x \equiv f_2 \circ x \circ \langle\,\rangle_{Y_1} \circ f_1 \circ x$.

4 Effect categories and Arrows

Starting from [8, 12], *monads* are used in Haskell for dealing with computational effects. A *Monad type* in Haskell is a unary type constructor that corresponds to a *strong monad*, in the categorical sense. Monads have been generalized on the categorical side to *Freyd categories* [10] and on the functional programming side to *Arrows* [7]. A precise statement of the facts that Arrows generalize Monads and that Arrows are Freyd categories can be found in [6], where each of the three notions is seen as a monoid in a relevant category. Now we prove that cartesian effect categories determine Arrows. In section 5 our approach is compared with the Monads approach, for two fundamental examples. In this section, all effect categories are strict: the congruence $\equiv$ is the equality.

4.1 Arrows

According to [9], Arrows in Haskell are defined as follows.

Definition 4.1. An *Arrow* is a binary type constructor class `A` of the form:

```
class Arrow A where
    arr :: (X → Y) → A X Y
    (⋙) :: A X Y → A Y Z → A X Z
    first :: A X Y → A (X, Z) (Y, Z)
```

satisfying the following equations:

(1)	$\texttt{arr}\ \mathrm{id} \ggg f$	$=$	f
(2)	$f \ggg \texttt{arr}\ \mathrm{id}$	$=$	f
(3)	$(f \ggg g) \ggg h$	$=$	$f \ggg (g \ggg h)$
(4)	$\texttt{arr}\ (w.v)$	$=$	$\texttt{arr}\ v \ggg \texttt{arr}\ w$
(5)	$\texttt{first}\ (\texttt{arr}\ v)$	$=$	$\texttt{arr}\ (v \times \mathrm{id})$
(6)	$\texttt{first}\ (f \ggg g)$	$=$	$\texttt{first}\ f \ggg \texttt{first}\ g$
(7)	$\texttt{first}\ f \ggg \texttt{arr}\ (\mathrm{id} \times v)$	$=$	$\texttt{arr}\ (\mathrm{id} \times v) \ggg \texttt{first}\ f$
(8)	$\texttt{first}\ f \ggg \texttt{arr}\ \texttt{fst}$	$=$	$\texttt{arr}\ \texttt{fst} \ggg f$
(9)	$\texttt{first}\ (\texttt{first}\ f) \ggg \texttt{arr}\ \texttt{assoc}$	$=$	$\texttt{arr}\ \texttt{assoc} \ggg \texttt{first}\ f$

where the functions $(\times)$, `fst` and `assoc` are defined as:

$(\times) ::$	$(X \to X') \to (Y \to Y') \to (X, Y) \to (X', Y')$	$(f \times g)(x, y) = (f\ x, g\ y)$
$\texttt{fst} ::$	$(X, Y) \to X$	$\texttt{fst}(x, y) = x$
$(\texttt{assoc}) ::$	$((X, Y), Z) \to (X, (Y, Z))$	$\texttt{assoc}((x, y), z) = (x, (y, z))$

4.2 Cartesian effect categories determine Arrows

Let $\mathbf{V}_H$ denote the category of Haskell types and ordinary functions, so that the Haskell notation $(\texttt{X} \to \texttt{Y})$ represents $\mathbf{V}_H(X, Y)$, made of the Haskell ordinary functions from X to Y. An arrow `A` contructs a type `A` X Y for all types X and Y. We slightly modify the definition of Arrows by allowing $(\texttt{X} \to \texttt{Y})$ to represent $\mathbf{V}(X, Y)$ for any cartesian category $\mathbf{V}$ and by requiring that `A` X Y is a set rather than a type. In addition, we use categorical notations instead of Haskell syntax.

So, from now on, for any cartesian category $\mathbf{V}$, an *Arrow A on* $\mathbf{V}$ associates to each points X, Y of $\mathbf{V}$ a set $A(X, Y)$, together with three operations:

$$\begin{aligned}
&\texttt{arr} : \mathbf{V}(X, Y) \to A(X, Y)\\
&\ggg : A(X, Y) \to A(Y, Z) \to A(X, Z)\\
&\texttt{first} : A(X, Y) \to A(X \times Z, Y \times Z)
\end{aligned}$$

that satisfy the equations (1)-(9).

Basically, the correspondence between a cartesian effect category $\mathbf{C}$ extending $\mathbf{V}$ and an Arrow A on $\mathbf{V}$ identifies $\mathbf{C}(X, Y)$ with $A(X, Y)$ for all types X and Y. More precisely:

Theorem 4.2. *Every cartesian effect category* $\mathbf{C}$ *extending* $\mathbf{V}$ *gives rise to an Arrow A on* $\mathbf{V}$*, according to the following table:*

Cartesian effect categories	*Arrows*
$\mathbf{C}(X, Y)$	$A(X, Y)$
$\mathbf{V}(X, Y) \subseteq \mathbf{C}(X, Y)$	$\texttt{arr} : \mathbf{V}(X, Y) \to A(X, Y)$
$f \mapsto (g \mapsto g \circ f)$	$\ggg : A(X, Y) \to A(Y, Z) \to A(X, Z)$
$f \mapsto f \times \mathrm{id}$	$\texttt{first} : A(X, Y) \to A(X \times Z, Y \times Z)$

Proof. The first and second line in the table say that $A(X, Y)$ is made of the functions from X to Y in $\mathbf{C}$ and that `arr` is the convertion from pure functions to arbitrary functions. The third and fourth lines say that $\ggg$ is the (reverse) composition of functions and that `first` is the semi-product with the identity. Let us check that A is an Arrow; the following table translates each property (1)-(9) in terms of cartesian effect categories (where $\rho_X : X \times U \to X$ is the projection), and gives the argument for its proof.

(1)	$f \circ \mathrm{id}$	$=$	f	unitarity in $\mathbf{C}$
(2)	$\mathrm{id} \circ f$	$=$	f	unitarity in $\mathbf{C}$
(3)	$h \circ (g \circ f)$	$=$	$(h \circ g) \circ f$	associativity in $\mathbf{C}$
(4)	$w \circ v$ in $\mathbf{V}$	$=$	$w \circ v$ in $\mathbf{C}$	$\mathbf{V} \subseteq \mathbf{C}$ is a functor
(5)	$v \times \mathrm{id}$ in $\mathbf{V}$	$=$	$v \times \mathrm{id}$ in $\mathbf{C}$	non-ambiguity of "$\times$"
(6)	$(g \circ f) \times \mathrm{id}$	$=$	$(g \times \mathrm{id}) \circ (f \times \mathrm{id})$	proposition 3.7
(7)	$(\mathrm{id} \times v) \circ (f \times \mathrm{id})$	$=$	$(f \times \mathrm{id}) \circ (\mathrm{id} \times v)$	proposition 3.7
(8)	$\rho \circ (f \times \mathrm{id}_U)$	$=$	$f \circ \rho$	definition 3.5
(9)	$\alpha^{-1} \circ ((f \times \mathrm{id}) \times \mathrm{id})$	$=$	$(f \times \mathrm{id}) \circ \alpha^{-1}$	proposition 3.9

□

The translation of the Arrow combinators follows easily, using $\langle f, g\rangle_l = (f \ltimes g) \circ \langle \mathrm{id}, \mathrm{id}\rangle$ as in section 3.4:

Cartesian effect categories	Arrows
$(id \times f) = \gamma \circ (f \times id) \circ \gamma$	`second` f = `arr swap` $\ggg$ `first` f $\ggg$ `arr swap`
$f \ltimes g = (\mathrm{id} \times g) \circ (f \times \mathrm{id})$	f *** g = `first` f $\ggg$ `second` g
$\langle f, g\rangle_l = (f \ltimes g) \circ \langle \mathrm{id}, \mathrm{id}\rangle$	f&&&g = `arr`$(\lambda b \to (b, b))$ $\ggg$ $(f$ *** $g)$

For instance, in [7], the author states that &&& is not a categorical product since in general $(f$&&&$g)$ $\ggg$ `arr fst` is different from f. We can state this more precisely in the effect category, where $(f$&&&$g)$ $\ggg$ `arr fst` corresponds to $q_1 \circ \langle f, g\rangle_l$. Indeed, according to corollary 3.22:

$$q_1 \circ \langle f, g\rangle_l \lesssim f \ .$$

5 Examples

Here are presented some examples of strict cartesian effect categories. Several versions are given, some of them rely on monads.

5.1 Partiality

Let $\mathbf{V} = \mathbf{Set}$ be the category of sets and maps, and $\mathbf{C} = \mathbf{Part}$ the category of sets and partial maps, so that $\mathbf{V}$ is a wide subcategory of $\mathbf{C}$. Let $\lesssim$ denote the usual ordering on partial maps: $f \lesssim g$ if and only if $\mathcal{D}(f) \subseteq \mathcal{D}(g)$ (where $\mathcal{D}$ denotes the domain of definition) and $f(x) = g(x)$ for all $x \in \mathcal{D}(f)$. The restriction of $\lesssim$ to

$\mathbf{V}$ is the equality of total maps. Clearly $\lesssim$ is not symmetric, but it satisfies all the other properties of a congruence, in particular the replacement property with respect to all maps. So, $\lesssim$ is a semi-congruence (which satisfies replacement), that makes $\mathbf{C}$ a strict effect category extending $\mathbf{V}$. *Warning:* usually the notations are $v : X \to Y$ for a total map and $f : X \rightharpoonup Y$ for a partial map, but here we use respectively $v : X \rightsquigarrow Y$ (total) and $f : X \to Y$ (partial).

Let us define the pair $\langle f, v\rangle$ of a partial map $f : X \to Y_1$ and a total map $v : X \rightsquigarrow Y_2$ as the partial map $\langle f, v\rangle : X \to Y_1 \times Y_2$ with the same domain of definition as f and such that $\langle f, v\rangle(x) = \langle f(x), v(x)\rangle$ for all $x \in \mathcal{D}(f)$. It is easy to check that we get a cartesian effect category. For illustrating the semi-product $f \times v$, there are two cases: either $f(x_1)$ is defined, or not, in which case we note $f(x_1) = \bot$. We use the traditional notation $x \overset{f}{\longmapsto} y$ when $y = f(x)$ and its analog $x \overset{v}{\rightsquigarrow} y$ when $y = v(x)$ and v is pure.

$$\begin{tikzcd}
x_1 \arrow[r, mapsto, "f"] & y_1 \\
\langle x_1, x_2\rangle \arrow[u, squiggly] \arrow[r, mapsto, "f\times v"] \arrow[d, squiggly] \arrow[ur, phantom, "="] & \langle y_1, y_2\rangle \arrow[u, squiggly] \arrow[d, squiggly] \\
x_2 \arrow[r, mapsto, squiggly, "v"] \arrow[ur, phantom, "="] & y_2
\end{tikzcd}
\qquad \text{or} \qquad
\begin{tikzcd}
x_1 \arrow[r, mapsto, "f"] & \bot \\
\langle x_1, x_2\rangle \arrow[u, squiggly] \arrow[r, mapsto, "f\times v"] \arrow[d, squiggly] \arrow[ur, phantom, "="] & \bot \arrow[u, squiggly] \arrow[d, squiggly] \\
x_2 \arrow[r, mapsto, squiggly, "v"] \arrow[ur, phantom, "\gtrsim"] & y_2 \neq \bot
\end{tikzcd}$$

It can be noted that, in the previous example, $\mathbf{C}$ is a 2-category, with a 2-cell from f to g if and only if $f \lesssim g$. More generally, let $\mathbf{C}$ be a 2-category and $\mathbf{V}$ a sub-2-category where the unique 2-cells are the identities. Then by defining $f \lesssim g$ whenever there is a 2-cell from f to g, we get a strict effect category. In such effect categories, the replacement property holds with respect to all functions in $\mathbf{C}$, but the semi-congruence is usually not symmetric.

Let us come back to the partiality example, from the slightly different point of view of the *Maybe monad*. First, let us present this point of view in a naive way, without monads. Let $U = \{\bot\}$ be a singleton, let "+" denote the disjoint union of sets, and for each set X let $GX = X + U$ and let $\eta_X : X \to GX$ be the inclusion. Each partial map f from X to Y can be extended as a total map Gf from X to GY, such that $Gf(x) = f(x)$ for $x \in \mathcal{D}(f)$ and $Gf(x) = \bot$ otherwise. This defines a bijection between the partial maps from X to Y and the total maps from X to GY. Let $\mathbf{C}$ be the category such that its points are the sets, and a function $X \to Y$ in $\mathbf{C}$ is a function $X \to GY$ in $\mathbf{Set}$; we say that $X \to Y$ in $\mathbf{C}$ *stands for* $X \to GY$ in $\mathbf{Set}$. Let $J : \mathbf{Set} \to \mathbf{C}$ be the functor that is the identity on points and associates to each map $v_0 : X \to Y$ the map $\eta_Y \circ v_0$. Let $\mathbf{V} = J(\mathbf{Set})$. Then $\mathbf{V}$ is a wide subcategory of $\mathbf{C}$. For all $f, g : X \to Y$ in $\mathbf{C}$, that stand for $f, g : X \to GY$ in $\mathbf{Set}$, let:

$$f \lesssim g \iff \forall x \in X\ (f(x) \neq \bot \Rightarrow (g(x) \neq \bot \wedge g(x) = f(x))\ .$$

This yields a strict effect category $\mathbf{C}$ extending $\mathbf{V}$, with the semi-congruence $\lesssim$, and as above the replacement property holds with respect to all functions in $\mathbf{C}$

but $\lesssim$ is not symmetric. Let $f : X \to Y_1$ in $\mathbf{C}$ and $v : X \to Y_2$ in $\mathbf{V}$, they stand respectively for $f : X \to GY_1$ and $v = \eta_{Y_2} \circ v_0$ with $v_0 : X \to Y_2$. Then, in **Set**, the pair $\langle f, v_0 \rangle : X \to GY_1 \times Y_2$ can be composed with:

$$t : GY_1 \times Y_2 = (Y_1 + U) \times Y_2 \to (Y_1 \times Y_2) + U = G(Y_1 \times Y_2) \ ,$$

that maps $\langle y_1, y_2 \rangle$ to itself and $\langle \perp, y_2 \rangle$ to $\perp$. Now, let $\langle f, v \rangle : X \to Y_1 \times Y_2$ in $\mathbf{C}$ stand for $\langle f, v \rangle = t \circ \langle f, v_0 \rangle : X \to G(Y_1 \times Y_2)$ in **Set**. Then $\langle f, v \rangle$ is a semi-product, so that $\mathbf{C}$ is a cartesian effect category. The diagrams for illustrating the semi-product $f \times v$ are the same as above.

This point of view can also be presented using the the *Maybe monad* for managing failures, as follows. We have defined a functor G : **Part** $\to$ **Set**, that is a right adjoint to the inclusion functor I : **Set** $\subseteq$ **Part**. The corresponding monad has endofunctor $M = GI$ on **Set**, the category $\mathbf{C}$ is the Kleisli category of M, and J : **Set** $\to$ $\mathbf{C}$ is the canonical functor associated to the monad. In addition, this monad M is *strong*, and t is the (Y_1, Y_2) component of the *strength* of M. But the definition of the semi-congruence $\lesssim$, as above, is not part of the usual framework of monads.

5.2 State

Let $\mathbf{V}_0$ be a cartesian category, with a distinguished point S for "the type of states"; for all X, let $\pi_X : S \times X \to X$ denotes the projection. Let $\mathbf{C}$ be the category with the same points as $\mathbf{V}_0$ and with a function $f : X \to Y$ for each function $f : S \times X \to S \times Y$ in $\mathbf{V}_0$; we say that $f : X \to Y$ in $\mathbf{C}$ *stands for* $f : S \times X \to S \times Y$ in $\mathbf{V}_0$. Let $J : \mathbf{V}_0 \to \mathbf{C}$ be the identity-on-points functor which maps each $v_0 : X \to Y$ in $\mathbf{V}_0$ to the function $J(v_0) : X \to Y$ in $\mathbf{C}$ that stands for $\mathrm{id}_S \times v_0 : S \times X \to S \times Y$ in $\mathbf{V}_0$. Let $\mathbf{V} = J(\mathbf{V}_0)$, it is a wide subcategory of $\mathbf{C}$. For all $f, g : X \to Y$ in $\mathbf{C}$, let:

$$f \lesssim g \iff \pi_Y \circ g = \pi_Y \circ f \ .$$

We get a strict effect category, where the semi-congruence $\lesssim$ is symmetric, but does not satisfy the replacement property with respect to all functions in $\mathbf{C}$. The semi-product of $f : X \to Y_1$ and $v : X \rightsquigarrow Y_2$ is defined as follows. Since $f : S \times X \to S \times Y_1$ in $\mathbf{V}_0$ and $v = \mathrm{id}_S \times v_0$ for some $v_0 : X \to Y$ in $\mathbf{V}_0$, the pair $\langle f, v_0 \circ \pi_X \rangle : S \times X \to (S \times Y_1) \times Y_2$ exists in $\mathbf{V}_0$. By composing it with the isomorphism $(S \times Y_1) \times Y_2 \to S \times (Y_1 \times Y_2)$ we get $\langle f, v \rangle : S \times X \to S \times (Y_1 \times Y_2)$ in $\mathbf{V}_0$, i.e., $\langle f, v \rangle : X \to Y_1 \times Y_2$ in $\mathbf{C}$. It is easy to check that this defines a semi-product, so that $\mathbf{C}$ is a cartesian effect category, where the characteristic

property of the semi-product $f \times v$ can be illustrated as follows:

$$\begin{tikzcd}
(s, x_1) \arrow[rr, mapsto, "f"] & & (s', y_1) \\
(s, x_1, x_2) \arrow[u, squiggly, "="'] \arrow[rr, mapsto, "f \times v"] \arrow[d, squiggly, "\gtrsim"] & & (s', y_1, y_2) \arrow[u, squiggly, "="'] \arrow[d, squiggly] \\
(s, x_2) \arrow[rr, mapsto, squiggly, "v"] & & (s, y_2) \neq (s', y_2) \arrow[r, mapsto, squiggly, "\pi_Y"] & y_2
\end{tikzcd}$$

The example above can be curried, thus recovering the *State monad*. A motivation for the introduction of Freyd categories in [10] is the possibility of dealing with state in a linear way, as above, rather than in the exponential way provided by the State monad. Now $\mathbf{V}_0$ is still a cartesian category with a distinguished point S, the "type of states", and in addition $\mathbf{V}_0$ has exponentials $(S \times X)^S$ for each X. Then the endofunctor $M(X) = (S \times X)^S$ defines the State monad on $\mathbf{V}_0$, with composition defined as usual. It is well-known that M is a strong monad, with strength $t_{Y_1,Y_2} = (S \times Y_1)^S \times Y_2 \to (S \times Y_1 \times Y_2)^S$ obtained from $\mathrm{app}_{S \times Y_1} \times \mathrm{id}_{Y_2} : S \times (S \times Y_1)^S \times Y_2 \to S \times Y_1 \times Y_2$, where "app" denotes the application function. Hence, from $f : X \to M(Y_1)$ and $v_0 : X \to Y_2$ in $\mathbf{V}_0$, we can build $\langle f, v \rangle = t_{Y_1,Y_2} \circ \langle f, v_0 \rangle : X \to M(Y_1 \times Y_2)$. Let $\mathbf{C}$ be the Kleisli category of the monad M, let $J : \mathbf{V}_0 \to \mathbf{C}$ be the canonical functor associated to the monad, and let $\mathbf{V} = J(\mathbf{V}_0)$, then $\mathbf{V}$ is a wide subcategory of $\mathbf{C}$. A function $f : X \to Y$ in $\mathbf{C}$ stands for a function $f : X \to (S \times Y)^S$ in $\mathbf{V}_0$. Now, in addition to the usual framework of monads, for all $f, g : X \to Y$ in $\mathbf{C}$, i.e., $f, g : X \to (S \times Y)^S$ in $\mathbf{V}_0$, let:

$$f \lesssim g \iff {\pi_Y}^S \circ g = {\pi_Y}^S \circ f \ ,$$

where ${\pi_Y}^S : (S \times Y)^S \to Y^S$ associates to each map $m : S \to S \times Y$ the map $\pi_Y \times m : S \to Y$. The relation $\lesssim$ defines a semi-conguence on $\mathbf{C}$, and $\langle f, v \rangle$ is a semi-product, so that $\mathbf{C}$ is a cartesian effect category. The characteristic property of the semi-product $f \times v$ can be illustrated as follows:

$$\begin{tikzcd}
x_1 \arrow[rr, mapsto, "f"] & & (s \mapsto (s', y_1)) \\
(x_1, x_2) \arrow[u, squiggly, "="'] \arrow[rr, mapsto, "f \times v"] \arrow[d, squiggly, "\gtrsim"] & & (s \mapsto (s', y_1, y_2)) \arrow[u, squiggly, "="'] \arrow[d, squiggly] \\
x_2 \arrow[rr, mapsto, squiggly, "v"] & & (s \mapsto (s, y_2)) \neq (s \mapsto (s', y_2)) \arrow[r, mapsto, squiggly, "\pi_Y^S"] & (s \mapsto y_2)
\end{tikzcd}$$

6 Conclusion

We have presented a new categorical framework, called a *cartesian effect category*, for dealing with the issue of multiple arguments in programming languages

with computational effects. The major new feature in cartesian effect categories is the introduction of a *semi-congruence*, which allows to define *semi-products* and to prove their properties by decorating the usual definitions, properties and proofs about products in a category. Forthcoming work should study the nesting of several effects.

In order to deal with other issues related to effects, we believe that the idea of *decorations* in logic can be more widely used. This is the case for dealing with exceptions [5] (note that a previous attempt to define decorated products can be found in [4]). The framework of decorations might be used for generalizing this work in the direction of closed Freyd categories [11]. or traced premonoidal categories [1]. Moreover, with one additional level of abstraction, decorations can be obtained from *morphisms between logics*, in the context of *diagrammatic logics* [3, 2].

References

[1] N. Benton and M. Hyland. Traced premonoidal categories. *Theoretical Informatics and Applications*, 37:273–299, 2003.

[2] Dominique Duval. Diagrammatic specifications. *Mathematical Structures in Computer Science*, 13(6):857–890, 2003.

[3] Dominique Duval and Christian Lair. Diagrammatic specifications. Rapport de Recherche 1043 -M-, LMC, 2002.

[4] Dominique Duval and Jean-Claude Reynaud. Diagrammatic logic and effects: the example of exceptions. Technical Report ccsd-00004129, December 2004.

[5] Dominique Duval and Jean-Claude Reynaud. Dynamic logic and exceptions: an introduction. In *Proc. MAP05, Mathematics, Algorithms, Proofs*, 2005.

[6] Chris Heunen and Bart Jacobs. Arrows, like monads, are monoids. *Electronic Notes in Theoretical Computer Science*, pages 219–236, 2006.

[7] John Hughes. Generalising monads to arrows. *Science of Computer Programming*, 37(1–3):67–111, 2000.

[8] Eugenio Moggi. Notions of computation and monads. *Information and Computation*, 93:55–92, 1991.

[9] Ross Paterson. A new notation for arrows. In *International Conference on Functional Programming*, pages 229–240. ACM, 2001.

[10] John Power and Edmund Robinson. Premonoidal categories and notions of computation. *Mathematical Structures in Computer Science*, 7(5):453–468, 1997.

[11] John Power and Hayo Thielecke. Closed Freyd- and κ-categories. In J. Wiedermann, P. van Emde Boas, and M. Nielsen, editors, *Proceedings 26th Int. Coll. on Automata, Languages and Programming, ICALP'99, Prague, Czech Rep., 11–15 July 1999*, volume 1644, pages 625–634. Springer-Verlag, Berlin, 1999.

[12] Philip Wadler. Monads for functional programming. In M. Broy, editor, *Program Design Calculi: Proceedings of the 1992 Marktoberdorf International Summer School*. Springer-Verlag, 1993.

A Proofs in cartesian effect categories

Here are proofs for some results in section 2.2, called *basic* proofs, followed by their *decorated* versions for the corresponding results in section 3.2. All basic proofs are straightforward. All proofs are presented in a formalized way: each property is preceded by its label and followed by its proof. For the basic proofs, the properties of the congruence are denoted *trans*, *sym*, *subst*, *repl*, for respectively transitivity, symmetry, substitution, replacement. For the decorated proofs, the properties of the congruence and the semi-congruence are still denoted *trans*, *sym*, *subst*, *repl*, with subscript either $\equiv$ or $\lesssim$. It should be reminded that $sym_{\lesssim}$ does *not* hold, and that $repl_{\lesssim}$ is allowed *only* with respect to a pure function: if $g_1 \lesssim g_2 : Y \to Z$ and $v : Z \rightsquigarrow W$ then $v \circ g_1 \lesssim v \circ g_2 : Y \to W$. In addition, *comp* means compatibiblity of $\lesssim$ with $\equiv$, which means that if either $f_1 \equiv f_2 \lesssim f_3$ or $f_1 \lesssim f_2 \equiv f_3$ then $f_1 \lesssim f_3$. In decorated proofs, "like *basic*" means that this part of the proof is exactly the same as in the basic proof. Proofs of propositions 2.9, 3.9(associativity) and 2.10, 3.10 (parallelism) are left to the reader.

Proof of proposition 2.6 (congruence).

1.	When $X_1 = X_2$	
(a_1)	$q_1 \circ \langle f_1, f_2 \rangle \equiv f_1$	
(b_1)	$f_1 \equiv f_1'$	
(c_1)	$q_1 \circ \langle f_1, f_2 \rangle \equiv f_1'$	(a_1), (b_1), *trans*
(c_2)	$q_2 \circ \langle f_1, f_2 \rangle \equiv f_2'$	like (c_1)
(d)	$\langle f_1, f_2 \rangle \equiv \langle f_1', f_2' \rangle$	(c_1), (c_2)
2.	In all cases	
(e_1)	$f_1 \equiv f_1'$	
(f_1)	$f_1 \circ p_1 \equiv f_1' \circ p_1$	(e_1), *subst*
(f_2)	$f_2 \circ p_2 \equiv f_2' \circ p_2$	like (f_1)
(g)	$\langle f_1 \circ p_1, f_2 \circ p_2 \rangle \equiv \langle f_1' \circ p_1, f_2' \circ p_2 \rangle$	(f_1), (f_2), (1)

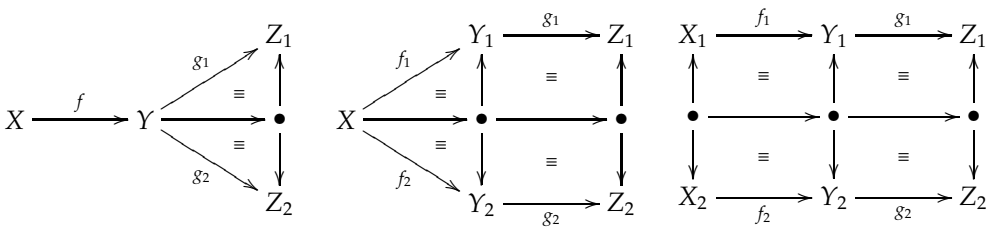
□

Proof of proposition 3.6 (congruence).

1.	When $X_1 = X_2$	
(c_1)	$q_1 \circ \langle f_1, f_2 \rangle \equiv f_1'$	like *basic*
(a_2)	$q_2 \circ \langle f_1, f_2 \rangle \lesssim f_2$	
(b_2)	$f_2 \equiv f_2'$	
(c_2)	$q_2 \circ \langle f_1, f_2 \rangle \lesssim f_2'$	(a_2), (b_2), *comp*
(d)	$\langle f_1, f_2 \rangle \equiv \langle f_1', f_2' \rangle$	(c_1), (c_2)
2.	In all cases	
(g)	$\langle f_1 \circ p_1, f_2 \circ p_2 \rangle \equiv \langle f_1' \circ p_1, f_2' \circ p_2 \rangle$	like *basic*

□

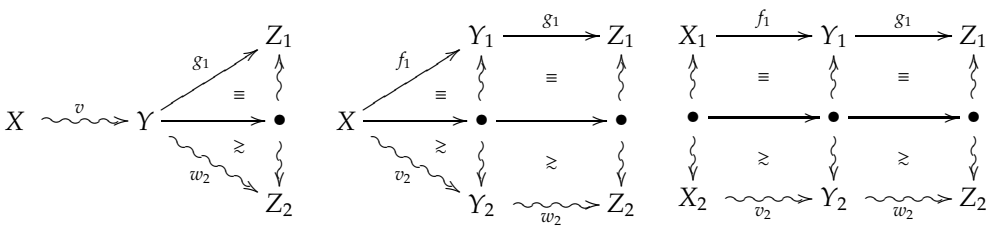
Proof of proposition 2.7 (composition).
The three left handsides can be illustrated as follows:



1.	When $f_1 = f_2 (= f)$	
(a_1)	$r_1 \circ \langle g_1, g_2 \rangle \equiv g_1$	
(b_1)	$r_1 \circ \langle g_1, g_2 \rangle \circ f \equiv g_1 \circ f$	(a_1), *subst*
(b_2)	$r_2 \circ \langle g_1, g_2 \rangle \circ f \equiv g_2 \circ f$	like (b_1)
(c)	$\langle g_1, g_2 \rangle \circ f \equiv \langle g_1 \circ f, g_2 \circ f \rangle$	(b_1), (b_2)
2.	When $X_1 = X_2$	
(d)	$(g_1 \times g_2) \circ \langle f_1, f_2 \rangle \equiv \langle g_1 \circ q_1 \circ \langle f_1, f_2 \rangle, g_2 \circ q_2 \circ \langle f_1, f_2 \rangle \rangle$	(1)
(e_1)	$q_1 \circ \langle f_1, f_2 \rangle \equiv f_1$	
(f_1)	$g_1 \circ q_1 \circ \langle f_1, f_2 \rangle \equiv g_1 \circ f_1$	*repl*
(f_2)	$g_2 \circ q_2 \circ \langle f_1, f_2 \rangle \equiv g_2 \circ f_2$	like (f_1)
(g)	$\langle g_1 \circ q_1 \circ \langle f_1, f_2 \rangle, g_2 \circ q_2 \circ \langle f_1, f_2 \rangle \rangle \equiv \langle g_1 \circ f_1, g_2 \circ f_2 \rangle$	(f_1), (f_2), prop. 2.6
(h)	$(g_1 \times g_2) \circ \langle f_1, f_2 \rangle \equiv \langle g_1 \circ f_1, g_2 \circ f_2 \rangle$	(d), (g), *trans*
3.	In all cases	
(k)	$(g_1 \times g_2) \circ \langle f_1 \circ p_1, f_2 \circ p_2 \rangle \equiv \langle g_1 \circ f_1 \circ p_1, g_2 \circ f_2 \circ p_2 \rangle$	(2)

□

Proof of proposition 3.7 (composition).
The three left handsides can be illustrated as follows:



1.	When $f_1 = v_2 (= v)$	
(b_1)	$r_1 \circ \langle g_1, w_2 \rangle \circ v \equiv g_1 \circ v$	like *basic*
(a_2)	$r_2 \circ \langle g_1, w_2 \rangle \lesssim w_2$	
(b_2)	$r_2 \circ \langle g_1, w_2 \rangle \circ v \lesssim w_2 \circ v$	(a_1), $subst_{\lesssim}$
(c)	$\langle g_1, w_2 \rangle \circ v \equiv \langle g_1 \circ v, w_2 \circ v \rangle$	(b_1), (b_2)
2.	When $X_1 = X_2$	
(d)	$(g_1 \times w_2) \circ \langle f_1, v_2 \rangle \equiv \langle g_1 \circ q_1 \circ \langle f_1, v_2 \rangle, w_2 \circ q_2 \circ \langle f_1, v_2 \rangle \rangle$	(1)
(f_1)	$g_1 \circ q_1 \circ \langle f_1, v_2 \rangle \equiv g_1 \circ f_1$	like *basic*
(e_2)	$q_2 \circ \langle f_1, v_2 \rangle \lesssim v_2$	
(f_2)	$w_2 \circ q_2 \circ \langle f_1, v_2 \rangle \lesssim w_2 \circ v_2$	$repl_{\lesssim}$ (w_2 is pure)
(g)	$\langle g_1 \circ q_1 \circ \langle f_1, v_2 \rangle, w_2 \circ q_2 \circ \langle f_1, v_2 \rangle \rangle \equiv \langle g_1 \circ f_1, w_2 \circ v_2 \rangle$	(f_1), (f_2), prop. 2.6
(h)	$(g_1 \times w_2) \circ \langle f_1, v_2 \rangle \equiv \langle g_1 \circ f_1, w_2 \circ v_2 \rangle$	(d), (g), $trans_{\equiv}$
3.	In all cases	
(k)	$(g_1 \times w_2) \circ \langle f_1 \circ p_1, v_2 \circ p_2 \rangle \equiv \langle g_1 \circ f_1 \circ p_1, w_2 \circ v_2 \circ p_2 \rangle$	(2)

□

Proof of proposition 2.8 (swap).
The two left handsides can be illustrated as follows:

$$
\begin{tikzcd}
 & Y_1 \arrow[r, "\mathrm{id}"] & Y_1 \\
X \arrow[ur, "f_1"] \arrow[r, "\equiv"] \arrow[dr, "f_2"'] & Y_2 \times Y_1 \arrow[u] \arrow[r, "\equiv"] \arrow[d, "\equiv"] & Y_1 \times Y_2 \arrow[u] \arrow[d] \\
 & Y_2 \arrow[r, "\mathrm{id}"'] & Y_2
\end{tikzcd}
\qquad
\begin{tikzcd}
X_1 \arrow[r, "\mathrm{id}"] & X_1 \arrow[r, "f_1"] & Y_1 \arrow[r, "\mathrm{id}"] & Y_1 \\
X_1 \times X_2 \arrow[u] \arrow[r, "\equiv"] \arrow[d] & X_2 \times X_1 \arrow[u] \arrow[r, "\equiv"] \arrow[d, "\equiv"] & Y_2 \times Y_1 \arrow[u] \arrow[r, "\equiv"] \arrow[d, "\equiv"] & Y_1 \times Y_2 \arrow[u] \arrow[d] \\
X_2 \arrow[r, "\mathrm{id}"'] & X_2 \arrow[r, "f_2"'] & Y_2 \arrow[r, "\mathrm{id}"'] & Y_2
\end{tikzcd}
$$

1.	When $X_1 = X_2$	
(a_1)	$q_1 \circ \gamma_Y \equiv q'_1$	
(b_1)	$q_1 \circ \gamma_Y \circ \langle f_2, f_1 \rangle \equiv q'_1 \circ \langle f_2, f_1 \rangle$	(a_1), *subst*
(c_1)	$q'_1 \circ \langle f_2, f_1 \rangle \equiv f_1$	
(d_1)	$q_1 \circ \gamma_Y \circ \langle f_2, f_1 \rangle \equiv f_1$	(b_1), (c_1), *trans*
(d_2)	$q_2 \circ \gamma_Y \circ \langle f_2, f_1 \rangle \equiv f_2$	like (d_1)
(e)	$\gamma_Y \circ \langle f_2, f_1 \rangle \equiv \langle f_1, f_2 \rangle$	(d_1), (d_2)
2.	In all cases	
(f)	$\langle f_2 \circ p'_2, f_1 \circ p'_1 \rangle \circ \gamma_X^{-1} \equiv \langle f_2 \circ p'_2 \circ \gamma_X^{-1}, f_1 \circ p'_1 \circ \gamma_X^{-1} \rangle$	prop. 2.7, *sym*
(g)	$\gamma_Y \circ \langle f_2 \circ p'_2, f_1 \circ p'_1 \rangle \circ \gamma_X^{-1} \equiv \gamma_Y \circ \langle f_2 \circ p'_2 \circ \gamma_X^{-1}, f_1 \circ p'_1 \circ \gamma_X^{-1} \rangle$	*repl*
(h)	$\gamma_Y \circ \langle f_2 \circ p'_2 \circ \gamma_X^{-1}, f_1 \circ p'_1 \circ \gamma_X^{-1} \rangle \equiv \langle f_1 \circ p'_1 \circ \gamma_X^{-1}, f_2 \circ p'_2 \circ \gamma_X^{-1} \rangle$	(1)
(i_1)	$p'_1 \circ \gamma_X^{-1} \equiv p_1$	
(j_1)	$f_1 \circ p'_1 \circ \gamma_X^{-1} \equiv f_1 \circ p_1$	(i_1), *repl*
(j_2)	$f_2 \circ p'_2 \circ \gamma_X^{-1} \equiv f_2 \circ p_2$	like (j_1)
(k)	$\langle f_1 \circ p'_1 \circ \gamma_X^{-1}, f_2 \circ p'_2 \circ \gamma_X^{-1} \rangle \equiv \langle f_1 \circ p_1, f_2 \circ p_2 \rangle$	(j_1), (j_2), prop. 2.6
(l)	$\gamma_Y \circ \langle f_2 \circ p'_2, f_1 \circ p'_1 \rangle \circ \gamma_X^{-1} \equiv \langle f_1 \circ p_1, f_2 \circ p_2 \rangle$	(g), (h), (k), *trans*

□

Proof of proposition 3.8 (swap).

The two left handsides can be illustrated as follows:

$$
\begin{tikzcd}
 & Y_1 \arrow[r, squiggly, "\mathrm{id}"] \arrow[d, squiggly, phantom, "\equiv"] & Y_1 \\
X \arrow[ur, "f_1"] \arrow[r, "\gtrsim"'] \arrow[dr, squiggly, "v_2"'] & Y_2 \times Y_1 \arrow[r, squiggly] \arrow[d, squiggly, phantom, "\equiv"] & Y_1 \times Y_2 \\
 & Y_2 \arrow[r, squiggly, "\mathrm{id}"'] & Y_2
\end{tikzcd}
\qquad
\begin{tikzcd}
X_1 \arrow[r, squiggly, "\mathrm{id}"] \arrow[d, squiggly, phantom, "\equiv"] & X_1 \arrow[r, "f_1"] \arrow[d, squiggly, phantom, "\equiv"] & Y_1 \arrow[r, squiggly, "\mathrm{id}"] \arrow[d, squiggly, phantom, "\equiv"] & Y_1 \\
X_1 \times X_2 \arrow[r, squiggly] \arrow[d, squiggly, phantom, "\equiv"] & X_2 \times X_1 \arrow[r] \arrow[d, squiggly, phantom, "\gtrsim"] & Y_2 \times Y_1 \arrow[r, squiggly] \arrow[d, squiggly, phantom, "\equiv"] & Y_1 \times Y_2 \\
X_2 \arrow[r, squiggly, "\mathrm{id}"'] & X_2 \arrow[r, squiggly, "v_2"'] & Y_2 \arrow[r, squiggly, "\mathrm{id}"'] & Y_2
\end{tikzcd}
$$

1.	When $X_1 = X_2$	
(d_1)	$q_1 \circ \gamma_Y \circ \langle v_2, f_1 \rangle \equiv f_1$	like *basic*
(a_2)	$q_2 \circ \gamma_Y \equiv q'_2$	
(b_2)	$q_2 \circ \gamma_Y \circ \langle v_2, f_1 \rangle \equiv q'_2 \circ \langle v_2, f_1 \rangle$	(a_2), $subst_{\equiv}$
(c_2)	$q'_2 \circ \langle v_2, f_1 \rangle \lesssim v_2$	
(d_2)	$q_2 \circ \gamma_Y \circ \langle v_2, f_1 \rangle \lesssim v_2$	(b_2), (c_2), *comp*
(e)	$\gamma_Y \circ \langle v_2, f_1 \rangle \equiv \langle f_1, v_2 \rangle$	(d_1), (d_2)
2.	In all cases	
(l)	$\gamma_Y \circ \langle f_2 \circ p'_2, f_1 \circ p'_1 \rangle \circ \gamma_X^{-1} \equiv \langle f_1 \circ p_1, f_2 \circ p_2 \rangle$	like *basic*

□