Graph-based Logic and Sketches I: The General Framework

Atish Bagchi* and Charles Wells

October 12, 1994

1 Introduction

Sketches as a method of specification of mathematical structures are an alternative to the string-based specification employed in mathematical logic. Sketch theory has been criticized as being lacunary when contrasted with logic because it apparently has nothing corresponding to proof theory. This article describes an approach to associating a proof-theoretic structure with a sketch. It is written in an innovative style using some of the ideas for presenting mathematics outlined in [Bagchi and Wells, 1993b].

1.1 Prerequisites

This article requires familiarity with the basic notions of mathematical logic as in Chapters 2 through 5 of [Ebbinghaus *et al.*, 1984], and with category theory and sketches as in Chapters 1 through 10 of [Barr and Wells, 1990]. We specifically presuppose that finite-limit sketches are known. Some notation for these ideas is established in Section 4.

*The first author is grateful to the Mathematical Sciences Research Institute for providing the opportunity to work on this paper there. Research at MSRI is supported in part by NSF grant #DMS-9022140.

1.2 Rationale

Traditional treatments of formal logic provide:

1. A syntax for formulas. The formulas are typically defined recursively by a production system (grammar).
2. Inference relations between sets of formulas. This may be given by structural induction on the construction of the formulas.
3. Rules for assigning meaning to formulas (semantics) that are sound with respect to the inference relation. The semantics may also be given by structural induction on the construction of the formulas.

First order logic, the logic and semantics of programming languages, and the languages that have been formulated for various kinds of categories are all commonly described in this way. The formulas in those logics are strings of symbols that are ultimately modeled on the sentences mathematicians speak and write when proving theorems.

Mathematicians with a category-theoretic point of view frequently state and prove theorems using graphs and diagrams. The graphs, diagrams, cones and other data of a sketch are formal objects that correspond to the graphs and diagrams used by such mathematicians in much the same way as the formulas of traditional logic correspond to the sentences mathematicians use in proofs. The functorial semantics of sketches is sound in the informal sense that it preserves by definition the structures given in the sketch. The analogy to the semantics of traditional model theory is close enough that sketches and their models fit the definition of "institution" ([Goguen and Burstall, 1986]), which is an abstract notion of a logical system having syntactic and semantic components. This is described in detail in [Barr and Wells, 1990], Section 10.3. Note that the soundness of functorial semantics appears trivial when contrasted with the inductive proofs of soundness that occur in string-based logic because the semantics functor is not defined recursively.

1.3 Overview

This paper exhibits a structure in sketch theory that corresponds to items 1 and 2 in the description of logic in Section 1.2. The data making up the structure we give do not correspond in any simple way to the data involved in items 1 and 2 of traditional logic; we will discuss the differences *in situ*.

The data in this structure are parametrized by the type of categorial theory being considered. Let E denote a finite-limit sketch that presents a type of category as essentially algebraic over the theory of categories. Let $\mathbf{E}$ denote the finite-limit theory generated by E. $\mathbf{E}$-categories are then the models of $\mathbf{E}$ in the category of sets. (This is described in Section 5.) The kinds of categories that can be described in this way include categories with finite products, categories with limits or colimits over any particular set of diagrams, cartesian closed categories, regular categories, toposes, and many others. Sketches for several specific instances of $\mathbf{E}$ are given in Appendix B.

An $\mathbf{E}$-sketch S (defined in Section 5.3.1 below) is a sketch that allows the specification of any kind of construction that can be made in an $\mathbf{E}$-category. As an example, let $\mathbf{CCC}$ be a finite-limit theory for Cartesian closed categories. It is possible to require that a certain object in a $\mathbf{CCC}$-sketch be the function space A^B of two other objects A and B of the sketch. This generalization greatly enhances the expressive power of sketches as originally defined by Ehresmann (in which only limits and colimits can be specified).

Such an $\mathbf{E}$-sketch is described in Section 5.3.1 as a formally adjoined global element of the limit vertex of a certain diagram in $\mathbf{E}$. This diagram forces the value of the limit vertex in a model to contain the graph, diagrams and other structures in the sketch. The $\mathbf{E}$-theory of this sketch, which corresponds roughly, but not precisely, to the deductive closure of a list of axioms in traditional logic, is the initial model of $\mathbf{E}[S]$, which is $\mathbf{E}$ with the global element adjoined.

An assertion in this setting is a potential factorization (PF) of an arrow of the $\mathbf{E}$-theory of S through an arrow into its codomain (defined precisely in Section 6.1). The assertion is valid if the PF does indeed factorize in every model of $\mathbf{E}[S]$.

Instead of the set of rules of deduction of a traditional theory, we have a set of rules of construction. More precisely, we give in Appendix A a system of construction rules that produce the categorial theory of a finite-limits sketch. We say that the potential factorization is **deducible** if there is an actual factorization in the categorial theory of the sketch. Such an arrow must be constructible by the rules in Appendix A. Thus the usual system of inference is replaced by a system of construction of arrows in a finite-limit category (no matter what type of category is sketched by E). This system is sound and complete with respect to models (Section 6.2).

The fact that we have assumed finite-limit sketches as given prior to the general definition of E-sketch is basic to the strategy of the paper, which is to make finite-limit logic the basic logic for all E-sketches (Section 8). The variation in what can be proved, for example for finite-limit sketches (E-sketches where $\mathsf{E} = \mathsf{FinLim}$) as contrasted with cartesian closed sketches (in which case E is a finite-limit theory for cartesian closed categories such as CCC in Appendix B.6) is entirely expressed by the choice of E and has no effect on the rules of construction.

The system that we describe may not facilitate human manipulation. However, as we discuss in Remark A.1.2, the system should not be difficult for computer implementation. Our approach may seem unusual from the point of view of symbolic logic, but it is in keeping with practices in computer implementation of complex data structures (a proof is a complex data structure) in that much more detail about the relationships between different parts of the structure are necessarily made explicit than is customary for systems that are designed to be understood by human beings, who have sophisticated built-in pattern recognition abilities.

1.4 Acknowledgments

This article is better because of conversations we have had with Robin Cockett and Colin McLarty. The names "string-based logic" and "graph-based logic" were suggested by Peter Freyd. The diagrams were prepared using K. Rose's `xypic`.

2 Notation

The notation established in this section is used throughout the article.

2.1 Lists

Given a set A, $\mathsf{List}[A]$ denotes the set of lists of elements of A, including the empty list. The kth entry in a list w of elements of A is denoted by w_k and the length of w is denoted by $\mathsf{Length}[w]$. The **range** of w, denoted by $\mathsf{Rng}[w]$, is the set of elements of A occurring as entries in w. If $f : A \to B$ is a function, $\mathsf{List}[f] : \mathsf{List}[A] \to \mathsf{List}[B]$ is by definition f "mapped over" $\mathsf{List}[A]$: If w is a list of elements of A, then the kth entry of $\mathsf{List}[f](w)$ is by definition $f(w_k)$. This makes List a functor from the category of sets to itself.

2.2 Graphs

For a graph G, the set of nodes of G is denoted by $\mathsf{Nodes}[G]$ and the set of arrows is $\mathsf{Arrows}[G]$. The underlying graph of a category $\mathscr{C}$ is denoted by $\mathsf{UndGr}[\mathscr{C}]$. A subgraph H of a graph G is said to be **full** if every arrow $f : h_1 \to h_2$ of G between nodes of H is an arrow of H.

2.3 Diagrams

We define an equivalence relation on the set of graph homomorphisms into a graph G as follows: If $\delta : I \to G$ and $\delta' : I' \to G$ are graph homomorphisms, then δ is equivalent to δ' if and only if there is a graph isomorphism $\phi : I \to I'$ such that

$$\begin{tikzcd} I \arrow[rr, "\phi"] \arrow[dr, "\delta"'] & & I' \arrow[dl, "\delta'"] \\ & G & \end{tikzcd} \tag{1}$$

commutes. A **diagram** in G is then by definition an equivalence class of graph homomorphisms $\delta : I \to G$. As is the practice when an object is defined to be an equivalence class, we will refer to a diagram by any member of the equivalence class.

If $\delta : I \to G$ is a diagram, I is said to be a **shape graph** of the diagram, denoted by $\mathsf{ShpGr}(\delta)$, and G is said to be the **ambient space** of the diagram, denoted by $\mathsf{AmbSp}(\delta)$. Observe that the ambient space of the diagram is determined absolutely, but the shape graph is determined only up to an isomorphism that makes Diagram (1) commute.

If I is a graph, $\mathscr{C}$ is a category, and $\delta : I \to \mathsf{UndGr}[\mathscr{C}]$ is a diagram, we will also write it as $\delta : I \to \mathscr{C}$ (unless there is danger of confusion).

2.3.1 Convention on drawing diagrams

It is customary to draw a diagram without naming its shape graph. We adopt the following convention: If a diagram is represented by a drawing on the page, the shape graph of the diagram is the graph that has one node for each object shown and one arrow for each arrow shown, with source and target as shown. Two objects at different locations on the page correspond to two different nodes of the shape graph, *even if they have the same label*, and an analogous remark applies to arrows.

2.3.2 Example

The diagram (2)

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "g"'] & A \arrow[d, "g"] \\ B \arrow[r, "\mathsf{Id}[B]"'] & B \end{tikzcd} \tag{2}$$

called δ, has shape graph

$$\begin{tikzcd} h \arrow[r, "t"] \arrow[d, "u"'] & i \arrow[d, "v"] \\ j \arrow[r, "x"'] & k \end{tikzcd} \tag{3}$$

so that $\delta(h) = A$, $\delta(i) = A$, $\delta(v) = g$, $\delta(x) = \mathsf{Id}[B]$ and so on. Diagram (4) below also has shape graph (3) (or one isomorphic to it, of course):

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "g"'] & B \arrow[d, "h"] \\ C \arrow[r, "k"'] & D \end{tikzcd} \tag{4}$$

On the other hand, Diagram (5) below

$$\begin{tikzcd} A \arrow[rr, "f"] \arrow[dr, "g"'] & & A \arrow[dl, "g"] \\ & B & \end{tikzcd} \tag{5}$$

is not the same diagram as (2). It has shape graph

$$\begin{tikzcd} i \arrow[rr, "u"] \arrow[dr, "v"'] & & j \arrow[dl, "w"] \\ & k & \end{tikzcd} \tag{6}$$

2.4 Cones

For any graph G and diagram $\delta : I \to G$, a cone $\Theta : v \lhd (\delta : I \to G)$ (also written $\Theta : v \lhd \delta$ if the context makes this clear) has **vertex** v denoted by $\mathsf{Vertex}[\Theta]$ and **base diagram** δ denoted by $\mathsf{BsDiag}[\Theta]$. For each node i of $\mathsf{ShpGr}[\delta]$, the formal projection of the cone Θ from $\mathsf{Vertex}[\Theta]$ to $\delta(i)$ is denoted by $\mathsf{Proj}[\Theta, i] : v \to \delta(i)$. For a category $\mathscr{C}$, a cone $\Theta : v \lhd (\delta : I \to \mathscr{C})$ is **commutative** if, for every arrow $f : i \to j$ in I, the diagram

$$\begin{tikzcd} & \mathsf{Vertex}[\Theta] \arrow[dl, "{\mathsf{Proj}[\Theta, i]}"'] \arrow[dr, "{\mathsf{Proj}[\Theta, j]}"] & \\ \delta(i) \arrow[rr, "\delta(f)"'] & & \delta(j) \end{tikzcd} \tag{7}$$

commutes. The limit cone for a diagram δ will be denoted by $\mathsf{LimCone}[\delta] : \mathsf{Lim}[\delta] \lhd \delta$.

2.5 Fonts

In general, variable objects are given in slant or script notation and specific objects (given by proper names) are given in upright notation. In more detail, we have the following systematic notation.

1. Specific data constructors, such as List, and specific fieldnames for complex objects, such as $\mathsf{Nodes}[G]$, are given in sans serif and are capitalized as shown.

2. Specific objects and arrows of sketches are also given in sans serif.

3. Specific constructor spaces, such as **FinLim** and **CCC**, are given in **bold sans serif**. We use **E** to denote a variable constructor space because of the unavailability of bold slanted sans serif.

4. Specific categories other than constructor spaces, such as **Set**, are given in **boldface**.

5. Diagrams (specific and variable) are named by lowercase Greek letters.

6. Cones (specific and variable) are named by uppercase Greek letters.

7. Models (specific and variable) are given in uppercase fraktur, for example $\mathfrak{M}$, $\mathfrak{C}$.

8. Variable sketches, and objects and arrows therein, are given in *slanted sans serif*.

9. Variable categories other than constructor spaces are given in script, for example $\mathscr{A}$, $\mathscr{B}$, $\mathscr{C}$.
10. Other variable objects are given in math italics, such as a, b, c or (especially arrows) in lowercase Greek letters.

3 Adjoining nodes to diagrams

We now state and prove some technical lemmas that are used many times herein.

3.1 Restrictions of diagrams

3.1.1 Definition Let $\delta : I \to \mathscr{C}$ be a diagram and $\mathsf{Incl}[J \subseteq I] : J \to I$ an inclusion of graphs. Let $\Theta : v \lhd \delta$ be a cone. The **base-restriction of Θ to J** is defined to be the cone $\Theta|^J : v \lhd (\delta \circ \mathsf{Incl}[J \subseteq I])$ with vertex v and projections defined by $\mathsf{Proj}[\Theta|^J, j] := \mathsf{Proj}[\Theta, j] : v \to \delta(j)$ for all nodes j of J. In this case, we also say that Θ is a **base-extension** of $\Theta|^J$.

3.1.2 Remark Observe that if Θ is commutative, then so is $\Theta|^J$.

3.1.3 Definition Let $\delta : I \to \mathscr{C}$ be a diagram and J a subgraph of I. Then the subdiagram $\delta|_J$ is said to **dominate** δ, or to be **dominant in** δ, if every commutative cone $\Theta : v \to (\delta|_J)$ has a base extension to a commutative cone $\Theta' : v \to \delta$ with the same vertex.

3.1.4 Remark Tholen and Tozzi [1989] give a condition ("confinality") on I and J such that any diagram based on I is dominated by its base restriction to J. One type of dominance that their condition does not cover is the case in which δ is obtained from $\delta|_J$ by adjoining a limit cone over a subdiagram of $\delta|_J$. This occurs in Section 7.1.1.

3.2 Limits of subdiagrams

3.2.1 Lemma *Let $\delta : I \to \mathscr{C}$ be a diagram and let J be a subgraph of I with inclusion $\mathsf{Incl}[J \subseteq I]$. Let $\gamma = \delta|_J$. Then there is a unique arrow*

$\phi : \mathsf{Lim}[\delta] \to \mathsf{Lim}[\gamma]$ *such that for all nodes j of J,*

$$\begin{tikzcd} \mathsf{Lim}[\delta] \arrow[rr, "\phi"] \arrow[dr, "{\mathsf{Proj}[\mathsf{LimCone}[\delta], j]}"'] & & \mathsf{Lim}[\gamma] \arrow[dl, "{\mathsf{Proj}[\mathsf{LimCone}[\gamma], j]}"] \\ & \delta(j) & \end{tikzcd} \tag{8}$$

commutes. Moreover, if γ is a dominant subdiagram of δ, then ϕ is an isomorphism.

Proof The base-restriction of $\mathsf{LimCone}[\delta]$ to J is a commutative cone Θ with vertex $\mathsf{Lim}[\delta]$. We choose ϕ to be $\mathsf{Fillin}[\Theta, \gamma]$, the arrow produced by Rule ∃FIA of Appendix A. It follows from Rule CFIA that for each node j of J, the diagram below

$$\begin{tikzcd} \mathsf{Lim}[\delta] \arrow[rr, "{\mathsf{Fillin}[\Theta, \gamma]}"] \arrow[dr, "{\mathsf{Proj}[\Theta, j]}"'] & & \mathsf{Lim}[\gamma] \arrow[dl, "{\mathsf{Proj}[\mathsf{LimCone}[\gamma], j]]}"] \\ & \gamma(j) & \end{tikzcd} \tag{9}$$

commutes. Because they do, rule !FIA implies that ϕ is the only arrow that makes them all commute, as required.

Assume that γ dominates δ. Then we may choose an extension $\Psi : \mathsf{Lim}[\gamma] \to \delta$ of $\mathsf{LimCone}[\gamma]$ to δ. Using ∃FIA, we define

$$\psi := \mathsf{Fillin}[\Psi, \delta] : \mathsf{Lim}[\gamma] \to \mathsf{Lim}[\delta]$$

It follows from !FIA that ψ is the only arrow from $\mathsf{Lim}[\gamma]$ to $\mathsf{Lim}[\delta]$ that makes all diagrams of the form

$$\begin{tikzcd} \mathsf{Lim}[\gamma] \arrow[rr, "\psi"] \arrow[dr, "{\mathsf{Proj}[\Psi, j]}"'] & & \mathsf{Lim}[\delta] \arrow[dl, "{\mathsf{Proj}[\mathsf{LimCone}[\delta], j]]}"] \\ & \gamma(j) & \end{tikzcd} \tag{10}$$

commute for each node j of J. Since $\phi \circ \psi : \mathsf{Lim}[\gamma] \to \mathsf{Lim}[\gamma]$ and $\mathsf{Id}[\mathsf{Lim}[\gamma]] : \mathsf{Lim}[\gamma] \to \mathsf{Lim}[\gamma]$ both commute with all the projections, it follows from !FIA that $\phi \circ \psi = \mathsf{Id}[\mathsf{Lim}[\gamma]]$, A similar argument shows that $\psi \circ \phi = \mathsf{Id}[\mathsf{Lim}[\delta]]$, so that ϕ is an isomorphism. □

3.3 Special cases of extending diagrams

Here we define some special cases of dominance that are easy to recognize.

3.3.1 Definition Let I and J be graphs that have the same nodes, and suppose that I has exactly one arrow $a : j \to k$ not in J. Let $\delta : I \to \mathscr{C}$ be a diagram with the property that for all nodes j' of J and all arrows $f : j \to j'$ and $g : j' \to k$,

$$\begin{tikzcd} j \arrow[rr, "a"] \arrow[dr, "f"'] & & k \\ & j' \arrow[ur, "g"'] & \end{tikzcd} \tag{11}$$

commutes. Then we say δ **extends** $\delta|_J$ **by adjoining a composite**.

3.3.2 Definition Let I and J be graphs with the following properties:

AC.1 I has exactly one node v not in J.

AC.2 Every arrow in I not in J has target v.

Suppose that $\delta : I \to \mathscr{C}$ is a diagram with the property that if $a : i \to v$, $b : j \to v$ and $f : i \to j$ are arrows of I, then

$$\begin{tikzcd} i \arrow[rr, "a"] \arrow[dr, "f"'] & & v \\ & j \arrow[ur, "b"'] & \end{tikzcd} \tag{12}$$

commutes. Then we say δ **extends** $\delta|J$ by **adjoining a commutative cocone**.

3.3.3 Definition Let I, J and J' be graphs with $J' \subseteq J \subseteq I$, such that J' is full in J, I contains exactly one node v not in J, and for each node j of J', I contains exactly one arrow $p_j : v \to j$ and no other arrows not in J. Let $\delta : J \to \mathscr{C}$ be a diagram, and suppose $\delta' : I \to \mathscr{C}$ is a diagram that extends δ in such a way that $\delta'(v)$ and the arrows $\delta'(p_j)$ constitute a limit cone to $\delta|_{J'}$. Then we say that δ' **extends** δ **by adjoining a limit**.

3.3.4 Lemma *Suppose that $\delta' : I \to \mathscr{C}$ extends $\delta : J \to \mathscr{C}$ by adjoining a composite, a commutative cocone, or a limit. Then*

$$\mathsf{Fillin}[\mathsf{LimCone}[\delta']|_J, \delta] : \mathsf{Lim}[\delta'] \to \mathsf{Lim}[\delta]$$

is an isomorphism.

Proof It is easy to see that, in all three cases, J is dominant in I. □

4 Finite limit sketches

We use a general concept of sketch described in Section 5 that is based on the concept of finite limit sketch, a particular case of projective sketch due to Ehresmann. In this section, we review briefly some aspects of finite limit sketches that are relevant later.

4.1 Categorial theories

The categorial theory generated by a finite-limit sketch $\mathcal{S}$ is denoted by $\mathsf{CatTh}[\mathbf{FinLim}, \mathcal{S}]$. It is a category with finite limits together with a sketch morphism

$$\mathsf{UnivMod}[\mathbf{FinLim}, \mathcal{S}] : \mathcal{S} \to \mathsf{CatTh}[\mathbf{FinLim}, \mathcal{S}]$$

called the **universal model** of the sketch. It has the following property: For each model $\mathfrak{M}$ of $\mathcal{S}$, there exists a functor $F_{\mathfrak{M}} : \mathsf{CatTh}[\mathbf{FinLim}, \mathcal{S}] \to \mathbf{Set}$, determined uniquely up to natural isomorphism, with the property that

$$U(F_{\mathfrak{M}}) \circ \mathsf{UnivMod}[\mathbf{FinLim}, \mathcal{S}] = \mathfrak{M}$$

where U is the forgetful functor from the category of finite-limit categories and finite-limit-preserving functors to the category of finite-limit sketches and morphisms thereof. It follows from these properties that $\mathsf{CatTh}[\mathbf{FinLim}, \mathcal{S}]$ is determined up to equivalence of categories and $\mathsf{UnivMod}[\mathbf{FinLim}, \mathcal{S}]$ is determined up to natural isomorphism.

In this article, for a given sketch $\mathcal{S}$, we assume given a particular instance of $\mathsf{CatTh}[\mathbf{FinLim}, \mathcal{S}]$: that constructed in [Barr and Wells, 1993a]. It has the following properties (which are not preserved by equivalence of categories):

T.1 $\mathsf{CatTh}[\mathbf{FinLim}, \mathcal{S}]$ is a category with canonical finite limits. (The construction explicitly produces the canonical limits.)

T.2 Every object of $\mathsf{CatTh}[\mathbf{FinLim}, S]$ is the canonical limit of a diagram of the form $\mathsf{UnivMod}[\mathbf{FinLim}, S] \circ \delta$ where δ is a diagram in the graph of S.

T.3 Every arrow of $\mathsf{CatTh}[\mathbf{FinLim}, S]$ is a composite of projections from canonical limits and arrows of the form $\mathsf{UnivMod}[\mathbf{FinLim}, S](f)$ for arrows f of the graph of S.

$\mathsf{CatTh}[\mathbf{FinLim}, S]$ is clearly closed under all the constructions of Appendix A. The properties listed in the preceding paragraph imply that the category $\mathsf{CatTh}[\mathbf{FinLim}, S]$ is minimal with respect to the constructions of that Appendix, so that in fact those constructions can be taken as an recursive definition of $\mathsf{CatTh}[\mathbf{FinLim}, S]$.

These observations are summed up in the following proposition.

4.1.1 Proposition *For a given sketch S, every object and every arrow of* $\mathsf{CatTh}[\mathbf{FinLim}, S]$ *is constructible by repeated applications of the constructions of Appendix A.*

It is proved in [Barr and Wells, 1993a] that every object of $\mathsf{CatTh}[\mathbf{FinLim}, S]$ is the limit of a diagram in the graph of the sketch S. Once this is known, the proof is a straightforward induction showing that the least fixed point of the constructions in Appendix A is indeed a category with finite limits.

4.2 Remark concerning models

Let Cat be some given finite-limit sketch for categories. We have to consider the following three mathematical entities.

1. A category $\mathscr{C}$.
2. A model $\mathfrak{C}$ of the sketch Cat in the category of sets which "is" or "represents" $\mathscr{C}$. $\mathfrak{C}$ is a morphism of finite-limit sketches from Cat to the underlying finite-limit sketch of **Set**. $\mathfrak{C}$ is determined uniquely by $\mathscr{C}$ if one assumes that **Set** has canonical finite limits. It is determined up to natural isomorphism in any case.
3. The model $\mathsf{CatTh}[\mathbf{FinLim}, \mathfrak{C}]$ of $\mathsf{CatTh}[\mathbf{FinLim}, \mathit{Cat}]$ induced by $\mathfrak{C}$. This is a finite-limit-preserving functor from $\mathsf{CatTh}[\mathbf{FinLim}, \mathit{Cat}]$ to **Set**.

For many category theorists, $\mathscr{C}$, $\mathfrak{C}$ and $\mathsf{CatTh}[\mathbf{FinLim}, \mathfrak{C}]$ denote the "same thing", or perhaps $\mathscr{C}$ and $\mathsf{CatTh}[\mathbf{FinLim}, \mathfrak{C}]$ are the "same thing" and $\mathfrak{C}$

consists of presentation data for $\mathscr{C}$. Other mathematicians would disagree, saying $\mathscr{C}$ is a category (presumably for them "category" has some meaning other than "model for the sketch for categories") and $\mathsf{CatTh}[\mathbf{FinLim}, \mathit{Cat}]$ is a functor, so how could they be the same thing? In the sequel, we will distinguish these constructions typographically, but we do not take a stand on this difference of opinion. Note that in any case the following three categories are equivalent:

1. The category of small categories and functors.
2. $\mathbf{Mod}[\mathit{Cat}, \mathbf{Set}]$, the category of models of the sketch Cat in the category $\mathbf{Set}$ with natural transformations as morphisms.
3. The category of finite-limit-preserving functors from $\mathsf{CatTh}[\mathbf{FinLim}, \mathit{Cat}]$ to $\mathbf{Set}$ with natural transformations between them as morphisms.

5 Sketches in general

We provide here a definition of "sketch" based on [Wells, 1990] (some of the terminology has been changed.) This constitutes a generalization of the concept of sketch invented by Charles Ehresmann and described in [Bastiani and Ehresmann, 1972] or in [Barr and Wells, 1990]. This definition presupposes the concept of finite-limit sketch (see Sections 1.1 and 4).

5.1 Constructor spaces

5.1.1 Assumptions

We will assume given a fixed finite-limit sketch Cat whose category of models is the category of small categories and functors. A specific such sketch is given explicitly in Appendix B.2. The presentation that follows has Cat as an implicit parameter.

5.1.2 Notation For any finite-limit sketch S and morphism $\eta : \mathit{Cat} \to S$ of sketches, the induced functor between categorial theories will be denoted by

$$\mathsf{CatTh}[\mathbf{FinLim}, \eta] : \mathsf{CatTh}[\mathbf{FinLim}, \mathit{Cat}] \to \mathsf{CatTh}[\mathbf{FinLim}, S]$$

5.1.3 Definition A finite-limit sketch E together with a morphism η : $\mathit{Cat} \to \mathit{E}$ of sketches is called a **constructor space sketch** provided that every object in $\mathsf{CatTh}[\mathbf{FinLim}, \mathit{E}]$ is the limit of a finite diagram whose nodes are of the form $\mathsf{CatTh}[\mathbf{FinLim}, \eta](\mathsf{n})$, where n is a node of Cat.

5.1.4 Remark In all the examples in this article, $\mathsf{CatTh}[\mathbf{FinLim}, \eta]$ is injective on objects. We will often assume that it is an inclusion and write n for $\mathsf{CatTh}[\mathbf{FinLim}, \eta](\mathsf{n})$.

5.1.5 Definition A category of the form $\mathsf{CatTh}[\mathbf{FinLim}, \mathit{E}]$ for some constructor space sketch E is called a **constructor space**. We will normally denote the constructor space $\mathsf{CatTh}[\mathbf{FinLim}, \mathit{E}]$ by $\mathbf{E}$.

5.1.6 Definition A model of a constructor space $\mathbf{E}$ in **Set** is called an **$\mathbf{E}$-category**, and a morphism of such models is called an **$\mathbf{E}$-functor**.

5.1.7 Discussion

Definitions 5.1.3, 5.1.5 and 5.1.6 generalize those in [Wells, 1990]. They conform to the two-dimensional version given in [Power and Wells, 1992].

The value in a model $\mathfrak{M}$ of an object v in a constructor space is the set of all examples of a particular construction that is possible in the $\mathbf{E}$-category $\mathfrak{M}$. Extended examples of this are discussed in Section 5.2 and Section 7. Thus each object of $\mathbf{E}$ represents a *type of construction* possible in $\mathbf{E}$, hence the name "constructor space".

Examples of constructor space sketches include the sketches for categories with finite products, categories with finite limits, cartesian closed categories, toposes, and so on. Several constructor space sketches are given in detail in Appendix B.

5.2 Notation for diagrams in a constructor space

The object of $\mathsf{CatTh}[\mathbf{FinLim}, \mathit{Cat}]$ whose value in a model is the set of all not necessarily commutative diagrams of the form

$$\begin{tikzcd} A \arrow[r, "h"] \arrow[d, "f"'] & B \arrow[d, "k"] \\ C \arrow[r, "g"'] \arrow[ur, "x"] & D \end{tikzcd} \tag{13}$$

is the limit of the diagram

$$
\begin{tikzcd}
 & \mathsf{ar} \arrow[r, "\mathsf{target}"] \arrow[dl, "\mathsf{source}"'] & \mathsf{ob} & \mathsf{ar} \arrow[l, "\mathsf{source}"'] \arrow[dr, "\mathsf{target}"] & \\
\mathsf{ob} & & \mathsf{ar} \arrow[u, "\mathsf{source}"] \arrow[d, "\mathsf{target}"] & & \mathsf{ob} \\
 & \mathsf{ar} \arrow[r, "\mathsf{target}"'] \arrow[ul, "\mathsf{source}"] & \mathsf{ob} & \mathsf{ar} \arrow[l, "\mathsf{source}"] \arrow[ur, "\mathsf{target}"'] &
\end{tikzcd}
\tag{14}
$$

We now describe this in more detail and introduce some notation that makes the discussion of such diagrams easier to follow. We use the notation $D(n)$ to refer to the diagram shown herein with label (n), and $I(n)$ for its shape graph. For example, the limit of the diagram above is $\mathsf{Lim}[D(14)]$.

Every node of $D(14)$ is either the object ob (the object that becomes the set of objects in a model) or the object ar (the object that becomes the set of arrows in a model) of $\mathsf{CatTh}[\mathbf{FinLim}, \mathit{Cat}]$. For a model $\mathfrak{C}$ of $\mathsf{CatTh}[\mathbf{FinLim}, \mathit{Cat}]$ in $\mathbf{Set}$, an element of $\mathfrak{C}(\mathsf{Lim}[D(14)]$ is a diagram in $\mathfrak{C}$, not necessarily commutative, of the form of Diagram (13).

In order to make the relation between Diagrams (13) and (14) clear, we give the shape graph of (14):

$$
\begin{tikzcd}
 & f \arrow[r, "t"] \arrow[dl, "s"'] & C & g \arrow[l, "s"'] \arrow[dr, "t"] & \\
A & & x \arrow[u, "s"] \arrow[d, "t"] & & D \\
 & h \arrow[r, "t"'] \arrow[ul, "s"] & B & k \arrow[l, "s"] \arrow[ur, "t"'] &
\end{tikzcd}
\tag{15}
$$

We have named the nodes of Diagram (15) by the objects and arrows that occur in Diagram (13) in such a way that the node named by an object or arrow of Diagram (13) will inhabit the value of that node in the model $\mathfrak{C}$. For example, the object A of $\mathfrak{C}$ is at the upper left corner of Diagram (13) and the projection arrow from $\mathfrak{C}(\mathsf{Lim}[D(14)])$ to $\mathfrak{C}(\mathsf{ob})$ determined by the node labeled A of the shape graph (15) is a function from the set of diagrams in $\mathfrak{C}$

of the form of Diagram (13) to the set of objects of $\mathfrak{C}$ that takes a diagram to the object in its upper left corner.

We will combine diagrams such as Diagram (14) and their shape graphs into one graph by labeling the nodes of the diagram by superscripts naming the corresponding node of the shape graph. In the case of Diagram (14), doing this gives the following **annotated diagram**:

$$\begin{tikzcd}
 & \mathsf{ar}^f \arrow[dl, "\mathsf{source}"'] \arrow[r, "\mathsf{target}"] & \mathsf{ob}^C & \mathsf{ar}^g \arrow[l, "\mathsf{source}"'] \arrow[dr, "\mathsf{target}"] & \\
\mathsf{ob}^A & & \mathsf{ar}^x \arrow[u, "\mathsf{source}"] \arrow[d, "\mathsf{target}"] & & \mathsf{ob}^D \\
 & \mathsf{ar}^h \arrow[ul, "\mathsf{source}"] \arrow[r, "\mathsf{target}"] & \mathsf{ob}^B & \mathsf{ar}^k \arrow[l, "\mathsf{source}"'] \arrow[ur, "\mathsf{target}"'] &
\end{tikzcd} \tag{16}$$

Here, the superscript A on the leftmost node indicates that the corresponding node of the shape graph is labeled A. Formally, the expression ob^A is used as the label for the node $\delta(A)$ and its use signifies that $\delta(A) = \mathsf{ob}$. That device helps the reader to see that Diagram (13) is indeed an element of $\mathfrak{C}(\mathsf{Lim}[D(14)])$.

For example, the particular arrow h of Diagram (13) is an element of ar, and the label ar^h in Diagram (16) helps one see that it is that node that projects to h in the model $\mathfrak{C}$ and that the source of h is A and that the target is B.

It is important to understand that an annotated diagram such as (16) denotes precisely the same diagram as (14). The fact that one node is labeled ob^A and another ob^B does not change the fact that *both nodes* are ob. The superscript merely gives information about the relation between Diagram (14) and Diagram (13).

Diagram (16) could also be drawn as the base of a limit cone Θ with limit

$\mathsf{Lim}[D(16)]$ (which of course is the same as $\mathsf{Lim}[D(14)]$) as follows.

$$\begin{tikzcd}
 & & \mathsf{Lim}[\delta(14)] \arrow[dll, "{\mathsf{Proj}[\Theta, g]}"'] \arrow[dl, "{\mathsf{Proj}[\Theta, f]}"] \arrow[d, "{\mathsf{Proj}[\Theta, x]}"] \arrow[dr, "{\mathsf{Proj}[\Theta, k]}"'] \arrow[drr, "{\mathsf{Proj}[\Theta, h]}"] & & \\
\mathsf{ar}^g \arrow[d, "\mathsf{source}"'] \arrow[ddr, "\mathsf{target}"] & \mathsf{ar}^f \arrow[dl, "\mathsf{source}"'] \arrow[ddrr, "\mathsf{target}"] & \mathsf{ar}^x \arrow[dll, "\mathsf{source}"] \arrow[drr, "\mathsf{target}"'] & \mathsf{ar}^k \arrow[ddll, "\mathsf{source}"'] \arrow[dr, "\mathsf{target}"] & \mathsf{ar}^h \arrow[d, "\mathsf{target}"] \arrow[ddl, "\mathsf{source}"'] \\
\mathsf{ob}^C & & & & \mathsf{ob}^B \\
 & \mathsf{ob}^D & & \mathsf{ob}^A &
\end{tikzcd} \tag{17}$$

Because of the typographical complexity of doing this for diagrams more complicated than Diagram (16), we will usually give diagrams whose limits we discuss in the form of Diagram (16), without showing the cone, instead of in the form of Diagram (17).

Showing the cone explicitly as in Diagram (17) nevertheless has an advantage. It makes it clear that many of the projection arrows from $\mathsf{Lim}[D(16)]$ are induced by others; in the particular case of Diagram (17), all the arrows to nodes of the form ob are induced by composing arrows to some node of the form ar with source or target. Diagram (16) does not exhibit this property. A systematic method of translating from graphical expressions such as Diagram (17) to a string-based expression could presumably be based on this. In the case of Diagram (17), the string-based expression would be something like this:

$$\begin{aligned}
\{\langle g, f, x, k, h\rangle \mid \mathrm{source}(g) = C, \mathrm{target}(g) = D, \quad \mathrm{source}(f) = A,\\
\mathrm{target}(f) = C, \mathrm{source}(x) = C, \mathrm{target}(x) = B, \mathrm{source}(k) = B,\\
\mathrm{target}(k) = D, \mathrm{source}(h) = A, \mathrm{target}(h) = B\}
\end{aligned}$$

or in more familiar terms,

$$\{\langle g, f, x, k, h\rangle \;\mid\; g : C \to D, f : A \to C, x : C \to B, k : B \to D, h : A \to B\}$$

5.3 Sketches

5.3.1 Definition Let E be a constructor space. For any node v of E, we may freely adjoin a global element $\mathsf{S}:1 \to \mathsf{v}$ to obtain a finite-limit theory $\mathsf{E}[\mathsf{S}]$. S is called an **E-sketch** and $\mathsf{E}[\mathsf{S}]$ is called the **syntactic category** of S; we will denote it $\mathsf{SynCat}[\mathsf{S}]$. It should be distinguished from the categorial theory $\mathsf{CatTh}[\mathsf{E}, \mathsf{S}]$, which is the initial model of $\mathsf{E}[\mathsf{S}]$. Details are in [Wells, 1990] or [Power and Wells, 1992].

5.3.2 Remark If $\mathsf{E} = \mathsf{FinLim}$, then a finite-limit sketch S in the traditional sense (a graph with diagrams and cones) corresponds to a **FinLim**-sketch in the sense of Definition 5.3.1 that we denote by $\mathsf{Form}[S]$ (but see Remark 5.3.4). The traditional sketch S is an element of the value in the initial model of a certain (not uniquely determined) node v of **FinLim** which is the limit of a generally large and complicated diagram (not uniquely determined) that specifies the graph, diagrams and cones of S. Example 5.3.3 illustrates this correspondence. These remarks apply to other types of Ehresmann sketches by replacing **FinLim** by the suitable constructor space.

5.3.3 Example As an example, consider the **FinLim**-sketch S with graph

$$\begin{tikzcd} A \arrow[r, shift left, "v"] & B \arrow[l, shift left, "u"] \arrow[r, "f"] & C \end{tikzcd} \tag{18}$$

one diagram

$$\begin{tikzcd} B \arrow[r, "u"] \arrow[d, "\mathsf{Id}[B]"'] & A \arrow[dl, "v"] \\ B & \end{tikzcd} \tag{19}$$

and one cone

$$\begin{tikzcd} & B \arrow[dl, "u"'] \arrow[dr, "f"] & \\ A & & C \end{tikzcd} \tag{20}$$

This is in fact a finite-product sketch, but any such sketch is also a **FinLim**-sketch. This is achieved, for instance, by setting $\mathsf{S} := \mathsf{Form}[S] : 1 \to \mathsf{v}$, where

v is the limit of this diagram:

$$
\begin{tikzcd}
 & & \mathsf{ob}^A \times \mathsf{ob}^C \arrow[d, "\mathsf{prod}"] & & \\
 & \mathsf{ar}^u \arrow[dl, "\mathsf{target}"'] \arrow[drr, "\mathsf{source}"] & \mathsf{cone} \arrow[l, "\mathsf{lproj}"'] \arrow[r, "\mathsf{rproj}"] & \mathsf{ar}^f \arrow[r, "\mathsf{target}"] \arrow[d, "\mathsf{source}"] & \mathsf{ob}^C \\
\mathsf{ob}^A & \mathsf{ar}_2 \arrow[u, "\mathsf{rfac}"] \arrow[r, "\mathsf{comp}"] \arrow[d, "\mathsf{lfac}"] & \mathsf{ar} & \mathsf{ob}^B \arrow[l, "\mathsf{unit}"'] & \\
 & \mathsf{ar}^v \arrow[ul, "\mathsf{source}"] \arrow[urr, "\mathsf{target}"'] & & &
\end{tikzcd}
\tag{21}
$$

This is however not the only way of viewing S as a **FinLim**-sketch as explained below.

5.3.4 Remark Note that the identification between traditional sketches and sketches in the new sense is many-to-one in both directions. A traditional finite-limit sketch can be described as an element of the limit of more than one diagram in **FinLim**. In the other direction, because every node v of **FinLim** is the limit of a diagram in a sketch for **FinLim** (for example, the one in Appendix B.5), any element of the value of v in an initial model can be described in terms of a graph, diagrams and cones, but in general not in only one way, and in any case such a description requires an arbitrary choice for the names of the nodes and arrows of the graph.

5.4 Models of sketches

Let $\mathcal{S}$ be an **E**-sketch as in Definition 5.3.1. The category of models (finite-limit-preserving functors) of $\mathbf{E}[\mathcal{S}]$ has an underlying functor $\mathsf{UndCat}[\mathbf{E}, \mathcal{S}]$ to the category of **E**-categories induced by the embedding of **E** into $\mathbf{E}[\mathcal{S}]$.

5.4.1 Definition Let $\mathcal{S}$ be an **E**-sketch and $\mathscr{C}$ an **E**-category. An **E**-functor from $\mathsf{CatTh}[\mathbf{E}, \mathcal{S}]$ to $\mathscr{C}$ is called a **model** of $\mathcal{S}$. A **morphism of models** in $\mathscr{C}$ is a natural transformation. The category of models of $\mathcal{S}$ in $\mathscr{C}$ is denoted by $\mathbf{Mod}[\mathcal{S}, \mathscr{C}]$.

5.4.2 Remark Each model of an **E**-sketch $\mathcal{S}$ is the unique functor induced (because $\mathsf{CatTh}[\mathbf{E}, \mathcal{S}]$ is an initial object) by a model $\mathfrak{M}$ of $\mathbf{E}[\mathcal{S}]$

in **Set** (a finite-limit-preserving functor from $\mathsf{E}[S]$ to **Set**) that has $\mathscr{C}$ as its underlying E-category. This passage from model of S to model of $\mathsf{E}[S]$ induces an equivalence of categories between the category of E-functors from $\mathsf{CatTh}[\mathsf{E}, S]$ to $\mathscr{C}$ and models (FinLim-functors) $\mathfrak{M}$ of $\mathsf{E}[S]$ in **Set** that have $\mathscr{C}$ as underlying E-category.

5.4.3 Example In Example 5.3.3, a model of the traditional sketch S in a finite-limit category $\mathscr{C}$ in the sense of Definition 5.4.1 is a finite-limit preserving functor from $\mathsf{CatTh}[\mathsf{FinLim}, S]$ (which is the initial model of $\mathsf{FinLim}[S]$ that takes S to S) to $\mathscr{C}$. The finite-limit functor from $\mathsf{FinLim}[S]$ to **Set** mentioned in Remark 5.4.2 is the extension of the model $\mathfrak{C}\colon \mathsf{FinLim} \to \mathbf{Set}$ that "corresponds to" or "is" (see Remark 4.2) $\mathscr{C}$ obtained by setting the value of the global element S to be the traditional sketch S.

6 Assertions in graph-based logic

6.1 Potential factorizations

In a constructor space E, a diagram of the form

$$\begin{tikzcd} & C \arrow[d, "j"] \\ A \arrow[r, "f"'] & B \end{tikzcd} \tag{22}$$

is called a **potential factorization** or **PF**. B is called the **workspace** of the potential factorization, and j is called the **claim** of the potential factorization. In many examples, including all theories in this paper, j is monic and corresponds to a formal selection of a subset of those objects formally denoted by B.

6.1.1 Notation

We will use suggestive notation for a potential factorization, as exhibited in the diagram below.

$$\begin{tikzcd} & \mathsf{claim} \arrow[d, "\mathsf{claimcon}"] \\ \mathsf{hyp} \arrow[r, "\mathsf{hypcon}"'] & \mathsf{wksp} \end{tikzcd} \tag{23}$$

The names hypcon, $\mathit{claimcon}$ and wksp respectively abbreviate "hypothesis construction", "claim construction" and "workspace". The reason for the names of the arrows and nodes is discussed in 7.3.

6.1.2 Actual factorizations

If an actual factorization arrow $\mathit{verif} : \mathit{hyp} \to \mathit{claim}$ exists in $\mathbf{E}$ that makes

$$\begin{tikzcd} & \mathit{claim} \arrow[d, "\mathit{claimcon}"] \\ \mathit{hyp} \arrow[ur, "\mathit{verif}"] \arrow[r, "\mathit{hypcon}"'] & \mathit{wksp} \end{tikzcd} \tag{24}$$

commute, then we say that the potential factorization is **deducible**. If for some model $\mathfrak{M}$ of $\mathbf{E}$ there is an arrow ξ of **Set** that makes

$$\begin{tikzcd} & \mathfrak{M}(\mathit{claim}) \arrow[d, tail, "\mathfrak{M}(\mathit{claimcon})"] \\ \mathfrak{M}(\mathit{hyp}) \arrow[ur, "\xi"] \arrow[r, "\mathfrak{M}(\mathit{hypcon})"'] & \mathfrak{M}(\mathit{wksp}) \end{tikzcd} \tag{25}$$

commute, then we say the model $\mathfrak{M}$ **satisfies** the potential factorization. If for *every* model $\mathfrak{M}$ there is such an arrow ξ then we say that the potential factorization is **valid**.

We give examples of potential factorizations in Section 7, and then discuss the general concept of potential factorization in Section 7.3.

6.2 Completeness

6.2.1 Theorem *In any categorial theory* $\mathsf{CatTh}[\mathbf{E}, S]$*, a potential factorization is deducible if and only if it is valid.*

Proof That deducibility implies validity follows from the fact that functors preserve factorizations.

For the converse, let

$$\begin{tikzcd} & \mathit{claim} \arrow[d, "\mathit{claimcon}"] \\ \mathit{hyp} \arrow[r, "\mathit{hypcon}"'] & \mathit{wksp} \end{tikzcd} \tag{26}$$

be a potential factorization in $\mathsf{CatTh}[\mathbf{E}, S]$. Suppose it is valid. Because the functor $\mathrm{Hom}(\mathit{hyp}, -)$ is a model, the hypothesis of the theorem implies that we may choose an arrow ξ of **Set** such that the diagram

$$\begin{tikzcd} & \mathrm{Hom}(\mathit{hyp}, \mathit{claim}) \arrow[d, "\mathrm{Hom}(\mathit{hyp}, \mathit{claimcon})"] \\ \mathrm{Hom}(\mathit{hyp}, \mathit{claim}) \arrow[ur, "\xi"] \arrow[r, "\mathrm{Hom}(\mathit{hyp}, \mathit{hypcon})"'] & \mathrm{Hom}(\mathit{hyp}, \mathit{wksp}) \end{tikzcd} \tag{27}$$

commutes. Define $\mathit{verif} := \xi(\mathrm{id}_{\mathit{hyp}})$. Then $\mathit{claimcon} \circ \mathit{verif} = \mathit{hypcon}$, so that the potential factorization is deducible.

Any model $\mathfrak{M}$ is a functor that preserves finite limits, so that if $\mathit{claimcon}$ is monic, so is $\mathfrak{M}(\mathit{claimcon})$. In that case, $\mathfrak{M}(\mathit{verif})$ necessarily equals ξ. □

Section B.7.2 gives an example of how completeness can be used.

6.2.2 Remark Traditional proofs of completeness contain an induction which is missing in the preceding argument. In the present system, a theorem can be identified with an arrow of $\mathsf{CatTh}[\mathbf{FinLim}, S]$. Proposition 4.1.1 describes the recursive construction of arrows and so is the analog of the inductive part of traditional proofs of completeness.

6.2.3 Proposition *Let $D : I \to \mathsf{CatTh}[\mathbf{E}, S]$ be a diagram. Suppose for every model $\mathfrak{M}$ of $\mathsf{CatTh}[\mathbf{E}, S]$, $\mathfrak{M} \circ D$ commutes. Then D commutes.*

Proof Suppose

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[dr, "h"] & B \arrow[d, "g"] \\ & C \end{tikzcd} \tag{28}$$

is a diagram in $\mathsf{CatTh}[\mathbf{E}, S]$. Then, because $\mathrm{Hom}(A, -)$ is a model,

$$\begin{tikzcd} \mathrm{Hom}(A, A) \arrow[r, "\mathrm{Hom}(A, f)"] \arrow[dr, "\mathrm{Hom}(A, h)"'] & \mathrm{Hom}(A, B) \arrow[d, "\mathrm{Hom}(A, g)"] \\ & \mathrm{Hom}(A, C) \end{tikzcd}$$

commutes. By chasing $\mathsf{Id}[A]$ around the diagram both ways, we get $g \circ f = h$, so Diagram (28) commutes as well.□

7 Examples of potential factorizations

We give two examples of potential factorizations in some detail. The example in 7.1 holds for any category and the example in 7.2 holds in any Cartesian closed category. We construct actual factorizations for these PF's in Section 9.

7.1 A fact about diagrams in any category

The following proposition holds in any category.

7.1.1 Proposition *In any category, given the following diagram*

$$\begin{tikzcd} A \arrow[r, "h"] \arrow[d, "f"'] & B \arrow[d, "k"] \\ C \arrow[r, "g"'] \arrow[ur, "x"] & D \end{tikzcd} \tag{29}$$

if the two triangles commute then so does the outside square.

Proof $k \circ h = k \circ (x \circ f) = (k \circ x) \circ f = g \circ f$.□

We construct here the potential factorization in $\mathsf{CatTh}[\mathbf{FinLim}, \mathit{Cat}]$ that corresponds to this theorem.

As we pointed out in Section 5.2, the value in a model of $\mathsf{Lim}[D(16)]$ is a diagram such as Diagram (29). However, Diagram (16) only carries the information as to the source and target of the arrows in Diagram (29). The structure we must actually work with includes the information as to which pairs are composable. There are four composable pairs in Diagram (29), and each one inhabits the value of ar_2, the node of formal composable pairs in a category (see Appendix B.2).

The following diagram thus contains the basic information about sources, targets and composability that are required for stating the theorem, and so its limit is suitable to be ***wksp***.

$$
\begin{tikzcd}
 & & \mathsf{ar}_2^{\langle g,f\rangle} \arrow[dl, "\mathsf{rfac}"'] \arrow[dr, "\mathsf{lfac}"] & & \\
 & \mathsf{ar}^f \arrow[r, "\mathsf{target}"] \arrow[dl, "\mathsf{source}"'] & \mathsf{ob}^C & \mathsf{ar}^g \arrow[l, "\mathsf{source}"'] \arrow[dr, "\mathsf{target}"] & \\
\mathsf{ob}^A & \mathsf{ar}_2^{\langle x,f\rangle} \arrow[u, "\mathsf{rfac}"] \arrow[r, "\mathsf{lfac}"] & \mathsf{ar}^x \arrow[u, "\mathsf{source}"] \arrow[d, "\mathsf{target}"] & \mathsf{ar}_2^{\langle k,x\rangle} \arrow[l, "\mathsf{rfac}"] \arrow[d, "\mathsf{lfac}"] & \mathsf{ob}^D \\
 & \mathsf{ar}^h \arrow[r, "\mathsf{target}"] \arrow[ul, "\mathsf{source}"] & \mathsf{ob}^B & \mathsf{ar}^k \arrow[l, "\mathsf{source}"'] \arrow[ur, "\mathsf{target}"'] & \\
 & & \mathsf{ar}_2^{\langle k,h\rangle} \arrow[ul, "\mathsf{rfac}"] \arrow[ur, "\mathsf{lfac}"'] & &
\end{tikzcd}
\tag{30}
$$

The statement that the two triangles in Diagram (29) commute is: $x \circ f = h$ and $k \circ x = g$. Using the composition arrow $\mathsf{comp}{:}\mathsf{ar}_2 \to \mathsf{ar}$ of $\mathsf{CatTh}[\textbf{FinLim}, \mathit{Cat}]$, this statement amounts to saying that Diagram (29) is a member of $\mathfrak{C}(\mathsf{Lim}[D(31)])$, which we take as ***hyp***:

$$
\begin{tikzcd}
 & & \mathsf{ar}_2^{\langle g,f\rangle} \arrow[dl, "\mathsf{rfac}"'] \arrow[dr, "\mathsf{lfac}"] & & \\
 & \mathsf{ar}^f \arrow[r, "\mathsf{target}"] \arrow[dl, "\mathsf{source}"'] & \mathsf{ob}^C & \mathsf{ar}^g \arrow[l, "\mathsf{source}"'] \arrow[dr, "\mathsf{target}"] & \\
\mathsf{ob}^A & \mathsf{ar}_2^{\langle x,f\rangle} \arrow[u, "\mathsf{rfac}"] \arrow[r, "\mathsf{lfac}"] \arrow[d, "\mathsf{comp}"] & \mathsf{ar}^x \arrow[u, "\mathsf{source}"] \arrow[d, "\mathsf{target}"] & \mathsf{ar}_2^{\langle k,x\rangle} \arrow[l, "\mathsf{rfac}"] \arrow[d, "\mathsf{lfac}"] \arrow[u, "\mathsf{comp}"] & \mathsf{ob}^D \\
 & \mathsf{ar}^h \arrow[r, "\mathsf{target}"] \arrow[ul, "\mathsf{source}"] & \mathsf{ob}^B & \mathsf{ar}^k \arrow[l, "\mathsf{source}"'] \arrow[ur, "\mathsf{target}"'] & \\
 & & \mathsf{ar}_2^{\langle k,h\rangle} \arrow[ul, "\mathsf{rfac}"] \arrow[ur, "\mathsf{lfac}"'] & &
\end{tikzcd}
\tag{31}
$$

The statement that the outside of Diagram (29) commutes is that the diagram is a member of $\mathfrak{C}(\mathsf{Lim}[\delta(32)])$, which we take as *claim*:

$$
\begin{tikzcd}
 & & \mathsf{ar}_2^{\langle g,f\rangle} \arrow[dl, "\mathsf{rfac}"'] \arrow[dr, "\mathsf{lfac}"] \arrow[drrr, "\mathsf{comp}"] & & & \\
 & \mathsf{ar}^f \arrow[r, "\mathsf{target}"'] \arrow[dl, "\mathsf{source}"'] & \mathsf{ob}^C & \mathsf{ar}^g \arrow[l, "\mathsf{source}"] \arrow[dr, "\mathsf{target}"] & & \\
\mathsf{ob}^A & \mathsf{ar}_2^{\langle x,f\rangle} \arrow[u, "\mathsf{rfac}"] \arrow[r, "\mathsf{lfac}"] & \mathsf{ar}^x \arrow[u, "\mathsf{source}"] \arrow[d, "\mathsf{target}"] & \mathsf{ar}_2^{\langle k,x\rangle} \arrow[l, "\mathsf{rfac}"] \arrow[d, "\mathsf{lfac}"] & \mathsf{ob}^D & \mathsf{ar}^{k\circ x\circ f} \\
 & \mathsf{ar}^h \arrow[ul, "\mathsf{source}"] \arrow[r, "\mathsf{target}"] & \mathsf{ob}^B & \mathsf{ar}^k \arrow[l, "\mathsf{source}"'] \arrow[ur, "\mathsf{target}"] & & \\
 & & \mathsf{ar}_2^{\langle k,h\rangle} \arrow[ul, "\mathsf{rfac}"] \arrow[ur, "\mathsf{lfac}"] \arrow[urrr, "\mathsf{comp}"'] & & &
\end{tikzcd}
\tag{32}
$$

Because Diagram (30) and Diagram (31) are restrictions of Diagram (32), we may, by Lemma 3.2.1 choose a function

$$\mathit{claimcon} : \mathsf{Lim}[D(32)] \to \mathsf{Lim}[D(30)]$$

and similarly another function

$$\mathit{hypcon} : \mathsf{Lim}[D(31)] \to \mathsf{Lim}[D(30)]$$

producing a potential factorization

$$
\begin{tikzcd}
 & \mathsf{Lim}[D(32)] \arrow[d, "\mathit{claimcon}"] \\
\mathsf{Lim}[D(31)] \arrow[r, "\mathit{hypcon}"'] & \mathsf{Lim}[D(30)]
\end{tikzcd}
\tag{33}
$$

This potential factorization expresses the content of Proposition 7.1.1 in diagrammatic form. It should be clear that the node $\mathsf{Lim}[D(30)]$ could have been replaced by $\mathsf{Lim}[D(16)]$.

7.2 A fact about Cartesian closed categories

Proposition 7.1.1 holds in any category. We now discuss a theorem of Cartesian closed categories, to show how the system presented in this paper handles structure that cannot be expressed using Ehresmann sketches. The latter are equivalent in expressive power to ordinary first order logic. (An excellent presentation of the details of this fact may be found in [Adámek and Rosičky, 1994].) Thus this example in a certain sense requires higher-order logic.

7.2.1 Proposition *In any Cartesian closed category, if*

$$\begin{tikzcd} A \times B \arrow[r, "g"] \arrow[dr, "h"'] & C \arrow[d, "f"] \\ & D \end{tikzcd}$$

commutes, then so does

$$\begin{tikzcd} A \arrow[r, "\lambda g"] \arrow[dr, "\lambda h"'] & C^B \arrow[d, "f^B"] \\ & D^B \end{tikzcd}$$

See Appendix B.6 for notation. The arrow f^B is defined by

$$f^B := \lambda(C^B \times B \xrightarrow{\text{eval}} C \xrightarrow{f} D) : C^B \longrightarrow D^B \tag{34}$$

The proof follows from the fact that λ is invertible and the calculation

$$\begin{aligned} \text{eval} \circ \big((f^B \circ \lambda g) \times \mathsf{Id}[B]\big) &= \text{eval} \circ \big((f^B \times \mathsf{Id}[B]) \circ (\lambda g \times \mathsf{Id}[B])\big) \\ &= \big(\text{eval} \circ (f^B \times \mathsf{Id}[B])\big) \circ (\lambda g \times \mathsf{Id}[B]) \\ &= (f \circ \text{eval}) \circ (\lambda g \times \mathsf{Id}[B]) \\ &= f \circ (\text{eval} \circ (\lambda g \times \mathsf{Id}[B])) \\ &= f \circ g = h = \text{eval} \circ (\lambda h \times \mathsf{Id}[B]) \end{aligned}$$

The first equality is based on an assertion true in all categories that can be handled in our system in a manner similar to (but more complicated than) that of module (63) in Appendix B.4. The second and fourth equalities are associativity of composition and are proven using Figure (53) of Appendix B.2. The sixth equality is a hypothesis. The other three equalities are all based on Diagram (83) in Appendix B.6.

We present here the potential factorization corresponding to the third equality, which is the most complicated of those based on Diagram (83). In this presentation, unlike that of 7.1, we will use the modules developed in Appendix B to simplify the figures. The actual factorization corresponding to this potential factorization is given in 9.2.

The fact under discussion is that the diagram

$$\begin{tikzcd} A\times B \arrow[r, "\lambda g\times \mathsf{Id}[B]"] & C^B\times B \arrow[r, "f^B\times \mathsf{Id}[B]"] \arrow[d, "\mathrm{eval}"'] & D^B\times B \arrow[d, "\mathrm{eval}"] \\ & C \arrow[r, "f"] & D \end{tikzcd} \tag{35}$$

commutes.

Thus ***wksp*** will be the limit of the following diagram, which describes the objects and arrows in Diagram (35) but has no requirements on its commutativity.

$$\begin{tikzcd} \mathsf{ar}^{\lambda g\times \mathsf{Id}[B]} \arrow[r, "\mathsf{target}"] \arrow[d, "\mathsf{source}"'] & \mathsf{ob}^{C^B\times \mathsf{Id}[B]} & \mathsf{ar}^{f^B\times \mathsf{Id}[B]} \arrow[l, "\mathsf{source}"'] \arrow[r, "\mathsf{target}"] & \mathsf{ob}^{D^B\times \mathsf{Id}[B]} \\ \mathsf{ob}^{A\times B} & \mathsf{ar}^{\mathrm{eval}} \arrow[u, "\mathsf{source}"] \arrow[d, "\mathsf{target}"'] & & \mathsf{ar}^{\mathrm{eval}} \arrow[u, "\mathsf{source}"'] \arrow[d, "\mathsf{target}"] \\ & \mathsf{ob}^{C} & \mathsf{ar}^{f} \arrow[l, "\mathsf{source}"] \arrow[r, "\mathsf{target}"'] & \mathsf{ob}^{D} \end{tikzcd} \tag{36}$$

We define ***hyp*** to be the limit of the following diagram, in which $\phi =$

$(f \circ \mathrm{eval}) \circ (\lambda g \times \mathsf{Id}[B])$.

$$
\begin{tikzcd}
 & & \mathsf{ob}^{D^B \times \mathsf{Id}[B]} & & \\
 & \mathsf{ar}^{f^B \times \mathsf{Id}[B]} \arrow[ur, "\mathsf{target}"] \arrow[dl, "\mathsf{source}"'] & \mathsf{ar}_2^{\langle \mathrm{eval}, f^B \times \mathsf{Id}[B]\rangle} \arrow[l, "\mathsf{rfac}"'] \arrow[rr, "\mathsf{lfac}"] & & \mathsf{ar}^{\mathrm{eval}} \arrow[ull, "\mathsf{source}"'] \arrow[ddd, "\mathsf{target}"] \\
\mathsf{ob}^{C^B \times B} & \mathsf{ar}^{\lambda g \times \mathsf{Id}[B]} \arrow[l, "\mathsf{target}"'] \arrow[r, "\mathsf{source}"] & \mathsf{ob}^{A \times B} & \mathsf{ar}^{\phi} \arrow[l, "\mathsf{source}"'] & \\
\mathsf{ar}^{\mathrm{eval}} \arrow[u, "\mathsf{source}"] \arrow[d, "\mathsf{target}"'] & \mathsf{ar}^{f \circ \mathrm{eval}} \arrow[ul, "\mathsf{source}"] \arrow[drrr, "\mathsf{target}"] & \mathsf{ar}_2^{\langle f \circ \mathrm{eval}, \lambda g \times \mathsf{Id}[B]\rangle} \arrow[ul, "\mathsf{rfac}"'] \arrow[ur, "\mathsf{comp}"] \arrow[l, "\mathsf{lfac}"'] & & \\
\mathsf{ob}^{C} & \mathsf{ar}_2^{\langle f, \mathrm{eval}\rangle} \arrow[ul, "\mathsf{rfac}"] \arrow[u, "\mathsf{comp}"] \arrow[d, "\mathsf{lfac}"] & & & \mathsf{ob}^{D} \\
 & \mathsf{ar}^{f} \arrow[ul, "\mathsf{source}"] \arrow[urrr, "\mathsf{target}"] & & &
\end{tikzcd}
\tag{37}
$$

Then Diagram (36) is a subdiagram of Diagram (37) (the big rectangle in Diagram (36) is the perimeter of Diagram (37)) and we define ***hypcon*** to be the induced arrow from $\mathsf{Lim}[D(37)]$ to $\mathsf{Lim}[D(36)]$.

claim is the limit of a diagram we shall refer to as Diagram (37′), obtained from Diagram (37) by adjoining an arrow labeled comp from $\mathsf{ar}_2^{\langle \mathrm{eval}, f^B \times B\rangle}$ to $\mathsf{ar}^{f \circ \mathrm{eval}}$. Diagram (37′) includes Diagram (37) and hence Diagram (36), and we take the arrow *claimcon* to be the arrow from Diagram (37′) to Diagram (36) induced by this inclusion. The existence of both *claimcon* and *hypcon* follows from Lemma 3.2.1.

We now have a potential factorization

$$
\begin{tikzcd}
 & \mathsf{Lim}[D(37')] \arrow[d, "\mathit{claimcon}"] \\
\mathsf{Lim}[D(37)] \arrow[r, "\mathit{hypcon}"'] & \mathsf{Lim}[D(36)]
\end{tikzcd}
\tag{38}
$$

The above potential factorization expresses the content of the third equality in the calculation in the proof of Proposition 7.2.1.

7.3 Discussion of the examples

7.3.1 General discussion

In a potential factorization

$$\begin{tikzcd} & \mathsf{claim} \arrow[d, "\mathsf{claimcon}"] \\ \mathsf{hyp} \arrow[r, "\mathsf{hypcon}"'] & \mathsf{wksp} \end{tikzcd} \tag{39}$$

each of hyp, claim and wksp represents a type of entity that can be constructed in an $\mathbf{E}$-category. In this section, we discuss a way of thinking about these nodes that exhibits how they could represent a theorem about $\mathbf{E}$-categories. This discussion is relevant to potential factorizations that are motivated by theorems stated in the traditional manner; it may not be relevant to an arbitrary potential factorization, which after all can be any pair of arrows with a common codomain.

1. The node wksp (for "workspace") represents the data involved in both the hypothesis and the conclusion. For a given theorem, the choice of what is actually included in wksp may be somewhat arbitrary (see the discussion of Example 7.2 below).

2. The node hyp represents possible additional properties that are part of the assumptions in the theorem being represented.

3. The node claim represents the properties that the theorem asserts must hold given the assumptions.

4. The arrow hypcon represents the selection or construction necessary to see the hypothesis as part of the workspace. In both our examples, hypcon represents a simple forgetting of properties.

5. The arrow $\mathsf{claimcon}$ represents the selection or construction necessary to see the claim as part of the workspace.

6. The arrow verif in an actual factorization represents a specific way, *uniform in any model*, that any entity of type hyp can be transformed into, or recognized as, an entity of type claim.

7.3.2 Discussion of Example 7.1

In Example 7.1,

1. *wksp* represent squares of the form

$$\begin{tikzcd} A \arrow[r, "h"] \arrow[d, "f"'] & B \arrow[d, "k"] \\ C \arrow[ur, "x"] \arrow[r, "g"] & D \end{tikzcd} \tag{40}$$

   with no commutativity conditions.

2. *hyp* represents diagrams of the form Diagram (40) in which the two triangles commute;

3. *claim* represents diagram of the form Diagram (40) in which the outside square commutes.

4. *hypcon* represents forgetting that the two triangles commute. Because it represents forgetting a property in this case, *hypcon* is monic, but in general it need not be.

5. *claimcon* represents forgetting that the outside square commutes.

7.3.3 Discussion of Example 7.2

In Example 7.2,

1. *wksp* represent diagrams of the form Diagram (35) with no commutativity conditions and no recognition that any sequence of arrows is composable. The phrase "The form of Diagram (35)" refers to the source and target commonalities of the arrows in the diagram. Obviously, some sequences compose but we have not represented that in *wksp*, although we could have.

2. *hyp* represents diagrams of the form of Diagram (35), recognizing the composite $f \circ \mathrm{eval}$ and the fact that

$$f \circ \mathrm{eval},\ \lambda g \times \mathsf{Id}[B]$$

   and eval, $f^B \times \mathsf{Id}[B]$ are composable pairs.

3. *claim* represents diagram of the form Diagram (35) that commute.
4. *hypcon* represents forgetting the information concerning composition in *hyp*.
5. *claimcon* represents forgetting the information concerning composition in *claim*.

We discuss the meaning of the actual factorization arrows *verif* in 9.3.

7.3.4 Remark The representation of facts such as those of Example 7.1 and 7.2 as potential factorizations is *variable free* in the sense that in each statement, one does not refer to a particular diagram such as Diagram (29) or Diagram (35) which stands as a pattern for all such diagrams. Propositions 7.1.1 and 7.2.1 state the fact in question using those diagrams as patterns, and understanding their meaning calls on the reader's ability to recognize patterns. Our description of the fact in the examples as a potential factorization is much more complicated because the diagrams involved in the potential factorization are essentially explicit descriptions of the relations between the nodes and arrows of Diagram (29) and Diagram (35) respectively, relations which a knowledgeable reader grasps from seeing the diagrams without having them indicated explicitly.

8 Construction and Deduction

8.1 The rules of graph-based logic

In first-order logic, rules are given for constructing terms and formulas, and further rules (rules of inference) are given for deriving formulas from formulas. These rules are intended to preserve truth.

In Appendix A we give rules for constructing all the objects and arrows of $\mathsf{CatTh}[\mathbf{FinLim}, \mathsf{S}]$ for an arbitrary finite-limit sketch S, and for constructing a basis for all the commutative diagrams in $\mathsf{CatTh}[\mathbf{FinLim}, \mathsf{S}]$. These rules correspond to both the term and formula construction rules and the rules of inference of string-based logic. The tools of a typical string-based logic include constant symbols, variables, function symbols, logical operators and quantifiers. Here we have nodes, arrows and commutative diagrams. What corresponds to a sentence is a pair of arrows as in Diagram (39) (a potential factorization), and what corresponds to the satisfiability of the sentence in a model $\mathfrak{M}$ is the existence of an arrow ξ for which

$$\begin{tikzcd} & \mathfrak{M}(\mathit{hyp}) \arrow[d, "\mathfrak{M}(\mathit{claimcon})"] \\ \mathfrak{M}(\mathit{claim}) \arrow[ur, "\xi"] \arrow[r, "\mathfrak{M}(\mathit{hypcon})"'] & \mathfrak{M}(\mathit{wksp}) \end{tikzcd} \tag{41}$$

commutes. A demonstration of the deducibility of the sentence corresponds to the construction of an arrow $\mathit{verif} : \mathit{claim} \to \mathit{hyp}$ in $\mathsf{CatTh}[\mathbf{FinLim}, S]$ such that

$$\begin{tikzcd} & \mathit{hyp} \arrow[d, "\mathit{claimcon}"] \\ \mathit{claim} \arrow[ur, "\mathit{verif}"] \arrow[r, "\mathit{hypcon}"'] & \mathit{wksp} \end{tikzcd} \tag{42}$$

commutes. Thus the same rules suffice for constructing the sentence (the potential factorization) and for proving it (constructing the arrow that makes it an actual factorizaton). (See Section 7.3 for further discussion of these points.)

Each rule in Appendix A is actually a rule scheme, and each instance of the scheme is an assertion that, given certain arrows and commutative diagrams (see Remark 8.1.1 below) in $\mathsf{CatTh}[\mathbf{FinLim}, S]$, other arrows or commutative diagrams exist in $\mathsf{CatTh}[\mathbf{FinLim}, S]$. For example, if δ is a diagram with shape graph

$$\begin{tikzcd} & i \arrow[d, "u"] \\ j \arrow[r, "v"'] & k \end{tikzcd} \tag{43}$$

then the following rule is an instance of ∃LIM:

$$\dfrac{\begin{tikzcd} & \delta(i) \arrow[d, "\delta(u)"] \\ \delta(j) \arrow[r, "\delta(u)"'] & \delta(k) \end{tikzcd}}{\begin{tikzcd} \mathsf{Lim}[\delta] \arrow[r, "{\mathsf{Proj}[\mathsf{Lim}[\delta], i]}"] \arrow[d, "{\mathsf{Proj}[\mathsf{Lim}[\delta], j]}"'] \arrow[dr, "{\mathsf{Proj}[\mathsf{Lim}[\delta], k]}" description] & \delta(i) \arrow[d] \\ \delta(j) \arrow[r] & \delta(k) \end{tikzcd}} \tag{44}$$

8.1.1 Remark

Rules ∃FIA, !FIA and CFIA assume the existence of commutative cones, but a commutative cone is a collection of interrelated commutative diagrams, so the statement above that each scheme assumes the existence of certain arrows and commutative diagrams is correct. Thus given a cone Θ, an instance of ∃FIA is this rule:

$$\dfrac{\begin{tikzcd} \mathsf{Vertex}[\Theta] \arrow[r, "{\mathsf{Proj}[\Theta, i]}"] \arrow[d, "{\mathsf{Proj}[\Theta, j]}"'] \arrow[dr, "{\mathsf{Proj}[\Theta, k]}" description] & \delta(i) \arrow[d] \\ \delta(j) \arrow[r] & \delta(k) \end{tikzcd}}{\begin{tikzcd} \mathsf{Vertex}[\Theta] \arrow[r, "{\mathsf{Fillin}[\Theta, \delta]}"] & \mathsf{Lim}[\delta] \end{tikzcd}} \tag{45}$$

The point of this remark is that ∃FIA is a rule with a diagram as hypothesis and an arrow as conclusion. The hypothesis is the cone itself, not the string "$\Theta : v \lhd (\delta : I \to \mathscr{C})$" or any other description of it.

9 Examples of theorems

9.1 Proof of Proposition 7.1.1

We continue Example 7.1 by constructing and thereby deducing the existence of an arrow *verif* : $\mathsf{Lim}[D(31)] \to \mathsf{Lim}[D(32)]$ making Diagram (33) commute.

We first construct Diagram (31′) (not shown) by adjoining ar_3 to Diagram (31), along with arrows

$$\mathsf{lfac} : \mathsf{ar}_3 \to \mathsf{ar}^k$$
$$\mathsf{mfac} : \mathsf{ar}_3 \to \mathsf{ar}^x$$
$$\mathsf{rfac} : \mathsf{ar}_3 \to \mathsf{ar}^f$$

We further construct Diagram (31″) by adjoining

$$\langle \mathsf{lfac}, \mathsf{mfac} \rangle : \mathsf{ar}_3 \to \mathsf{ar}_2^{\langle k,x \rangle}$$
$$\langle \mathsf{mfac}, \mathsf{rfac} \rangle : \mathsf{ar}_3 \to \mathsf{ar}_2^{\langle x,f \rangle}$$
$$\mathsf{lass} : \mathsf{ar}_3 \to \mathsf{ar}^{\langle g,f \rangle}$$
$$\mathsf{rass} : \mathsf{ar}_3 \to \mathsf{ar}^{\langle k,h \rangle}$$

The first two are induced by the fact that ar_3 is a limit and the other two are defined in Appendix B.2.

Diagram (30) is a base restriction of each of Diagram (31′) and Diagram (31″), so, we may, using Lemma 3.2.1, choose arrows

$$\phi_1 : \mathsf{Lim}[D(31')] \to \mathsf{Lim}[D(30)]$$

and

$$\phi_2 : \mathsf{Lim}[D(31'')] \to \mathsf{Lim}[D(30)]$$

Diagram (31) is a dominant subdiagram of Diagram (31′) since the latter is obtained from the former by adjoining a limit of a subdiagram together with their projection arrows (lfac, mfac and rfac). Therefore, using Lemma 3.3.4, we may choose an isomorphism ψ_1 making

$$\begin{tikzcd} \mathsf{Lim}[D(31)] \arrow[rr, "\psi_1"] \arrow[dr, "hypcon"'] & & \mathsf{Lim}[D(31')] \arrow[dl, "\phi_1"] \\ & \mathsf{Lim}[D(30)] & \end{tikzcd} \tag{46}$$

commute.

Similarly, Diagram (31′) is a dominant subdiagram of Diagram (31″) since the latter is obtained from the former by adjoining four composites. Therefore by Lemma 3.3.4 we may choose an isomorphism ψ_2 making

$$\begin{tikzcd} \mathsf{Lim}[D(31')] \arrow[rr, "\psi_2"] \arrow[dr, "\phi_1"'] & & \mathsf{Lim}[D(31'')] \arrow[dl, "\phi_2"] \\ & \mathsf{Lim}[D(30)] & \end{tikzcd} \tag{47}$$

commute. We then construct Diagram (31‴) by adjoining arrows

$$\mathsf{comp} : \mathsf{ar}_2^{\langle g,f\rangle} \to \mathsf{ar}^{k\circ x\circ f}$$

and

$$\mathsf{comp} : \mathsf{ar}_2^{\langle k,h\rangle} \to \mathsf{ar}^{k\circ x\circ f}$$

where $\mathsf{ar}^{k\circ x\circ f}$ is a new node.

Because of associativity (the right diagram in Figure (53) of Appendix B.2), Diagram (31‴) extends Diagram (31″) by adjoining composites, so we may choose an isomorphism $\psi_3 : \mathsf{Lim}[D(31'')] \to \mathsf{Lim}[D(31''')]$ and an arrow $\phi_3 : \mathsf{Lim}[D(31''')] \to \mathsf{Lim}[D(30)]$ making

$$\begin{tikzcd} \mathsf{Lim}[D(31'')] \arrow[rr, "\psi_3"] \arrow[dr, "\phi_2"'] & & \mathsf{Lim}[D(31''')] \arrow[dl, "\phi_3"] \\ & \mathsf{Lim}[D(30)] & \end{tikzcd} \tag{48}$$

commute.

Finally, by Lemma 3.2.1, we may choose arrow $\psi_4 : \mathsf{Lim}[D(31''')] \to \mathsf{Lim}[D(32)]$ making

$$\begin{tikzcd} \mathsf{Lim}[D(31''')] \arrow[rr, "\psi_4"] \arrow[dr, "\phi_3"'] & & \mathsf{Lim}[D(32)] \arrow[dl, "\mathit{claim}"] \\ & \mathsf{Lim}[D(30)] & \end{tikzcd} \tag{49}$$

commute.

We next set

$$\mathit{verif} := \psi_4 \circ \psi_3 \circ \psi_2 \circ \psi_1 \tag{50}$$

whence the theorem follows.

9.2 Proof of Theorem 7.2.1

In this section, we provide a factorization of the potential factorization described in Section 7.2.

Diagram (37) contains the following as a subdiagram, in which, using Diagram (34),

$$\theta = \langle \mathrm{eval}, f^B \times \mathsf{Id}[B] \rangle = \langle \mathrm{eval}, \lambda(f \circ \mathrm{eval}) \times \mathsf{Id}[B] \rangle$$

$$
\begin{tikzcd}
\mathsf{ob}^{C^B \times B} & & \mathsf{ar}^{f^B \times \mathsf{Id}[B]} \arrow[ll, "\mathsf{source}"'] \arrow[rr, "\mathsf{target}"] & & \mathsf{ob}^{D^B \times B} \\
& & \mathsf{ar}_2^{\theta} \arrow[u, "\mathsf{rfac}"] \arrow[dll, "\mathsf{comp}"'] \arrow[drr, "\mathsf{lfac}"] & & \\
\mathsf{ar}^{f \circ \mathrm{eval}} \arrow[uu, "\mathsf{source}"] \arrow[rr, "\mathsf{target}"'] & & \mathsf{ob}^{D} & & \mathsf{ar}^{\mathrm{eval}} \arrow[uu, "\mathsf{source}"'] \arrow[ll, "\mathsf{target}"]
\end{tikzcd}
\tag{51}
$$

This diagram is an instance of Diagram (83), so it commutes. The arrow comp satisfies Definition 3.3.1, so Lemma 3.3.4 implies that there is an isomorphism $\mathit{verif} : \mathit{hyp} \to \mathit{claim}$. Now the inclusion of Diagram (36) into Diagram (37) followed by the inclusion of Diagram (37) into Diagram (37′) is precisely the inclusion of Diagram (36) into Diagram (37′). It follows that $\mathit{hypcon} \circ \mathit{verif}^{-1} = \mathit{claimcon}$, so that $\mathit{claimcon} \circ \mathit{verif} = \mathit{hypcon}$ as required.

9.3 Discussion of the proofs.

The factorization $\mathit{verif} : \mathsf{Lim}[D(31)] \to \mathsf{Lim}[D(32)]$ of Diagram (33) given in Equation (50) constitutes the recognition that if the two triangles commute, then so does the outside square. The fact that verif makes Diagram (33) commute is a codification of the fact that if the two triangles commute then so does the outside square *of the same diagram*. In general, the reason we require that actual factorizations be an arrow in a certain comma category instead of merely an arrow from one node to another is to allow us to assert hypotheses and conclusions that share data (in this case the data in Diagram (40)).

The factorization $\mathit{verif} : \mathit{hyp} \to \mathit{claim}$ constructed in 9.2 constitutes recognition that $\mathrm{eval} \circ (f^B \times \mathsf{Id}[B]) = f \circ \mathrm{eval}$ via the arrow $\mathsf{comp} : \mathsf{ar}_2^{\langle \mathrm{eval}, f^B \times \mathsf{Id}[B] \rangle} \to \mathsf{ar}^{f \circ \mathrm{eval}}$ in Diagram (37′). Because of this, the node $\mathsf{ar}_2^{\langle f \circ \mathrm{eval}, \lambda g \times \mathsf{Id}[B] \rangle}$ could also be labeled $\langle \mathrm{eval} \circ f^B \times \mathsf{Id}[B], \lambda g \times \mathsf{Id}[B] \rangle$. Thus the factorization also exhibits the fact that

$$
\mathrm{eval} \circ (f^B \times \mathsf{Id}[B]) \circ (\lambda g \times \mathsf{Id}[B]) = f \circ \mathrm{eval} \circ (\lambda g \times \mathsf{Id}[B])
$$

It is clear that there are many alternative formulations of Proposition 7.2.1. For example, instead of first constructing $\mathsf{ar}_2^{f \circ \mathrm{eval}}$ as in Diagram (37) (which is hyp in this case), we could have constructed a node

$\mathsf{ar}_2^{\mathrm{eval}\circ(f^B\times\mathsf{Id}[B])}$ and an arrow

$$\mathsf{comp} : \mathsf{ar}_2^{\langle \mathrm{eval},(f^B\times\mathsf{Id}[B])\rangle} \to \mathsf{ar}_2^{\mathrm{eval}\circ(f^B\times\mathsf{Id}[B])}$$

Then the construction of an arrow

$$\mathsf{comp} : \mathsf{ar}_2^{\langle f,\mathrm{eval}\rangle} \to \mathsf{ar}^{\mathrm{eval}\circ(f^B\times\mathsf{Id}[B])}$$

would have proved the theorem.

10 Future work

A second part of this work [Bagchi and Wells, 1993a] will discuss in detail the relationship between the graph-based logic for finite-product categories and the rules for equational deduction.

It is noteworthy that the rules of construction for constructor spaces given in Appendix A correspond to arrows of **FinLim**, although not in a one-to-one way (see Remark A.1.1). The rules are given here in a form that requires pattern recognition (recall the discussion in 7.3.4), but they clearly could be given at another level of abstraction as arrows or families of arrows of **FinLim**. We expect to make this explicit in a later work.

M. Makkai [1993a], [1993b] has produced an approach to explicating the logic of sketches that is quite different from that presented here. Both are attempts at codifying the process of diagram-manipulation used to prove results valid in particular structured categories. In both cases, structured categories or doctrines form the semantic universes with a pre-existing notion of validity. Both approaches are motivated by the desire to formulate a syntactic notion of deducibility. After that is said, the two approaches are very different and a detailed investigation of the relationship between them is desirable.

The following points are however worth mention.

- Both approaches require a generalization of Ehresmann sketches: Sketches as in Definition 5.3.1 here and sketch category over a category G in [Makkai, 1993b].
- In Makkai's approach, the sketch axioms are different for different doctrines and serve both as axioms and as rules of inference. In contrast,

here the rules of construction are the *same* for all doctrines, what distinguishes doctrines is their specification as CS-sketches. This feature is a departure from the usual practice in symbolic logic.

The rules of construction given herein take place in the doctrine of finite limits. Most of the syntax and rules of deduction of traditional logical theories are clearly expressible using context-sensitive grammars, which intuitively at least can be modeled using finite limits. (Context-free grammars can be modeled using only finite products [Wells and Barr, 1988].) However, one could imagine extensions of this doctrine; for example, the use of coproducts to allow the specification of conditional compilation or the use of doctrines involving epimorphic covering families that allow the deliberate description of ambiguous statements.

A Appendix: Rules of construction

These rules are given in two lists.

A.1 Rules that construct objects and arrows

This list gives all the rules that construct objects and arrows in the category $\mathsf{CatTh}[\mathbf{FinLim}, \mathcal{S}]$.

∃OB $\dfrac{}{c}$ for every object c of $\mathcal{S}$.

∃ARR $\dfrac{}{a \xrightarrow{f} b}$ for every arrow $f : a \to b$ of $\mathcal{S}$.

∃COMP $\dfrac{a \xrightarrow{f} b \xrightarrow{g} c}{a \xrightarrow{g \circ f} c}$ for every object b and pair of arrows $f : a \to b$ and $g : b \to c$ of $\mathsf{CatTh}[\mathbf{FinLim}, \mathcal{S}]$.

∃ID $$\frac{c}{\begin{tikzcd} c \arrow[loop left, "\mathsf{Id}[c]"] \end{tikzcd}}$$ for every object c of $\mathsf{CatTh}[\textbf{FinLim}, S]$.

∃LIM $$\frac{\delta : I \to \mathsf{CatTh}[\textbf{FinLim}, S]}{\mathsf{LimCone}[\delta] : \mathsf{Lim}[\delta] \multimap \delta}$$ for every diagram $\delta : I \to \mathsf{CatTh}[\textbf{FinLim}, S]$.

∃FIA $$\frac{\Theta : v \multimap \delta}{\mathsf{Fillin}[\Theta, \delta] : v \to \mathsf{Lim}[\delta]}$$ for every diagram δ and every cone $\Theta : v \multimap \delta$ in $\mathsf{CatTh}[\textbf{FinLim}, S]$.

A.1.1 Remark The first two rules are justified by the inclusion of the sketch S into $\mathsf{CatTh}[\textbf{FinLim}, S]$. The rules ∃COMP corresponds to the arrow comp and ∃ID to unit of the sketch for categories B.2. ∃LIM and ∃FIA correspond to arrows in **FinLim** but not specifically to arrows of the sketch B.5, because an arbitrary finite limit is constructed from a combination of products and equalizers.

A.1.2 Remark

The rules just given construct specific objects and arrows in $\mathsf{CatTh}[\textbf{FinLim}, S]$. Rule ∃LIM, for example, constructs a specific limit cone called $\mathsf{LimCone}[\delta]$, thus providing canonical limits for $\mathsf{CatTh}[\textbf{FinLim}, S]$. It is true that there are other limit cones in general for a given diagram δ, but $\mathsf{LimCone}[\delta]$ is a specific one.

Of course, in many cases, the entity constructed is the unique entity satisfying some property. For example, the arrow $g \circ f$ constructed by ∃COMP is (by definition of commutative diagram) the only one making the bottom diagram in COMPDIAG commute in $\mathsf{CatTh}[\textbf{FinLim}, S]$. The arrow constructed by ∃ID is (by an easy theorem of category theory) the only one making the bottom diagrams in IDL and IDR commute. The arrow constructed by ∃FIA is (because of !FIA) the only one making the bottom diagram in CFIA commute. In connection with the point that each rule constructs a specific arrow, these observations are red herrings: in fact, each rule constructs a specific arrow with the name given, independently of any uniqueness properties arising from any other source.

This point of view is contrary to the spirit of category theory. We follow it here because we are constructing syntax with an eye toward implementing it in a computer language. It may be possible to design a computer language in which a construction such as ∃LIM produces a pointer to an isomorphism class of structures rather than to a specific one of them, but we know of no such language. On the other hand, implementing the specific constructions defined above should be relatively straightforward using a modern object-oriented language. Note that we are not asserting that it would be straightforward to find a confluent and normalizing form of these rules for automatic theorem proving, only that there are no obvious difficulties in implementing them so that they could be applied in an *ad-hoc* manner.

A.2 Rules that construct diagrams

The following rules produce the existence of diagrams that must commute.

$$\text{REF}\quad \frac{a \xrightarrow{f} b}{a \overset{f}{\underset{f}{\rightrightarrows}} b}$$

for every arrow $f : a \to b$ of $\mathsf{CatTh}[\mathbf{FinLim}, S]$

$$\text{TRANS}\quad \frac{a \overset{f}{\underset{g}{\rightrightarrows}} b \qquad a \overset{g}{\underset{h}{\rightrightarrows}} b}{a \overset{f}{\underset{h}{\rightrightarrows}} b}$$

for all objects a and b and all arrows $f, g, h : a \to b$ of $\mathsf{CatTh}[\mathbf{FinLim}, S]$

$$\exists\text{DIAG}\quad \frac{}{\mathsf{UnivMod}[\mathbf{FinLim}, S] \circ \delta : I \to \mathsf{CatTh}[\mathbf{FinLim}, S]}$$

for every diagram δ in the set D_S of diagrams of S.

COMPDIAG

$$\frac{\begin{tikzcd} a \arrow[r, "f"] & b \arrow[d, "g"] \\ & c \end{tikzcd}}{\begin{tikzcd} a \arrow[r, "f"] \arrow[dr, "g\circ f"'] & b \arrow[d, "g"] \\ & c \end{tikzcd}}$$

for every pair of arrows $f : a \to b$ and $g : b \to c$ of $\mathsf{CatTh}[\textbf{FinLim}, S]$.

IDL

$$\frac{\begin{tikzcd} c \arrow[loop above, "\mathsf{Id}[c]"] \end{tikzcd} \qquad g : b \to c}{\begin{tikzcd} b \arrow[r, "g"] \arrow[dr, "g"'] & c \arrow[d, "\mathsf{Id}[c]"] \\ & c \end{tikzcd}}$$

for every object c and every arrow $g : b \to c$ of $\mathsf{CatTh}[\textbf{FinLim}, S]$.

IDR

$$\frac{\begin{tikzcd} c \arrow[loop above, "\mathsf{Id}[c]"] \end{tikzcd} \qquad f : c \to d}{\begin{tikzcd} c \arrow[r, "\mathsf{Id}[c]"] \arrow[dr, "f"'] & c \arrow[d, "f"] \\ & d \end{tikzcd}}$$

for every object c and every arrow $f : c \to d$ of $\mathsf{CatTh}[\textbf{FinLim}, S]$.

ASSOC

$$\frac{\begin{tikzcd} a \arrow[r, "f"] & b \arrow[r, "g"] & c \arrow[r, "h"] & d \end{tikzcd}}{\begin{tikzcd} a \arrow[r, "f"] \arrow[d, "g\circ f"'] & b \arrow[dl, "g"'] \arrow[d, "h\circ g"] \\ c \arrow[r, "h"'] & d \end{tikzcd}}$$

for all arrows $f : a \to b$, $g : b \to c$ and $h : c \to d$ of $\mathsf{CatTh}[\textbf{FinLim}, S]$.

$$
\frac{\begin{array}{c} i \in \mathsf{Nodes}[I] \\ \Theta : v \lhd \delta \end{array}}{\begin{tikzcd} v \arrow[rr, "{\mathsf{Fillin}[\Theta,\delta]}"] \arrow[dr, "{\mathsf{Proj}[\Theta,i]}"'] & & \mathsf{Lim}[\delta] \arrow[dl, "{\mathsf{Proj}[\mathsf{LimCone}[\delta],i]}"] \\ & \delta(i) & \end{tikzcd}}
$$

CFIA

for every diagram $\delta{:}I \to \mathsf{CatTh}[\textbf{FinLim}, \mathsf{S}]$, every node i of I, and every cone Θ with base diagram δ.

!FIA

$$
\frac{\begin{array}{c} \delta : I \to \mathsf{CatTh}[\textbf{FinLim}, \mathsf{S}] \\ \Theta : v \lhd \delta \\ h : v \to \mathsf{Lim}[\delta] \\ k : v \to \mathsf{Lim}[\delta] \\ \text{and each of the following diagrams for each node } i \text{ of } I\text{:} \\ \begin{tikzcd} v \arrow[rr, "h"] \arrow[dr, "{\mathsf{Proj}[\Theta,i]}"'] & & \mathsf{Lim}[\delta] \arrow[dl, "{\mathsf{Proj}[\mathsf{LimCone}[\delta],i]}"] \\ & \delta(i) & \end{tikzcd} \\ \begin{tikzcd} v \arrow[rr, "k"] \arrow[dr, "{\mathsf{Proj}[\Theta,i]}"'] & & \mathsf{Lim}[\delta] \arrow[dl, "{\mathsf{Proj}[\mathsf{LimCone}[\delta],i]}"] \\ & \delta(i) & \end{tikzcd} \end{array}}{\begin{tikzcd} v \arrow[r, shift left, "h"] \arrow[r, shift right, "k"'] & \mathsf{Lim}[\delta] \end{tikzcd}}
$$

for every diagram $\delta{:}I \to \mathsf{CatTh}[\textbf{FinLim}, \mathsf{S}]$, every cone Θ in $\mathsf{CatTh}[\textbf{FinLim}, \mathsf{S}]$ with base diagram δ, and every pair of arrows $h, k : v \to \mathsf{Lim}[\delta]$.

A.2.1 Remark Note that we do not need a rule of the form

$$\text{SYM}\qquad \frac{a \underset{g}{\overset{f}{\rightrightarrows}} b}{a \underset{f}{\overset{g}{\rightrightarrows}} b}$$

since the two diagrams exhibited are actually the same diagram (see 2.3).

B Appendix: Sketches for constructor spaces

Here we present constructor space sketches for certain types of categories. In each case the models are categories of the sort described and the morphisms of models are functors that preserve the structure on the nose. It is an old result that such categories can be sketched. See [Burroni, 1970a], [Burroni, 1970b], [McDonald and Stone, 1984], and [Coppey and Lair, 1988], for example.

The embedding η of Section 5.1.3 will in each case be inclusion.

B.1 Notation

We denote the ith projection in a product diagram of the form

$$\begin{tikzcd} & \mathsf{ob}^A \times \mathsf{ob}^B \arrow[dl, "p_1"'] \arrow[dr, "p_2"] & \\ \mathsf{ob}^A & & \mathsf{ob}^B \end{tikzcd}$$

as p_i, or $p_i^{A\times B}$ if the source or target is not shown. We use a similar device for the product of three copies of ob.

B.2 The sketch *Cat* for categories

This version of the sketch for categories is based on [Barr and Wells, 1990]. Another version is given in [Coppey and Lair, 1988], page 64. The first versions were done by Ehresmann [1966], [1968a] and [1968b].

B.2.1 The graph of *Cat*

The graph of the sketch for categories contains nodes as follows.

1. $\mathsf{1}$, the formal terminal object.
2. ob, the formal set of objects.
3. ar, the formal set of arrows.
4. ar_2, the formal set of composable pairs of arrows.
5. ar_3, the formal set of composable triples of arrows.

The arrows for the sketch for categories are

1. $\mathsf{unit} : \mathsf{ob} \to \mathsf{ar}$ that formally picks out the identity arrow of an object.
2. $\mathsf{source}, \mathsf{target} : \mathsf{ar} \to \mathsf{ob}$ that formally pick out the source and target of an arrow.
3. $\mathsf{comp} : \mathsf{ar}_2 \to \mathsf{ar}$ that picks out the composite of a composable pair.
4. $\mathsf{lfac}, \mathsf{rfac}{:}\mathsf{ar}_2 \to \mathsf{ar}$ that pick out the left and right factors in a composable pair.
5. $\mathsf{lfac}, \mathsf{mfac}, \mathsf{rfac} : \mathsf{ar}_3 \to \mathsf{ar}$ that pick out the left, middle and right factors in a composable triple of arrows.
6. $\mathsf{lass}, \mathsf{rass} : \mathsf{ar}_3 \to \mathsf{ar}_2$: lass formally takes $\langle h, g, f\rangle$ to $\langle h \circ g, f\rangle$ and rass takes it to $\langle h, g \circ f\rangle$.
7. $\mathsf{lunit}, \mathsf{runit} : \mathsf{ar} \to \mathsf{ar}_2$: lunit takes an arrow $f : A \to B$ to $\langle \mathsf{Id}[B], f\rangle$ and runit takes it to $\langle f, \mathsf{Id}[A]\rangle$.
8. Arrows $\mathsf{id} : x \to x$ as needed.

Observe that id, lfac and rfac, like p_1 and p_2, are overloaded. We will observe the same care with these arrows as with p_1 and p_2 as mentioned in Section B.1.

B.2.2 Cones of *Cat*

ar_2 and ar_3 are defined by these cones:

$$
\begin{tikzcd}
& \mathsf{ar}_2 \arrow[dl, "\mathsf{lfac}"'] \arrow[dr, "\mathsf{rfac}"] & \\
\mathsf{ar} \arrow[dr, "\mathsf{source}"'] & & \mathsf{ar} \arrow[dl, "\mathsf{target}"] \\
& \mathsf{ob} &
\end{tikzcd}
\qquad
\begin{tikzcd}
& \mathsf{ar}_3 \arrow[dl, "\mathsf{lfac}"'] \arrow[d, "\mathsf{mfac}"] \arrow[dr, "\mathsf{rfac}"] & \\
\mathsf{ar} \arrow[d, "\mathsf{source}"'] & \mathsf{ar} \arrow[dl, "\mathsf{target}"'] \arrow[dr, "\mathsf{source}"] & \mathsf{ar} \arrow[d, "\mathsf{target}"] \\
\mathsf{ob} & & \mathsf{ob}
\end{tikzcd}
$$

B.2.3 Diagrams of *Cat*

$$
\begin{tikzcd}
\mathsf{ar}_2 \arrow[d, "\mathsf{comp}"'] & \mathsf{ar}_3 \arrow[l, "\langle \mathsf{lfac}, \mathsf{mfac} \rangle"'] \arrow[d, "\mathsf{lass}"] \arrow[dr, "\mathsf{rfac}"] & \\
\mathsf{ar} & \mathsf{ar}_2 \arrow[l, "\mathsf{lfac}"] \arrow[r, "\mathsf{rfac}"'] & \mathsf{ar}
\end{tikzcd}
\qquad
\begin{tikzcd}
& \mathsf{ar}_3 \arrow[r, "\langle \mathsf{mfac}, \mathsf{rfac} \rangle"] \arrow[dl, "\mathsf{lfac}"'] \arrow[d, "\mathsf{rass}"] & \mathsf{ar}_2 \arrow[d, "\mathsf{comp}"] \\
\mathsf{ar} & \mathsf{ar}_2 \arrow[l, "\mathsf{lfac}"] \arrow[r, "\mathsf{rfac}"'] & \mathsf{ar}
\end{tikzcd}
\tag{52}
$$

$$
\begin{tikzcd}
\mathsf{ob} \arrow[d, "\mathsf{unit}"'] & \mathsf{ar} \arrow[l, "\mathsf{target}"'] \arrow[d, "\mathsf{lunit}"] \arrow[dr, "\mathsf{id}"] & \\
\mathsf{ar} & \mathsf{ar}_2 \arrow[l, "\mathsf{lfac}"] \arrow[r, "\mathsf{rfac}"'] & \mathsf{ar}
\end{tikzcd}
\qquad
\begin{tikzcd}
& \mathsf{ar} \arrow[r, "\mathsf{source}"] \arrow[dl, "\mathsf{id}"'] \arrow[d, "\mathsf{runit}"] & \mathsf{ob} \arrow[d, "\mathsf{unit}"] \\
\mathsf{ar} & \mathsf{ar}_2 \arrow[l, "\mathsf{lfac}"] \arrow[r, "\mathsf{rfac}"'] & \mathsf{ar}
\end{tikzcd}
$$

$$
\begin{tikzcd}
\mathsf{ar} \arrow[r, "\mathsf{runit}"] \arrow[dr, "\mathsf{id}"'] & \mathsf{ar}_2 \arrow[d, "\mathsf{comp}"] & \mathsf{ar} \arrow[l, "\mathsf{lunit}"'] \arrow[dl, "\mathsf{id}"] \\
& \mathsf{ar} &
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathsf{ar}_3 \arrow[r, "\mathsf{rass}"] \arrow[d, "\mathsf{lass}"'] & \mathsf{ar}_2 \arrow[d, "\mathsf{comp}"] \\
\mathsf{ar}_2 \arrow[r, "\mathsf{comp}"'] & \mathsf{ar}
\end{tikzcd}
\tag{53}
$$

B.3 The sketch for the constructor space **FinProd**

To get the sketch for categories with finite products, we must add the following nodes and arrows to the sketch for categories:

Nodes:

1. ta, the formal set of terminal arrows.
2. cone, the formal set of cones of the form

$$\begin{tikzcd} & V \arrow[dl, "p_1"'] \arrow[dr, "p_2"] & \\ A & & B \end{tikzcd} \tag{54}$$

3. fid, the formal set of fill-in diagrams ("sawhorses") of the form

$$\begin{tikzcd} V \arrow[r, "h"] \arrow[d] \arrow[dr] & L \arrow[d] \arrow[dl] \\ A & B \end{tikzcd} \tag{55}$$

where h commutes with the cone projections.

Arrows:

1. $\mathsf{ter} : 1 \to \mathsf{ob}$, that formally picks out a particular terminal object.
2. $! : \mathsf{ob} \to \mathsf{ta}$, that picks out the arrow from an object to the terminal object.
3. $\mathsf{inc} : \mathsf{ta} \to \mathsf{ar}$, the formal inclusion of the set of terminal arrows into the set of arrows.
4. $\mathsf{prod} : \mathsf{ob} \times \mathsf{ob} \to \mathsf{cone}$, that picks out the product cone over a pair of objects.
5. $\mathsf{soco} : \mathsf{fid} \to \mathsf{cone}$, that picks out the source cone of a fill-in arrow.
6. $\mathsf{taco} : \mathsf{fid} \to \mathsf{cone}$, that picks out the target cone of a fill-in arrow.
7. $\mathsf{ufid} : \mathsf{cone} \to \mathsf{fid}$, that takes a cone to the unique fill-in diagram that has the cone as source cone.
8. $\mathsf{fia} : \mathsf{fid} \to \mathsf{ar}$ that formally picks out the fill-in arrow in a fill-in diagram.

B.3.1 Cones for **FinProd**

FinProd has four cones in addition to those of the sketch for categories. One is the cone with vertex 1 over the empty diagram. The one below says that ta is the formal set of arrows to the terminal object:

$$\begin{tikzcd} & \mathsf{ta} \arrow[dl, "\mathsf{inc}"'] \arrow[dr] & \\ \mathsf{ar} \arrow[rr, "\mathsf{target}"] \arrow[dr] & & \mathsf{ob} \\ & 1 \arrow[ur, "\mathsf{ter}"'] & \end{tikzcd} \tag{56}$$

Note that in giving this cone, we are not only saying that ta is the limit of the diagram

$$\begin{tikzcd} \mathsf{ar} \arrow[rr, "\mathsf{target}"] \arrow[dr] & & \mathsf{ob} \\ & 1 \arrow[ur, "\mathsf{ter}"'] & \end{tikzcd} \tag{57}$$

but also that inc is one of the projection arrows. (Indeed, this is the only projection arrow that matters, since the other two are induced.)

The following cone makes cone the formal object of cones to a discrete diagrams consisting of a pair of objects.

$$\begin{tikzcd} & \mathsf{cone} \arrow[dl, "\mathsf{lproj}"'] \arrow[dr, "\mathsf{rproj}"] & \\ \mathsf{ar} \arrow[r, "\mathsf{source}"'] & \mathsf{ob} & \mathsf{ar} \arrow[l, "\mathsf{source}"] \end{tikzcd} \tag{58}$$

Finally, there must be a cone with vertex fid over Diagram 59 below, which is annotated to refer to Diagram (55), in which Γ is the cone with vertex V, Λ is the cone with vertex L and is a limit cone, and h is the fill-in arrow. In this case, the projection arrows of the cone are not shown.

$$
\begin{tikzcd}
\mathsf{ar}_2 \arrow[d, "\mathsf{comp}"] \arrow[ddd, bend right=60, "\mathsf{lfac}"'] \arrow[rrrr, bend left=20, "\mathsf{rfac}"] & & \mathsf{ob}^V & & \\
\mathsf{ar} \arrow[urr, "\mathsf{source}"] \arrow[d, "\mathsf{target}"] & \mathsf{cone}^\Gamma \arrow[l, "\mathsf{lproj}"] \arrow[r, "\mathsf{rproj}"'] & \mathsf{ar} \arrow[u, "\mathsf{source}"] \arrow[d, "\mathsf{target}"] & & \\
\mathsf{ob}^A & \mathsf{ob}^A \times \mathsf{ob}^B \arrow[l, "p_1"'] \arrow[r, "p_2"] \arrow[d, "\mathsf{prod}"] & \mathsf{ob}^B & \mathsf{ar}_2 \arrow[ul, "\mathsf{comp}"'] \arrow[r, "\mathsf{rfac}"] \arrow[dl, "\mathsf{lfac}"] & \mathsf{ar}^h \arrow[uull, "\mathsf{source}"'] \arrow[ddll, "\mathsf{target}"] \\
\mathsf{ar} \arrow[u, "\mathsf{target}"] \arrow[dr, "\mathsf{source}"'] & \mathsf{cone}^\Lambda \arrow[l, "\mathsf{lproj}"'] \arrow[r, "\mathsf{rproj}"] & \mathsf{ar} \arrow[u, "\mathsf{target}"] \arrow[d, "\mathsf{source}"] & & \\
& & \mathsf{ob}^L & &
\end{tikzcd}
\tag{59}
$$

In addition, we require:

1) The projection to cone^Γ must be soco.
2) The projection to cone^Λ must be taco.
3) The projection to ar^h must be fia.

B.3.2 Diagrams for **FinProd**

The following two diagrams make the arrow to the terminal object have the correct source and target.

$$
\begin{tikzcd}
\mathsf{ob} \arrow[r, "!"] \arrow[drr, "\mathsf{id}"'] & \mathsf{ta} \arrow[r, "\mathsf{inc}"] & \mathsf{ar} \arrow[d, "\mathsf{source}"] \\
& & \mathsf{ob}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathsf{ta} \arrow[r, "\mathsf{inc}"] \arrow[drr, "\mathsf{id}"'] & \mathsf{ar} \arrow[r, "\mathsf{source}"] & \mathsf{ob} \arrow[d, "!"] \\
& & \mathsf{ta}
\end{tikzcd}
\tag{60}
$$

The diagram below makes the fill-in arrow to a product unique.

$$
\begin{tikzcd}
\mathsf{cone} \arrow[r, shift left, "\mathsf{ufid}"] & \mathsf{fid} \arrow[l, shift left, "\mathsf{soco}"]
\end{tikzcd}
\tag{61}
$$

The diagram below forces the product cone projections to have the correct targets.

$$\begin{tikzcd}
& & \mathsf{ob} \times \mathsf{ob} \arrow[dll, "p_1"'] \arrow[d, "\mathsf{prod}"] \arrow[drr, "p_2"] & & \\
\mathsf{ob} & \mathsf{ar} \arrow[l, "\mathsf{target}"] & \mathsf{cone} \arrow[l, "\mathsf{lproj}"] \arrow[r, "\mathsf{rproj}"'] & \mathsf{ar} \arrow[r, "\mathsf{target}"'] & \mathsf{ob}
\end{tikzcd} \tag{62}$$

B.4 Modules

As we proceed to sketch more complicated constructions, we will need to use some device to communicate the nature of the necessary diagrams, which become too large to comprehend easily. Here we introduce the first of several **modules**: diagrams that occur frequently as subdiagrams because they are needed to force the value of a node in a model to contain certain types of constructions.

B.4.1 The module for the product of objects

Every occurrence of ob that is annotated $M \times N$ must be part of a subdiagram of the following form:

$$\begin{tikzcd}
\mathsf{ob}^M & \mathsf{ar}^{p_1^{M\times N}} \arrow[l, "\mathsf{target}"'] \arrow[dr, "\mathsf{source}"] & \\
\mathsf{ob}^M \times \mathsf{ob}^N \arrow[u, "p_1"] \arrow[d, "p_2"'] \arrow[r, "\mathsf{prod}"] & \mathsf{cone} \arrow[u, "\mathsf{lproj}"'] \arrow[d, "\mathsf{rproj}"] & \mathsf{ob}^{M\times N} \\
\mathsf{ob}^N & \mathsf{ar}^{p_2^{M\times N}} \arrow[l, "\mathsf{target}"] \arrow[ur, "\mathsf{source}"'] &
\end{tikzcd} \tag{63}$$

Henceforth, an occurrence of ob annotated $A \times B$ (for example) will be taken to *imply* the existence of a subdiagram of the form of Diagram (63) with M replaced with A and N replaced with B. The subdiagram will not necessarily be shown. If this is part of a diagram δ, the diagram can be reconstructed by taking the union of the shape graph of the module (63) and the shape graph of the part of δ that is shown on the page, and defining the diagram based on the resulting graph as the pushout of the diagram shown and the module. This is illustrated in Diagrams (65) and (66) in the next section.

B.4.2 The module for the product of arrows

In the commutative diagram

$$\begin{tikzcd}
K & K \times N \arrow[l, "p_1"'] \arrow[d] \arrow[dr, "p_2"] & \\
 & & N \\
M & M \times N \arrow[l, "p_2"] \arrow[ur, "p_2"'] &
\arrow[from=1-1, to=3-1, "u"']
\end{tikzcd} \tag{64}$$

the unlabeled arrow is necessarily $u \times \mathsf{Id}[N]: K \times N \to M \times N$. Such a diagram must be an element in a model of the value of Diagram (65) below, which is therefore a module for the product of an arrow and an identity arrow. In this diagram, $\phi := \langle p_2^{M\times N}, u \times \mathsf{Id}[N]\rangle$.

$$\begin{tikzcd}
 & & & & & \mathsf{ob}^K \times \mathsf{ob}^N \arrow[dl, "\mathsf{prod}"] \arrow[ddlllll, "p_1"'] \arrow[dddr, "p_2"] & \\
 & & & & \mathsf{cone} \arrow[dll, "\mathsf{lproj}"'] \arrow[dr, "\mathsf{rproj}"] & & \\
\mathsf{ob}^K & & \mathsf{ar}^{p_1^{K\times N}} \arrow[ll, "\mathsf{target}"'] \arrow[rr, "\mathsf{source}"] & & \mathsf{ob}^{K\times N} & \mathsf{ar}^{p_2^{K\times N}} \arrow[l, "\mathsf{source}"'] \arrow[dr, "\mathsf{target}"] & \\
\mathsf{ar}^u \arrow[u, "\mathsf{source}"] \arrow[d, "\mathsf{target}"'] & \mathsf{ar}_2^{\langle u,p_1\rangle} \arrow[l, "\mathsf{lfac}"'] \arrow[r, "\mathsf{comp}"] \arrow[ur, "\mathsf{rfac}"] & \mathsf{ar} & \mathsf{ar}_2^{\langle u\times \mathsf{Id}[N],p_2\rangle} \arrow[l, "\mathsf{comp}"'] \arrow[r, "\mathsf{rfac}"] \arrow[dl, "\mathsf{lfac}"'] & \mathsf{ar}^{u\times \mathsf{Id}[N]} \arrow[u, "\mathsf{source}"] \arrow[d, "\mathsf{target}"'] & \mathsf{ar}_2^{\phi} \arrow[l, "\mathsf{rfac}"'] \arrow[u, "\mathsf{comp}"] \arrow[d, "\mathsf{lfac}"'] & \mathsf{ob}^N \\
\mathsf{ob}^M & & \mathsf{ar}^{p_1^{M\times N}} \arrow[ll, "\mathsf{target}"'] \arrow[rr, "\mathsf{source}"] & & \mathsf{ob}^{M\times N} & \mathsf{ar}^{p_2^{M\times N}} \arrow[l, "\mathsf{source}"'] \arrow[ur, "\mathsf{target}"'] & \\
 & & & & \mathsf{cone} \arrow[ull, "\mathsf{lproj}"] \arrow[ur, "\mathsf{rproj}"'] & & \\
 & & & & & \mathsf{ob}^M \times \mathsf{ob}^N \arrow[ul, "\mathsf{prod}"'] \arrow[uulllll, "p_1"] \arrow[uuur, "p_2"'] &
\end{tikzcd} \tag{65}$$

More precisely, let x be an element of $\mathfrak{M}(\mathsf{Lim}[D(65)])$, for some category $\mathfrak{M}$ with finite limits. Then if $\mathsf{Proj}[\mathsf{Lim}[D(65)], h](x) = h$, then

$$\mathsf{Proj}[\mathsf{Lim}[D(65)], h\times \mathsf{Id}[A]](x) = h\times \mathsf{Id}[A]$$

as suggested by the notation. This will be used in Section B.6 below.

Diagram (65) contains two copies of Diagram (63), the module for the product of two objects. The copy at the bottom is precisely Diagram (63), and the copy at the top is Diagram (63) with M replaced with K. In the sequel, a diagram such as Diagram (65) will be drawn without the modules, as shown below.

$$\begin{tikzcd}
\mathsf{ob}^K & & \mathsf{ar}^{p_1^{K\times N}} \arrow[ll, "\mathsf{target}"'] \arrow[rr, "\mathsf{source}"] & & \mathsf{ob}^{K\times N} & \mathsf{ar}^{p_2^{K\times N}} \arrow[l, "\mathsf{source}"'] \arrow[dr, "\mathsf{target}"] & \\
\mathsf{ar}^u \arrow[u, "\mathsf{source}"] \arrow[d, "\mathsf{target}"'] & \mathsf{ar}_2^{\langle u,p_1^{K\times N}\rangle} \arrow[l, "\mathsf{lfac}"'] \arrow[ur, "\mathsf{rfac}"] \arrow[r, "\mathsf{comp}"] & \mathsf{ar} & \mathsf{ar}_2^{\langle u\times \mathsf{Id}[N],p_2\rangle} \arrow[l, "\mathsf{comp}"'] \arrow[r, "\mathsf{rfac}"] \arrow[dl, "\mathsf{lfac}"'] & \mathsf{ar}^{u\times \mathsf{Id}[N]} \arrow[u, "\mathsf{source}"] \arrow[d, "\mathsf{target}"'] & \mathsf{ar}_2^{\phi} \arrow[l, "\mathsf{rfac}"'] \arrow[u, "\mathsf{comp}"] \arrow[d, "\mathsf{lfac}"'] & \mathsf{ob}^N \\
\mathsf{ob}^M & & \mathsf{ar}^{p_1^{M\times N}} \arrow[ll, "\mathsf{target}"'] \arrow[rr, "\mathsf{source}"] & & \mathsf{ob}^{M\times N} & \mathsf{ar}^{p_2^{M\times N}} \arrow[l, "\mathsf{source}"'] \arrow[ur, "\mathsf{target}"'] &
\end{tikzcd} \tag{66}$$

Diagram (65) may be mechanically reconstructed from Diagram (66) and the annotations that include the symbols $M \times N$ and $K \times N$ (three of each). The shape diagram of Diagram (65) is the pushout of the shape diagram of Diagram (66) and the shape diagrams of the modules Diagram (63) and Diagram (63) with $M \leftarrow K$. Each of the latter two have four annotated nodes and six annotated arrows in common with Diagram (66), and the values of any two of the three smaller diagrams at a given common node or arrow is of course the same, so that Diagram (65) is the union of Diagram (66) and the two modules.

B.5 The sketch for the constructor space **FinLim**

We sketch the constructor space **FinLim** by adding data to the sketch for **FinProd** that ensure that a **FinLim**-category has equalizers of pairs of arrows. The sketch has the following nodes:

1. ppair is the formal set of parallel pairs of the form

$$
\begin{tikzcd} A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \end{tikzcd} \tag{67}
$$

2. econe is the formal set of diagrams

$$
\begin{tikzcd} E \arrow[r, "u"] & A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \end{tikzcd} \tag{68}
$$

   in which $f \circ u = g \circ u$. Of course, a cone to Diagram (67) also has a projection to B, but that is forced and need not be included in the data for the cone.

3. efid is the set of fill-in diagrams

$$
\begin{tikzcd} & X \arrow[dl, "v"'] \arrow[d, "u"] & \\ E \arrow[r, "e"] & A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \end{tikzcd} \tag{69}
$$

   in which $f \circ e = g \circ e$ and $u = e \circ v$.

The arrows of the sketch include:

1. $\mathsf{equ} : \mathsf{ppair} \to \mathsf{econe}$, that formally picks out the equalizer of the parallel pair.
2. $\mathsf{top}, \mathsf{bot} : \mathsf{ppair} \to \mathsf{ar}$, that pick out f and g in Diagram (67).
3. $\mathsf{etop}, \mathsf{ebot} : \mathsf{econe} \to \mathsf{ar}$, that pick out f and g in Diagram (68).
4. $\mathsf{esoco}, \mathsf{etaco} : \mathsf{efid} \to \mathsf{econe}$ that pick out the source and target cones of the fill-in arrow.
5. $\mathsf{eufid} : \mathsf{econe} \to \mathsf{efid}$ that takes a diagram of the form of Diagram (68) to the unique fill-in diagram that has this diagram as source cone.
6. $\mathsf{efia} : \mathsf{efid} \to \mathsf{ar}$ that picks out the fill-in arrow in a fill-in diagram.

B.5.1 Cones for **FinLim**

ppair is the limit of the diagram

$$\begin{tikzcd}
\mathsf{ar}^f \arrow[r, "\mathsf{target}"] \arrow[d, "\mathsf{source}"'] & \mathsf{ob}^B \\
\mathsf{ob}^A & \mathsf{ar}^g \arrow[l, "\mathsf{source}"'] \arrow[u, "\mathsf{target}"']
\end{tikzcd} \tag{70}$$

The following projections from ppair have names: $\mathsf{soob} : \mathsf{ppair} \to \mathsf{ob}^A$, $\mathsf{top} : \mathsf{ppair} \to \mathsf{ar}^f$, and $\mathsf{bot} : \mathsf{ppair} \to \mathsf{ar}^g$.

econe is the limit of

$$\begin{tikzcd}
 & \mathsf{ar}_2 \arrow[r, "\mathsf{lfac}"] \arrow[dl, "\mathsf{comp}"'] \arrow[d, "\mathsf{rfac}"] & \mathsf{ar}^f \arrow[d, "\mathsf{source}"] \arrow[dr, "\mathsf{target}"] & \\
\mathsf{ar} & \mathsf{ar}^e \arrow[r, "\mathsf{target}"] & \mathsf{ob}^A & \mathsf{ob}^B \\
 & \mathsf{ar}_2 \arrow[r, "\mathsf{lfac}"'] \arrow[ul, "\mathsf{comp}"] \arrow[u, "\mathsf{rfac}"'] & \mathsf{ar}^g \arrow[u, "\mathsf{source}"'] \arrow[ur, "\mathsf{target}"'] &
\end{tikzcd} \tag{71}$$

Two projections have names: $\mathsf{etop} : \mathsf{econe} \to \mathsf{ar}^f$ and $\mathsf{ebot} : \mathsf{econe} \to \mathsf{ar}^g$.

efid is the limit of the pushout of Diagram (71) and the following diagram. Note that the common part of the two diagrams is

$$\mathsf{ar}^e \xrightarrow{t} \mathsf{ob}^A$$

We could have presented Diagram (59) as a pushout in much the same way (the common part would describe the arrow $h : V \to L$). We have deliberately varied the way we present the data in this article because we are not sure ourselves which approach communicates best.

$$\begin{tikzcd}
\mathsf{ar}^v \arrow[r, "\mathsf{source}"] \arrow[dd, "\mathsf{target}"'] & \mathsf{ob}^X & \mathsf{ar}^u \arrow[l, "\mathsf{source}"'] \arrow[dd, "\mathsf{target}"] \\
 & \mathsf{ar}_2 \arrow[ul, "\mathsf{rfac}"] \arrow[ur, "\mathsf{comp}"'] \arrow[d, "\mathsf{lfac}"] & \\
\mathsf{ob}^E & \mathsf{ar}^e \arrow[l, "\mathsf{source}"'] \arrow[r, "\mathsf{target}"] & \mathsf{ob}^A
\end{tikzcd} \tag{72}$$

The named projections are $\mathsf{esoco}\colon\mathsf{efid} \to \mathsf{ar}^u$, $\mathsf{etaco}\colon\mathsf{efid} \to \mathsf{ar}^e$ and $\mathsf{efia}\colon\mathsf{efid} \to \mathsf{ar}^v$.

B.5.2 Diagrams for FinLim

The following diagram makes the fill-in arrow unique.

$$\begin{tikzcd} \mathsf{econe} \arrow[r, shift left, "\mathsf{eufid}"] & \mathsf{efid} \arrow[l, shift left, "\mathsf{esoco}"] \end{tikzcd} \tag{73}$$

These two diagrams ensure that the equalizer cone be a cone to the correct diagram.

$$\begin{tikzcd} \mathsf{ppair} \arrow[r, "\mathsf{equ}"] \arrow[dr, "\mathsf{top}"'] & \mathsf{econe} \arrow[d, "\mathsf{etop}"] \\ & \mathsf{ar} \end{tikzcd} \qquad \begin{tikzcd} \mathsf{ppair} \arrow[r, "\mathsf{equ}"] \arrow[dr, "\mathsf{bot}"'] & \mathsf{econe} \arrow[d, "\mathsf{ebot}"] \\ & \mathsf{ar} \end{tikzcd} \tag{74}$$

B.6 The sketch for the constructor space CCC

B.6.1 Definition A **Cartesian closed category** is a category $\mathscr{C}$ with the following structure:

CCC.1 $\mathscr{C}$ has binary products.

CCC.2 For each pair of objects A and B of $\mathscr{C}$, there is an object B^A and an arrow $\mathrm{eval}\colon B^A \times A \to B$.

CCC.3 For each triple of objects A, B and C of $\mathscr{C}$, there is a map

$$\lambda : \mathrm{Hom}(B \times A, C) \to \mathrm{Hom}(B, C^A) \tag{75}$$

such that for every arrow $f : B \times A \to C$,

$$\begin{tikzcd} B \times A \arrow[r, "\lambda f \times \mathsf{Id}[A]"] \arrow[dr, "f"'] & C^A \times A \arrow[d, "\mathrm{eval}"] \\ & C \end{tikzcd} \tag{76}$$

commutes.

CCC.4 For any arrow $g : B \to C^A$, $\lambda(\mathrm{eval} \circ (g \times \mathsf{Id}[A])) = g$.

Using this definition, the sketch for the constructor space for Cartesian closed categories may be built on the sketch for **FinProd** by adding the following nodes and arrows.

The nodes are:

1. twovf, the formal set of "functions of two variables", that is, arrows of the form $B \times A \to C$.
2. curry, the formal set of "curried functions" $B \to C^A$.

The sketch for **CCC** has arrows

1. $\mathsf{fs} : \mathsf{ob}^B \times \mathsf{ob}^A \to \mathsf{ob}^{B^A}$ that picks out the function space B^A of two objects B and A.
2. $\mathsf{ev} : \mathsf{ob}^B \times \mathsf{ob}^A \to \mathsf{ar}$ that picks out the arrow $\mathrm{eval} : B^A \times A \to B$.
3. $\mathsf{lam} : \mathsf{ar} \to \mathsf{ar}$, the formal version of the mapping λ of Diagram (75).
4. $\mathsf{tsource} : \mathsf{twovf} \to \mathsf{ob}^{B\times A}$, that picks out the source of a function $f : A \times B \to C$.
5. $\mathsf{ttarget} : \mathsf{twovf} \to \mathsf{ob}^C$, that picks out the target of a function $f : A \times B \to C$.
6. $\mathsf{arrow} : \mathsf{twovf} \to \mathsf{ar}^f$, that picks out the arrow f itself.
7. $\mathsf{csource} : \mathsf{curry} \to \mathsf{ob}^B$, that picks out the source of a curried function $g : B \to C^A$.
8. $\mathsf{ctarget} : \mathsf{curry} \to \mathsf{ob}^{C^A}$, that picks out the target of a curried function $f : B \to C^A$.
9. $\mathsf{arrow} : \mathsf{curry} \to \mathsf{ar}^g$, that picks out the arrow g itself.

B.6.2 Cones for CCC

CCC must have two cones

$$\begin{tikzcd} & \mathsf{twovf} \arrow[dl, "\mathsf{tsource}"'] \arrow[d, "\mathsf{arrow}"] \arrow[dr, "\mathsf{ttarget}"] & \\ \mathsf{ob}^{B\times A} & \mathsf{ar}^f \arrow[l, "\mathsf{source}"] \arrow[r, "\mathsf{source}"'] & \mathsf{ob}^C \end{tikzcd} \tag{77}$$

$$
\begin{tikzcd}
 & \mathsf{curry} \arrow[dl] \arrow[d, "\mathsf{ctarget}"'] \arrow[dr, "\mathsf{arrow}"] \arrow[drr, "\mathsf{csource}"] & & \\
\mathsf{ob}^C \times \mathsf{ob}^A \arrow[r, "\mathsf{fs}"'] & \mathsf{ob}^{c^A} & \mathsf{ar}^g \arrow[l, "\mathsf{target}"] \arrow[r, "\mathsf{source}"'] & \mathsf{ob}^B
\end{tikzcd}
\tag{78}
$$

B.7 The module for function spaces

Henceforth, we will assume the module

$$
\begin{tikzcd}
\mathsf{ob}^M & \mathsf{ob}^M \times \mathsf{ob}^N \arrow[l, "p_1"'] \arrow[r, "\mathsf{fs}"] \arrow[d, "p_2"] & \mathsf{ob}^{M^N} \\
 & \mathsf{ob}^N &
\end{tikzcd}
\tag{79}
$$

is attached whenever an occurrence of ob is annotated by M^N. Note that this occurred in Diagram (78).

B.7.1 Diagrams for CCC

Diagram (80) below forces eval to have the correct domain and codomain.

$$
\mathsf{ob}^{B^A \times A} \xleftarrow{\mathsf{source}} \mathsf{ar}^{\mathrm{eval}} \xrightarrow{\mathsf{target}} \mathsf{ob}^B
\tag{80}
$$

Expanding this diagram using the required modules is a two stage process, giving

$$
\begin{tikzcd}
\mathsf{ob}^A & \mathsf{ob}^{B^A} \times \mathsf{ob}^A \arrow[l, "p_2"'] \arrow[r, "p_1"] \arrow[d, "\mathsf{prod}"] & \mathsf{ob}^{B^A} & & \\
\mathsf{ar} \arrow[u, "\mathsf{rproj}"] \arrow[dr, "\mathsf{source}"'] & \mathsf{cone} \arrow[l, "\mathsf{lproj}"'] \arrow[r, "\mathsf{rproj}"] & \mathsf{ar} \arrow[u, "\mathsf{lproj}"] \arrow[dl, "\mathsf{source}"] & \mathsf{ob}^B \times \mathsf{ob}^A \arrow[ul, "\mathsf{fs}"'] \arrow[r, "p_1"] \arrow[d, "p_2"] & \mathsf{ob}^A \\
 & \mathsf{ob}^{B^A \times A} & \mathsf{ar}^{\mathrm{eval}} \arrow[l, "\mathsf{source}"] \arrow[r, "\mathsf{target}"'] & \mathsf{ob}^B &
\end{tikzcd}
\tag{81}
$$

Diagram (82) below forces λf to have the correct domain and codomain.

$$
\begin{tikzcd}
& \mathsf{ob}^{C} & \mathsf{ob}^{C^A} \\
\mathsf{ob}^{B\times A} & \mathsf{twovf}^{f} \arrow[l, "\mathsf{tsource}"'] \arrow[u, "\mathsf{ttarget}"] \arrow[r, "\mathsf{lam}"] & \mathsf{curry}^{\lambda f} \arrow[u, "\mathsf{csource}"] \arrow[d, "\mathsf{ctarget}"] \\
& & \mathsf{ob}^{B}
\end{tikzcd}
\tag{82}
$$

Diagram (83) below forces Diagram (76) to commute.

$$
\begin{tikzcd}
\mathsf{ob}^{B\times A} & \mathsf{ar}^{\lambda f\times \mathsf{Id}[A]} \arrow[l, "\mathsf{source}"'] \arrow[r, "\mathsf{target}"] & \mathsf{ob}^{C^A\times A} \\
& \mathsf{ar}_2 \arrow[u, "\mathsf{rfac}"] \arrow[dl, "\mathsf{comp}"'] \arrow[dr, "\mathsf{lfac}"] & \\
\mathsf{ar}^{f} \arrow[uu, "\mathsf{source}"] \arrow[r, "\mathsf{target}"'] & \mathsf{ob}^{C} & \mathsf{ar}^{\mathrm{eval}} \arrow[l, "\mathsf{target}"] \arrow[uu, "\mathsf{source}"']
\end{tikzcd}
\tag{83}
$$

Diagram (84) below ensures that requirement CCC–4 holds.

$$
\begin{tikzcd}
\mathsf{ob}^{C^A\times A} & \mathsf{ar}^{\mathrm{eval}} \arrow[l, "\mathsf{source}"'] \arrow[r, "\mathsf{target}"] & \mathsf{ob}^{C} & & \mathsf{ob}^{B} & \\
\mathsf{ar}^{g\times \mathsf{Id}[A]} \arrow[u, "\mathsf{target}"] \arrow[d, "\mathsf{source}"] & \mathsf{ar}_2^{\langle \mathrm{eval}, g\times \mathsf{Id}[A]\rangle} \arrow[l, "\mathsf{rfac}"'] \arrow[u, "\mathsf{lfac}"] \arrow[dr, "\mathsf{comp}"] & & & \mathsf{ar}^{g} \arrow[u, "\mathsf{source}"] \arrow[r, "\mathsf{target}"] & \mathsf{ob}^{C^A} \\
\mathsf{ob}^{B\times A} & & \mathsf{ar}^{\mathrm{eval}\circ(g\times \mathsf{Id}[A])} \arrow[ll, "\mathsf{source}"'] \arrow[uu, "\mathsf{target}"] & & \mathsf{curry} \arrow[u, "\mathsf{arrow}"] & \\
& & & & \mathsf{twovf} \arrow[u, "\mathsf{lam}"] \arrow[ull, "\mathsf{arrow}"] \arrow[uuull, "\mathsf{ttarget}"] \arrow[ullll, "\mathsf{tsource}"] &
\end{tikzcd}
\tag{84}
$$

B.7.2 Invertibility of λ

It follows from CCC-4 that if $\mathscr{C}$ is any Cartesian closed category corresponding to a model $\mathfrak{C}$, then $\mathfrak{C}(\lambda)$ is a bijection. The Completeness Theorems 6.2.1 and 6.2.3 then imply that there is an arrow λ^{-1} in CCC that, as its name suggests, is a formal inverse to λ.

References

At the end of each entry, the pages on which that entry is cited are listed in parentheses.

[Adámek and Rosičky, 1994] Jiří Adámek and Jiří Rosičky. **Locally Presentable and Accessible Categories**. Cambridge University Press, 1994. (26)

[Bagchi and Wells, 1993a] Atish Bagchi and Charles Wells. *Graph-based logic and sketches II: Finite product categories and equational logic.* Available by gopher from `gopher.cwru.edu`, path=`/class/mans/math/pub/wells/`, 1993. (37)

[Bagchi and Wells, 1993b] Atish Bagchi and Charles Wells. *The varieties of mathematical prose.* Available by gopher from `gopher.cwru.edu`, path=`/class/mans/math/pub/wells/`, 1993. (1)

[Barr and Wells, 1985] Michael Barr and Charles Wells. **Toposes, Triples and Theories**, volume 278 of *Grundlehren der mathematischen Wissenschaften.* Springer-Verlag, New York, 1985. A list of corrections and additions is maintained in [Barr and Wells, 1993c]. (59)

[Barr and Wells, 1990] Michael Barr and Charles Wells. **Category Theory for Computing Science**. Prentice-Hall International Series in Computer Science. Prentice-Hall International, New York, 1990. The document [Barr and Wells, 1993b] contains corrections and additions to the book. (1, 2, 13, 43, 59)

[Barr and Wells, 1992] Michael Barr and Charles Wells. *On the limitations of sketches.* Canadian Mathematical Bulletin, **35**:287–294, 1992.

[Barr and Wells, 1993a] Michael Barr and Charles Wells. *The category of diagrams*, 1993. To appear. (11, 12)

[Barr and Wells, 1993b] Michael Barr and Charles Wells. *Category Theory for Computing Science: Update.* Available by anonymous FTP from `triples.math.mcgill.ca` in directory `pub/barr`. The file is called `ctcsupdate.dvi`, 1993. Corrections and additions to [Barr and Wells, 1990]. (58)

[Barr and Wells, 1993c] Michael Barr and Charles Wells. *Corrections to Toposes, Triples and Theories.* Available by anonymous FTP from `triples.math.mcgill.ca` in directory `pub/barr`, 1993. Corrections and additions to [Barr and Wells, 1985]. (58)

[Bastiani and Ehresmann, 1972] Andrée Bastiani and Charles Ehresmann. *Categories of sketched structures.* Cahiers de Topologie et Géométrie Différentielle, **13**:104–213, 1972. (13)

[Burroni, 1970a] Albert Burroni. *Esquisses des catégories à limites et des quasi-topologies.* Esquisses Mathématiques, **5**, 1970. (43)

[Burroni, 1970b] E. Burroni. *Catégories discretement structurées.* Esquisses Mathématiques, **5**, 1970. (43)

[Coppey and Lair, 1988] L. Coppey and C. Lair. *Leçons de théorie des esquisses, partie II.* Diagrammes, **19**, 1988. (43)

[Ebbinghaus *et al.*, 1984] H.-D. Ebbinghaus, J. Flum, and W. Thomas. **Mathematical Logic**. Springer-Verlag, 1984. (1)

[Ehresmann, 1966] Charles Ehresmann. *Introduction to the theory of structured categories.* Technical Report 10, University of Kansas, 1966. (43)

[Ehresmann, 1968a] Charles Ehresmann. *Catégories structurées généralisés.* Cahiers de Topologie et Géométrie Différentielle, **X**:139–168, 1968. (43)

[Ehresmann, 1968b] Charles Ehresmann. *Esquisses et types des structures algébriques.* Bul. Inst. Polit. Iaşi, **XIV**, 1968. (43)

[Fourman and Vickers, 1986] Michael Fourman and S. Vickers. *Theories as categories.* In **Category Theory and Computer Programming**, D. Pitt et al., editors, volume 240 of *Lecture Notes in Computer Science*, pages 434–448. Springer-Verlag, 1986.

[Goguen and Burstall, 1986] Joseph A. Goguen and R. M. Burstall. *A study in the foundations of programming methodology: Specifications, institutions, charters and parchments*. In **Category Theory and Computer Programming**, D. Pitt et al., editors, volume 240 of *Lecture Notes in Computer Science*, pages 313–333. Springer-Verlag, 1986. (2)

[Goguen and Meseguer, 1982] Joseph A. Goguen and José Meseguer. *Completeness of many-sorted equational logic*. Technical Report CSL-135, SRI International Computer Science Laboratory, 333 Ravenswood Ave., Menlo Park, CA 94025, USA, 1982.

[Guitart and Lair, 1980] René Guitart and Christian Lair. *Calcul syntaxique des modèles et calcul des formules internes*. Diagrammes, **4**, 1980.

[Guitart and Lair, 1982] René Guitart and Christian Lair. *Limites et colimites pour representer les formules*. Diagrammes, **7**, 1982.

[Lambek and Scott, 1984] Joachim Lambek and P. Scott. *Aspects of higher order categorical logic*. In **Mathematical Applications of Category Theory**, J.W. Gray, editor, volume 30 of *Contemporary Mathematics*, pages 145–174. American Mathematical Society, 1984.

[Lambek and Scott, 1986] Joachim Lambek and P. Scott. **Introduction to Higher Order Categorical Logic**, volume 7 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1986.

[Makkai and Paré, 1990] Michael Makkai and R. Paré. **Accessible Categories: the Foundations of Categorical Model Theory**, volume 104 of *Contemporary Mathematics*. American Mathematical Society, 1990.

[Makkai and Reyes, 1977] Michael Makkai and Gonzalo Reyes. **First Order Categorical Logic**, volume 611 of *Lecture Notes in Mathematics*. Springer-Verlag, 1977.

[Makkai, 1993a] Michael Makkai. *Generalized sketches as a framework for completeness theorems*, 1993. Preprint, McGill University. (37)

[Makkai, 1993b] Michael Makkai. *The syntax of categorical logic*, 1993. Preprint, McGill University. (37)

[McDonald and Stone, 1984] John L. McDonald and Arthur Stone. *Topoi over graphs.* Cahiers de Topologie et Géométrie Différentielle, **XXV**, 1984. (43)

[Meseguer, 1989] José Meseguer. *General logics.* Technical Report CSL-89-5, SRI International Computer Science Laboratory, 333 Ravenswood Ave., Menlo Park, CA 94025, USA, 1989.

[Power and Wells, 1992] A.J. Power and Charles Wells. *A formalism for the specification of essentially algebraic structures in 2-categories.* Mathematical Structures in Computer Science, **2**:1–28, 1992. (14, 18)

[Sanella and Tarlecki, 1988] D. Sanella and A. Tarlecki. *Building specifications in an arbitrary institution.* Information and Control, **76**:165–210, 1988.

[Tholen and Tozzi, 1989] Walter Tholen and Anna Tozzi. *Completions of categories and initial completions.* Cahiers de Topologie et Géométrie Différentielle Catégorique, pages 127–156, 1989. (8)

[Wells and Barr, 1988] Charles Wells and M. Barr. *The formal description of data types using sketches.* In **Mathematical Foundations of Programming Language Semantics**, Michael Main et al., editors, volume 298 of *Lecture Notes in Computer Science.* Springer-Verlag, 1988. (38)

[Wells, 1990] Charles Wells. *A generalization of the concept of sketch.* Theoretical Computer Science, **70**:159–178, 1990. (13, 14, 18)

Atish Bagchi
226 West Rittenhouse Square
#702
Philadelphia, PA 19103
atish@math.upenn.edu

Charles Wells
Department of Mathematics
Case Western Reserve University
University Circle
Cleveland, OH 44106-7058, USA
cfw2@po.cwru.edu