HAL
open science

# Diagrammatic Inference

Dominique Duval

**To cite this version:**

Dominique Duval. Diagrammatic Inference. 2009. hal-00177075v2

**HAL Id: hal-00177075**

**https://hal.science/hal-00177075v2**

Preprint submitted on 20 Nov 2009

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

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

# Diagrammatic Inference

Dominique Duval
LJK, Université de Grenoble, France, Dominique.Duval@imag.fr

November 20, 2009

**Abstract**

Diagrammatic logics were introduced in 2002, with emphasis on the notions of specifications and models. In this paper we improve the description of the inference process, which is seen as a Yoneda functor on a bicategory of fractions. A diagrammatic logic is defined from a morphism of limit sketches (called a propagator) which gives rise to an adjunction, which in turn determines a bicategory of fractions. The propagator, the adjunction and the bicategory provide respectively the syntax, the models and the inference process for the logic. Then diagrammatic logics and their morphisms are applied to the semantics of side effects in computer languages.

## 1 Introduction

The framework of *diagrammatic logics* was introduced in [Du03], after [DL02]. It relies on well-known categorical notions like Ehresmann's sketches [Eh68] and Gabriel and Zisman's categories of fractions [GZ67]. Diagrammatic logics have been influenced by Lair's "*trames*" [Lai87] and by Makkai's *sketch entailments* [Mak97]. They share many common features with other approaches of categorical logic, among which [Lam68, Law69a, Law69b, Mi75, Se79, BW94, Bj05, Gu07]. While categorical logic traditionally relies on viewing logical theories as categories, or sometimes as 2-categories, diagrammatic logic allows more general kinds of theories. This is motivated by applications to computer science, like the application to side-effects that is presented at the end of the paper.

Diagrammatic logics were introduced in order to deal with some unusal kinds of logics and with morphisms between logics. They can be used for proving properties of computational languages with effects, in a natural and powerful way, and for providing a notion of model for these languages, with guaranteed soundness properties [DR06, DDLR06, DDR07]. A diagrammatic logic $\mathcal{L}$ is defined from a special kind of *adjunction*, which itself comes from a special kind of *morphism of limit sketches*. The adjunction provides the *models* and the *inference process* for the logic $\mathcal{L}$, while the limit sketches and their morphism provide a *syntax* for $\mathcal{L}$. In this paper, the inference process for a diagrammatic logic is defined as a *Yoneda functor on a bicategory*. Bicategories, as introduced by Bénabou [Be67], did not

appear explicitly in [Du03]. In this paper, we show that they play a major role in clarifying the notion of diagrammatic inference.

We define a *logical adjunction* $F \dashv U : \mathbf{S} \rightleftarrows \mathbf{T}$ (where $F : \mathbf{S} \to \mathbf{T}$ and $U : \mathbf{T} \to \mathbf{S}$) as an adjunction where $\mathbf{S}$ is cocomplete and $U$ is full and faithful. Then $\mathbf{S}$ is called the category of *specifications* and $\mathbf{T}$ the category of *theories*. Typically, a specification $\Sigma$ is a collection of axioms and a theory $\Theta$ is a collection of theorems that is closed under inference. The *models* of $\Sigma$ with values in $\Theta$ are defined from the bijection $\mathbf{S}(\Sigma, U\Theta) \cong \mathbf{T}(F\Sigma, \Theta)$ (natural in $\Sigma$ and $\Theta$) in section 2, as in [Du03].

For a given logical adjunction $F \dashv U : \mathbf{S} \rightleftarrows \mathbf{T}$, the aim of inference is to determine, for a given specification $\Sigma$, some *generalized elements* of $F\Sigma$, which means, some morphisms in $\mathbf{T}$ with codomain $F\Sigma$. Let $\mathbf{S1}$ be the category of *classes of fractions* $\tau\backslash\sigma$ of $\mathbf{S}$, with numerator any morphism $\sigma$ in $\mathbf{S}$ and with denominator an *entailment* $\tau$, *i.e.*, a morphism in $\mathbf{S}$ such that $F\tau$ is invertible in $\mathbf{T}$. Then there is an equivalence $F_1 \sim U_1 : \mathbf{S1} \rightleftarrows \mathbf{T}$ with $F_1\Sigma = F\Sigma$, so that there is a bijection $\mathbf{S1}(U_1\Theta, \Sigma) \cong \mathbf{T}(\Theta, F\Sigma)$, natural in $\Theta$ (in $\mathbf{T}$) and $\Sigma$ (in $\mathbf{S1}$). The category $\mathbf{S1}$ of classes of fractions is obtained from the *bicategory of fractions* $\mathbf{S2}$. Let $C$ denote the *connectivity functor* from $\mathbf{Cat}$ to $\mathbf{Set}$, which identifies the connected objects in every category. Functors $U_2 : \mathbf{S2} \to \mathbf{T}$ and $F_2 : \mathbf{T} \to \mathbf{S2}$ are defined, with $F_2\Sigma = F\Sigma$, such that the previous bijection can be stated, more precisely, as $C(\mathbf{S2}(U_2\Theta, \Sigma)) \cong \mathbf{T}(\Theta, F\Sigma)$, naturally in $\Theta$ (in $\mathbf{T}$) and $\Sigma$ (in $\mathbf{S2}$). This is the key for defining the inference process: the *inference rules* are fractions, from which all *proofs* can be derived, an *inference step* is defined as a composition in $\mathbf{S2}$, so that the *inference process* can be seen as the Yoneda covariant functor on the bicategory $\mathbf{S2}$. It should be noted that the composition in $\mathbf{S2}$, for each inference step, requires a pushout in $\mathbf{S}$. Inference is studied, along these lines, in section 3.

A *limit sketch* $\mathbf{E}$ is made of a graph together with some potential identities, composites and limits, which turn $\mathbf{E}$ into a generator for a complete category. Here, as in [Du03], morphisms of limit sketches are called *propagators*. Each limit sketch $\mathbf{E}$ gives rise to a category $\mathbf{Real}(\mathbf{E})$ of realizations, or "loose models", which is cocomplete. It is known from [Eh68] that each propagator $P : \mathbf{E}_S \to \mathbf{E}_T$ determines an adjunction $F \dashv U : \mathbf{S} \rightleftarrows \mathbf{T}$, where $\mathbf{S} = \mathbf{Real}(\mathbf{E}_S)$ and $\mathbf{T} = \mathbf{Real}(\mathbf{E}_T)$. According to [Du03], any propagator $P$ can be modified in a reasonable way (reminded in theorem 4.5) in order to get $U$ full and faithful. Then $P$ is called a *logical propagator*, and $F \dashv U$ is a logical adjunction. A *diagrammatic logic* $\mathcal{L}$ is defined as an equivalence class of logical propagators. An *inference system* for $\mathcal{L}$ is a propagator in the class $\mathcal{L}$ which consists of adding inverses to some arrows, it provides the inference rules. This is studied in section 4, where in addition it is checked that an inference system may satisfy relevant *finiteness conditions* to be called a *syntax*.

The paper ends up, in section 5, with an application to side-effects in computer languages, which provides a categorical base for [DDR07].

Some familiarity with category theory is assumed: most of it can be found in [Mac98], and in [Le98] for bicategories. We use the category $\mathbf{Set}$ of sets, the 2-category $\mathbf{Cat}$ of categories, and so on, without mentioning the size issues. Sometimes we use the symbol "$\rightarrowtail\!\!\!\!\!\!\twoheadrightarrow$" for contravariant functors, so that $\mathbf{C} \rightarrowtail\!\!\!\!\!\!\twoheadrightarrow \mathbf{D}$ means either $\mathbf{C}^{\mathrm{op}} \to \mathbf{D}$ or $\mathbf{C} \to \mathbf{D}^{\mathrm{op}}$, and we

denote by $\mathbf{D}^{\mathbf{C},\mathrm{op}}$ the category of contravariant functors from $\mathbf{C}$ to $\mathbf{D}$. For an introduction to the theory of sketches ("*esquisses*", in french), see [CL84, CL88, BW99], and [We93] for additional references. There are many kinds of sketches (linear sketches, finite product sketches, limit sketches,...), which correspond to different kinds of logic. In addition, there are many variants for each kind of sketches, however the choice of some variant only matters for the technical details.

# 2 Models

## 2.1 Adjunctions

An *adjunction* is a pair of functors $F : \mathbf{S} \to \mathbf{T}$ (*the left adjoint*) and $U : \mathbf{T} \to \mathbf{S}$ (*the right adjoint*) together with a bijection, natural in $\Sigma$ (in $\mathbf{S}$) and $\Theta$ (in $\mathbf{T}$):

$$\mathbf{S}(\Sigma, U\Theta) \cong \mathbf{T}(F\Sigma, \Theta) \tag{1}$$

This is denoted $F \dashv U$, or more precisely $F \dashv U : \mathbf{S} \rightleftarrows \mathbf{T}$. An adjunction defines two natural transformations, the *unit* $\eta : \mathrm{Id}_{\mathbf{S}} \Rightarrow UF$ and the *counit* $\varepsilon : FU \Rightarrow \mathrm{Id}_{\mathbf{T}}$. When both $\eta$ and $\varepsilon$ are natural isomorphisms, the adjunction is called an *equivalence (of categories)*, which is denoted $F \sim U : \mathbf{S} \rightleftarrows \mathbf{T}$. A *morphism of adjunctions*, from $F \dashv U : \mathbf{S} \rightleftarrows \mathbf{T}$ to $F' \dashv U' : \mathbf{S}' \rightleftarrows \mathbf{T}'$, is a pair of adjunctions $F_S \dashv U_S : \mathbf{S} \rightleftarrows \mathbf{S}'$ and $F_T \dashv U_T : \mathbf{T} \rightleftarrows \mathbf{T}'$ such that $U \circ U_T = U_S \circ U'$, from which follows a natural isomorphism $F_T \circ F \cong F' \circ F_S$. A morphism made of two equivalences is an *equivalence of adjunctions*.

## 2.2 Logical adjunctions

**Definition 2.1** A *logical adjunction* is an adjunction $F \dashv U : \mathbf{S} \rightleftarrows \mathbf{T}$ such that the category $\mathbf{S}$ is cocomplete and the functor $U$ is full and faithful. For instance, a *full reflection* between cocomplete categories is a logical adjunction.

From now on in section 2, a logical adjunction $F \dashv U : \mathbf{S} \rightleftarrows \mathbf{T}$ is chosen.

**Definition 2.2** The category of *specifications* and the category of *theories* are $\mathbf{S}$ and $\mathbf{T}$, respectively. For each specification $\Sigma$ and theory $\Theta$, the set of *models* of $\Sigma$ with *values* in $\Theta$ is $\mathrm{Mod}(\Sigma, \Theta) = \mathbf{T}(F\Sigma, \Theta)$, so that $\mathrm{Mod}(\Sigma, \Theta) \cong \mathbf{S}(\Sigma, U\Theta)$.

This gives rise to the functor $\mathrm{Mod} : \mathbf{S} \times \mathbf{T}^{\mathrm{op}} \to \mathbf{Set}$. It may happen that this functor takes its values in $\mathbf{Cat}$. The next result is a direct consequence of adjunction, it will be used in section 5.

**Proposition 2.3** *Let us consider a morphism $\langle F_S \dashv U_S, F_T \dashv U_T \rangle$ from $F \dashv U$ to a logical adjunction $F' \dashv U' : \mathbf{S}' \rightleftarrows \mathbf{T}'$. Then there is a bijection, natural in $\Sigma$ (in $\mathbf{S}$) and $\Theta'$ (in $\mathbf{T}'$):*

$$\mathrm{Mod}_{F \dashv U}(\Sigma, U_T\Theta') \cong \mathrm{Mod}_{F' \dashv U'}(F_S\Sigma, \Theta') \ .$$

**Example 2.4 (Equational logic)** Let **Gr** denote the category of graphs and **FpCat** the category of categories with chosen **F**inite **P**roducts. The inclusion of **FpCat** in **Gr** gives rise to a reflection, that is not full. The finite product sketches are defined now, as a kind of intermediate notion between graphs and categories with chosen finite products. First, a *linear sketch* **E** is a graph where for some points $X$ there is a loop $\mathrm{id}_X : X \to X$ called *the (potential) identity of* $X$, for some consecutive arrows $f : X \to Y$, $g : Y \to Z$ there is an arrow $g \circ f : X \to Z$ called *the (potential) composite of* $f$ *and* $g$. Then, a *finite product sketch* **E** is a linear sketch where for some finite families of points $(X_1, \ldots, X_k)$ (with $k \geq 0$) there is a discrete cone $(p_j : \prod_{i=1}^{n} X_i \to X_j)_{1 \leq j \leq k}$ in **E** called *the (potential) product* of $X_1, \ldots, X_k$). No additional axiom has to be satisfied. A *morphism of finite product sketches* is a morphism of graphs which preserves all potential features. This yields the category **FpSk** of finite product sketches. The variants for this definition include: a potential identity for each point, and/or a potential composite for each pair of consecutive arrows, or "diagrams" instead of composites, and/or any number of potential products (often called *distinguished cones*) for each finite discrete base. A category with chosen finite products can be seen as a finite product sketch, with its chosen products as potential products; this inclusion of **FpCat** in **FpSk** gives rise to a full reflection. Since the category **FpSk** is cocomplete, we get a logical adjunction $F_{\mathrm{Fp}} \dashv U_{\mathrm{Fp}} : \mathbf{FpSk} \rightleftarrows \mathbf{FpCat}$. The category of sets, with some choice for the finite products of sets, defines a theory with respect to this logical adjunction. Every equational specification Spec can be seen as a finite product sketch $\Sigma$: the sorts, operations and equations become points, arrows and equalities of arrows, respectively [BW99]. The diagrammatic models of $\Sigma$ with values in the theory of sets can be identified with models of Spec, which can be called the "strict" models of Spec.

**Example 2.5 (Limit logic)** It is easy to generalize example 2.4 by replacing finite products with limits. The resulting logic will be called the *limit logic*. Let **LCat** denote the category of categories with chosen **L**imits. We define *limit sketches* as an intermediate notion between graphs and categories with chosen limits; they are similar to *projective sketches* [CL84]. A *limit sketch* **E** is a linear sketch where for some diagrams $b : \mathbf{J} \to \mathbf{E}$ there is a commutative cone $(p_J : \mathrm{Lim}(b) \to X_J)_J$ with base $b$ in **E** (*i.e.*, the $J$'s are the points of **J** and $b(j) \circ p_J = p_K$ for each arrow $j : J \to K$ in **J**), called *the (potential) limit of* $b$. No additional axiom has to be satisfied. A *morphism of limit sketches* is a morphism of graphs which preserves all potential features. This yields the category **LSk** of limit sketches. There exist also many variants for this definition. The inclusion of **LCat** in **Gr** gives rise to a reflection, that is not full. A category with chosen limits can be seen as a limit sketch, with its chosen limits as potential limits; this inclusion of **LCat** in **LSk** gives rise to a full reflection. Since the category **LSk** is cocomplete, we get a logical adjunction $F_{\mathrm{Lim}} \dashv U_{\mathrm{Lim}} : \mathbf{LSk} \rightleftarrows \mathbf{LCat}$. For every limit sketch $\Sigma$, a diagrammatic model of $\Sigma$ with values in the category of sets, with some choice for the limits of sets, maps the points of $\Sigma$ to sets, its arrows to functions, and its potential limits to the chosen limits: so, the diagrammatic models are "strict" models.

Examples 2.4 and 2.5 are easily generalized to other kinds of sketches and categories with structure. Quoting [We93], following Lawvere [Law69b] (in this paper we say "kind"

instead of "type"): *"Let $E$ be a type of sketch, determined by what sorts of cones and cocones are allowed in the sketch. [...] Corresponding to each type $E$ there is a type of category, required to have all limits, respectively colimits, of the type of cones, respectively cocones, allowed by $E$. Likewise, there is a type of functor, required to preserve that type of limits or colimits. [...] Given a type $E$, we will refer to $E$-sketches, $E$-categories and $E$-functors. Following Lawvere, we will refer to $E$ as a doctrine. [...] A doctrine can be a type of category requiring other structure besides limits and colimits (precisely, any type of category definable essentially algebraically over the category of categories)."* Moreover, in section 5, we define a logical adjunction such that its theories are not defined as "categories with structure".

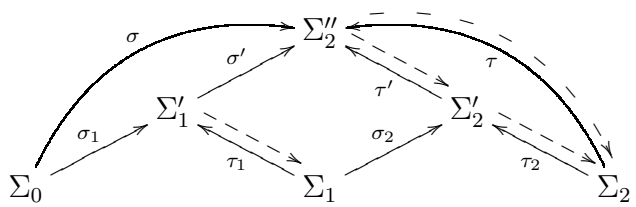
# 3 Inference

## 3.1 Fractions

This section relies on chapter 1 of [GZ67]. We insist on the fact that fractions form the objects of a *bicategory* [Be67, Le98]. Let $\mathbf{S}$ be a cocomplete category and $F : \mathbf{S} \to \mathbf{T}$ a functor preserving colimits, which is satisfied whenever there is a logical adjunction $F \dashv U : \mathbf{S} \rightleftarrows \mathbf{T}$.

A morphism $\tau : \Sigma \to \Sigma'$ in $\mathbf{S}$ is a *$F$-isomorphism* if $F\tau$ is an isomorphism in $\mathbf{T}$; this is denoted $\tau : \Sigma \overset{\leftarrow\!-\!-}{\to} \Sigma'$. A *(left $F$-)fraction* from $\Sigma$ to $\Sigma_1$ is a cospan $\tau\backslash\sigma : \Sigma \to \Sigma'_1 \overset{-\!-\!\to}{\leftarrow} \Sigma_1$ where $\sigma : \Sigma \to \Sigma'_1$ is any morphism in $\mathbf{S}$ and $\tau : \Sigma_1 \overset{\leftarrow\!-\!-}{\to} \Sigma'_1$ is a $F$-isomorphism. A *morphism of fractions* from $\Sigma$ to $\Sigma_1$, say $\alpha : \tau_1\backslash\sigma_1 \Rightarrow \tau_2\backslash\sigma_2$ where $\tau_1\backslash\sigma_1 : \Sigma \to \Sigma'_1 \overset{-\!-\!\to}{\leftarrow} \Sigma_1$ and $\tau_2\backslash\sigma_2 : \Sigma \to \Sigma'_2 \overset{-\!-\!\to}{\leftarrow} \Sigma_1$, is a morphism $\alpha : \Sigma'_1 \to \Sigma'_2$ in $\mathbf{S}$ such that $\alpha \circ \sigma_1 = \sigma_2$ and $\alpha \circ \tau_1 = \tau_2$. This last equality implies that $\alpha$ is a $F$-isomorphism. The fractions from $\Sigma$ to $\Sigma_1$ together with their morphisms form a category $\mathbf{S2}(\Sigma, \Sigma_1)$.

The *composite* of two consecutive fractions $\tau_1\backslash\sigma_1 : \Sigma_0 \to \Sigma'_1 \overset{-\!-\!\to}{\leftarrow} \Sigma_1$ and $\tau_2\backslash\sigma_2 : \Sigma_1 \to \Sigma'_2 \overset{-\!-\!\to}{\leftarrow} \Sigma_2$ is the fraction $(\tau_2\backslash\sigma_2) \circ (\tau_1\backslash\sigma_1) : \Sigma_0 \to \Sigma''_2 \overset{-\!-\!\to}{\leftarrow} \Sigma_2$ with numerator $\sigma = \sigma' \circ \sigma_1$ and denominator $\tau = \tau' \circ \tau_2$, where $\sigma'$ and $\tau'$ are obtained from the pushout of $\tau_1$ and $\sigma_2$. Since $F$ preserves colimits, the properties of pushouts prove that $\tau'$ is a $F$-isomorphism, so that $\tau$ also is a $F$-isomorphism:




Together with the identities id\id, this forms the *bicategory of fractions* $\mathbf{S2}$: it has the same objects as $\mathbf{S}$, the fractions as morphisms (or 1-cells), and the morphisms of fractions as 2-cells. Every morphism $\sigma : \Sigma \to \Sigma_1$ in $\mathbf{S}$ can be identified to the fraction $\mathrm{id}_{\Sigma_1}\backslash\sigma$, so that $\mathbf{S}$ is a wide subcategory of the bicategory $\mathbf{S2}$ (*wide* means they have the same objects); the inclusion functor is denoted $I : \mathbf{S} \to \mathbf{S2}$.

Let $b : \mathbf{J} \to \mathbf{S}$ be a diagram in $\mathbf{S}$. A *lax commutative cocone* in $\mathbf{S2}$ with base $b$ is a cocone $(\tau_J \backslash \sigma_J : b(J) \to \Sigma)_J$, for all objects $J$ in $\mathbf{J}$, together with a 2-cell $\alpha_j : \tau_J \backslash \sigma_J \Rightarrow \tau_K \backslash \sigma_K \circ b(j)$ for each arrow $j : J \to K$ in $\mathbf{J}$. A cocone $(h_J : b(J) \to \mathcal{H})_J$ in $\mathbf{S}$ with base $b$ is a *lax colimit* in $\mathbf{S2}$ if it is commutative and if for each lax commutative cocone $(\,(\tau_J \backslash \sigma_J)_J, (\alpha_j)_j\,)$ in $\mathbf{S2}$ with base $b$, there is a fraction $\tau \backslash \sigma : \mathcal{H} \to \Sigma$ with a 2-cell $\beta_J : \tau_J \backslash \sigma_J \Rightarrow \tau \backslash \sigma \circ h_J$ for each object $J$ in $\mathbf{J}$ such that $\beta_K \circ \alpha_j = \beta_J$ for each arrow $j : J \to K$ in $\mathbf{J}$, and such a $\tau \backslash \sigma$ is unique up to an in vertible 2-cell. Then $\tau \backslash \sigma$ is called "the" *lax cotuple* of the $\tau_J \backslash \sigma_J$'s. This is illustrated below:

$$\begin{tikzcd}
 & & & \mathcal{H} \arrow[dd, "\sigma"] \\
 & b(J) \arrow[rr, "b(j)"'] \arrow[urrr, "h_J"] \arrow[d, "\sigma_J"'] & & b(K) \arrow[ur, "h_K"'] \arrow[d, "\sigma_K"] & \\
 & \Sigma'_J \arrow[rr, "\alpha_j"] \arrow[urrr, "\beta_J"] & & \Sigma'_K \arrow[ur, "\beta_K"'] & \Sigma' \\
\Sigma \arrow[ur, "\tau_J"] \arrow[urrr, "\tau_K"'] \arrow[urrrrr, bend right, "\tau"'] & & & &
\end{tikzcd}$$

It can be proved by diagram-chasing that the inclusion functor $I : \mathbf{S} \to \mathbf{S2}$. maps colimits in $\mathbf{S}$ to lax colimits in $\mathbf{S2}$.

In any category $\mathbf{A}$, two objects $X$ and $X'$ are *connected* if they are related by a chain of morphisms. Since every functor preserves the connectivity relation, the *connectivity functor* $C : \mathbf{Cat} \to \mathbf{Set}$ maps each category to its set of connected components and each functor to the induced map.

The bicategory $\mathbf{S2}$ gives rise to the *category of classes of fractions* $\mathbf{S1}$ by identifying the connected fractions. The *quotient functor* $Q : \mathbf{S2} \to \mathbf{S1}$ is the identity on objects and maps each morphism to its connectivity class. In addition, it maps lax colimits to colimits. Given two specifications $\mathcal{H}$ and $\Sigma$, the set of classes of fractions from $\mathcal{H}$ to $\Sigma$ can be seen from two points of view, since there is a bijection, natural in $\mathcal{H}$ and $\Sigma$ (both in $\mathbf{S2}$):

$$C(\mathbf{S2}(\mathcal{H}, \Sigma)) \cong \mathbf{S1}(Q\mathcal{H}, Q\Sigma) \tag{2}$$

The *localization* of a category $\mathbf{A}$ with respect to a set of morphisms $M$ of $\mathbf{A}$ is the functor $L_M : \mathbf{A} \to \mathbf{A}[M^{-1}]$ that consists of adding inverses to the morphisms in $M$ [GZ67]. It is easy to check that the functor $L = Q \circ I : \mathbf{S} \to \mathbf{S1}$ is the localization of $\mathbf{S}$ with respect to the set of $F$-isomorphisms. In addition, $L$ preserves colimits.

**Remark 3.1** Let us emphasize that the (bi)categories $\mathbf{S}$, $\mathbf{S2}$, $\mathbf{S1}$ have the same objects (the specifications), which are preserved by the functors $I$, $Q$, $L$. In addition, the composition of morphisms in $\mathbf{S1}$ is obtained from the composition of morphisms in $\mathbf{S2}$, which itself makes use of composition and pushouts in $\mathbf{S}$.

If $\tau_1 \backslash \sigma_1$ and $\tau_2 \backslash \sigma_2$ are connected, then clearly $F\tau_1^{-1} \circ F\sigma_1 = F\tau_2^{-1} \circ F\sigma_2$. So, a functor $F_1 : \mathbf{S1} \to \mathbf{T}$ is defined by $F_1(\Sigma) = F(\Sigma)$ for each object $\Sigma$ in $\mathbf{S}$ and $F_1(Q(\tau \backslash \sigma)) = F\tau^{-1} \circ F\sigma$

for each fraction $\tau\backslash\sigma$. It is such that $F_1 \circ L = F$. The functor $U_1 : \mathbf{T} \to \mathbf{S1}$ is defined as $U_1 = L \circ U$. The next result comes from [GZ67].

**Theorem 3.2** *Let $F \dashv U : \mathbf{S} \rightleftarrows \mathbf{T}$ be an adjunction where the category $\mathbf{S}$ is cocomplete. Then $U$ is full and faithful if and only if $F_1$ and $U_1$ form an equivalence $F_1 \sim U_1 : \mathbf{S1} \rightleftarrows \mathbf{T}$.*

Hence, every logical adjunction $F \dashv U : \mathbf{S} \rightleftarrows \mathbf{T}$ gives rise to an equivalence $F_1 \sim U_1 : \mathbf{S1} \rightleftarrows \mathbf{T}$. It follows that every theory $\Theta$ is isomorphic to $F_1\mathcal{H}$ for some specification $\mathcal{H}$, and that there is a bijection, natural in $\Sigma$ and $\mathcal{H}$ (both in $\mathbf{S1}$):

$$\mathbf{S1}(\mathcal{H}, \Sigma) \cong \mathbf{T}(F_1\mathcal{H}, F_1\Sigma) \tag{3}$$

The functors $F_2 : \mathbf{S2} \to \mathbf{T}$ and $U_2 : \mathbf{T} \to \mathbf{S2}$ are now defined by $F_2 = F_1 \circ Q$ and $U_2 = I \circ U$, so that $Q \circ U_2 = L \circ U = U_1$. From the natural bijections (2) and (3), we get the following bijection, natural in $\mathcal{H}$ and $\Sigma$ (both in $\mathbf{S2}$):

$$C(\mathbf{S2}(\mathcal{H}, \Sigma)) \cong \mathbf{T}(F_2\mathcal{H}, QF_2\Sigma) \tag{4}$$

## 3.2 Inference steps

Given a logical adjunction $F \dashv U : \mathbf{S} \rightleftarrows \mathbf{T}$, we have defined the bicategory $\mathbf{S2}$ of fractions, with an inclusion functor $I : \mathbf{S} \to \mathbf{S2}$ and a quotient functor $Q : \mathbf{S2} \to \mathbf{S1}$, such that the localization of $\mathbf{S}$ with respect to the $F$-isomorphisms is $L = Q \circ I : \mathbf{S} \to \mathbf{S1}$.

**Definition 3.3** An *entailment* is a $F$-isomorphism $\tau : \Sigma \overset{\leftarrow}{\to} \Sigma'$, and an *instance of* $\Sigma$ *in* $\Sigma_1$ is a fraction $\tau\backslash\sigma : \Sigma \to \Sigma_1' \overset{\to}{\leftarrow} \Sigma_1$ (also written $\tau\backslash\sigma : \Sigma \to \Sigma_1$, in $\mathbf{S2}$). Let us consider a fraction $\rho : \mathcal{C} \to \mathcal{H}' \overset{\to}{\leftarrow} \mathcal{H}$, called an *inference rule* with *hypothesis* $\mathcal{H}$ and *conclusion* $\mathcal{C}$. Given a specification $\Sigma$, the *inference step in* $\Sigma$ *along* $\rho$ is the functor $\mathbf{S2}(\rho, \Sigma) : \mathbf{S2}(\mathcal{H}, \Sigma) \to \mathbf{S2}(\mathcal{C}, \Sigma)$ of composition on the right with $\rho$.

**Remark 3.4** Each entailment $\tau$ gives rise to an isomorphism $F\tau$ and to a bijection $\mathrm{Mod}(\tau, \Theta)$ for each theory $\Theta$, which proves the *soundness* of the logical adjunction. Each instance $\tau\backslash\sigma$ of $\Sigma$ in $\Sigma'$ gives rise to a morphism $F(\tau\backslash\sigma) : F\Sigma \to F\Sigma'$ and to a function $\mathrm{Mod}(\tau\backslash\sigma, \Theta) : \mathrm{Mod}(\Sigma', \Theta) \to \mathrm{Mod}(\Sigma, \Theta)$ for each $\Theta$.

The inference step in $\Sigma$ along $\rho$ maps each instance $\kappa$ of $\mathcal{H}$ in $\Sigma$ to the instance $\gamma = \kappa \circ \rho$ of $\mathcal{C}$ in $\Sigma$. The composition is performed in $\mathbf{S2}$, which means that it requires a pushout in $\mathbf{S}$. This is illustrated below. We have chosen an illustration that is different from the illustration of composition of fractions in section 3.1, because it better reflects the semantics of inference: the top line is made of the rule, with its hypothesis on the left, the bottom line is made of the given specification $\Sigma$ and its entailments; the square on the left is a pushout, and the diagram is commutative. The numerator and denominator of $\rho$ are denoted $\sigma\rho$ and

$\tau\rho$, and similarly for $\kappa$ and $\gamma$.

in **S2** :

$$\begin{tikzcd} \mathcal{H} \arrow[d, "\kappa"'] & \mathcal{C} \arrow[l, "\rho"'] \arrow[d, "\gamma"] \\ \Sigma \arrow[r, equal] & \Sigma \end{tikzcd}$$

in **S** :

$$\begin{tikzcd} & \mathcal{H} \arrow[r, "\tau\rho"'] \arrow[d, "\sigma\kappa"'] & \mathcal{H}' \arrow[l, dashed, bend right] \arrow[d] & \mathcal{C} \arrow[l, "\sigma\rho"'] \arrow[d, "\sigma\gamma"] \\ \Sigma \arrow[r, "\tau\kappa"'] \arrow[rrr, bend right, dashed, "\tau\gamma"'] & \Sigma_\kappa \arrow[l, dashed, bend right] \arrow[r] & \Sigma_\gamma \arrow[l, dashed, bend right] \arrow[r, equal] & \Sigma_\gamma \arrow[lll, dashed, bend right] \end{tikzcd}$$

**Example 3.5 (Modus ponens)** Let us consider the logical adjunction for which a specification is a pair $\Sigma = \langle \Sigma_F, \Sigma_P \rangle$ made of a set $\Sigma_F$ of *formulas* with a partial binary operator "$\Rightarrow$" and a subset $\Sigma_P \subseteq \Sigma_F$ of *provable formulas*, and a theory $\Theta$ is a specification that satisfies two properties: when $A$ and $B$ are formulas then $A \Rightarrow B$ is a formula, and when $A \Rightarrow B$ and $A$ are provable then $B$ also is provable. The second property corresponds to the *modus ponens* inference rule $\mathcal{C}_P \to \mathcal{H}'_P \overset{\dashrightarrow}{\leftarrow} \mathcal{H}_P$ for building provable formulas, where $\mathcal{H}_P = \langle \{A, B, A \Rightarrow B\}, \{A, A \Rightarrow B\} \rangle$, $\mathcal{H}'_P = \langle \{A, B, A \Rightarrow B\}, \{A, A \Rightarrow B, B\} \rangle$, $\mathcal{C}_P = \langle \{C\}, \{C\} \rangle$, the entailment $\mathcal{H}_P \overset{\dashleftarrow}{\to} \mathcal{H}'_P$ is the inclusion and the morphism $\mathcal{C}_P \to \mathcal{H}'_P$ maps $C$ to $B$. Classically, only the provable formulas of $\mathcal{H}_P$ and of the image of $\mathcal{C}_P$ in $\mathcal{H}'_P$ are mentioned, and the modus ponens rule is written $\frac{A \quad A \Rightarrow B}{B}$. The hypothesis $\mathcal{H}_P = \langle \{A, B, A \Rightarrow B\}, \{A, A \Rightarrow B\} \rangle$ contains two provable formulas. It is made of two simpler hypothesis $\mathcal{H}_1 = \langle \{A\}, \{A\} \rangle$ and $\mathcal{H}_2 = \langle \{A, B, A \Rightarrow B\}, \{A \Rightarrow B\} \rangle$. More precisely, $\mathcal{H}_P$ is the colimit of the diagram $\mathcal{H}_1 \leftarrow \mathcal{H}_0 \to \mathcal{H}_2$, where $\mathcal{H}_0 = \langle \{A\}, \emptyset \rangle$ and the morphisms are the inclusions. So, each instance of $\mathcal{H}_P$ can be built as a lax cotuple of instances.

## 3.3 Inference process

**Definition 3.6** With respect to a set $\mathcal{R}$ of inference rules, a *proof* is a fraction in the sub cocomplete bicategory of **S2** which is generated by $\mathcal{R}$.

In this section, for simplicity, it is assumed that the proofs are all the fractions; this assumption is discussed in section 4. The inference process is defined now by allowing every ingredient (either $\mathcal{H}$, $\rho$, $\Sigma$, or $\kappa$) of the inference step to vary in the relevant (bi)category. When **C** and **D** are bicategories, we define a *functor* from **C** to **D** as a homomorphism in [Le98]. Essentially, a functor $G : \mathbf{C} \to \mathbf{D}$ maps objets to objects, morphisms to morphisms, 2-cells to 2-cells, it preserves composites of 2-cells, and it preserves composites of morphisms only up to an invertible 2-cell. A *contravariant functor* $G : \mathbf{C} \to\!\!\!\!\!\times\, \mathbf{D}$ is contravariant on morphisms and covariant on 2-cells.

The inference steps in $\Sigma$ can be composed, in two slightly different ways. Let $\rho_1 : \mathcal{C}_1 \to \mathcal{H}_1$ and $\rho_2 : \mathcal{C}_2 \to \mathcal{H}_2$ with $\mathcal{C}_1 = \mathcal{H}_2$ in **S2**, and let $\kappa$ be an instance of $\mathcal{H}_1$ in $\Sigma$. On the one hand, $\kappa$ is mapped by $\mathbf{S2}(\rho_2, \Sigma) \circ \mathbf{S2}(\rho_1, \Sigma)$ to $(\kappa \circ \rho_1) \circ \rho_2$. On the other hand, $\kappa$ is mapped by $\mathbf{S2}(\rho_1 \circ \rho_2, \Sigma)$ to $\kappa \circ (\rho_1 \circ \rho_2)$. These are two instances of $\mathcal{C}_2$ in $\Sigma$ related by an invertible 2-cell. The following definition corresponds to the second point of view.

**Definition 3.7** The *inference process in* $\Sigma$ is the contravariant functor $\mathbf{S2}(-,\Sigma) : \mathbf{S2} \longrightarrow \times \ \mathbf{Cat}$.

The contravariant functor $\mathbf{S2}(-,\Sigma)$ maps each specification $\mathcal{H}$ to the category $\mathbf{S2}(\mathcal{H},\Sigma)$ and each proof $\rho : \mathcal{C} \to \mathcal{H}$ to the functor $\mathbf{S2}(\rho,\Sigma)$ of composition on the right with $\rho$, as for primitive inference rules: this means that a proof is seen as a *derived inference rule* $\rho$.

Now $\Sigma$ itself may vary. Let $\mathbf{Cat}^{\mathbf{S2},\mathrm{op}}$ denote the 2-category of contravariant functors from $\mathbf{S2}$ to $\mathbf{Cat}$ and $\mathcal{Y}'_{\mathbf{S2}} : \mathbf{S2} \longrightarrow \mathbf{Cat}^{\mathbf{S2},\mathrm{op}}$ the Yoneda covariant functor of $\mathbf{S2}$, which maps each specification $\Sigma$ to the contravariant functor $\mathcal{Y}'\Sigma = \mathbf{S2}(-,\Sigma)$ [Le98].

**Definition 3.8** The *inference process* is the Yoneda functor $\mathcal{Y}'_{\mathbf{S2}} : \mathbf{S2} \to \mathbf{Cat}^{\mathbf{S2},\mathrm{op}}$.

# 4 Syntax

## 4.1 Propagators

In order to define a *syntax* for some logical adjunctions, we use limit sketches. Among all kinds of sketches, the limit sketches play a very special role in this paper, since they are used to define the "logic for logics", *i.e.*, the *meta logic* for defining all diagrammatic logics, as explained below. As in example 2.5, a *limit sketch* $\mathbf{E}$ is a graph where some points $X$ have a *(potential) identity* $\mathrm{id}_X : X \to X$, some consecutive arrows $f : X \to Y$, $g : Y \to Z$ have a *(potential) composite* $g \circ f : X \to Z$, and some diagrams $b : \mathbf{J} \to \mathbf{E}$ have a *(potential) limit* $(p_J : \mathrm{Lim}(b) \to X_J)_J$ with base $b$. A morphism of limit sketches is a morphism of graphs which preserves all potential features, which gives rise to the category $\mathbf{LSk}$ of limit sketches.

**Definition 4.1** A *realization* of a limit sketch $\mathbf{E}$ in any category $\mathbf{C}$ is a morphism of graphs which maps the potential features in $\mathbf{E}$ to actual features in $\mathbf{C}$. A *morphism of realizations of* $\mathbf{E}$ *in* $\mathbf{C}$ is a natural transformation. This yields the category $\mathbf{Real}(\mathbf{E},\mathbf{C})$ of realizations of $\mathbf{E}$ in $\mathbf{C}$.

The realizations of $\mathbf{E}$ could be called its "loose models". The category $\mathbf{Real}(\mathbf{E}) = \mathbf{Real}(\mathbf{E},\mathbf{Set})$ is cocomplete. When a category $\mathbf{A}$ is equivalent to $\mathbf{Real}(\mathbf{E})$, we say that $\mathbf{E}$ is *a sketch for* $\mathbf{A}$.

**Definition 4.2** A *propagator* $P : \mathbf{E}_S \to \mathbf{E}_T$ is a morphism of limit sketches. A *morphism of propagators* $\ell : P \to P'$, where $P : \mathbf{E}_S \to \mathbf{E}_T$ and $P' : \mathbf{E}'_S \to \mathbf{E}'_T$, is made of two propagators $\ell_S : \mathbf{E}_S \to \mathbf{E}'_S$ and $\ell_T : \mathbf{E}_T \to \mathbf{E}'_T$ such that $\ell_T \circ P = P' \circ \ell_S$. This yields the category of propagators.

For each propagator $P : \mathbf{E}_S \to \mathbf{E}_T$, the *underlying functor* $U_P = \mathbf{Real}(P) : \mathbf{Real}(\mathbf{E}_T) \to \mathbf{Real}(\mathbf{E}_S)$ is neither full nor faithful, in general. A fundamental result about limit sketches [Eh68] is that this underlying functor has a left adjoint $F_P : \mathbf{Real}(\mathbf{E}_S) \to \mathbf{Real}(\mathbf{E}_T)$, called the *freely generating functor*. So, every propagator gives rise to an adjunction $F_P \dashv U_P : \mathbf{Real}(\mathbf{E}_S) \rightleftarrows \mathbf{Real}(\mathbf{E}_T)$ where $\mathbf{Real}(\mathbf{E}_S)$ is cocomplete.

**Definition 4.3** A propagator $P$ is an *equivalence of limit sketches* when the adjunction $F_P \dashv U_P$ is an equivalence of categories. A morphism of propagators $\ell : P \to P'$ is an *equivalence* when both $\ell_S$ and $\ell_T$ are equivalences of limit sketches.

The Yoneda contravariant functor can be generalized to limit sketches, as follows [LD01, Du03]. Let $\mathbf{E}$ be a limit sketch and $\mathbf{Proto}(\mathbf{E})$ its *prototype*, *i.e.*, the category generated by $\mathbf{E}$ in such a way that all potential features in $\mathbf{E}$ become actual features in $\mathbf{Proto}(\mathbf{E})$. The Yoneda contravariant functor $\mathcal{Y}_{\mathbf{Proto}(\mathbf{E})}$ is such that its restriction to $\mathbf{E}$ forms a contravariant realization of $\mathbf{E}$ with values in $\mathbf{Real}(\mathbf{E})$. This is the *Yoneda contravariant realization of* $\mathbf{E}$, denoted $\mathcal{Y}_{\mathbf{E}}$. The *density property* of $\mathcal{Y}_{\mathbf{E}}$ states that every realization $\Sigma$ of $\mathbf{E}$ is the colimit of a diagram in the image of $\mathcal{Y}_{\mathbf{E}}$. When $P : \mathbf{E}_S \to \mathbf{E}_T$ is a propagator, the Yoneda contravariant realizations $\mathcal{Y}_S$ and $\mathcal{Y}_T$ of $\mathbf{E}_S$ and $\mathbf{E}_T$, respectively, are such that there is a natural isomorphism $F_P \circ \mathcal{Y}_S \cong \mathcal{Y}_T \circ P$.

## 4.2 Logical propagators

**Definition 4.4** A *logical propagator* is a propagator $P : \mathbf{E}_S \to \mathbf{E}_T$ such that the underlying functor $U_P$ is full and faithful.

Since the category $\mathbf{Real}(\mathbf{E}_S)$ is always cocomplete, every logical propagator $P : \mathbf{E}_S \to \mathbf{E}_T$ gives rise to a logical adjunction $F_P \dashv U_P : \mathbf{Real}(\mathbf{E}_S) \rightleftarrows \mathbf{Real}(\mathbf{E}_T)$.

For instance, for each set of arrows $A$ of $\mathbf{E}$, the *localizer* of $\mathbf{E}$ with respect to $A$ is the propagator with source $\mathbf{E}$ which, for each $a : E' \to E$ in $A$, adds an arrow $a^{-1} : E \to E'$ and the composites $a \circ a^{-1} = \mathrm{id}_E$ and $a^{-1} \circ a = \mathrm{id}_{E'}$. It is is easy to check that when $P$ is a localizer then it is a logical propagator and the functor $F_P$ is a localization. In addition, a morphism of propagators $\ell : P \to P'$ where $P$ is logical is characterized by $\ell_S : \mathbf{E}_S \to \mathbf{E}'_S$ such that each $P$-entailment in $\mathbf{E}_S$ is mapped to a $P'$-entailment in $\mathbf{E}'_S$.

As an instance of a (generally) non-logical propagator, let us say that a propagator $P : \mathbf{E} \to \mathbf{E}'$ is a *swelling propagator* if it is an inclusion and if every arrow of $\mathbf{E}'$ with its source in $\mathbf{E}$ is an arrow in $\mathbf{E}$, every composition of $\mathbf{E}'$ with its source in $\mathbf{E}$ is a composition in $\mathbf{E}$, every limit of $\mathbf{E}'$ with its vertex in $\mathbf{E}$ is a limit in $\mathbf{E}$, and every other limit of $\mathbf{E}'$ has at least one projection entirely outside $\mathbf{E}$ (which means that the vertex of the limit, at least one of its projection, and the target of this projection, are outside $\mathbf{E}$). Then the freely generating functor $F_P$ consists of "adding nothing", in the following sense: let $\Sigma$ be a realization of $\mathbf{E}$, then $F_P\Sigma$ is such that its restriction to $\mathbf{E}$ coincides with $\Sigma$ and $F_P\Sigma(E') = \emptyset$ for each point $E'$ not in $\mathbf{E}$, so that $F_P\Sigma(e')$ is the unique map with source $\emptyset$ for each arrow $e'$ not in $\mathbf{E}$. It is clear that $F_P$ is full and faithful.

From now on, for simplicity, it is assumed that each limit sketch has "enough" identities: either there is an identity for each point, or at least there is an identity whenever we need it (adding identities is an equivalence). The next result is the *decomposition theorem* from [Du03]. Such a decomposition is not uniquely determined.

**Theorem 4.5** *For each propagator $P : \mathbf{E}_S \to \mathbf{E}_T$ there is a swelling propagator $P_S$ and a logical propagator $P'$ such that $P = P' \circ P_S$. In addition, there is such a decomposition where $P'$ is composed of a localizer followed by an equivalence.*

As a typical example, here is a decomposition of a propagator $P$ which adds an arrow between two given points: $P_S$ is the inclusion and $P'$ maps $t_H$ and $t_C$ to $\mathrm{id}_H$ and $\mathrm{id}_C$, respectively. So, $P'$ is composed of the localizer with respect to $\{t_H, t_C\}$ followed by the equivalence that maps $t_H$, $t_C$ and their inverses to identities.

$$\begin{tikzcd}
H \arrow[rr, phantom, "\quad"] & & C \\
\end{tikzcd}
\xrightarrow{P_S}
\begin{tikzcd}
H' \arrow[r] \arrow[d, "t_H"] & C' \arrow[d, "t_C"] \\
H & C
\end{tikzcd}
\xrightarrow{P'}
\begin{tikzcd}
H \arrow[loop above, "\mathrm{id}_H"] \arrow[r] & C \arrow[loop above, "\mathrm{id}_C"]
\end{tikzcd}$$

## 4.3 Diagrammatic logics

Clearly, a propagator that is equivalent to a logical propagator is also logical.

**Definition 4.6** A *diagrammatic logic* $\mathcal{L}$ is an equivalence class of logical propagators. Hence the morphisms of diagrammatic logics are defined from the morphisms of propagators, which yields the *category* **DiaLog** of diagrammatic logics. An *inference system* for a diagrammatic logic $\mathcal{L}$ is a localizer in the class $\mathcal{L}$.

Theorem 4.5 provides a diagrammatic logic and an inference system for this logic, from any propagator. Let $\mathcal{L}$ be a diagrammatic logic, and $P$ a chosen logical propagator in the class $\mathcal{L}$. The following notions are defined with respect to $\mathcal{L}$ and $P$.

**Definition 4.7** The *specifications*, *theories* and *models* are the specifications, theories and models with respect to the logical adjunction $F_P \dashv U_P$. When in addition $P$ is an inference system for $\mathcal{L}$, a *syntactic inference rule* is a (right) fraction $r = s/t : H \overset{\dashrightarrow}{\leftarrow} H' \to C$ where $s : H' \to C$ is any arrow in $\mathbf{E}_S$ and $t : H' \overset{\dashleftarrow}{\to} H$ is an arrow in $\mathbf{E}_S$ such that $P(t)$ is invertible in $\mathbf{E}_T$.

So, the image of a syntactic inference rule by the Yoneda contravariant functor of $\mathbf{E}_S$ is an instance with respect to the logical adjunction $F_P \dashv U_P$, in coherence with definition 3.3.

The *type* $\mathbf{Type}(\mathbf{E})$ of a limit sketch $\mathbf{E}$ is the complete category generated by $\mathbf{E}$ in such a way that all potential features in $\mathbf{E}$ become actual features in $\mathbf{Type}(\mathbf{E})$. Each propagator $P : \mathbf{E}_S \to \mathbf{E}_T$ gives rise to a limit-preserving functor $\mathbf{Type}(P) : \mathbf{Type}(\mathbf{E}_S) \to \mathbf{Type}(\mathbf{E}_T)$. A *syntactic proof* is defined as a syntactic inference rule, with respect to $\mathbf{Type}(P)$ instead of $P$. So, the image of a syntactic proof by the Yoneda contravariant functor is a proof with respect to the logical adjunction $F_P \dashv U_P$, as in definition 3.6. The density property of the Yoneda contravariant functor shows that every proof with respect to $F_P \dashv U_P$ is isomorphic to the image of a syntactic proof, which justifies the assumption that "the proofs are all the fractions" in section 3.3.

**Remark 4.8** Our definition of diagrammatic logics and their deduction processes provides a new point of view about the notion of inference process in a specification $\Sigma$ (definition 3.7), which now can be seen as *a realization of* $\mathbf{E}_S$ *with values in* $\mathbf{Cat}$. Indeed, the Yoneda contravariant realization $\mathcal{Y}_S$ of $\mathbf{E}_S$ takes its values in the category $\mathbf{S} = \mathbf{Real}(\mathbf{E}_S)$, so that it can be composed with the inference process in $\Sigma$, *i.e.*, with the contravariant functor $\mathbf{S2}(-,\Sigma) : \mathbf{S2} \nrightarrow \mathbf{Cat}$, which maps colimits to limits. This gives rise to the realization $\mathbf{S2}(-,\Sigma) \circ \mathcal{Y}_S$ of $\mathbf{E}_S$ with values in $\mathbf{Cat}$.

**Example 4.9 (Equational logic: syntax)** As reminded in example 2.4, the equational logic can be defined from the full reflection of $\mathbf{FpSk}$ in $\mathbf{FpCat}$. This logical adjunction comes from a logical propagator $P_{\mathrm{Eq}} : \mathbf{E}_{\mathrm{FpSk}} \to \mathbf{E}_{\mathrm{FpCat}}$, that defines the *diagrammatic equational logic* $\mathcal{L}_{\mathrm{Eq}}$. This propagator can be obtained from a propagator $P_{\mathrm{FpCat}} : \mathbf{E}_{\mathrm{Gr}} \to \mathbf{E}_{\mathrm{FpCat}}$ for the reflection of $\mathbf{Gr}$ in $\mathbf{FpCat}$, by a decomposition satisfying the properties of theorem 4.5. This is also the case for the limit logic in example 2.5 and for other kinds of doctrines.

## 4.4 Finiteness issues

No finiteness condition has been assumed until now. However, a syntax is used for writing things down with a finite number of symbols. So that we have to check some finiteness properties, in order to ensure that an inference system for a diagrammatic logic does define a syntax. This issue is outlined now.

Let $P : \mathbf{E}_S \to \mathbf{E}_T$ be an inference system, and let us assume that the sketches $\mathbf{E}_S$ and $\mathbf{E}_T$ are finite (the categories $\mathbf{S} = \mathbf{Real}(\mathbf{E}_S)$ and $\mathbf{T} = \mathbf{Real}(\mathbf{E}_T)$ are usually infinite). Then, each specification $\Sigma$ is defined from the finite number of sets $\Sigma(E)$, for all points $E$ of $\mathbf{E}_S$, and there is a finite number of elementary inference rules.

In addition, let us assume that $\Sigma$ is finite, in the sense that each set $\Sigma(E)$ is finite (the generated theory $F\Sigma$ is usually infinite). The inference process builds new specifications $\Sigma'$, which are entailed from $\Sigma$. It has to be checked that these specifications also can be assumed finite. Each $\Sigma'$ is built as the vertex of a pushout in $\mathbf{S}$, so that is is finite as soon as *the finite colimits in* $\mathbf{S}$ *preserve finiteness*, which means that the colimit of a finite base made of finite specifications is a finite specification. This need not be true, in general. This issue can be solved thanks to an assumption about the *acyclicity* of sketches, as explained below. This assumption is presumably rather strong.

Usually a *cycle* in a graph is a loop, distinct from an identity, in the generated category. Let us define a *cycle* in a limit sketch as loop, distinct from an identity, in the generated type. When a limit sketch $\mathbf{E}$ is acyclic, then the finite colimits in $\mathbf{Real}(\mathbf{E})$ preserve finiteness. Let us assume that $P : \mathbf{E}_S \to \mathbf{E}_T$ is built thanks to theorem 4.5, from a propagator $P_0 : \mathbf{E}_{S,0} \to \mathbf{E}_{T,0}$, with $\mathbf{E}_{S,0}$ acyclic (like for instance $\mathbf{E}_{\mathrm{Gr}}$). Then the construction in the proof of theorem 4.5 can be modified in such a way that $\mathbf{E}_S$ also is acyclic: basically, every arrow $f : X \to Y$ in $\mathbf{E}_{T,0}$ gives rise to a span $X \xleftarrow{c} X' \xrightarrow{f'} Y$ in $\mathbf{E}_S$, which is mapped to $X \xleftarrow{\mathrm{id}} X \xrightarrow{f} Y$ in $\mathbf{E}_T$. The idea is that $c$ stands for an injection and $f'$ for a partial version of $f$ (it can be added that $c$ is a potential monomorphism). Then the finite colimits in $\mathbf{S}$ preserve finiteness, which is the required property.

# 5 Applications

## 5.1 Decorations

The notion of *decoration*, as defined below, can be used for studying the semantics of computer languages. For instance, for dealing with multivariate functions in imperative programming, as explained below and in [DDR07]. It may also be used for formalizing the mechanism of exceptions [DR06]. The idea of *decoration* is based upon a span of diagrammatic logics:

$$\mathcal{L}_{\text{sim}} \xleftarrow{\ \ell_{\text{sim}}\ } \mathcal{L}_{\text{dec}} \xrightarrow{\ \ell_{\text{exp}}\ } \mathcal{L}_{\text{exp}}$$

where the three logics $\mathcal{L}_{\text{dec}}$, $\mathcal{L}_{\text{sim}}$ and $\mathcal{L}_{\text{exp}}$ are called respectively the *decorated*, *simplified* and *explicit* logics, and the morphisms $\ell_{\text{sim}}$ and $\ell_{\text{exp}}$ are the *simplification* and *explicitation* morphisms. The subscripts in the notations are simplified, for instance the adjunctions (either on specifications or on theories) with respect to $\ell_{\text{sim}}$ are denoted $F_{\text{sim}} \dashv U_{\text{sim}}$, and so on. It is assumed that both freely generating functors on specifications $F_{\text{sim}}$ and $F_{\text{exp}}$ are easy to compute. The idea is that the logics $\mathcal{L}_{\text{sim}}$ and $\mathcal{L}_{\text{exp}}$ are well-known, while the decorated logic $\mathcal{L}_{\text{dec}}$ is not. The morphisms $\ell_{\text{sim}}$ and $\ell_{\text{exp}}$ are used for building proofs and models, respectively, for any given decorated specification $\Sigma_{\text{dec}}$.

On the models side, it is assumed that the set of intended models of $\Sigma_{\text{dec}}$ is $\text{Mod}_{\text{exp}}(\Sigma_{\text{exp}}, \Theta_{\text{exp}})$ for some given explicit theory $\Theta_{\text{exp}}$, where $\Sigma_{\text{exp}} = F_{\text{exp}}\Sigma_{\text{dec}}$. Then, according to proposition 2.3, the set of intended models of $\Sigma_{\text{dec}}$ can be identified with $\text{Mod}_{\text{dec}}(\Sigma_{\text{dec}}, \Theta_{\text{dec}})$ where $\Theta_{\text{dec}} = U_{\text{exp}}\Theta_{\text{exp}}$. This ensures the soundness of the intended models of $\Sigma_{\text{dec}}$ with respect to the proofs in the decorated logic.

On the proofs side, a decorated proof $p_{\text{dec}}$ is mapped by $\ell_{\text{sim},T}$ to a simplified proof $p_{\text{sim}}$. This property provides a method for building decorated proofs in two steps: first a simplified proof $p_{\text{sim}}$ is built in the well-known simplified logic, then, if possible, a decorated proof $p_{\text{dec}}$ is built such that $p_{\text{sim}} = \ell_{\text{sim},T}p_{\text{dec}}$.

## 5.2 Multivariate functions in imperative programming

Multivariate functions in functional (effect-free) programming can be formalized via categorical products: a term $f(t_1, t_2)$ is composed of the pair $\langle t_1, t_2\rangle$ followed by the bivariate function $f$, so that $t_1$ and $t_2$ play symmetric roles. This cannot be done in imperative programming, where the value of $f(t_1, t_2)$ may depend on the order of evaluation of $t_1$ and $t_2$. A major contribution of [DDR07] is the definition of the *sequential product* of morphisms in an effect category, for formalizing "first $t_1$, then $t_2$". Then a *cartesian effect category* is an effect category with sequential products, it provides a semantics for computational languages with effects. This is shortly reminded below, by looking at the diagrammatic logics that are involved.

The *simplified logic* $\mathcal{L}_{\text{sim}}$ is the equational logic $\mathcal{L}_{\text{Eq}}$, defined as the class of the logical propagator $P_{\text{Eq}}$ in example 4.9.

Let us define the *decorated logic* $\mathcal{L}_{\text{dec}}$. Let $\mathbf{V}$ be category, a *(strict) effect category* extending $\mathbf{V}$ is a category $\mathbf{C}$ such that $\mathbf{V}$ is a wide subcategory of $\mathbf{C}$ (the morphisms in $\mathbf{V}$

are called *pure*) and $\mathbf{C}$ is endowed with a *semi-congruence* $\lesssim$, *i.e.*, a reflexive and transitive relation between parallel morphisms in $\mathbf{C}$ which satisfies the substitution property and only a "pure" version of the replacement property: if $g_1 \lesssim g_2 : Y \to Z$ then $g_1 \circ f \lesssim g_2 \circ f$ *for all* $f : X \to Y$ *in* $\mathbf{C}$ and $v \circ g_1 \lesssim v \circ g_2$ *for all* $v : Z \to W$ *in* $\mathbf{V}$. In a cartesian category, the product $t_1 \times t_2$ of two morphisms is the unique morphism such that $q_1 \circ (t_1 \times t_2) = t_1 \circ p_1$ and $q_2 \circ (t_1 \times t_2) = t_2 \circ p_2$, where the $p_i$'s and $q_i$'s are the relevant projections. In a cartesian effect category, such a product is defined for pure morphisms. When $t_2$ is pure but $t_1$ is not, the *semi-product* $t_1 \times t_2$ is characterized by $q_1 \circ (t_1 \times t_2) = t_1 \circ p_1$ and *only* $q_2 \circ (t_1 \times t_2) \lesssim t_2 \circ p_2$. Then the *sequential product* of two morphisms $t_1$ and $t_2$, when maybe neither is pure, is the composition of the semi-products $(t_1 \times \mathrm{id})$ and $(\mathrm{id} \times t_2)$. For dealing with the side-effects due to modifications of a global state, the relation $f \lesssim g$ means that the functions $f$ an $g$ return the same result, but they may modify the state in two different ways (so that, in this case, $\lesssim$ is interpreted as an equivalence relation). Like $P_{\mathrm{Eq}}$ is obtained from a decomposition of $P_{\mathrm{FpCat}}$ in example 4.9, the logical propagator $P_{\mathrm{EEq}} : \mathbf{E}_{\mathrm{FpESk}} \to \mathbf{E}_{\mathrm{FpECat}}$ is obtained from a decomposition of a propagator $P_{\mathrm{FpECat}} : \mathbf{E}_{\mathrm{EGr}} \to \mathbf{E}_{\mathrm{FpECat}}$, where $\mathbf{E}_{\mathrm{EGr}}$ is the limit sketch for *effect graphs*:

$$\begin{tikzcd} \mathtt{Point} \arrow[r, bend right, "\mathtt{target}"'] & \mathtt{Arrow} \arrow[l, bend right, "\mathtt{source}"'] & \mathtt{PureArrow} \arrow[l, "\mathtt{inj}"'] \end{tikzcd}$$

where `PureArrow` and `inj` stand respectively for the set of *pure arrows* and for the conversion (it can be added that `inj` is a potential monomorphism). This gives rise to the *diagrammatic equational logic with effects* $\mathcal{L}_{\mathrm{dec}}$.

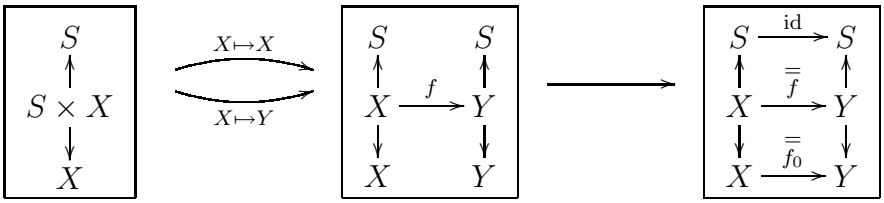
The *simplification morphism* $\ell_{\mathrm{sim}} : \mathcal{L}_{\mathrm{dec}} \to \mathcal{L}_{\mathrm{sim}}$ maps `PureArrow` to `Arrow` and `inj` to $\mathtt{id}_{\mathtt{Arrow}}$, which means that it blurs the distinction between pure and non-pure morphisms. Similarly, $\ell_{\mathrm{sim}}$ maps the semi-congruence to the equality. It follows that $\ell_{\mathrm{sim}}$ maps sequential products to ordinary products. So, each proof in $\mathcal{L}_{\mathrm{dec}}$ is mapped to a proof in $\mathcal{L}_{\mathrm{sim}}$. This property is used in the appendix of [DDR07] for building proofs in equational logic with effects, by decorating proofs in equational logic. The intended models of the decorated specifications are *not* preserved by the simplification morphism.

The *explicit logic* $\mathcal{L}_{\mathrm{exp}}$ is the *pointed equational logic*, made of the equational logic together with a distinguished sort $S$ of *states*. The morphisms of pointed equational specifications (*resp.* theories) must preserve $S$. It is easy to build $\mathcal{L}_{\mathrm{exp}}$ from the equational logic $\mathcal{L}_{\mathrm{Eq}}$.

The *explicitation morphism* $\ell_{\mathrm{exp}} : \mathcal{L}_{\mathrm{dec}} \to \mathcal{L}_{\mathrm{exp}}$ is based on the idea that a morphism $f : X \to Y$ in a decorated specification is mapped to a morphism $f : S \times X \to S \times Y$, and that when $f$ is pure then $f = \mathrm{id}_S \times f_0$ for some $f_0 : X \to Y$. This informal description corresponds to the formal description of $\ell_{\mathrm{exp}}$ via the Yoneda contravariant realizations, using the natural isomorphism $F_P \circ \mathcal{Y}_S \cong \mathcal{Y}_T \circ P$ (for every propagator $P$). The image of $\mathbf{E}_{\mathrm{EGr}}$ by the Yoneda contravariant realization of $\mathbf{E}_{\mathrm{EGr}}$ in the category of effect graphs is as follows (pure morphisms are represented as dashed arrows):

$$\boxed{X} \quad \begin{tikzcd} X \arrow[r, bend left, "X \mapsto X"] \arrow[r, bend right, "X \mapsto Y"'] & \end{tikzcd} \quad \boxed{X \xrightarrow{f} Y} \longrightarrow \boxed{X \overset{f}{\dashrightarrow} Y}$$

This is mapped by $F_{\mathrm{exp}}$ to the following diagram in the category of pointed finite product sketches (the vertical arrows are the projections):




For a fixed set of states $\mathbb{S}$, the category of sets together with $\mathbb{S}$ forms a theory $\mathbf{Set}_{\mathbb{S}}$ with respect to the pointed equational logic. The intended models of a decorated specification $\Sigma_{\text{dec}}$ can be defined as the models of the explicit specification $\Sigma_{\text{exp}} = F_{\text{exp}}\Sigma_{\text{dec}}$ with values in $\mathbf{Set}_{\mathbb{S}}$. As explained in section 5.1, this ensures the soundness of the intended models of $\Sigma_{\text{dec}}$ with respect to the proofs in the decorated logic.

# References


[BW94] Atish Bagchi and Charles Wells. Graph-Based Logic and Sketches I: The General Framework. `oai:arXiv.org:math/9410206` (1994).

[BW99] Michael Barr and Charles Wells. Category Theory for Computing Science, 3rd edition. PM023, CRM, Université de Montréal. `http://crm.umontreal.ca/pub/Ventes/desc/PM023.html` (1999).

[Be67] Jean Bénabou. Introduction to bicategories. Reports of the Midwest Category Seminar. Lecture Notes in Mathematics 47, Springer, p. 1–77 (1967).

[Bj05] Marie Bjerrum. A sketch theoretical survey. Masters Thesis for the Cand. Scient. Degree in Mathematics at the Univeristy of Copenhagen (2005).

[CL84] Laurent Coppey and Christian Lair. Leçons de théorie des esquisses I. Diagrammes 12 (1984).

[CL88] Laurent Coppey and Christian Lair. Leçons de théorie des esquisses II. Diagrammes 19 (1988).

[DDLR06] Cesar Dominguez, Dominique Duval, Laureano Lamban and Julio Rubio. Towards Diagrammatic Specifications of Symbolic Computation Systems. MAP'05. `http://drops.dagstuhl.de/opus/volltexte/2006/292/` (2006).

[DDR07] Jean-Guillaume Dumas, Dominique Duval and Jean-Claude Reynaud. Sequential products in effect categories. `http://front.math.ucdavis.edu/0707.1432` (2007).

[Du03] Dominique Duval. Diagrammatic Specifications. Mathematical Structures in Computer Science 13 p. 857–890 (2003).

[DL02] Dominique Duval and Christian Lair. Diagrammatic Specifications. Rapport de Recherche du LMC 1043M (2002).

[DR06] Dominique Duval and Jean-Claude Reynaud. Diagrammatic logic and exceptions: an introduction. MAP'05. http://drops.dagstuhl.de/opus/volltexte/2006/293/ (2006).

[Eh68] Charles Ehresmann. Esquisses et types des structures algébriques. Bul. Inst. Polit. Iaşi XIV (1968).

[GZ67] Pierre Gabriel and Michel Zisman. Calculus of fractions and homotopy theory. Springer (1967).

[Gu07] René Guitart. Personal communication (2007).

[Lai87] Christian Lair. Trames et Sémantiques Catégoriques des Systèmes de Trames. Diagrammes 18 (1987).

[LD01] Christian Lair and Dominique Duval. Fibrations et éclatements, Lemmes de Yoneda et Modèles Engendrés. Rapport de Recherche du LACO 2001-03. http://www.unilim.fr/laco/rapports (2001).

[Lam68] Joachim Lambek. Deductive systems and categories I: Syntactic calculus and residuated categories. Math. Systems Theory 2 p. 287–318 (1968).

[Law69a] F. William Lawvere. Adjointness in Foundations. Dialectica 23 p. 281–295 (1969). Reprints in Theory and Applications of Categories 16 http://www.tac.mta.ca/tac/reprints/articles/16/tr16.pdf p. 1–16 (2006).

[Law69b] F. William Lawvere. Ordinal sums and equational doctrines. Springer Lecture Notes in Mathematics 80, Springer-Verlag, p. 141–155 (1969).

[Le98] Tom Leinster. Basic bicategories. http://arxiv.org/abs/math/9810017 (1998).

[Mac98] Saunders Mac Lane. Categories for the Working Mathematician. 2nd edition. Graduate Texts in Mathematics 5 Springer (1998).

[Mak97] Michael Makkai. Generalized sketches as a framework for completeness theorems (I). Journal of Pure and Applied Algebra 115 p. 49–79 (1997).

[Mi75] Richard Mijoule. Le forcing pour l'étude des fondements. Cahiers de Topologie et Géométrie Différentielle XVI-3. p. 291–294 (1975).

[Se79] R.A.G. Seely. Weak adjointness in proof theory. Lecture Notes in Math. 753, Springer (1979).

[We93] Charles Wells. Sketches: Outline with References. http://www.cwru.edu/artsci/math/wells/pub/papers.html (1993).