Category Theory for Linear Logicians

Richard Blute * Philip Scott †

September 10, 2003

Abstract

This paper presents an introduction to category theory with an emphasis on those aspects relevant to the analysis of the model theory of linear logic. With this in mind, we focus on the basic definitions of category theory and categorical logic.

An analysis of cartesian and cartesian closed categories and their relation to intuitionistic logic is followed by a consideration of symmetric monoidal closed, linearly distributive and $*$-autonomous categories and their relation to multiplicative linear logic. We examine nonsymmetric monoidal categories, and consider them as models of noncommutative linear logic. We introduce traced monoidal categories, and discuss their relation to the geometry of interaction. The necessary aspects of the theory of monads is introduced in order to describe the categorical modelling of the exponentials. We conclude by briefly describing the notion of *full completeness*, a strong form of categorical completeness, which originated in the categorical model theory of linear logic.

No knowledge of category theory is assumed, but we do assume knowledge of linear logic sequent calculus and the standard models of linear logic, and modest familiarity with typed lambda calculus.

0 Introduction

Category theory arose as an organizing framework for expressing the *naturality* of certain constructions in algebraic topology. Its subsequent applicability, both as a language for simply expressing complex relationships between mathematical structures and as a mathematical theory in its own right, is remarkable. Categorical principles have been put to good use in virtually every branch of mathematics, in most cases leading to profound new understandings.

Roughly a *category* is an abstraction of the principle that the morphisms between objects of interest are just as important as the objects themselves. So a category will consist of two classes, the class of objects and the class of morphisms between objects. One must have a

*Department of Mathematics, University of Ottawa, Ottawa, Ontario, K1N 6N5, CANADA `rblute@mathstat.uottawa.ca`

†Department of Mathematics, University of Ottawa, Ottawa, Ontario, K1N 6N5, CANADA `phil@site.uottawa.ca`

composition law, and each object must come equipped with a specified identity morphism. This data must satisfy some evident axioms. From this simple definition, an enormous theory follows. For example, one next defines morphisms between categories; these are *functors*. One can go on to define morphisms between functors; these are *natural transformations*, and on and on. There is a remarkably rich interaction between these structures. As expositions of this theory, we highly recommend [Mac, Borc].

Categorical logic begins with the idea that, given a logic, one can form a category whose objects are formulas and whose morphisms are (equivalence classes of) proofs. The question of the proper notion of equivalence is extremely important and delicate. We will examine it in some detail below. There are several benefits to the formation of this category. First, under this interpretation, the logic's connectives are naturally exhibited as functors, and the logic's inference rules are exhibited as natural transformations. Then models of the logic can be simply defined as structure-preserving functors from this syntactic category to a category with the appropriate structure. Second, the category so formed will typically be freely generated in a certain sense, and can thus be used to derive general information about all categories of the same structure. The most well-developped examples of this idea are the relations between intuitionistic logic and cartesian closed categories, and between linear logic and $*$-autonomous categories. Both of these relationships will be described below.

The goal of this paper is to establish sufficient categorical background to understand these relationships and their consequences. We will introduce cartesian closed categories (cccs) and describe the translation between cccs and intuitionistic logic. This is the most well-established example of categorical logic, and is the subject of the book [LS]. This is followed by a consideration of monoidal, symmetric monoidal closed, linearly distributive and $*$-autonomous categories and the translation between these structures and multiplicative linear logic. One of the most intriguing aspects of linear logic is that it is sufficiently flexible as a logical system to allow one to define noncommutative versions. With this in mind, we examine nonsymmetric monoidal categories, and consider them as models of noncommutative linear logic. We will especially focus on examples arising from the representation theory of *Hopf algebras*.

We also introduce traced monoidal categories, which arose independently of linear logic, but were subsequently seen to provide the appropriate framework for the analysis of Girard's geometry of interaction. Computationally, the most important fragment of linear logic is the exponential fragment, and its categorical structure leads one to the notion of *Seely model*. The necessary aspects of the theory of monads is introduced in order to describe the categorical modelling of the exponentials. We conclude by briefly describing the notion of *full completeness*, a strong form of categorical completeness, which originated in the categorical model theory of linear logic [AJ]. Full completeness is an excellent example of the influence of categorical principles on logical semantics, not just for linear logic, but for general logics.

No knowledge of category theory is assumed, but we do assume knowledge of linear logic sequent calculus and the standard models of linear logic. Also it would help to have a modest familiarity with typed lambda calculus (as in Girard's [GLT]). This paper may be considered a companion to the article [Sco], but stressing the linear logic aspects. We note that we only focus on aspects of category theory of immediate relevance to linear logic. So important topics like limits and colimits are omitted.

1 Categories, Functors, Natural Transformations

1.1 Basics of Categories

A *category* $\mathcal{C}$ consists of two classes, *Objects* and *Arrows*, together with two functions *Arrows* $\underset{cod}{\overset{dom}{\rightrightarrows}}$ *Objects* satisfying the following properties (we write $A \xrightarrow{f} B$ for: $f \in$ *Arrows*, $dom(f) = A$ and $cod(f) = B$):

- There are *identity* arrows $A \xrightarrow{id_A} A$, for each object A,
- There is a partially-defined binary *composition* operation on arrows, denoted by juxtaposition,

$$\frac{A \xrightarrow{f} B \quad B \xrightarrow{g} C}{A \xrightarrow{gf} C}$$

(defined only when $dom(g) = cod(f)$) satisfying the following equations:

(i) $f id_A = f = id_B f$, where $A \xrightarrow{f} B$,

(ii) $h(gf) = (hg)f$, where $A \xrightarrow{f} B \xrightarrow{g} C \xrightarrow{h} D$.

A category is called *large* or *small* depending upon whether its class of objects is respectively a proper class or a set, in the sense of Gödel-Bernays set theory. We denote by $\mathcal{C}(A, B)$ the collection of arrows $A \to B$ in the category $\mathcal{C}$. A category is *locally small* if $\mathcal{C}(A, B)$ is a set, for all objects A, B. It is convenient to represent arrows graphically. Equations in categories are also typically represented graphically and are called *commutative diagrams.*

Many familiar classes of structures in mathematics and logic can be organized into categories. Here are some basic examples. Verification that the set of arrows is closed under composition as well as satisfying the axioms of a category is left as an exercise.

Set: This (large) category has the class of all sets as Objects, with all set-theoretic functions as Arrows. Identity arrows and composition of arrows are defined in the usual way.

Rel: This has the same objects as **Set**, but an arrow $A \xrightarrow{R} B$ is a binary relation $R \subseteq A \times B$. Here composition is given by *relational product*, i.e. given $A \xrightarrow{R} B \xrightarrow{S} C$,

$$A \xrightarrow{SR} C = \{(a, c) \in A \times C \mid \exists b \in B \text{ such that } (a, b) \in R \ \&\ (b, c) \in S\}$$

while the identity arrows $A \xrightarrow{id_A} A$ are given by the diagonal relations: $id_A = \{(a, a) \mid a \in A\}$.

Universal Algebras: Here Objects can be any equational class of algebras (e.g. semigroups, monoids, groups, rings, lattices, heyting or boolean algebras, $\cdots$). Arrows are *homomorphisms*, i.e. set-theoretic functions preserving the given structure. Composition and identities are induced from **Set**. We use boldface notation for the names of the associated categories, e.g. **Group**, **Lat**, **Heyt**, for the categories of groups, lattices, and Heyting algebras, resp.

$\mathbf{Vec_k}$: Here Objects are vector spaces over a field $\mathbf{k}$, and Arrows are $\mathbf{k}$-linear maps. We usually omit the subscript $\mathbf{k}$, and write **Vec** for short. An important subcategory of **Vec** is the category $\mathbf{Vec}_{fd}$ of finite dimensional vector spaces and linear maps. Of course, one can also consider various classes of topological vector spaces and normed spaces, with appropriate notions of map.

Top: Here objects are topological spaces and morphisms are continuous maps. One can also consider various homotopy categories, i.e. where the morphisms are homotopy equivalence classes of continuous maps. It is from this sort of example that category theory originally arose.

Poset: Here the objects are partially-ordered sets and morphisms are monotone maps. A particularly important example arising in theoretical computer science is the category ω-**CPO** of posets in which ascending countable chains $\cdots a_i \leq a_{i+1} \leq a_{i+2} \leq \cdots$ have suprema, and in which morphisms are poset maps preserving suprema of countable chains. Composition and identities are inherited from **Set**. For an introduction to this and other aspects of domain theory, see [AC].

Of course, those areas of mathematics that heavily use category theory, e.g. algebraic topology, algebraic geometry, and homological algebra, are replete with many more sophisticated examples.

The previous examples were large categories, i.e. in which the collections of objects form proper classes in the sense of set theory. We now present some "small" categories, based on much smaller collections of objects and arrows:

One: The category with one object and one (identity) arrow.

Discrete categories: These are categories where the only arrows are identities. A set X becomes a discrete category, by letting the objects be the elements of X, and adding an identity arrow $x \to x$ for each $x \in X$. All (small) discrete categories arise in this way.

A monoid: A monoid M gives a category with one object, call it $\mathcal{C}_M$, as follows: if the single object is $*$, we define $\mathcal{C}_M(*,*) = M$. Composition of maps is multiplication in the monoid. Conversely, note that every category $\mathcal{C}$ with one object corresponds to a monoid, namely $\mathcal{C}(*,*)$.

A preorder: A preordered set $\mathbb{P} = (P, \leq)$ (where $\leq$ is a reflexive & transitive relation) may be considered as a category, whose objects are just the elements of $\mathbb{P}$ and in which we define $\mathbb{P}(a,b) = \{*\}$ if $a \leq b$ and $\mathbb{P}(a,b) = \emptyset$ if $a \not\leq b$. Thus, given two objects $a, b \in \mathbb{P}$, there is at most one arrow from a to b; moreover, there is an arrow $a \to b$ in $\mathbb{P}$ exactly when $a \leq b$. In this case, the category laws are exactly the preorder conditions.

Graphs and finite categories: A *graph* (more precisely, a directed multigraph), consists of a pair of sets, called *Objects* and *Arrows*, together with two functions $Arrows \underset{cod}{\overset{dom}{\rightrightarrows}} Objects$. Every (small) category has an underlying graph, obtained by simply ignoring the other data beyond dom, cod. In particular, any finite category can be represented by simply drawing its underlying graph and assuming the existence of all well-defined compositions of arrows. Notice that all vertices in the underlying graph of a category have loops (given by identity arrows). Indeed, another way of looking at a category is as a kind of graph with additional structure (i.e. identity edges, a composition law and equations).

Graphs form a category **Graph** whose objects are graphs and whose arrows are pairs of functions $Arrows_0 \xrightarrow{f} Arrows_1$ and $Objects_0 \xrightarrow{g} Objects_1$ such that $gdom_0 = dom_1 f$ and $gcod_0 = cod_1 f$.

1.2 Deductive systems as categories

In the 1960's, Lambek introduced the novel idea of using Gentzen's methods in category theory and linguistics. His new approach involved the use of proof-theoretical methods in constructing free categories and for solving coherence (i.e. decision) problems. At the same time he emphasized a fundamental new idea: arrows in (freely generated) categories are equivalence classes of proofs. Lambek's work raises a question of particular relevance to linear logicians: what should the equations between proofs be? There is no ultimate answer except that Lambek's work would seem to say that the equations should be elegant and natural from the viewpoint of category theory. This section will follow [LS] closely. For more on the history of this area, see [LS] and the references therein.

Definition 1.1 A *deductive system* is a labelled directed graph (whose objects are called *formulas* and whose arrows are called *labelled sequents*). There are certain specified arrows (called *axioms*) among which are arrows $A \xrightarrow{id_A} A$, for all formulas A, and certain specified rules (called "inference rules") for generating new arrows from old ones, among which is the composition rule called "cut": $\dfrac{A \xrightarrow{f} B \quad B \xrightarrow{g} C}{A \xrightarrow{gf} C}\,cut$, for all formulas A, B, C.

A deductive system freely generates "labelled proof trees" by the following procedure:

- Axioms are proof trees.
- The set of proof trees must be closed under the inference rules.

The root of a proof tree is called a "provable sequent", or "proof" for short, while the leaves of the tree are axioms.

Example 1.2 Let $\mathcal{G}$ be a graph. The *deductive system freely generated from* $\mathcal{G}$ is defined as follows:

1. The formulas are the objects of $\mathcal{G}$ (also called *atomic formulas*).
2. The axioms consist of a distinguished *identity* axiom $A \xrightarrow{id_A} A$, for each formula A, together with all the arrows of $\mathcal{G}$ (the latter are sometimes called *nonlogical axioms*).

3. Cut is the only rule of inference.

A deductive system freely generated from $\mathcal{G}$ forms a category $F(\mathcal{G})$, *the category freely generated from* $\mathcal{G}$, whose objects are all the formulas and whose arrows are equivalence classes of proofs. Namely, we impose equations between proof trees by taking the congruence relation generated by the following equations:

$$\frac{A \xrightarrow{f} B \quad B \xrightarrow{id_B} B}{A \xrightarrow{id_B f} B} = A \xrightarrow{f} B$$

$$\frac{A \xrightarrow{id_A} A \quad A \xrightarrow{f} B}{A \xrightarrow{f id_A} B} = A \xrightarrow{f} B$$

$$\frac{A \xrightarrow{f} B \qquad \dfrac{B \xrightarrow{g} C \quad C \xrightarrow{h} D}{B \xrightarrow{hg} D}}{A \xrightarrow{(hg)f} D} = \frac{\dfrac{A \xrightarrow{f} B \quad B \xrightarrow{g} C}{A \xrightarrow{gf} C} \qquad C \xrightarrow{h} D}{A \xrightarrow{h(gf)} D}$$

An important special case is the following:

Example 1.3 (Deductive systems generated by discrete graphs) A graph $\mathcal{G}_0$ is *discrete* if it has no arrows: it may be identified with a set (of objects). The deductive system generated from the set $\mathcal{G}_0$ has only atomic formulas (objects of $\mathcal{G}_0$) and for axioms it has only identity axioms $A \xrightarrow{id_A} A$, for atomic formulas A. $F(\mathcal{G}_0)$ is called the *free category generated from the set of objects* $\mathcal{G}_0$.

We will later consider freely generated categories with additional structure (i.e. with additional operations on formulas, and additional axioms and rules of inference). It is possible conversely to *define* a category as a certain kind of deductive system, although in that case it will not necessarily be freely generated: the class of objects may simply be specified and the class of arrows merely closed under appropriate operations and equations (see [LS]). Moreover, if the category is large, like **Set**, the objects and arrows form proper classes, which is not exactly what logicians are familiar with.

1.3 Operations on categories

There are many ways of forming new categories out of old ones. Two basic operations are the following:

Dualization: If $\mathcal{C}$ is a category, so is its *dual* $\mathcal{C}^{op}$, with the same objects, but whose arrows are reversed (i.e. interchange *dom* and *cod*). Clearly $(\mathcal{C}^{op})^{op} = \mathcal{C}$ and reversing all arrows changes commutative diagrams in $\mathcal{C}$ to commutative diagrams in $\mathcal{C}^{op}$. In other words, we have the following bijective correspondence:

$$\frac{f\colon A \to B \text{ in } \mathcal{C}}{f\colon B \to A \text{ in } \mathcal{C}^{op}}$$

Products: If $\mathcal{C}$, $\mathcal{D}$ are categories, so is their cartesian product $\mathcal{C} \times \mathcal{D}$, with the obvious structure: objects are pairs of objects, arrows are pairs of arrows, composition and identities are defined componentwise.

Finally, we end with a useful notion:

Definition 1.4 A *subcategory* $\mathcal{C}$ of a category $\mathcal{B}$ is any category whose class of objects and arrows are contained in those of $\mathcal{B}$, respectively, and which is closed under the "operations" in $\mathcal{B}$ of *domain, codomain, composition, and identity.* $\mathcal{C}$ is a *full* subcategory of $\mathcal{B}$ if for all objects $A, B \in \mathcal{C}$, $\mathcal{C}(A, B) = \mathcal{B}(A, B)$. In other words, a full subcategory is determined by just restricting the class of objects, since the arrows are predetermined by $\mathcal{B}$.

For example, we often consider small subcategories whose objects are of "bounded size" within the large examples above: e.g. the full subcategories of (i) finite sets and (ii) finite dimensional vector spaces, and more generally, for a fixed infinite cardinal κ, sets (resp. vector spaces) of cardinality (resp. dimension) bounded by κ.

1.4 Functors

If $\mathcal{C}, \mathcal{D}$ are categories, a *functor* $F : \mathcal{C} \to \mathcal{D}$ is a pair $F = (F_{ob}, F_{arr})$, where $F_{ob} : Objects(\mathcal{C}) \to Objects(\mathcal{D})$, and similarly for arrows, satisfying the following (we omit the subscripts ob, arr): if $A \xrightarrow{f} B$ then $FA \xrightarrow{F(f)} FB$ with: $F(gf) = F(g)F(f)$ and $F(id_A) = id_{FA}$. A functor $F : \mathcal{C}^{op} \to \mathcal{D}$ is sometimes called *contravariant.* From the definition of the opposite category, a contravariant functor F preserves the identity arrows, but reverses composition: $F(gf) = F(f)F(g)$.

Examples 1.5

1. *Forgetful (also called Underlying) Functors.* These include forgetful functors U : **Posets** $\to$ **Set**, U : **Top** $\to$ **Set**, U : **Alg** $\to$ **Set** (where **Alg** is any category of universal algebras and homomorphisms between them). U maps objects and arrows to their underlying set (omitting the other structure).

   Sometimes, one only forgets part of the structure, e.g. there are several forgetful functors on **TopGrp**; we have U_1 : **TopGrp** $\to$ **Grp** which maps a topological group and a continuous group homomorphism to its underlying group (and the underlying group homomorphism), and similarly there is U_2 : **TopGrp** $\to$ **Top**.

2. *Representable (or Hom) Functors.* If $A \in \mathcal{C}$, we have the dual co- and contravariant homs:

   1. *Covariant hom* : $\mathcal{C}(A, -) : \mathcal{C} \to \mathbf{Set}$ given by:

$$\begin{array}{rcl} B & \mapsto & \mathcal{C}(A, B) \\ B \xrightarrow{f} C & \mapsto & \mathcal{C}(A, f) : \mathcal{C}(A, B) \to \mathcal{C}(A, C) \\ & & \text{where } \mathcal{C}(A, f)(g) = fg. \end{array}$$

2. *Contravariant hom*: $\mathcal{C}(-,A) : \mathcal{C}^{op} \to \mathbf{Set}$ given by:

$$\begin{array}{rcl} B & \mapsto & \mathcal{C}(B,A) \\ B \xrightarrow{f} C & \mapsto & \mathcal{C}(f,A) : \mathcal{C}(C,A) \to \mathcal{C}(B,A) \\ & & \text{where } \mathcal{C}(A,f)(g) = gf. \end{array}$$

3. *Powerset Functors.* There are co- and contravariant powerset functors on **Set**:

   1. *Covariant* $\mathcal{P} : \mathbf{Set} \to \mathbf{Set}$ given by:

$$\begin{array}{rcl} A & \mapsto & \mathcal{P}(A) = \{X \mid X \subseteq A\} \\ A \xrightarrow{f} B & \mapsto & \mathcal{P}(f) : \mathcal{P}(A) \to \mathcal{P}(B) \;\; \text{where } \mathcal{P}(f)(X) = f[X] \end{array}$$

   2. *Contravariant* $\mathcal{P} : \mathbf{Set}^{op} \to \mathbf{Set}$ given by:

$$\begin{array}{rcl} A & \mapsto & \mathcal{P}(A) = \{X \mid X \subseteq A\} \\ A \xrightarrow{f} B & \mapsto & \mathcal{P}(f) : \mathcal{P}(B) \to \mathcal{P}(A) \;\; \text{where } \mathcal{P}(f)(Y) = f^{-1}[Y] \end{array}$$

4. *Free Algebra Functors.* $F : \mathbf{Set} \to \mathbf{Alg}$, where $F(X)$ = the free algebra generated by set X (e.g. **Alg** can be **Mon**, **Grp**, **Vec**, etc.)

5. *Identity and Inclusion Functors*: For example, $Id : \mathbf{Set} \to \mathbf{Set}$, and the evident inclusion $Inc : \mathbf{Vec}_{fd} \hookrightarrow \mathbf{Vec}$ of finite dimensional vector spaces among all vector spaces.

6. *Dual Spaces*: Let $V \in \mathbf{Vec}$ and $V^{\perp} = Lin(V, \mathbf{k})$, the dual space of V. Exercise: show there are two functors: $(-)^{\perp} : \mathbf{Vec}^{op} \to \mathbf{Vec}$ and $(-)^{\perp\perp} : \mathbf{Vec} \to \mathbf{Vec}$.

   Typically, this functor would be denoted V^*, but we will suggest that in some settings, this notion of linear dual actually models linear negation quite successfully, hence our choice of notation.

7. Let P and P' be posets, viewed as categories. We leave it to the reader to verify that a functor $F\colon P \to P'$ is the same thing as an order-preserving function from P to P'.

1.5 Natural Transformations

Given functors $F, G : \mathcal{C} \to \mathcal{D}$, a *natural transformation* is a family of arrows, indexed by the objects of $\mathcal{C}$, $\{\theta_C : FC \to GC \mid C \in \mathcal{C}\}$ such that for every $f : C \to D$, the following diagram commutes:

$$\begin{tikzcd} FC \arrow[r, "\theta_C"] \arrow[d, "Ff"'] & GC \arrow[d, "Gf"] \\ FD \arrow[r, "\theta_D"] & GD \end{tikzcd}$$

Given n-ary functors $F, G : \mathcal{C}^n \to \mathcal{D}$, a family of arrows, indexed by n-tuples of objects of $\mathcal{C}$, $\alpha_{A_1,\cdots,A_n} : F(A_1, \cdots, A_n) \to G(A_1, \cdots, A_n)$ is said to be *natural in* A_i if fixing all the other arguments $A_j, j \neq i$, the resultant family $\alpha_{\cdots,A_i,\cdots} : F(\cdots, A_i, \cdots) \to G(\cdots, A_i, \cdots)$ determines a natural transformation between functors $\mathcal{C} \to \mathcal{D}$ with respect to the ith argument as variable.

Examples 1.6

1. *Double Dual*: Define $\theta : Id \to (-)^{\perp\perp} : \mathbf{Vec} \to \mathbf{Vec}$, where $\theta_V : V \to V^{\perp\perp}$ is given by:

$$\theta_V(x)(f) = f(x) \quad \text{for} \quad f \in V^{\perp}, x \in V.$$

   Exercise: Verify that θ is well-defined, and that the appropriate natural transformation diagram commutes. It may be shown that θ_V is an isomorphism if and only if V is finite dimensional. However, note that if V is indeed finite dimensional, there is no *natural* isomorphism $\eta : Id \to (-)^{\perp}$, even though for each V, $V \cong V^{\perp}$ in this case. The reason is that any such isomorphism depends on a choice of basis.

2. *Functor Categories*: Let $\mathcal{C}, \mathcal{D}$ be categories, with $\mathcal{C}$ small. Let $Funct(\mathcal{C}, \mathcal{D})$ be the category whose objects are functors from $\mathcal{C}$ to $\mathcal{D}$, and whose arrows are natural transformations between them, where we compose natural transformations as follows: given $F, G, H \in Funct(\mathcal{C}, \mathcal{D})$, define

$$FA \xrightarrow{(\psi\theta)_A} HA \;=\; FA \xrightarrow{\theta_A} GA \xrightarrow{\psi_A} HA$$

   for each object $A \in \mathcal{C}$. In particular, if $\mathcal{C}$ is small, and $\mathcal{D} = \mathbf{Set}$, the category $Funct(\mathcal{C}^{op}, \mathcal{D}) = \mathbf{Set}^{\mathcal{C}^{op}}$ is called the category of *presheaves on* $\mathcal{C}$.

   If $\mathcal{C}$ is the small category with two objects and two non-identity arrows, $\bullet \rightrightarrows \bullet$, one can identify $\mathbf{Set}^{\mathcal{C}^{op}}$ with the category **Graph** of small graphs.

3. There is a category **Cat** of small categories and functors between them. There is a forgetful functor $U : \mathbf{Cat} \to \mathbf{Graph}$ which associates to every small category $\mathcal{C}$ its underlying graph.

1.6 Adjoints and Equivalences

An arrow in a category is an *isomorphism* or *iso* if it has a two-sided inverse. This corresponds to the usual mathematical notion of "isomorphism" in most familiar categories. In the case of functor categories, we obtain the following related notions:

- *Natural Isomorphisms*: A natural transformation $F \xrightarrow{\theta} G$ is a natural isomorphism if, for each A, $FA \xrightarrow{\theta_A} GA$ is an iso. We often write $F \cong G$ (leaving θ implicit) to denote such a natural isomorphism.

- *Isos of Categories*: A pair of functors $\mathcal{C} \underset{G}{\overset{F}{\rightleftarrows}} \mathcal{D}$ is an isomorphism of categories if $GF = Id_{\mathcal{C}}$ and $FG = Id_{\mathcal{D}}$. This is much too strong and rarely occurs in mathematics. A much more reasonable notion is the following:

- *Natural Equivalence*: A pair of functors $\mathcal{C} \underset{G}{\overset{F}{\rightleftarrows}} \mathcal{D}$ is a natural equivalence of categories if there are natural isomorphisms $GF \cong Id_{\mathcal{C}}$ and $FG \cong Id_{\mathcal{D}}$. We shall see many examples of this notion below.

Most mathematical duality theories, as in the case of the famous representation theorems of Stone, Gelfand, and Pontrjagin, amount to "contravariant" natural equivalences $\mathcal{C} \cong \mathcal{D}^{op}$. Barr's book [Barr80] on $*$-autonomous categories, which analyzes such duality theories, is an important source of concrete models for (fragments of) linear logic. We shall discuss this later.

One of the most important concepts in category theory is that of *adjoint functors*. Given functors $\mathcal{D} \underset{U}{\overset{F}{\leftrightarrows}} \mathcal{C}$, we say *$F$ is left adjoint to U* (denoted $F \dashv U$) if there is a natural isomorphism

$$\mathcal{D}(FC, D) \cong \mathcal{C}(C, UD) \ .$$

That is, there is a family of arrows $\{\alpha_{C,D} : \mathcal{D}(FC,D) \to \mathcal{C}(C,UD)\}$ which determines a natural isomorphism of functors (natural in C and D), $\alpha_{-,-} : \mathcal{D}(F-,-) \xrightarrow{\cong} \mathcal{C}(-,U-)$. This isomorphism determines a natural bijection of arrows

$$\begin{array}{ll} FC \to D & \text{in } \mathcal{D} \\ \hline\hline C \to UD & \text{in } \mathcal{C}. \end{array}$$

Indeed, the statement that F is left adjoint to U is equivalent to the following *universal mapping property* of the functor F: for each object $C \in \mathcal{C}$, there is an object $FC \in \mathcal{D}$ and an arrow $\eta_C : C \to UFC$, such that for any arrow $f : C \to UD \in \mathcal{C}$, there is a unique $f^* : FC \to D \in \mathcal{D}$ satisfying: $U(f^*)\eta_C = f$, i.e.

$$\begin{tikzcd} UFC \arrow[dr, "U(f^*)"] & \\ C \arrow[u, "\eta_C"] \arrow[r, "f"'] & UD \end{tikzcd}$$

Exercise 1.7 (Adjoints)

1. Prove the equivalence of F being left adjoint to U and the universal mapping property above.

2. Given adjoints $F \dashv U$ as above, show there are natural transformations $\eta : Id_{\mathcal{C}} \to UF$ and $\varepsilon : FU \to Id_{\mathcal{D}}$ satisfying $(U\varepsilon)(\eta U) = id_U$ and $(\varepsilon F)(F\eta) = id_F$, where id denotes an identity natural transformation.

3. Show that there is a one-one correspondence between solutions $(F, \eta, (-)^*)$ of the universal mapping problem in part (i) with quadruples $(F, U, \eta, \varepsilon)$ satisfying the equations in part (ii).

Notions defined by universal mapping properties are unique up to isomorphism. For example, adjoint functors determine each other uniquely up to natural isomorphisms.

Adjoint functors abound in mathematics. Lawvere has used this in an attempted axiomatic foundation for large parts of mathematics.

Examples 1.8 (Adjoints) 1. *Galois Correspondences:* Consider two pre-orders as categories, with a pair of adjoint functors (order-preserving maps) between them : $(P, \leq) \underset{G}{\overset{F}{\rightleftarrows}} (Q, \leq)$. Then $F \dashv G$ means: $F(a) \leq b$ iff $a \leq G(b)$, for all $a \in P, b \in Q$. Let $j = GF : Q \to Q$. This gives a monotone *closure operator* satisfying: (i) $a \leq j(a)$ and (ii) $j^2(a) \leq j(a)$, for all $a \in P$

2. *Free Algebras:* In categories of universal algebras, the left adjoint to U exists, where $\mathbf{Alg} \underset{U}{\overset{F}{\leftrightarrows}} \mathbf{Set}$. Here $F(X)$ is the free (universal) algebra generated by the set X (for a still more concrete example, replace **Alg** by the category **Mon** of monoids. Then $F(X) = X^*$, the free monoid on the set X.) In general, $\eta_X : X \to UFX$ is the "inclusion of generators" which maps the set X into the underlying set of the free algebra $F(X)$. The universal property of adjoint functors reduces to the familiar universal property of free algebras.

3. *Topological examples* There is an evident forgetful functor U: **Top**$\to$**Set**. This functor has both left and right adjoints. We leave it as an exercise to find them.

4. *Free Structures and Free Categories*: Generalizing the previous examples, left adjoints to forgetful functors typically determine "free" structures. A special case fundamental to categorical logic is the construction of free (structured) categories on graphs (see Exercise 1.9 below). We have an adjoint situation $\mathbf{Cat} \underset{U}{\overset{F}{\leftrightarrows}} \mathbf{Graph}$ in which the forgetful functor U has a left adjoint F, where $F(\mathcal{G})$ is the free (small) category generated by a (small) graph $\mathcal{G}$. More generally, we will later introduce free cartesian, cartesian closed, and $*$-autonomous categories. The point is that categorical logic allows us to construct such free categories directly from the formulas and proofs of certain logics.

Exercise 1.9 Prove $F \dashv U$, where $\mathbf{Cat} \underset{U}{\overset{F}{\leftrightarrows}} \mathbf{Graph}$ and $F(\mathcal{G})$ is the free (small) category generated by a (small) graph $\mathcal{G}$. In particular describe the η and the ε of the adjointness.

The next exercise is important: it illustrates Lawvere's slogan: *many categorical notions arise as adjoints to previously defined functors.*

Exercise 1.10 (Categorical Structure via Adjoints) 1. For any category $\mathcal{C}$, there is a unique functor to the one-object category: $\mathcal{C} \xrightarrow{\,!\,} \mathbf{One}$. Postulating that $\mathcal{C}$ has a left (resp. right) adjoint corresponds to saying $\mathcal{C}$ has an *initial object* $\perp$ (resp. a *terminal object* $\top$). The universal properties say: for any object $C \in \mathcal{C}$ there is a unique arrow $C \xrightarrow{!_C} \top$ (resp. a unique arrow $\perp \xrightarrow{0_C} C$). Letting $\{*\}$ be any one-element set, this says: $\mathcal{C}(C, \top) \cong \{*\}$ and $\mathcal{C}(\perp, C) \cong \{*\}$.

E.g. In **Set**, $\perp = \emptyset$ and $\top = \{*\}$ (any one element set). In **Vec**, $\perp = \top = \{0\}$, the trivial space.

2. *Products and Coproducts* : For any category $\mathcal{C}$, there is a diagonal functor $\mathcal{C} \xrightarrow{\Delta} \mathcal{C} \times \mathcal{C}$. If we postulate a right adjoint $\Delta \dashv R$, then for all $C, A, B \in \mathcal{C}$, $\mathcal{C} \times \mathcal{C}(\Delta(C), (A, B)) \cong \mathcal{C}(C, R(A, B))$. Writing $R(A, B) = A \times B$, show that we have a natural isomorphism $\mathcal{C}(C, A) \times \mathcal{C}(C, B) \cong \mathcal{C}(C, A \times B)$. We say $A \times B$ is the *cartesian product* of A and B. Dually, postulating a left adjoint $L \dashv \Delta$ determines a *coproduct* (= product in $\mathcal{C}^{op}$). Writing $L(A, B) = A + B$, show that this satisfies $\mathcal{C}(A + B, C) \cong \mathcal{C}(A, C) \times \mathcal{C}(B, C)$.

   E.g. In **Set**, $A \times B$ exists and is the usual cartesian product, while $A + B$ exists and is the disjoint union. In $\mathbf{Vec}_{fd}$ and Abelian Groups, products and coproducts exist and coincide: $V \times W \cong V \oplus W$. In **Top**, we obtain the usual product topology.

3. *Exponentials*: If $\mathcal{C}$ has products, prove that there is an induced functor $\mathcal{C} \times \mathcal{C} \xrightarrow{\times} \mathcal{C}$. Fix an object $A \in \mathcal{C}$. Consider the induced functor $\mathcal{C} \xrightarrow{-\times A} \mathcal{C}$, given on objects by $C \mapsto C \times A$. Suppose that $- \times A$ has a right adjoint $(-)^A$. Show that this means there is a natural isomorphism $\mathcal{C}(C \times A, B) \cong \mathcal{C}(C, B^A)$. This property is called *cartesian closedness*, and will be considered in the next section. Verify that in **Set**, the exponentials B^A exist, where B^A is the set of all functions from A to B. Investigate cartesian closedness in some of the other categories we have mentioned. What does it mean for a poset, considered as a category, to have products and exponentials? For a difficult problem, which categories of topological spaces have exponentials? For a discussion of the existence of topological cartesian closed categories, see [Mac].

1.7 Cartesian and Cartesian Closed Categories

We shall now begin a process of equationally axiomatizing categories with products and function spaces, as introduced in Exercise (1.10) above. These categories have significant connections to the proof theory of certain intuitionist propositional calculi.

Definition 1.11 A *cartesian category* is a category with finite products, i.e. binary products together with a terminal object.

Thus we have natural bijections:

$$\mathcal{C}(C, \top) \cong \{*\} \tag{1}$$
$$\mathcal{C}(C, A) \times \mathcal{C}(C, B) \cong \mathcal{C}(C, A \times B) \tag{2}$$

We shall be interested in categories with a *specified* cartesian structure. The following is a standard technique in categorical logic. Clearly we must postulate that there is a terminal object $\top$ and a binary operation on objects denoted $A \times B$. What about arrows? We shall chase the identity arrow starting from the RHS of isomorphism (2), since this is the only distinguished structure we have at hand. So, letting $C = A \times B$, the identity arrow $id_{A\times B}$ on the RHS maps to a pair of arrows on the LHS, which we call *projections* $A \times B \xrightarrow{\pi_1} A, A \times B \xrightarrow{\pi_2} B$. Conversely, going from the LHS to the RHS of (2), we wish to internalize pairing: given a pair of arrows $C \xrightarrow{f} A$ and $C \xrightarrow{g} B$, define a "pairing" $C \xrightarrow{\langle f,g\rangle} A \times B$. Using these operations, we will then impose equations specifying the bijections (1), (2).

So, on arrows we postulate the following distinguished structure, for all objects A, B, C:

- *Terminal*: An arrow $C \xrightarrow{!_C} \top$
- *Projections*: $A \times B \xrightarrow{\pi_1^{A,B}} A\ ,\ A \times B \xrightarrow{\pi_2^{A,B}} B$
- *Pairing*: $\dfrac{C \xrightarrow{f} A \quad C \xrightarrow{g} B}{C \xrightarrow{\langle f,g\rangle} A \times B}$

The isomorphisms (1) , (2) above may be given equationally by imposing the following identities (for all objects A, B, C):

$$!^C = f, \quad \text{for any } f : C \to \top \tag{3}$$

$$\begin{gathered}\pi_1\langle f, g\rangle = f\ ,\quad \pi_2\langle f, g\rangle = g\ ,\quad \langle \pi_1 h, \pi_2 h\rangle = h \\ \text{where } f : C \to A,\ g : C \to B,\ h : C \to A \times B\end{gathered} \tag{4}$$

Thus, a *cartesian category with specified structure* is given by the above data: an object $\top$, a binary operation $\times$ on objects, distinguished families of arrows $!_C, \pi_i^{A,B}, \langle f, g\rangle$ for all objects A, B, C, satisfying the above equations.

Exercise 1.12

1. Work out specified cartesian structure for **Set**, $\mathbf{Vec}_{fd}$ and **Top**.
2. Show that a poset $\mathbb{P}$, considered as a category, is a cartesian category iff it is a $\wedge$-semilattice with top element.

Example 1.13 (Deductive system for $\{\wedge, \top\}$ generated by $\mathcal{G}_0$) Let $\mathcal{G}_0$ be a discrete graph (cf. Exercise 1.3). We will now give an explicit description of the deductive system generated by $\mathcal{G}_0$. Formulas are freely generated from atoms (i.e. objects of $\mathcal{G}_0$), using $\{\wedge, \top\}$. Sequents are freely generated from the following "axioms" and "rules":

Axioms: $A \xrightarrow{id} A,\ \ A \wedge B \xrightarrow{\pi_1} A,\ \ A \wedge B \xrightarrow{\pi_2} B,\ \ C \xrightarrow{!^C} \top.$

Rules: $\dfrac{A \xrightarrow{f} B \quad B \xrightarrow{g} C}{A \xrightarrow{gf} C}\ cut \qquad \dfrac{C \xrightarrow{f} A \quad C \xrightarrow{g} B}{C \xrightarrow{\langle f,g\rangle} A \wedge B}\ pairing$

Finally, we obtain a cartesian category $F(\mathcal{G}_0)$ (*the free cartesian category generated by the discrete graph* $\mathcal{G}_0$) by letting the objects be the formulas and letting arrows be equivalence classes of proofs, where we impose the smallest congruence relation forcing the equations (3), (4) of a cartesian category to hold.

We generalize this example to arbitrary graphs in Exercise 1.15 below.

In general, categorical constructions given by universal mapping properties are only determined up to isomorphism. However in categorical logic and proof theory, it is natural to consider categories with specified structure (as above) and "strict" functors, i.e. those preserving the structure on the nose [LS].

Definition 1.14 **Cart** is the category of cartesian categories and *strict* cartesian functors, i.e. those preserving the structure on-the-nose: $F(A \times B) = F(A) \times F(B), F(\top) = \top$.

There is a forgetful functor **Cart** $\xrightarrow{U}$ **Graph**. This functor has a left adjoint **Graph** $\xrightarrow{F}$ **Cart**, where for any graph $\mathcal{G}$, $F(\mathcal{G})$ is *the cartesian category freely generated from* $\mathcal{G}$.

Exercise 1.15 (Free Cartesian Categories) Let $\mathcal{G}$ be a graph. Following Examples 1.2 and 1.13, construct $F(\mathcal{G})$ as a deductive system and prove that $F \dashv U$. [**Hint:** Objects of $F(\mathcal{G})$ are formulas in the language $\{\top, \wedge\}$ freely generated from the objects of $\mathcal{G}$ (i.e. consider the objects of $\mathcal{G}$ as atomic formulae). Proofs are freely generated from the Axioms in Example 1.13 along with all the arrows $A \xrightarrow{f} B \in \mathcal{G}$ (considered as nonlogical axioms) using the rules. Finally, impose the smallest congruence relation on proofs making $F(\mathcal{G})$ a cartesian category.]

Definition 1.16 A *cocartesian* category is the dual of a cartesian category, i.e. a category with binary coproducts and an initial object. A *cocartesian category with distinguished structure* is obtained by dualizing the cartesian case, i.e. we postulate the following distinguished structure, for all objects A, B, C:

- *Initial*: An arrow $\perp \xrightarrow{O_C} C$
- *Injections*: $A \xrightarrow{in_1^{A,B}} A + B$, $B \xrightarrow{in_2^{A,B}} A + B$
- *Copairing*: $\dfrac{A \xrightarrow{f} C \quad B \xrightarrow{g} C}{A + B \xrightarrow{[f,g]} C}$

The relevant equations specifying the isomorphisms in Exercise 1.10(2) are obtained by dualizing equations (3), (4). Free cocartesian categories may be obtained by setting up an appropriate deductive system for $\{\perp, \vee\}$ using the above structure (cf. Exercise 1.15). It is common to denote initial objects by 0, rather than $\perp$.

Cartesian Closed Categories

We now wish to equationally axiomatize those cartesian categories with specified exponentials (cf. Exercise 1.10).

Definition 1.17 A category $\mathcal{C}$ is *cartesian closed* (or a *ccc*) if it is cartesian and, for each object A, the endofunctor $\mathcal{C} \xrightarrow{- \times A} \mathcal{C}$ has a right adjoint, denoted $(-)^A$.

The adjointness says that for each A, we have an isomorphism $\mathcal{C}(C \times A, B) \cong \mathcal{C}(C, B^A)$, natural in C and B.

To axiomatize *specified* exponential structure (on top of specified cartesian structure) we specify: (i) there is also a binary operation B^A on objects; (ii) on arrows we postulate the following arrow schema and unary rule-schema for generating new arrows from old, in addition to the cartesian structure (for all objects A, B, C).

- *Evaluation:* $ev_{AB} : B^A \times A \longrightarrow B$
- *Currying:* $$\frac{C \times A \xrightarrow{f} B}{C \xrightarrow{f^*} B^A}$$

Finally, we impose the following equations in addition to the cartesian equations:

- (*Beta*) $ev\langle f^*\pi_1, \pi_2 \rangle = f : C \times A \to B$
- (*Eta*) $(ev\langle g\pi_1, \pi_2\rangle)^* = g : C \to B^A$.

Thus, a *cartesian closed category with specified structure* is given by the following data: a specified object $\top$, two binary operations $\times$ and $(-)^{(-)}$ on objects, the basic arrows $A \xrightarrow{id_A} A$, $A \xrightarrow{!_A} \top$, $A \times B \xrightarrow{\pi_1^{A,B}} A$, $A \times B \xrightarrow{\pi_2^{A,B}} B$, $B^A \times A \xrightarrow{ev_{A,B}} B$, the unary rule of Currying, and two binary rules of composition and pairing. Finally we postulate the equations of cartesian categories with (Beta) and (Eta).

Exercise 1.18 Check that the equations guarantee the bijection
$\mathcal{C}(C \times A, B) \cong \mathcal{C}(C, B^A)$, and this bijection is natural in C and B.

The category **CCC** is defined as follows: its objects are cartesian closed categories with specified structure and its morphisms are functors preserving the structure on the nose. There is a forgetful functor $\mathbf{CCC} \xrightarrow{U} \mathbf{Graph}$.

Examples 1.19

1. **Set** (see Exercise 1.10) and more generally functor categories (presheaves) $\mathbf{Set}^{\mathcal{C}^{op}}$. In the case of **Set**, B^A is the set of all functions from A to B, ev is *evaluation*: $ev(g, a) = g(a)$, and currying is: $f^*(c)(a) = f(c, a)$, for all $g \in B^A, f \in B^{C\times A}, a \in A, c \in C$.

2. ω-**CPO**: *Objects* are posets in which ascending ω-chains have suprema. *Arrows* are functions preserving suprema of chains (hence, monotone). Products are cartesian products, with pointwise order structure, $\top = \{*\}$, $B^A = \omega\text{-CPO}(A, B)$ with order and sups defined pointwise (e.g. $(\bigvee_n f_n)(a) = \bigvee f_n(a)$). The rest of the structure is induced from **Set**.

3. **Heyting Semilattices**: A cartesian closed poset $= (P, \leq, \top, \wedge, \Rightarrow)$ is a poset satisfying, for all $a, b, c \in P$,

$$a \leq \top \qquad\qquad \frac{c \leq a \quad c \leq b}{c \leq a \wedge b}$$

$$a \wedge b \leq a \quad a \wedge b \leq b, \qquad c \wedge a \leq b \text{ iff } c \leq a \Rightarrow b$$

So, $b^a = a \Rightarrow b$ is the largest element whose meet with a is less than or equal to b. A *Heyting Algebra* $(P, \leq, \top, \wedge, \Rightarrow, \vee, \bot)$ is a cartesian closed poset with finite coproducts and an initial object. These are the posetal models of intuitionistic propositional calculus.

The canonical example is due to Stone and Tarski: Let $X \in \mathbf{Top}$ be a topological space. Then $\mathcal{O}(X)$, the poset of open subsets of X, is a Heyting algebra: for $U, V \in \mathcal{O}(X)$, $U \wedge V = U \cap V$, $U \vee V = U \cup V$, $U \Rightarrow V = int((X \setminus U) \cup V)$, $\top = X$, $\bot = \emptyset$.

4. **Cat**: The category of (small) categories in Example 1.6 is cartesian closed. We have already introduced the notion of product of two categories, and we leave it as an exercise to verify that the appropriate functor category acts as an exponential in this setting.

5. **Deductive Systems for $\{\wedge, \Rightarrow, \top\}$ and free ccc's**: In general, ccc's will correspond to labelled deductions in intuitionistic $\{\wedge, \Rightarrow, \top\}$-logic. We add to the cartesian $\{\wedge, \top\}$-fragment one new axiom schema $ev_{A,B}$ (evaluation) and one new rule of inference (*Currying*):

$$(A \Rightarrow B) \wedge A \xrightarrow{ev_{A,B}} B \qquad \text{and} \qquad \frac{C \wedge A \xrightarrow{f} B}{C \xrightarrow{f^*} (A \Rightarrow B)}\ \text{Curry} \quad .$$

We form $F(\mathcal{G})$, *the free ccc generated from graph* $\mathcal{G}$, as follows. Formulas are generated from the atomic formulas (i.e. objects of $\mathcal{G}$) using $\{\top, \wedge, \Rightarrow\}$. Proofs are generated from the nonlogical axioms (i.e. arrows of $\mathcal{G}$) together with the axioms (identity), (terminal), (projections), (evaluation) using the rules: (pairing), and (currying). We impose the equations of ccc's between proofs.

The operation $F(-)$ is functorial. Indeed, the forgetful functor U has a left adjoint F, $\mathbf{CCC} \underset{U}{\overset{F}{\rightleftarrows}} \mathbf{Graph}$, with $F(\mathcal{G})$ the free ccc as described above.

Labels on proofs may be encoded by typed lambda terms, in the familiar manner. This is detailed in [LS]. For example, in the currying rule above, $f^* = \lambda_{x:A} f(\langle z, x\rangle)$ where $z : C$.

Finally, the universal property of $F(\mathcal{G})$ says the following: for any ccc $\mathcal{C}$ and graph morphism $J : \mathcal{G} \to U(\mathcal{C})$, there is a unique extension to a strict ccc-functor $[\![-]\!]_J : F(\mathcal{G}) \to \mathcal{C}$.

Exercise 1.20 (For λ-calculus hackers) Verify in what sense the equations (Beta) and (Eta) above correspond to their λ-calculus counterparts. Actually, (Beta)–as written–corresponds to a restricted version of β-conversion, where we substitute a *variable* rather than an arbitrary term.

6. **CCC's = Typed Lambda Calculi**: This example is basic to categorical logic and proof theory. Cartesian closed categories are equivalent to typed lambda calculi (with product types) in a strong sense. Let **CCC** be the category of ccc's with specified structure and strict ccc functors. Similarly, we may define the category of typed λ-calculi, whose objects are (not necessarily freely generated) typed lambda-calculi, and whose morphisms are *translations*, i.e. interpretations strictly preserving the lambda

structure (see [LS]). There is a natural equivalence of categories:

$$\mathbf{CCC} \underset{L}{\overset{C}{\leftrightarrows}} \textbf{Typed } \boldsymbol{\lambda}\textbf{-Calculus}$$

Here, associated to every ccc $\mathcal{C}$ there is a typed lambda calculus $L(\mathcal{C})$, the *internal language* of $\mathcal{C}$. Roughly speaking, the types of $L(\mathcal{C})$ are the objects of $\mathcal{C}$ and the terms are freely generated, using the arrows of $\mathcal{C}$ as new term-forming operations (where currying corresponds to λ-introduction). The equations are generated by the equalities in $\mathcal{C}$. Conversely, $C(\mathcal{L})$, the ccc *syntactically generated by a lambda theory* $\mathcal{L}$, is essentially the closed term model, viewed as a ccc (for details, see [LS]). We remark that for this to go through, we require that our languages (in this case typed λ-calculi) need not be freely generated (in the same sense that deductive systems can be generalized). Moreover, $F(\mathcal{G})$, the free ccc generated by graph $\mathcal{G}$, is equivalent to $C(L(\mathcal{G}))$, where $L(\mathcal{G})$ is the typed lambda calculus generated by the graph $\mathcal{G}$ (analogous to $L(\mathcal{C})$).

This categorical equivalence of ccc's, typed lambda calculi, and equivalence classes of proofs in intuitionistic deductive systems is the ultimate categorical form of the Curry-Howard isomorphism, and is due essentially to Lambek.

7. **Presheaves** $Set^{\mathcal{C}^{op}}$, the category of *presheaves on* $\mathcal{C}$, is the functor category whose objects are contravariant functors $\mathcal{C}^{op} \to \mathbf{Set}$ and whose maps are natural transformations between them. **Set** is the special case when $\mathcal{C} = \mathbf{One}$. The ccc structure of presheaves is given as follows:

$\top(A) = \{*\}$	$G^F \times F \xrightarrow{ev} G$ is defined by: $ev_C(\theta, c) = \theta_C(id_C, c)$, $c \in F(C)$.
$(F \times G)(A) = F(A) \times G(A)$	$\theta^* : H \to G^F$ is defined by:
$G^F(A) = nat(\mathcal{C}(-, A) \times F, G)$	$\theta^*_A(a)_C(h, c) = \theta_C(H(h)(a), c)$, where $h : C \to A, a \in H(A)$.

For some purposes, it is slightly more convenient to consider "covariant" presheaf categories $Set^{\mathcal{D}}$, which of course are included in the previous case, by observing that $Set^{\mathcal{D}} = Set^{(\mathcal{D}^{op})^{op}}$.

8. **Special Case:** G-Sets as presheaves.

Let G be a group, X a set. Let $Sym(X) =$ the group of all bijections of X. A G*-set* is a group homomorphism $G \to Sym(X)$. Equivalently, a G*-set* is a *left action map* $\cdot : G \times X \to X$, denoted $(g, x) \mapsto g{\cdot}x$ satisfying:

(i) $e_G{\cdot}x = x$, for all $x \in X$;

(ii) $g_1{\cdot}(g_2{\cdot}x) = (g_1 g_2){\cdot}x$, for all $g_i \in G$, $x \in X$.

The category G-**Set** of G-sets and G-set maps is defined as follows. Objects are G-sets. A G-set arrow $X \to Y$ is an *equivariant function*, i.e. a **Set**-function $f : X \to Y$ such that for all $g \in G, x \in X,\ f(g{\cdot}x) = g{\cdot}f(x)$.

Exercise: G-$\mathbf{Set} \cong \mathbf{Set}^G$, where in the right-hand-side, the group G is considered as a category with one object (in which all arrows are isos).

Hence G-**Set** is a ccc. The *ccc*-structure can be described as follows. Let X, Y be two G-sets.

Product: $X \times Y$, with action $g{\cdot}(x, y) = (g{\cdot}x, g{\cdot}y)$.

Exponentials: Y^X (all set maps), with action $(g{\cdot}f)(a) = g{\cdot}(f(g^{-1}{\cdot}a))$. In particular, we have $ev(g{\cdot}(f, x)) = g{\cdot}ev(f, x)$, for all $f \in Y^X, x \in X, g \in G$.

9. **Per(ℕ)**

A *per* (partial equivalence relation) is a symmetric, transitive relation on a set. We shall consider the category of pers on a functionally complete partial combinatory algebra. For example, consider the Kleene algebra $(\mathbb{N}, \cdot)$, in which $m{\cdot}n = \{m\}(n)$ is the application of the m partial recursive function to input n. We form the category **Per(ℕ)** as follows: the *objects* of **Per(ℕ)** are the pers on ℕ, denoted $R, S, T, \cdots$. The *arrows* of **Per(ℕ)** are equivalence classes of certain partial recursive functions, denoted by their gödel number. Given a partial recursive function $\{e\} : \mathbb{N} \rightharpoonup \mathbb{N}$, e represents an arrow $R \to S$ iff $\forall m, n[mRn \Rightarrow e{\cdot}m \downarrow, e{\cdot}n \downarrow$ and $e{\cdot}mSe{\cdot}n]$. Two indices representing arrows $e, e' : R \to S$ are equivalent, denoted $e \sim e' : R \to S$, iff $\forall m, n[mRn \to e{\cdot}m \downarrow, e'{\cdot}n \downarrow$ and $e{\cdot}mSe'{\cdot}n]$.

This structure forms a ccc. For products, the recursive bijection $\mathbb{N} \times \mathbb{N} \cong \mathbb{N}$, induces a pairing function $\langle -, - \rangle$. Define $\langle a, b\rangle R \times S \langle a', b'\rangle$ iff aRa' and bSb'. For exponentials, define $(S^R, \sim_{S^R}) = (\mathbf{Per}(\mathbb{N})(R, S), \sim)$, where $\sim$ is the above equivalence relation on indices. Getting the ccc structure, notably the operation of Currying, requires some elementary recursion theory (Kleene's s-m-n theorem) [BFSS]. This example admits many generalizations.

10. **Coherence Spaces and Stable Maps.** A *coherence space* $\mathcal{A}$ is a family of sets satisfying:

- (i) $a \in \mathcal{A}$ and $b \subseteq a$ implies $b \in \mathcal{A}$.
- (ii) if $B \subseteq \mathcal{A}$ and if $\forall c, c' \in B(c \cup c' \in \mathcal{A})$ then $\cup B \in \mathcal{A}$.

In particular, $\emptyset \in \mathcal{A}$. Morphisms are *stable maps*, i.e. monotone maps preserving pullbacks and filtered colimits. That is, $f : \mathcal{A} \to \mathcal{B}$ is a stable map if (i) $b \subseteq a \in \mathcal{A}$ implies $f(b) \subseteq f(a)$, (ii) $f(\cup_{i\in I} a_i) = \cup_{i\in I} f(a_i)$, for I directed, and (iii) $a \cup b \in \mathcal{A}$ implies $f(a \cap b) = f(a) \cap f(b)$. This gives a category **Coh**. Every coherence space $\mathcal{A}$ yields a reflexive-symmetric (undirected) graph $(|\mathcal{A}|, \frown\!\!\!\!\smile)$ where $|\mathcal{A}| = \{a \mid \{a\} \in \mathcal{A}\}$ and $a \frown\!\!\!\!\smile b$ iff $\{a, b\} \in \mathcal{A}$. Moreover, there is a bijective correspondence between such graphs and coherence spaces.

Given two coherence spaces $\mathcal{A}, \mathcal{B}$ their product $\mathcal{A} \times \mathcal{B}$ is defined via the associated graphs as follows: $(|\mathcal{A}\times\mathcal{B}|, \frown_{\mathcal{A}\times\mathcal{B}})$, with $|\mathcal{A}\times\mathcal{B}| = |\mathcal{A}|+|\mathcal{B}| = (\{1\}\times|\mathcal{A}|)\cup(\{2\}\times|\mathcal{B}|)$ where $(1,a) \frown_{\mathcal{A}\times\mathcal{B}} (1,a')$ iff $a \frown_{\mathcal{A}} a'$, $(2,b) \frown_{\mathcal{A}\times\mathcal{B}} (2,b')$ iff $b \frown_{\mathcal{B}} b'$, and $(1,a) \frown_{\mathcal{A}\times\mathcal{B}} (2,b)$ for all $a \in |\mathcal{A}|, b \in |\mathcal{B}|$. The function space $\mathcal{B}^{\mathcal{A}} = \mathbf{Coh}(\mathcal{A},\mathcal{B})$ of stable maps can be given the structure of a coherence space, ordered by Berry's order: $f \preceq g$ iff for all $a, a' \in \mathcal{A},\ a' \subseteq a$ implies $f(a') = f(a) \cap g(a')$. For details, see [GLT]. This class of domains led to the discovery of linear logic (cf. Example 2.15).

A *bicartesian closed category* (biccc) is a ccc with binary coproducts and an initial object (often denoted by 0). It corresponds to the proof theory of full intuitionistic propositional logic, i.e. of the connectives $\{\wedge, \vee, \Rightarrow, \top, \bot\}$.

Exercise 1.21 A bicartesian closed category satisfies $A^{B+C} \cong A^B \times A^C$, $A^0 \cong 1$ and the distributive law: $(A+B)\times C \cong (A\times C) + (B \times C)$.

Observe that until now we have been discussing the proof theory of *intuitionistic* logics. What can we say about the proof theory of classical logic? Writing $\neg A = 0^A = A \Rightarrow 0$, notice that in any biccc there is a canonical arrow or proof $A \to \neg\neg A$. A naive guess for a model for classical logic is to demand that this arrow should be an isomorphism, so $A \cong \neg\neg A$. Let us call such biccc's *Boolean categories.*

The following surprising theorem about biccc's also characterizes Boolean categories. For a proof and discussion of this theorem, see [LS].

Theorem 1.22 (Joyal) *In any biccc, there is at most one arrow $A \to 0$. In particular, in the associated intuitionistic propositional calculus, there is at most one proof of $A \to \bot$ and hence at most one proof of $\neg A$, up to equivalence of proofs.*

Thus Boolean categories are necessarily preorders and, up to equivalence of categories, the only such are boolean algebras!

So to understand the proof theory of classical logic requires a more sophisticated approach. It turns out that this involves categorical versions of Parigot's $\lambda\mu$-calculus and ideas arising from the notion of continuations in programming language theory. The appropriate categorical framework, called *control categories*, was developed by P. Selinger in [Sel].

2 Monoidal and $*$-Autonomous Structures

Definition 2.1 A *monoidal (or tensored) category* $(\mathcal{C}, I, \otimes, \alpha, \ell, r)$ is a category $\mathcal{C}$, with functor $\otimes : \mathcal{C}\times\mathcal{C} \to \mathcal{C}$, unit object $I \in ob(\mathcal{C})$, and specified isos: $\alpha_{ABC} : (A\otimes B)\otimes C \xrightarrow{\cong} A\otimes(B\otimes C)$, $\ell_A : I\otimes A \xrightarrow{\cong} A$, $r_A : A\otimes I \xrightarrow{\cong} A$ satisfying the following: $\ell_I = r_I : I\otimes I \to I$, as well as:

$$\begin{tikzcd}
A\otimes(I\otimes C) \arrow[r, "\alpha"] \arrow[d, "1\otimes\ell_C"'] & (A\otimes I)\otimes C \arrow[d, "r_A\otimes 1"] \\
A\otimes C \arrow[r, phantom, "="] & A\otimes C
\end{tikzcd}
\qquad
\begin{tikzcd}
A(B(CD)) \arrow[r, "\alpha"] \arrow[d, "1\otimes\alpha"] & (AB)(CD) \arrow[r, "\alpha"] & ((AB)C)D \\
A((BC)D) \arrow[rr, "\alpha"] & & (A(BC))D \arrow[u, "\alpha\otimes 1"]
\end{tikzcd}$$

where we omit $\otimes$'s in the second diagram for typographical reasons. This diagram is known as the Mac Lane pentagon.

A monoidal category is a very basic structure. There are any number of additional structures one may add to this basic definition. The structures of relevance to this paper are *symmetric* structure, *closed* structure, or *traced* structure. We now begin the description of these structures.

Suppose first that there is a natural isomorphism $s_{AB} : A \otimes B \to B \otimes A$ satisfying the following three diagrams:

$$(1)\quad \begin{tikzcd} A \otimes B \arrow[r, "s_{A,B}"] \arrow[dr, "id"'] & B \otimes A \arrow[d, "s_{B,A}"] \\ & A \otimes B \end{tikzcd} \qquad (2)\quad \begin{tikzcd} B \otimes I \arrow[rr, "s_{B,I}"] \arrow[dr, "\simeq"'] & & I \otimes B \arrow[dl, "\simeq"] \\ & B & \end{tikzcd}$$

$$(3)\quad \begin{tikzcd} A \otimes (B \otimes C) \arrow[r, "\alpha"] \arrow[d, "id_A \otimes s_{B,C}"'] & (A \otimes B) \otimes C \arrow[r, "s_{A\otimes B,C}"] & C \otimes (A \otimes B) \arrow[d, "\alpha"] \\ A \otimes (C \otimes B) \arrow[r, "\alpha"] & (A \otimes C) \otimes B \arrow[r, "s_{A,C} \otimes id"] & (C \otimes A) \otimes B \end{tikzcd}$$

$\mathcal{C}$ is *symmetric* if diagrams (1), (2), and (3) commute.

Examples 2.2

1. Any cartesian category, with $\otimes = \times$.

2. Any co-cartesian category (= finite coproducts), with $\otimes = +$

3. $\mathbf{Rel}_\times$. This is the category $\mathbf{Rel}$ whose objects are sets and whose arrows are binary relations. The functor $\otimes : \mathbf{Rel} \times \mathbf{Rel} \to \mathbf{Rel}$ is defined as follows. On objects, $\otimes = \times$, while on maps, $A \otimes B \stackrel{R\otimes S}{\longrightarrow} C \otimes D$ is given by: $(a, b)R \otimes S(c, d)$ iff aRc & bSd. The tensor unit $I = \{*\}$, any one element set.

4. $\mathbf{Rel}_+$. This is again the category $\mathbf{Rel}$, except $\otimes = +$ (disjoint union), $I = \emptyset$, and where $A \otimes B \stackrel{R\otimes S}{\longrightarrow} C \otimes D$ is given by:

$$(a, 0)R \otimes S(c, 0) \text{ if and only if } aRc$$

$$(b, 1)R \otimes S(d, 1) \text{ if and only if } bSd$$

where disjoint union in $\mathbf{Set}$ is given by: $X + Y = X \times \{0\} \cup Y \times \{1\}$

5. Two important monoidal subcategories of $\mathbf{Rel}_+$ are:

   1. **Pfn**: Sets and partial functions.
   2. **PInj**: Sets and partial injective functions.

6. $\mathbf{Vec}_{fd}$ and **Vec**: (finite dimensional) vector spaces over $\mathbf{k}$, where $\mathbf{k}$ is a field. Here $V \otimes W$ is taken to be the usual tensor product, and $I = \mathbf{k}$.

Next, it is natural to ask that the tensor product have an appropriate adjoint, and this leads us to our next definition.

Definition 2.3 A *symmetric monoidal closed category* (smcc) $\mathcal{C}$ is a symmetric monoidal category such that for all $A \in \mathcal{C}$, $- \otimes A : \mathcal{C} \to \mathcal{C}$ has a right adjoint $A \multimap -$, i.e. there is an isomorphism, natural in B, C, satisfying $\mathcal{C}(C \otimes A, B) \cong \mathcal{C}(C, A \multimap B)$. This is the monoidal analog of cartesian closed category; $A \multimap B$ is the "linear exponential" or "linear function space". In particular there are *evaluation* and *coevaluation* maps $(A \multimap B) \otimes A \to B$ and $C \to (A \multimap (C \otimes A))$, satisfying the adjoint equations.

Examples 2.4

1. Any ccc, with $A \otimes B = A \times B$ and $A \multimap B = A \Rightarrow B$.

2. A poset $\mathcal{P} = (P, \leq)$ is an smcc iff there are operations $\otimes, \multimap : P^2 \to P$, $1 \in P$ satisfying:

   1. $(P, \otimes, 1)$ is a commutative monoid.
   2. $\otimes, \multimap$ are functorial in the posetal sense: i.e. $x \leq x', y \leq y'$ implies $x \otimes y \leq x' \otimes y'$ and $x' \multimap y \leq x \multimap y'$
   3. (Closedness) $x \otimes y \leq z$ iff $x \leq y \multimap z$.

3. *Girard's Phase Semantics*: This is a posetal smcc, in the sense of Example 2 above. Let $M = (M, \cdot, e)$ be a commutative monoid. Consider the poset $\mathcal{P}(M)$, the powerset of M. We view $\mathcal{P}(M)$ as a poset ordered by inclusion. For $X, Y \in \mathcal{P}(M)$, define

$$
\begin{aligned}
X \otimes Y &= \{x \cdot y \mid x \in X, y \in Y\} =_{def} X \cdot Y \\
X \multimap Y &= \{z \in M \mid z \cdot X \subseteq Y\} \\
I &= \{e\}
\end{aligned}
$$

4. **Vec**, where $V \otimes W$ is the usual algebraic tensor product and $V \multimap W = Lin(V, W)$. More generally, consider $\mathcal{R}$-Modules over a commutative ring $\mathcal{R}$, with the standard algebraic notions of $V \otimes_{\mathcal{R}} W$ and $V \multimap W = Hom(V, W)$.

5. $\mathcal{MOD}(G)$. This example extends groups acting on sets to groups acting linearly on vector spaces. Let G be a group and V a vector space. A *representation of G on V* is a group homomorphism $\rho : G \to Aut(V)$; equivalently, it is a left G-action $G \times V \stackrel{\cdot}{\longrightarrow} V$ (satisfying the same equations as a G-set) such that $v \mapsto g{\cdot}v$ is a linear automorphism, for each $g \in G$. The pair (ρ, V) is called a *G-module* or *G-space*. $\mathcal{MOD}(G)$ has as objects the G-modules and as morphisms the linear maps commuting with the G-actions. Define the smcc structure of $\mathcal{MOD}(G)$ as follows:

$$\begin{array}{rcl} V \otimes W & = & \text{the usual tensor product, with action determined by} \\ & & g{\cdot}(v \otimes w) = g{\cdot}v \otimes g{\cdot}w \\ V \multimap W & = & Lin(V, W), \ \text{with action } (g{\cdot}f)(v) = g{\cdot}f(g^{-1}{\cdot}v) \ , \\ & & \textit{the contragredient action.} \end{array}$$

We recommend [FH] as an introduction to group representation theory.

Exercise 2.5 Formulate intuitionistic linear logic (ILL) as a deductive system, and show (with appropriate equations between proofs) it forms an smcc.

We now come to the fundamental definition, which will correspond to the proof theory of MLL. To model classical linear logic, we need an involutive negation. *In what follows, $\perp$ should not be confused with its use in the previous chapter (as an initial object).* The idea is that one chooses an object, which will be called $\perp$, and then defines (linear) negation via the formula $A^\perp = A \multimap \perp$. However to make this negation involutive, we should have $A = A^{\perp\perp}$, or since we are approaching this categorically, $A \cong A^{\perp\perp}$. This leads to the definition of *$*$-autonomous category*.

Definition 2.6 A *$*$-autonomous category* $(\mathcal{C}, \otimes, I, \multimap, \perp)$ is an smcc with a distinguished *dualizing* object $\perp$, such that (letting $A^\perp = A \multimap \perp$), the canonical map $\mu_A : A \to A^{\perp\perp}$ is an iso, for all A (i.e. "all objects are reflexive").

Facts about $*$-autonomous categories $\mathcal{C}$:

- We get a dualizing functor $\mathcal{C}^{op} \xrightarrow{(\ )^\perp} \mathcal{C}$ s.t. $\mathcal{C}(A, B) \cong \mathcal{C}(B^\perp, A^\perp)$ which is a natural iso.
- $\mathcal{C}$ is closed under duality of categorical constructions: e.g. $\mathcal{C}$ has products iff it has coproducts, pullbacks iff pushouts, $\mathcal{C}$ is complete iff co-complete, etc.
- $(A \multimap B)^\perp \cong A \otimes B^\perp$ and $I \cong \perp^\perp$ Also $A \multimap B \cong B^\perp \multimap A^\perp$.
- We may define $A \invamp B = (A^\perp \otimes B^\perp)^\perp$. In general, $\otimes \neq \invamp$, and (in general) there is not even a $\mathcal{C}$-morphism $A \otimes B \to A \invamp B$.
- As we shall see below, categorical models of **MALL** (multiplicative, additive linear logic) will be $*$-autonomous categories with products (and hence coproducts).

Example 2.7 *Sets and relations.* The category $\mathbf{Rel}_\times$, with its usual monoidal structure, is probably the simplest $*$-autonomous category. The dualizing object is any one element set. We leave the details as an exercise. We will consider this example further below when we introduce compact closed categories.

Example 2.8 *Finite-dimensional vector spaces.* The category $\mathbf{Vec}_{fd}$, with its usual monoidal structure, is also a $*$-autonomous category. The dualizing object is the base field. We will also consider this example further when we introduce compact closed categories.

Example 2.9 *$*$-autonomous posets and lattices.* Girard's phase semantics [G1] gives examples of $*$-autonomous lattices, i.e. structures $(P, \leq, \otimes, I, \multimap, \perp, \wedge, \vee)$ which are posetal $*$-autonomous categories. One method of construction is to consider closure operators $j : P \to P$ such that $j(x) \otimes j(y) \leq j(x \otimes y)$. We consider the j-closed elements, i.e. the fixed points $Fix(j) = \{p \in P \mid j(p) = p\}$. We then seek to define a $*$-autonomous structure on $Fix(j)$.

For example, consider Girard's phase semantics (Example 2.4). Observe that the powerset of a monoid $\mathcal{P}(M)$ has both a lattice as well as a phase semantics structure. Pick an arbitrary $\perp \in \mathcal{P}(M)$, and consider $(-)^{\perp} : \mathcal{P}(M) \to \mathcal{P}(M)$ given by: $X^{\perp} = \{p \mid p{\cdot}X \subseteq \perp\}$. Let $j = (-)^{\perp\perp}$. On the set $Fix(j) = (-)^{\perp\perp}$-closed elements of $\mathcal{P}(M)$ (Girard calls them *facts*), we define: $G \otimes H = (G{\cdot}H)^{\perp\perp}$, $G \parr H = (G^{\perp}{\cdot}H^{\perp})^{\perp}$,
$G \wedge H = G \cap H$, $G \vee H = (G \cup H)^{\perp\perp}$.

Example 2.10 *Finiteness spaces.* This example is due to T. Ehrhard [Ehr3]. It is an elaboration of the category **Rel**. Let X be a set and $u, v \subseteq X$ subsets of X. Say that u and v are *orthogonal*, written $u \perp v$, if $u \cap v$ is finite. If $\mathcal{F}$ is a set of subsets of X, write $\mathcal{F}^{\perp} = \{v \subseteq X | u \perp v \text{ for all } u \in \mathcal{F}\}$.

Ehrhard defines a *finiteness space* to be a pair $(X, \mathcal{F})$ where X is a set and $\mathcal{F}$ is a set of subsets of X such that $\mathcal{F}^{\perp\perp} = \mathcal{F}$. A morphism $R{:}\, (X, \mathcal{F}) {\to} (Y, \mathcal{G})$ is a subset $R \subseteq X \times Y$ such that for all $u \in \mathcal{F}$ we have $R(u) \in \mathcal{G}$, and for all $v \in \mathcal{G}$ we have $R^{op}(v) \in \mathcal{F}$. Here R^{op} is the reciprocal of R. It is straightforward to verify that this is indeed a category, with composition the usual relational composition. Then define:

- $(X, \mathcal{F})^{\perp} = (X, \mathcal{F}^{\perp})$
- $(X, \mathcal{F}) \otimes (Y, \mathcal{G}) = (X \times Y, \{u \times v | u \in \mathcal{F}, v \in \mathcal{G}\}^{\perp\perp})$
- $\perp = (\emptyset, \{\emptyset\})$

The rest of the details that we indeed have a $*$-autonomous category are straightforward.

Example 2.11 *Poset-valued sets.* This is a class of models constructed by de Paiva and Schalk [PS], which can also be thought of as a generalization of **Rel**. One considers a $*$-autonomous poset, P, for example a Girard quantale or a phase space, as described above. Then a *P-valued set* is defined to be a pair (A, f) where A is a set and $f{:}\, A {\to} P$ is a function. A morphism between P-valued sets $R{:}\, (A, f) {\to} (B, g)$ is a relation $R{:}\, A {\to} B$ such that xRy implies $f(x) \leq g(y)$. Then one defines $(A, f) \otimes (B, g) = (A \times B, f \otimes g)$, where $f \otimes g$ is defined using the monoidal structure of P. The rest of the $*$-autonomous structure of P similarly lifts to the category.

Example 2.12 *Topological vector spaces.* We have already mentioned that the category of finite-dimensional vector spaces is $*$-autonomous, with the usual notion of dual space acting as negation. If one only wishes to consider discrete vector spaces, this is the best one can do. Indeed it is a standard result that a vector space is isomorphic to its second dual if and only if it is finite-dimensional, the problem being that the second dual of an infinite-dimensional space is substantially larger than the original space. If one wishes to consider infinite-dimensional spaces, one must add an additional topological structure.

So one passes to a category in which the objects are topological vector spaces and the morphisms are linear *continuous* maps. The hope in doing this is that in requiring continuity, one will decrease the size of the dual space to such an extent that one will be able to obtain additional objects isomorphic to their second dual and still retain the closed structure. It was the consideration of such spaces by Barr that led to the original axiomatization of $*$-autonomy. One appropriate notion of topology, introduced by Lefschetz, is the *linear topology*. This is the notion that led Barr to his axiomatization. See [Barr80, Bl96] for the details of the following.

Definition 2.13 Let V be a vector space. A topology τ on V is a *linear topology* if it satisfies the following three properties:

- τ is hausdorff.
- The topology τ makes V a topological vector space, i.e. addition and scalar multiplication are continuous.
- The origin has a neighborhood basis of open linear subspaces.

We get a category **TVec** when one takes as morphisms the linear, continuous maps. It can be shown that this is a symmetric monoidal closed category. The tensor product in **TVec** is given by an appropriate topology on the tensor of the underlying spaces, and the internal hom is given by the space of linear, continuous maps, again with an appropriate topology. This notion of topology is ideal in that one can show that the usual embedding of V into its second dual is always a bijection. So we have indeed reduced the second dual space to the appropriate size. If one restricts to the category of objects for which the embedding is also a homeomorphism, one obtains a category **RTVec**, and Barr demonstrates:

Theorem 2.14 **RTVec** *is a complete, cocomplete $*$-autonomous category.*

Example 2.15 *Coherence spaces and linear maps.* This example led to linear logic. Recall the ccc **Coh** of coherence spaces and stable maps was discussed at the end of Section 1.7, Example 10. A morphism $f : \mathcal{A} \to \mathcal{B}$ in **Coh** is *linear* if for any $X \subseteq \mathcal{A}$ such that for all $b, c \in X$, $b \cup c \in \mathcal{A}$, we have $f(\bigcup X) = \bigcup\{f(b) \mid b \in X\}$. Let $\mathbf{Coh}_{lin}$ be the subcategory of **Coh** consisting of coherence spaces and linear maps. This is $*$-autonomous, via the familiar constructions [GLT].

Example 2.16 *And many more....* The above list is by no means comprehensive. $*$-autonomous categories appear in many guises, in many branches of mathematics. There are at least three additional examples which should certainly be mentioned.

- Game semantics. An extremely important class of examples arises from game theory, with important computational properties. We recommend [Abr97] as an introduction.
- The Chu construction. This is a simple construction, which applied to a symmetric monoidal closed category (with pullbacks) canonically yields a $*$-autonomous category. Despite being straightforward, it would seem to have a number of remarkable properties. The construction is due to Barr and Chu [Barr80], and has been studied extensively by Pratt. See [DHPP] for one example of the applicability of Chu spaces.
- Recent work of Ehrhard and Regnier on Köthe spaces and the differential lambda-calculus suggests a whole new avenue to explore in the categorical semantics of linear logic. See [Ehr2, ER].

Summary of necessary structure

To aid comparison with proof theory, let us finally sum up the situation so far:

SMCC's with Products

Arrow-generating Rules	Equations
$A \xrightarrow{id} A \qquad \dfrac{A \xrightarrow{f} B \quad B \xrightarrow{g} C}{A \xrightarrow{gf} C}$	equations of a category
$\dfrac{A \xrightarrow{f} B \quad A' \xrightarrow{g} B'}{A \otimes A' \xrightarrow{f \otimes g} B \otimes B'}$	$\otimes$ is a functor : $ff' \otimes gg' = (f \otimes g)(f' \otimes g')$ $id \otimes id = id$
$A \otimes (B \otimes C) \xrightarrow{\alpha} (A \otimes B) \otimes C$ $A \otimes B \xrightarrow{s} B \otimes A$ $I \otimes A \xrightarrow{\ell} A$	α, s, ℓ are natural isos and equations for smcc's
$\dfrac{A \otimes B \xrightarrow{f} C}{A \xrightarrow{f^*} (B \multimap C)} \multimap\mathrm{R}$ $(A \otimes B) \otimes A \xrightarrow{ev} B$	equations for monoidal closedness
$\dfrac{\Gamma \xrightarrow{f} A \quad \Gamma \xrightarrow{g} B}{\Gamma \xrightarrow{\langle f,g \rangle} A \times B}$ $A \times B \xrightarrow{\pi_1} A \quad A \times B \xrightarrow{\pi_2} B$ $A \xrightarrow{!_A} \top$	cartesian products

This gives us symmetric monoidal closed categories (smcc's) with products. At this point we could also add coproducts, denoted $+$ (or in linear logic $\oplus$), and their associated equations, dual to products. But as mentioned previously, once we have $*$-autonomous categories, we get duality for free, essentially by De Morgan duality.

Adding Negation

Arrow-generating Rules	Equations
$\dfrac{A \xrightarrow{f} B}{B^\perp \xrightarrow{f^\perp} A^\perp}$	$(-)^\perp$ is contravariant functor
$A^\perp \longrightarrow (A \multimap \perp)$ $(A \multimap \perp) \longrightarrow A^\perp$	These are natural isos
$(A \multimap B) \to (B^\perp \multimap A^\perp)$	Natural strength iso
$A \to ((A \multimap \perp) \multimap \perp)$	natural iso

These equations specify that the action of the functor $(-)^\perp$ is given by a dualizing object $\perp$, and a natural iso $(-)^{\perp\perp} \cong id$.

The next notion is much more familiar mathematically, although logically it corresponds to a rather degenerate case of linear logic: the case where $\otimes = \invamp$:

Definition 2.17 A *compact closed category* [KL] is a symmetric monoidal category such that for each object A there exists a dual object A^*, and canonical morphisms:

$$\nu: I \to A \otimes A^*$$
$$\psi: A^* \otimes A \to I$$

such that evident equations hold. In the case of a strict monoidal category, these equations reduce to the usual adjunction triangles.

Lemma 2.18

- *Compact closed categories are $*$-autonomous, with the tensor unit as dualizing object.*
- *As in any $*$-autonomous category in which the tensor unit is the dualizing object, there is a canonical morphism $A \otimes B \to A \invamp B$. (This is an instance of a more general observation. Such categories validate the* **Mix** *rule, which states:*

$$\frac{\vdash \Gamma \quad \vdash \Delta}{\vdash \Gamma, \Delta}\, Mix$$

This rule is not valid in linear logic, but the theory with **Mix** *added is of great interest.) For compact closed categories, this map is an isomorphism.*

Examples 2.19

- $\mathbf{Rel}_\times$ is compact. On objects, define $A \otimes B = A \multimap B = A \times B$, $I = \perp = \{*\}$ (any 1-element set) and $A^\perp = A$. On morphisms, if $A \xrightarrow{R} B$, define $B^\perp \xrightarrow{R^\perp} A^\perp = B \xrightarrow{R^{op}} A$. It is easy to check that

$$\mathbf{Rel}_\times(C \otimes A, B) \cong \mathbf{Rel}_\times(C, A \multimap B) \cong \mathcal{P}(C \times A \times B)$$

$$\mathbf{Rel}_\times(A, B) \cong \mathbf{Rel}_\times(B^\perp, A^\perp)$$
$$(A \otimes B)^\perp = A \times B = A^\perp \otimes B^\perp$$

- $\mathbf{Vec}_{fd}$ (finite dimensional vector spaces over field $\mathbf{k}$) is also compact: Here $\perp = \mathbf{k}$, $V \multimap W = Lin(V, W)$, so $V^\perp = V \multimap \perp = Lin(V, \mathbf{k}) = V^*$, the dual space of V. There is a natural isomorphism $V \cong V^{**}$ given by the canonical map $V \to V^{**}$. Indeed, an arbitrary vector space is finite dimensional iff this canonical map $V \to V^{**}$ is an isomorphism.

- Let P and P' be posets. An *order ideal* from P to P' is a relation $R \subseteq P \times P'$ satisfying

$$x_1 \leq_P x_2 \;\&\; x_2 R y_2 \;\&\; y_2 \leq_{P'} y_1 \implies x_1 R y_1$$

  One readily verifies that order ideals do indeed form a category with the inequality itself, viewed as a binary relation, acting as identity. The compact closed structure of $\mathbf{Rel}_\times$ extends readily, except now $P^\perp = P^{op}$. See [RW] for a detailed discussion of this category.

What are monoidal functors between monoidal categories? Here there may be several notions. Let us pick an important one:

Definition 2.20 A *monoidal functor* between monoidal categories is a 3-tuple (F, m_I, m) where $F : \mathcal{C} \to \mathcal{D}$ is a functor, together with two natural transformations $m_I : I \longrightarrow F(I)$ and $m_{UV} : F(U) \otimes F(V) \longrightarrow F(U \otimes V)$ satisfying some coherence diagrams (which we omit). F is *strict* if m_I, m_{UV} are identities. A monoidal functor is *symmetric* if m commutes with the symmetries: $m_{B,A} s_{FA,FB} = F(s_{A,B}) m_{A,B}$, for all A, B.

Finally, we need an appropriate notion of natural transformation for monoidal functors.

Definition 2.21 A natural transformation $\alpha : F \to G$ is *monoidal* if it is compatible with both m_I and m_{UV}, for all U, V, in the sense that the following equations hold:

$$\begin{aligned} \alpha_I \circ m_I &= m_I \\ m_{UV} \circ (\alpha_U \otimes \alpha_V) &= \alpha_{U \otimes V} \circ m_{UV} \end{aligned}$$

Let $*$-$\mathbf{Aut}_{st}$ be the category of $*$-autonomous categories and strict $*$-autonomous functors. We wish to construct *free* such categories, i.e. to find a left adjoint F to the forgetful functor $U : *$-$\mathbf{Aut}_{st} \to \mathbf{Graph}$, so $F(\mathcal{G})$ will be the free $*$-autonomous category generated by the graph $\mathcal{G}$. The procedure is now familiar: one sets up an appropriate deductive system (generated by $\mathcal{G}$) for MLL (cf. Example 5 in Section 1.7) and imposes the relevant equations between proofs. A related, but more delicate issue is to set up the fundamental equivalence of categorical logic, as in Example 6 in Section 1.7 , between $*$-autonomous categories and their *internal logics*, which are calculi of proof-terms. Thorny categorical questions like dealing with the units and coherence equations must also be taken into account.

These and related issues are discussed in the work of Cockett, Seely and others, see [CS1, CS2, BCST, BCS1, BCS2], which begins with an alternate approach to $*$-autonomous categories and the model theory of linear logic. The starting point is the notion of *linearly distributive category*[1]. Roughly, *LDC*s axiomatize multiplicative linear logic in terms of tensor and par, as opposed to tensor and negation. So an *LDC* is a category with two monoidal structures which interact via a linear distribution. One may then add negation as an additional structure.

Definition 2.22 A *symmetric linearly distributive category* (SLDC) is a category $\mathcal{C}$ equipped with

- Two bifunctors $\otimes, \parr : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$, together with objects and isomorphisms endowing $\mathcal{C}$ with two monoidal structures.
- Linear distributivity natural transformations
  1. $\omega^L_{ABC} : A \otimes (B \parr C) \to (A \otimes B) \parr C$
  2. $\omega^R_{ABC} : A \otimes (B \parr C) \to (A \otimes C) \parr B$
- a number of coherence conditions.

A *symmetric linearly distributive category with negation* is an SLDC together with an object function $(-)^\perp$ on $ob(\mathcal{C})$ and natural maps $A \otimes A^\perp \xrightarrow{\gamma_A} \perp$, $I \xrightarrow{\tau_A} A \parr A^\perp$ satisfying:

$$\begin{tikzcd}
I \otimes A \arrow[r, "\tau \otimes A"] \arrow[dd, "\ell^{\otimes}_A"'] & (A \parr A^\perp) \otimes A \arrow[dr, "sw^R s"] & \\
& & A \parr (A^\perp \otimes A) \arrow[dl, "A \parr \gamma s"] \\
A & A \parr \perp \arrow[l, "\ell^{\parr}"] &
\end{tikzcd}$$

Theorem 2.23 (Cockett-Seely) *SLDC's with negation are the same as $*$-autonomous categories.*

Let us give an example from the above reference.

[1]Originally referred to as weakly distributive categories.

Example 2.24 A *shift monoid* $(M, 0, +, a)$ is a commutative monoid with an invertible element a. Let $x{\cdot}y = x + y - a$. Then note that $x{\cdot}(y+z) = (x{\cdot}y)+z$, which is an instance of a linear distributivity. Shift monoids are exactly discrete linearly distributive categories. *Shift groups* , i.e. shift monoids which are groups, are exactly discrete $*$-autonomous categories. In the latter case, linear negation is defined by $x^{\perp} = a - x$.

The references cited above, beginning with the notion of linearly distributive category, extend the categorical analysis of linear logic in several directions. Using (two-sided) proof nets, a natural deduction system for various fragments of linear logic, the authors give explicit constructions of free weakly distributive and $*$-autonomous structures [BCST], and extend this idea to the exponential fragment [CS1]. In [CS2, BCS2], these ideas are extended to include functors between linearly distributive categories and a logic for the analysis of such functors is presented.

3 Monads and Exponentials

3.1 Monads

Monads (also called triples or standard constructions) arose in the 1960's from the theory of adjoint functors and have played a central role in category theory ever since. It turns out that many categories of interest are "monadic", i.e. equivalent to categories of (Eilenberg-Moore) algebras of a monad, and thus arise from a pair of adjoint functors, as we discuss below. The general theory has many consequences, for example "monadicity" is a far-reaching generalization of the concept of "equationality" which includes not only traditional equational varieties of universal algebras but also theories with infinitary operations as well as certain topological categories (e.g. compact Hausdorff spaces).

More recently, monads have played an important role in theoretical computer science. For example, the power set monad discussed below (and its domain-theoretic variants introduced by Plotkin) are now a standard tool in modelling nondeterminacy, while coalgebraic methods are fundamental in concurrency theory. The influential work of E. Moggi [Mog] introduced monads and comonads into programming language semantics as a kind of structuring tool: they permit a modular treatment of such important programming features as exceptions, side-effects, non-determinism, resumptions, dynamic allocation, etc.

Definition 3.1 A *monad* on a category $\mathcal{C}$ is a 3-tuple (T, η, μ), where $T : \mathcal{C} \to \mathcal{C}$ is a functor, $\eta : Id \longrightarrow T$ (unit) and $\mu : T^2 \longrightarrow T$ (multiplication) are two natural transformations satisfying the following equations:

$$\begin{tikzcd} T \arrow[r, "T\eta"] \arrow[d, "\eta T"'] \arrow[dr, "id"'] & T^2 \arrow[d, "\mu"] \\ T^2 \arrow[r, "\mu"'] & T \end{tikzcd} \qquad \begin{tikzcd} T^3 \arrow[r, "\mu T"] \arrow[d, "T\mu"'] & T^2 \arrow[d, "\mu"] \\ T^2 \arrow[r, "\mu"'] & T \end{tikzcd}$$

A *comonad* on $\mathcal{D}$ is a monad on $\mathcal{D}^{op}$. Thus a comonad is a functor $G : \mathcal{D} \to \mathcal{D}$, together with natural transformations $\varepsilon : G \to Id$ (counit) and $\delta : G \to G^2$ (comultiplication) satisfying the dual of the diagrams above.

Examples 3.2

1. *Power Set Monad*: The covariant power set functor $\mathcal{P} : \mathbf{Set} \to \mathbf{Set}$ determines a monad $(\mathcal{P}, \mu, \eta)$, where $\eta : Id \to \mathcal{P}$ is given by: $\eta_X(x) = \{x\}$ and $\mu : \mathcal{P}^2 \to \mathcal{P}$ is given by: $\mu_A(\mathcal{F}) = \cup\mathcal{F}$, where $\mathcal{F} \subseteq \mathcal{P}(A)$

2. *Adjoint Functors*: The canonical examples of monads and comonads arise from a pair of adjoint functors $\mathcal{D} \underset{U}{\overset{F}{\leftrightarrows}} \mathcal{C}$, where $F \dashv U$. Let $T = UF : \mathcal{C} \to \mathcal{C}$. Following Exercise 1.7, there are natural transformations $\eta : Id \to UF$ and $\mu = U\varepsilon F : UFUF \to UF$. We leave it as an exercise to check the monad equations.

   Continuing this example, we also obtain a comonad $G = FU : \mathcal{D} \to \mathcal{D}$, where $\varepsilon : FU \to Id$ and $\delta = F\eta U : FU \to FUFU$.

In fact, *every* monad $T : \mathcal{C} \to \mathcal{C}$ arises from a pair of adjoint functors $\mathcal{D} \underset{U}{\overset{F}{\leftrightarrows}} \mathcal{C}$. Although the category $\mathcal{D}$ is not unique, there are two natural choices:

Theorem 3.3 (Kleisli,Eilenberg-Moore) Every monad (T, η, μ) on $\mathcal{C}$ arises from a pair of adjoint functors $\mathcal{D} \underset{U}{\overset{F}{\leftrightarrows}} \mathcal{C}$ for two choices of $\mathcal{D}$, now called the Kleisli and the Eilenberg-Moore categories, respectively.

- *The Kleisli Category* of T :

  $\mathcal{D} = Kleisli(T)$ is defined as follows: the objects of $\mathcal{D}$ are the same as the objects of $\mathcal{C}$. The hom-sets are defined as $\mathcal{D}(A, B) = \mathcal{C}(A, TB)$. We define the categorical structure of $\mathcal{D}$ as follows:

  1. *Identity arrows* are defined by setting $id_A : A \to A$ in $\mathcal{D} = \eta_A : A \to TA$ in $\mathcal{C}$.
  2. *Composition* in $\mathcal{D}$ is defined via composition in $\mathcal{C}$ as:

$$\frac{A \xrightarrow{f} B \quad B \xrightarrow{g} C}{A \xrightarrow{gf} C} \text{ in } \mathcal{D} \quad = \quad \cfrac{\cfrac{A \xrightarrow{f} TB \quad \cfrac{B \xrightarrow{g} TC}{TB \xrightarrow{Tg} T^2C}}{A \xrightarrow{(Tg)f} T^2C} \qquad T^2C \xrightarrow{\mu_C} TC}{A \xrightarrow[\mu_C(Tg)f]{} TC}$$

  We leave it as an exercise to verify that the associated pair of adjoint functors $\mathcal{D} \underset{U}{\overset{F}{\leftrightarrows}} \mathcal{C}$ are given by: $U(A) = T(A)$, $U(h) = \mu_B T(h)$ for any object A and any arrow $h : A \to B$ in $\mathcal{D} = Kleisli(T)$ and $F(A) = A$, $F(f) = \eta_B f$ for objects A and arrows $f : A \to B$ in $\mathcal{C}$.

- *The Eilenberg-Moore Category*

  The Eilenberg-Moore category $\mathcal{D} = \mathcal{C}^T$ of a monad T is defined as follows: its objects are arrows $TA \xrightarrow{\alpha} A$ (called T-*algebras*) satisfying: $\alpha\eta_A = id_A$ and $\alpha\mu_A = \alpha T(\alpha)$. Morphisms of T-algebras are arrows $A \xrightarrow{f} B \in \mathcal{C}$ commuting with the T-algebra structure. The definition of objects and arrows in $\mathcal{C}^T$ is illustrated by the following commuting diagrams:

$$\begin{tikzcd} A \arrow[r, "\eta_A"] \arrow[dr, "id"'] & TA \arrow[d, "\alpha"] \\ & A \end{tikzcd} \qquad \begin{tikzcd} T^2A \arrow[r, "\mu_A"] \arrow[d, "T(\alpha)"'] & TA \arrow[d, "\alpha"] \\ TA \arrow[r, "\alpha"] & A \end{tikzcd} \qquad \begin{tikzcd} TA \arrow[r, "Th"] \arrow[d, "\alpha"'] & TB \arrow[d, "\beta"] \\ A \arrow[r, "h"] & B \end{tikzcd}$$

  We leave it to the reader to check that $\mathcal{C}^T$ is a category, with functors $\begin{tikzcd} \mathcal{C}^T \arrow[r, shift left, "U"'] & \mathcal{C} \arrow[l, shift left, "F"'] \end{tikzcd}$ given by: $U(TA \xrightarrow{\alpha} A) = A$ and $U(h) = h$ for objects and arrows in $\mathcal{C}^T$, and $F(A) = T^2(A) \xrightarrow{\mu_A} TA$ and $F(f) = T(f)$ for objects and arrows in $\mathcal{C}$.

Exercise 3.4 1. The Kleisli category of the power set monad $\mathcal{P}$ on **Set** has sets for objects and functions $A \xrightarrow{r} \mathcal{P}(B)$ for arrows. The arrows may be identified with relations $R \subseteq A \times B$. Check that composition in the Kleisli category corresponds to relational composition. Conclude that $Kleisli(\mathcal{P}) = \mathbf{Rel}$.

2. The Eilenberg-Moore category of $\mathcal{P}$ is exactly the category of sup-complete lattices and sup-preserving maps.

It may be shown that the Kleisli category of a monad is equivalent to the full subcategory of the Eilenberg-Moore category consisting of all free algebras, where free algebras in the Eilenberg-Moore category are those of the form $\mu_A : T^2A \to TA$. .

We can dualize the entire discussion above and speak of the co-Kleisli category of a comonad $G : \mathcal{D} \to \mathcal{D}$, of Eilenberg-Moore categories of coalgebras, etc. In this case, $Co - Kleisli(G)$ will be equivalent to the full subcategory of cofree coalgebras of the Eilenberg-Moore category of G. This will be relevant for linear logicians, as we now show.

3.2 Adding Exponentials to Linear Logic

The deductive system for MALL, and the equations between proofs we postulated previously, correspond to the theory of $*$-autonomous categories with products and coproducts. Although minor variations are possible (e.g. weak vs ordinary products), the story so far seems to yield a natural and satisfying categorical modelling of MALL proof theory.

Unfortunately, the exponentials are less clear: the structure seems less canonical. Work by many categorical logicians has refined the original Seely model (e.g. see [See, Bier95, BCS1]), resulting in interesting and reasonable equations between proofs. We begin with seven basic derivations and postulate equations which arise directly from the categorical point of view.

Functoriality	$\dfrac{A \xrightarrow{f} B}{!A \xrightarrow{!f} !B}$
Monoidalness	$I \xrightarrow{m_I} !I \qquad !A \otimes !B \xrightarrow{m_{AB}} !(A \otimes B)$
Products	$I \xrightarrow{n_I} !\top \quad !A \otimes !B \xrightarrow{n_{AB}} !(A \& B)$
Dereliction	$!A \xrightarrow{\varepsilon_A} A$
Weakening	$!A \xrightarrow{\varepsilon'_A} I$
Contraction	$!A \xrightarrow{\delta'_A} !A \otimes !A$
Digging (Storage)	$!A \xrightarrow{\delta_A} !!A$

Let $\mathcal{C}$ be a model of MALL proofs, i.e. a $*$-autonomous category with products (and hence coproducts). We postulate the following additional equational data:

- $(!, m_I, m_{AB}) : \mathcal{C} \to \mathcal{C}$ is a monoidal endofunctor
- $!A \xrightarrow{\varepsilon_A} A$ and $!A \xrightarrow{\delta_A} !!A$ are monoidal natural transformations.
- $(!, \delta, \varepsilon)$ is a monoidal comonad.
- n_I, n_{AB} are natural isomorphisms.
- The associated adjunction structure $\langle F, U, \eta, \varepsilon \rangle$ between the co-Kleisli category of ! and $\mathcal{C}$ is monoidal.
- Various coherence equations [BCS1, HS].

Having products and the canonical isomorphism $!(A \& B) \cong\ !A \otimes !B$ gives added features that must be postulated in weaker fragments (cf [Bier95, BCS1]). For example, the following are a consequence of the above properties:

Examples 3.5

(i) The endofunctor ! establishes an isomorphism of the following cocommutative comonoids: $I \xleftarrow{\varepsilon'_A} !A \xrightarrow{\delta'} !A \otimes !A \cong\ !(\top \longleftarrow A \xrightarrow{\Delta} A \& A)$

(ii) The forgetful functor from the category of $\otimes$-comonoids in $\mathcal{C}$, say $\otimes$-Comonoids$(\mathcal{C}) \xrightarrow{U} \mathcal{C}$ has a right adjoint $U \dashv\ !$ rendering $(!A, \varepsilon'_A, \delta'_A)$ a cofree,cocommutative $\otimes$-comonoid object in $\mathcal{C}$.

Finally, we remark that the essence of Girard's original translation of intuitionistic logic into CLL is the following observation:

Proposition 3.6 (Seely) *The co-Kleisli category of the comonad* $(!, \delta, \varepsilon)$ *is a cartesian closed category, in which* $A \Rightarrow B =\ !A \multimap B$

Indeed, recall that in the co-Kleisli category of $!$, $Hom(A, B)$ is defined to be $\mathcal{C}(!A, B)$.

Examples 3.7

1. The category **Rel** has exponentials. Let X be a set, and define $!X$ to be the set of all finite multisets on X. We leave the remaining details to the reader. But we note that this is an instance of a more general construction. Finite multisets on X could be written as follows:

$$!X = 0 \oplus X \oplus X \otimes_s X \oplus X \otimes_s X \otimes_s X \ldots$$

   Here 0 is the initial object, $\oplus$ is disjoint union, which acts as both product and coproduct in this category, and $\otimes_s$ is the *symmetrized tensor*, and is expressed as a certain quotient (coequalizer). For example, $X \otimes_s X$ is the coequalizer of the identity and the symmetry map. In general, the n'th symmetric group acts on the n-fold tensor of X with itself, and the symmetrized tensor is the coequalizer of all of these maps.

   It was an observation of Barr that this formula works frequently, but certainly not always. We recommend [HS] as a reference which considers these issues.

2. The inclusion $\mathbf{Coh}_{lin} \hookrightarrow \mathbf{Coh}$ has a left adjoint $! \dashv \hookrightarrow$. Thinking of a coherence space as a graph, $!\mathcal{A} = (\mathcal{A}_{fin}, \frown)$, where $\mathcal{A}_{fin}$ is the set of finite cliques in the graph $\mathcal{A}$ and where $a \frown b$ iff $a \cup b$ is a clique. The co-Kleisli category induced by this comonad is equivalent to **Coh**. (see for example [Tr]).

3. The category of finiteness spaces described above also has an exponential, as observed in [Ehr3]. Let $(X, \mathcal{F})$ be a finiteness space, then define its exponential by

$$!(X, \mathcal{F}) = (M(X), M(\mathcal{F}))$$

   where $M(X)$ is the set of all finite multisets on X, and $M(\mathcal{F}))$ is an appropriate set of subsets. See [Ehr3] for further details.

4 Traced monoidal categories and the geometry of interaction

Traced monoidal categories, introduced by Joyal, Street, and Verity [JSV], provide a convenient framework for discussing iteration, parametrized feedback and fixedpoints in computation, algebra of networks, and categorical aspects of Girard's Geometry of Interaction (GoI) program [AHS, Hagh, Hi].

Definition 4.1 A *traced symmetric monoidal category* is a symmetric monoidal category $(\mathcal{C}, \otimes, I, s)$ with a family of functions $\mathtt{Tr}^U_{X,Y} : \mathcal{C}(X \otimes U, Y \otimes U) \longrightarrow \mathcal{C}(X, Y)$ pictured in Figure 1, called a *trace*, subject to the following conditions:

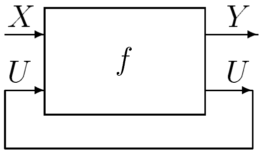


Figure 1: The trace $\mathtt{Tr}^U_{X,Y}(f)$

1. **Natural** in X, $\mathtt{Tr}^U_{X,Y}(f)g = \mathtt{Tr}^U_{X',Y}(f(g \otimes 1_U))$, where $f : X \otimes U \longrightarrow Y \otimes U$, $g : X' \longrightarrow X$,
2. **Natural** in Y, $g\mathtt{Tr}^U_{X,Y}(f) = \mathtt{Tr}^U_{X,Y'}((g \otimes 1_U)f)$, where $f : X \otimes U \longrightarrow Y \otimes U$, $g : Y \longrightarrow Y'$,
3. **Dinatural** in U, $\mathtt{Tr}^U_{X,Y}((1_Y \otimes g)f) = \mathtt{Tr}^{U'}_{X,Y}(f(1_X \otimes g))$, where $f : X \otimes U \longrightarrow Y \otimes U'$, $g : U' \longrightarrow U$,
4. **Vanishing (I, II)**, $\mathtt{Tr}^I_{X,Y}(f) = f$ and $\mathtt{Tr}^{U \otimes V}_{X,Y}(g) = \mathtt{Tr}^U_{X,Y}(\mathtt{Tr}^V_{X \otimes U, Y \otimes U}(g))$, for $f : X \otimes I \longrightarrow Y \otimes I$ and $g : X \otimes U \otimes V \longrightarrow Y \otimes U \otimes V$.
5. **Superposing**,
$$g \otimes \mathtt{Tr}^U_{X,Y}(f) = \mathtt{Tr}^U_{W \otimes X, Z \otimes Y}(g \otimes f)$$
for $f : X \otimes U \longrightarrow Y \otimes U$ and $g : W \longrightarrow Z$.
6. **Yanking**, $\mathtt{Tr}^U_{U,U}(\sigma_{U,U}) = 1_U$.

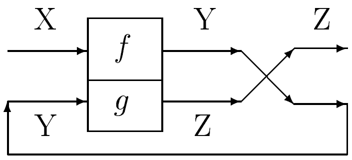
We think of $\mathtt{Tr}^U_{X,Y}(f)$ as "feedback along U", as in Figure 1. Similarly, the axioms of traced monoidal categories have a geometrical representation given in [AHS] (Appendix 1).

Examples 4.2

1. The category $\mathbf{Rel}_\times$ is traced. Let $R : X \times U \longrightarrow Y \times U$ be a morphism in $\mathbf{Rel}_\times$. Then $\mathtt{Tr}^U_{X,Y}(R) : X \longrightarrow Y$ is defined by: $\mathtt{Tr}^U_{X,Y}(R)(x, y) = \exists u.R(x, u, y, u)$.

2. The category $\mathbf{Vec}_{fd}$ is traced. Let $f : V \otimes U \longrightarrow W \otimes U$ be a linear map, where U, V, W are finite dimensional vector spaces with bases $\{u_i\}, \{v_j\}, \{w_k\}$. We define $\mathtt{Tr}^U_{V,W}(f) : V \longrightarrow W$ by:
$$\mathtt{Tr}^U_{V,W}(f)(v_i) = \sum_{j,k} a^{kj}_{ij} w_k \qquad \text{where } f(v_i \otimes u_j) = \sum_{k,m} a^{km}_{ij} w_k \otimes u_m.$$
This reduces to the usual trace of $f : U \longrightarrow U$ when V and W are one dimensional.

3. Note that both **Rel** and $\mathbf{Vec}_{fd}$ are compact closed categories. More generally [JSV], every compact closed category has a canonical trace:
$$Tr^U_{A,B}(f) = A \cong A \otimes I \stackrel{id \otimes \eta}{\longrightarrow} A \otimes U \otimes U^\perp \stackrel{f \otimes id}{\longrightarrow} B \otimes U \otimes U^\perp \stackrel{id \otimes ev'}{\longrightarrow} B \otimes I \cong B$$
where $ev' = ev{\circ}s$.



$$= \quad X \xrightarrow{f} Y \xrightarrow{g} Z$$

Figure 2: Generalized Yanking

4. The category ω-**CPO**$_\perp$ consists of objects of ω-**CPO** with a smallest element $\perp$, and maps of ω-**CPO** that do not necessarily preserve $\perp$. Here $\otimes = \times$, $I = \{\perp\}$. The (dinatural) family of least-fixed-point combinators $\mathsf{Y}_U : U^U \to U$ induces a trace, given as follows (using informal lambda calculus notation): for any $f : X \times U \to Y \times U$, $Tr^U_{X,Y}(f)(x) = f_1(x, \mathsf{Y}_U(\lambda u.f_2(x,u)))$, where $f_1 = \pi_1 \circ f : X \times U \to Y$, $f_2 = \pi_2 \circ f : X \times U \to U$ and $\mathsf{Y}_U(\lambda u.f_2(x,u))$ = the smallest element u' of U such that $f_2(x,u') = u'$.

In the above examples, $\otimes$ is based on (cartesian) product. Unfortunately, these examples do not really illustrate the notion of feedback as data flow: the movement of tokens through a network. This latter view, emphasized in work of Abramsky and later Haghverdi and Hines (cf. [AHS, Hagh, Hi]), is illustrated by examples based on coproduct-like traces, given below.

Exercise 4.3 (Generalized Yanking) Let $\mathcal{C}$ be a traced symmetric monoidal category, with arrows $f : X \to Y$ and $g : Y \to Z$. Then $g \circ f = Tr^Y_{X,Z}(s_{Y,Z} \circ (f \otimes g))$. Find an algebraic proof of this fact. Geometrically, the reader should stare at the diagram in Figure 2 , and do a "string-pulling" argument (cf. [JSV])

The next examples of traced monoidal categories arise in considering "coproduct-like" traces, and are related to dataflow interpretations of graphical networks. We illustrate this view with categories connected to **Rel**.

Examples 4.4

1. **Rel**$_+$, the category **Rel** with $\times = +$, disjoint union. Suppose $X+U \xrightarrow{R} Y+U$ is a relation. The coproduct injections induce four restricted relations : $R_{UU}, R_{UY}, R_{XY}, R_{XU}$ (for example, $R_{XY} \subseteq X \times Y$ is such that $R_{XY}(x,y) = R(in_1^{X,U}(x), in_1^{Y,U}(y))$. Let R^* be the reflexive, transitive closure of the relation R. A trace can be defined as follows:

$$\begin{aligned} Tr^U_{X,Y}(R) &= R_{XY} \cup \bigcup_{n \geq 0} R_{UY} \circ R^n_{UU} \circ R_{XU} \\ &= R_{XY} \cup R_{UY} \circ R^*_{UU} \circ R_{XU}. \end{aligned} \tag{5}$$

2. Consider the categories **Pfn** and **PInj** of sets and partial functions (resp. sets and partial injective functions), as monoidal subcategories of **Rel**$_+$. The tensor product is given by the disjoint union of sets, where we identify $A + B = \{1\} \times A \cup \{2\} \times B$ (note that this is not a coproduct in **PInj**, although it is a coproduct in **Pfn**). There

are the obvious injections $in_1^{A,B} : A \to A+B$ and $in_2^{A,B} : B \to A+B$ as well as "quasiprojections" $\rho_1 : A+B \longrightarrow A$ given by $\rho_1((1,a)) = a$ (where $\rho_1((2,b))$ is undefined) and similarly for $\rho_2 : A+B \longrightarrow B$.

Given a morphism $f : X+U \longrightarrow Y+U$, we may consider its four "components" $f_{XY} : X \to Y$, $f_{XU} : X \to U$, $f_{UX} : U \to X$, and $f_{UU} : U \to U$ obtained by pre- and post-composing with injections and quasiprojections: for example, $f_{XY} = X \xrightarrow{in_1} X+U \xrightarrow{f} Y+U \xrightarrow{\rho_1} Y$, (See Figure 3).

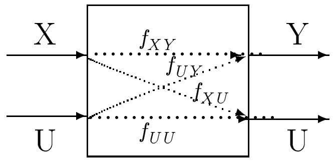


Figure 3: Components of $f : X+U \to Y+U$

Both **Pfn** and **Pinj** are traced, the trace being given by the following iterative formula

$$\mathtt{Tr}^U_{X,Y}(f) = f_{XY} + \sum_{n\in\omega} f_{UY} f^n_{UU} f_{XU}, \tag{6}$$

which we interpret as follows: a family $\{h_i\}_{i\in I} : X \longrightarrow Y$ is said to be *summable* if the h_i's have pairwise disjoint domains and codomains. In that case, we define their sum

$$(\sum_{i\in I} h_i)(x) = \begin{cases} h_j(x), & \text{if } x \in \mathtt{Dom}(h_j) \text{ for some } j \in I; \\ \text{undefined}, & \text{else.} \end{cases}$$

From a dataflow view, particles enter through X, travel around a loop on U some number n of times, then exit through Y. Numerous other examples of such "coproduct-like" traces are studied in [AHS].

The iterative trace formulas (5) and (6) are versions of Girard's *Execution Formula* from his GoI program. A general categorical framework for discussing such traces, and their connections to Girard's original work, is studied in Haghverdi's work [Hagh].

On a more general level, starting with a traced monoidal category $\mathcal{C}$, we now describe a compact closed category $\mathbf{Int}(\mathcal{C})$ described in [JSV] (also called $\mathcal{G}(\mathcal{C})$ in [Abr96]) which captures in abstract form many of the features of Girard's Geometry of Interaction program, as well as the general ideas behind game semantics. We follow the treatment in Abramsky [Abr96].

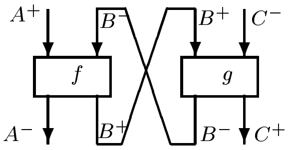
Definition 4.5 (The Int Construction) Given a traced monoidal category $\mathcal{C}$ we define a compact closed category, $\mathbf{Int}(\mathcal{C})$, as follows [JSV, Abr96]:

- Objects: Pairs of objects (A^+, A^-) where A^+ and A^- are objects of $\mathcal{C}$.
- Arrows: An arrow $f : (A^+, A^-) \longrightarrow (B^+, B^-)$ in $\mathbf{Int}(\mathcal{C})$ is an arrow $f : A^+ \otimes B^- \longrightarrow A^- \otimes B^+$ in $\mathcal{C}$.

- Identity: $1_{(A^+,A^-)} = s_{A^+,A^-}$.
- Composition: Arrows $f : (A^+, A^-) \longrightarrow (B^+, B^-)$ and $g : (B^+, B^-) \longrightarrow (C^+, C^-)$ have composite $g \circ f : (A^+, A^-) \longrightarrow (C^+, C^-)$ given by:

$$g \circ f = Tr^{B^- \otimes B^+}_{A^+ \otimes C^-, A^- \otimes C^+}(\beta(f \otimes g)\alpha)$$

where $\alpha = (1_{A^+} \otimes 1_{B^-} \otimes s_{C^-,B^+})(1_{A^+} \otimes s_{C^-,B^-} \otimes 1_{B^+})$ and $\beta = (1_{A^-} \otimes 1_{C^+} \otimes s_{B^+,B^-})(1_{A^-} \otimes s_{B^+,C^+} \otimes 1_{B^-})(1_{A^-} \otimes 1_{B^+} \otimes s_{B^-,C^+})$. Pictorially, $g \circ f$ is given by symmetric feedback:



- Tensor: $(A^+, A^-) \otimes (B^+, B^-) = (A^+ \otimes B^+, A^- \otimes B^-)$ and for $(A^+, A^-) \longrightarrow (B^+, B^-)$ and $g : (C^+, C^-) \longrightarrow (D^+, D^-)$, $f \otimes g = (1_{A^-} \otimes s_{B^+,C^-} \otimes 1_{D^+})(f \otimes g)(1_{A^+} \otimes s_{C^+,B^-} \otimes 1_{D^-})$
- Unit: (I, I).
- Duality: The dual of (A^+, A^-) is given by $(A^+, A^-)^\perp = (A^-, A^+)$ where the unit $\eta : (I, I) \longrightarrow (A^+, A^-) \otimes (A^+, A^-)^\perp =_{def} s_{A^-,A^+}$ and counit $\epsilon : (A^+, A^-)^\perp \otimes (A^+, A^-) \longrightarrow (I, I) =_{def} s_{A^-,A^+}$.
- Internal Homs: As usual, $(A^+, A^-) \multimap (B^+, B^-) = (A^+, A^-)^\perp \otimes (B^+, B^-) = (A^- \otimes B^+, A^+ \otimes B^-)$.

Translating the work of [JSV] in our setting we obtain that $\mathbf{Int}(\mathcal{C})$ is a kind of "free compact closure" of $\mathcal{C}$ at the bicategorical level (for which the reader is referred to [JSV]):

Proposition 4.6 *Let $\mathcal{C}$ be a traced symmetric monoidal category*

1. $\mathbf{Int}(\mathcal{C})$ *defined above is a compact closed category. Moreover, $F_{\mathcal{C}} : \mathcal{C} \longrightarrow \mathbf{Int}(\mathcal{C})$ defined by $F_{\mathcal{C}}(A) = (A, I)$ and $F_{\mathcal{C}}(f) = f$ is a full and faithful embedding.*
2. *The inclusion of 2-categories* $\mathbf{CompCl} \hookrightarrow \mathbf{TraMon}$ *of compact closed categories into traced monoidal ones has a left biadjoint with unit having component at $\mathcal{C}$ given by $F_{\mathcal{C}}$.*

Following Abramsky [Abr96], we interpret the objects of $\mathbf{Int}(\mathcal{C})$ in a game-theoretic manner: A^+ is the type of "moves by Player (the System)" and A^- is the type of "moves by Opponent (the Environment)". The composition of morphisms in $\mathbf{Int}(\mathcal{C})$ is connected to Girard's execution formula . In [Abr96] it is pointed out that $\mathcal{G}(\mathbf{Pinj})$ captures the essence of the original Girard model, while $\mathcal{G}(\omega\text{-CPO}_\perp)$ is the model of GoI in [AJ1].

Finally, we remark that in [AHS], a general analysis of such algebraic models of GoI is given. There it is shown how to use the above abstract GoI construction to obtain models of the $\{!, \multimap\}$ fragment of linear logic, presented in terms of *linear combinatary algebras*. These are certain combinatory algebras $(\mathcal{A}, \cdot)$ equipped with a map $\mathbf{!} : \mathcal{A} \to \mathcal{A}$ and constants $B, C, I, K, W, D, \delta, F$ satisfying the combinatory identities for a Hilbert-style axiomatization of $\{!, \multimap\}$ (see also [Tr]). The method is sketched as follows.

Let $\mathcal{C}$ be a traced smc, with an endofunctor $T : \mathcal{C} \to \mathcal{C}$ and an object (called a *reflexive object*) $U \in \mathcal{C}$ with retractions $U \otimes U \lhd U$, $I \lhd U$, and $TU \lhd U$. Then if T satisfies some reasonable axioms and setting $V = (U, U)$ and $I = (I, I)$, it is shown in [AHS] how the homset $\mathbf{Int}(\mathcal{C})(I, V) = \mathcal{C}(U, U)$ naturally inherits the structure of a linear combinatory algebra. For example, in the case of $\mathcal{C} = \mathbf{Pinj}$, $\mathbb{N}$ is such a reflexive object, with endofunctor $T(-) = \mathbb{N} \times (-)$. This example underlies the original Girard GoI constructions. The model in [AJ1] likewise arises from $\mathbf{Int}(\mathbf{CPO}_\perp)$. Moreover, Girard's original operator-theoretic models (in the category of Hilbert spaces), as well as Danos-Regnier's *small model* [DR2] are also captured in the above framework using some additional functorial structure (see [Hagh], Section 6).

5 Nonsymmetric monoidal categories

One of the most appealing features of linear logic is its flexibility; one can readily define variants of linear logic which either have a full exchange rule or a very limited exchange rule. These variants correspond to the varying degrees of symmetry that one gives to the tensor. In short, just as there are nonsymmetric monoidal categories, there is nonsymmetric linear logic. The most interesting examples of nonsymmetric monoidal categories occur in the representation theory of *Hopf algebras.* Hopf algebras arise in many areas of physics, computer science and combinatorics. In this section, we review the basics of nonsymmetric monoidal categories, how Hopf algebras provide examples, and how these examples correspond to various types of linear logic.

If we drop the requirement that the tensor be symmetric, then one should consider categories with two internal HOM's. Thus we should have adjunctions of the form:

$$HOM(A \otimes B, C) \cong HOM(B, A \multimap C)$$
$$HOM(A \otimes B, C) \cong HOM(A, C \multimapinv B)$$

This is the definition of *biautonomous* category, an obvious generalization of the symmetric case. Of course, if the tensor happens to be symmetric, this will induce an isomorphism between the two HOM's.

Analogously, to define a nonsymmetric analogue of categories with dualizing objects one needs two duals, $A^\perp$ and ${}^\perp A$. (The dualizing object for each will be the same.) These will be subject to the isomorphisms:

$${}^\perp(A^\perp) \cong ({}^\perp A)^\perp \cong A$$

More specifically, a biautonomous category has a canonical morphism:

$$A \longrightarrow {}^\perp(A^\perp) \cong ({}^\perp A)^\perp$$

and if this map is an isomorphism, then we have a *bi-$*$-autonomous* category. (In general, there will be no relationship between A and $A^{\perp\perp}$ in the nonsymmetric case.)

We now discuss a variant of these categories.

Definition 5.1 *If in a bi-$*$-autonomous category, the dualizing object, $\perp$, has the property that:*

$$^{\perp}A \cong A^{\perp}$$

or equivalently:

$$A \multimap \perp \cong \perp \multimapinv A$$

then $\perp$ is said to be cyclic. *A $*$-autonomous category with such a dualizing object is also said to be cyclic.*

In the posetal case, these are the Girard quantales, and were introduced by Yetter in [Yet] and studied by Rosenthal [Ros]. A notion of proof net for this theory is contained in [Yet].

Yetter's *cyclic linear logic* is obtained by replacing the usual exchange rule with:

$$\frac{\vdash A_1, A_2, \ldots, A_n}{\vdash A_{\sigma(1)}, A_{\sigma(2)}, \ldots, A_{\sigma(n)}}$$

where σ is a *cyclic* element of the symmetric group on n letters. It is straightforward to verify that a $*$-autonomous category with a cyclic dualizing object validates this rule.

Noncommutative linear logic, with the cyclic exchange rule would seem to be the optimal level of noncommutativity. The theory has an excellent semantics, sequent calculus and proof nets. Similarly well-behaved structures for fully noncommutative linear logic have proven to be much more problematic.

We note that the subject of noncommutative linear logic has not been explored as extensively as other aspects of linear logic. In addition to obtaining further noncommutative full completeness theorems, there are also a number of logical variants of cyclic linear logic that should be considered. Indeed, Ruet's recent variant, called simply *noncommutative logic*, ultimately suggests that the number of noncommutative variants may be almost endless. See [AR] for an analysis of its syntax, as well as a notion of proof net for this logic.

Ruet's logic is a mix of commutative and noncommutative elements. In it, there are two sets of connectives, one an ordinary commutative tensor and par and the other a cyclic noncommutative tensor and par. Interaction between the two systems is mediated by a structural rule called *entropy*.

It is hoped that for any possible version of noncommutative linear logic, there is a corresponding notion of Hopf algebra (see below). For Ruet's logic, there is the notion of *entropic Hopf algebra* developed in [BLR].

5.1 Representations of Hopf algebras

We now introduce Hopf algebras as a means of constructing examples of nonsymmetric monoidal closed categories. Hopf algebras are best considered as a nonsymmetric generalization of the category $\mathcal{MOD}(G)$ of G-modules (see Examples 2.4, Number 5). We recommend [K, Maj] as excellent introductions. We begin with some preliminaries. We assume throughout a fixed, but arbitrary field $\mathbf{k}$.

Definition 5.2 An *(associative) algebra* is a $\mathbf{k}$-vector space H equipped with maps $m\colon \mathsf{H}\otimes\mathsf{H}\to\mathsf{H}$ and $\eta\colon\mathbf{k}\to\mathsf{H}$ which are called the multiplication and unit, and these must satisfy the evident equations for associativity and unit. Dually one may define a *(coassociative) coalgebra* as a space with maps $\Delta\colon\mathsf{H}\to\mathsf{H}\otimes\mathsf{H}$ and $\varepsilon:\mathsf{H}\to k$ satisfying the dual axioms. Then a *Hopf algebra* is a k-vector space H equipped with an algebra structure, a compatible coalgebra structure and a map $S\colon\mathsf{H}\to\mathsf{H}$ called the antipode satisfying appropriate equations. The following chart summarizes the necessary structure. All maps shown are linear.

Structure		*Equations*
Algebra	$m\colon\mathsf{H}\otimes\mathsf{H}\to\mathsf{H}$ (multiplication) $\eta\colon k\to\mathsf{H}$ (unit)	Associativity and Unit: $m\circ(m\otimes id)=m\circ(id\otimes m)$ and $\eta(1)$ is 2-sided unit for m.
Coalgebra	$\Delta\colon\mathsf{H}\to\mathsf{H}\otimes\mathsf{H}$ (comultiplication) $\varepsilon:\mathsf{H}\to k$ (counit)	Coassociativity with counit for comultiplication (dual to algebra structure).
Bialgebra	Algebra + Coalgebra	Δ and ε are algebra homs. (Equivalently m,η are coalgebra homs.)
Antipode	$S\colon\mathsf{H}\to\mathsf{H}$	Inverse to $id_{\mathsf{H}}:\mathsf{H}\to\mathsf{H}$ under convolution

Here convolution refers to the operation on $Hom_{\mathbf{k}}(\mathsf{H},\mathsf{H})$ defined by $(f*g)(c)=m((f\otimes g)(\Delta c))$. The identity for the convolution operation is given by $\eta\varepsilon:\mathsf{H}\to\mathsf{H}$. We say a Hopf algebra is *(co)commutative* if the (co)multiplication is (co)commutative (i.e. the appropriate diagram or its dual commutes [K, Maj].) □

Example 5.3 A particular Hopf algebra which provides the semantics of cyclic linear logic is known as the *shuffle algebra.* It is an example of an *incidence algebra* and is of fundamental importance in several areas of mathematics. The terminology below is motivated by thinking of shuffling a deck of cards.

Let X be a set and X^* the free monoid generated by X. We denote words (= strings) in X^* by $w,w',\cdots$ and occasionally $z,z'\ldots$. Elements $x,y,\cdots\in X$ are identified with words of length 1, the empty word (= unit of the monoid) is denoted by ϵ, and the monoid multiplication is given by concatenation of strings. We denote the length of word w by $|w|$. Let $\mathbf{k}[X^*]$ be the free $\mathbf{k}$-vector space generated by X. We consider $\mathbf{k}[X^*]$ endowed with the following Hopf algebra structure (cf. [BS2]):

(i) $\mathcal{A}=\mathbf{k}[X^*]$ is an *algebra*, i.e. comes equipped with an associative $\mathbf{k}$-linear multiplication (with unit) $m:\mathcal{A}\otimes\mathcal{A}\to\mathcal{A}$:

$$w\otimes w' \quad\mapsto\quad w{\cdot}w'=\sum_{u\in Sh(w,w')} u \tag{7}$$

where $Sh(w, w')$ denotes the set of "shuffled" words of length $|w| + |w'|$ obtained from w and w'. Here, a *shuffle* of $w = a_1 \cdots a_m$ and $w' = a'_1 \cdots a'_n$ is a word of length $m + n$, say $w'' = c_1 \cdots c_{m+n}$ such that each of the a_i and a'_j occurs once in w'' ; moreover, within w'', a_i and a'_j occur in their original sequential order. For example, if $w = aba$ and $w' = bc$, we obtain the following set of shuffled words (where the letters from w' are underlined)

$$aba\underline{bc}, ab\underline{b}a\underline{c}, a\underline{b}ba\underline{c}, \underline{b}aba\underline{c}, ab\underline{bc}a, a\underline{b}b\underline{c}a, \underline{b}ab\underline{c}a, a\underline{bc}ba, \underline{b}a\underline{c}ba, \underline{bc}aba$$

Thus the summation $w \cdot w'$ is equal to

$$ababc + 2abbac + babac + 2abbca + babca + abcba + bacba + bcaba$$

Note that we always denote the shuffle multiplication with $\cdot$, as opposed to the monoid multiplication, for which we use concatenation.

The unit $\eta : \mathbf{k} \to \mathcal{A}$ arises by mapping $1 \mapsto \epsilon$.

(ii) $\mathcal{A} = \mathbf{k}[X^*]$ is a *coalgebra*, i.e. comes equipped with a coassociative comultiplication (with counit) $\Delta : \mathcal{A} \to \mathcal{A} \otimes \mathcal{A}$, defined as:

$$\Delta(w) = \sum_{w_1 w_2 = w} w_1 \otimes w_2 \tag{8}$$

Note that in the equation $w_1 w_2 = w$ we are using the original monoid multiplication of X^*. The above pair $w_1 w_2$ is called a *cut* of w.

The counit $\varepsilon : \mathcal{A} \to \mathbf{k}$ is defined by:

$$\varepsilon(w) = \begin{cases} 1 & \text{if } w = \epsilon \\ 0 & \text{else} \end{cases} \tag{9}$$

Finally, there is an *antipode* defined as

$$S(w) = (-1)^{|w|} \bar{w} \tag{10}$$

where $\bar{w}$ denotes the word w written backwards.

Proposition 5.4 $\mathcal{A} = \mathbf{k}[X^*]$ *with the above structure forms a Hopf algebra with involutive antipode.*

5.2 H-Modules

In analogy with the notion of G-space, we may speak of the action of a Hopf algebra H on a vector space V. This is a linear map $\rho : \mathsf{H} \otimes V \to V$ satisfying the analog of the action equations above:

Definition 5.5 Given a Hopf algebra H, a *module* over H is a vector space V, equipped with a linear map called an H-*action* $\rho \colon \mathsf{H} \otimes V \to V$ such that the following diagrams commute:

$$\begin{tikzcd}
\mathsf{H}\otimes\mathsf{H}\otimes V \arrow[r, "id\otimes\rho"] \arrow[d, "m\otimes id"] & \mathsf{H}\otimes V \arrow[d, "\rho"'] \\
\mathsf{H}\otimes V \arrow[r, "\rho"] & V
\end{tikzcd}
\qquad
\begin{tikzcd}
V & \mathsf{H}\otimes V \arrow[l, "\rho"'] \\
& k\otimes V \arrow[u, "\eta\otimes id"'] \arrow[ul, "\cong"]
\end{tikzcd}$$

We will generally denote an H-action by concatenation, e.g. $\rho(h\otimes v) = hv$. Then the above diagrams translate, respectively, to: $(hh')v = h(h'v)$ and $\eta(1)v = v$, for all $h, h' \in \mathsf{H}, v \in V$. We shall frequently denote $\eta(1)$ by 1_{H}.

If (V,ρ) and (W,τ) are modules, then *a map of modules,* sometimes called an H-*map,* is a k-linear map $f\colon V\to W$ such that the following commutes:

$$\begin{tikzcd}
\mathsf{H}\otimes V \arrow[r, "id\otimes f"] \arrow[d, "\rho"'] & \mathsf{H}\otimes W \arrow[d, "\tau"] \\
V \arrow[r, "f"] & W
\end{tikzcd}$$

i.e. in the above notation, $f(hv) = hf(v)$ for all $h\in\mathsf{H}, v\in V$. We thus obtain a category $\mathcal{MOD}(\mathsf{H})$.

The above definition is a straightforward generalization from group representations; indeed, the latter arises as the special case $\mathsf{H} = k[G]$. If U and V are modules, then $U\otimes V$ has a natural module structure given by:

$$\mathsf{H}\otimes U\otimes V \xrightarrow{\Delta\otimes id} \mathsf{H}\otimes\mathsf{H}\otimes U\otimes V \xrightarrow{c_{23}} \mathsf{H}\otimes U\otimes\mathsf{H}\otimes V \xrightarrow{\rho\otimes\rho} U\otimes V$$

Theorem 5.6 (See [Maj, K] for details.) *$\mathcal{MOD}(\mathsf{H})$ is a monoidal category. If the Hopf algebra is cocommutative, then the tensor product is symmetric. The unit for the tensor is given by the ground field with the module structure induced by the counit of H.*

Definition 5.7 Given an arbitrary Hopf algebra H with bijective antipode, and two H-modules, A and B, we will define two new H-modules, $A \multimap B$ and $B \circ\!\!- A$, as follows. In both cases, the underlying space will be $A \multimap_{\mathbf{k}} B$, the space of $\mathbf{k}$-linear maps.

The action on $B \circ\!\!- A$ is defined by:

$$(hf)(a) = \sum h_1 f(S(h_2)a) \tag{11}$$

and the action on $A\multimap B$ is defined by:

$$(hf)(a) = \sum h_2 f(S^{-1}(h_1)a) \tag{12}$$

where $\Delta(h) = \sum h_1\otimes h_2$.

A proof of the following can be found in [Maj]. See that reference also for the history of these constructions, many of which are due to that author.

Theorem 5.8 *Let* H *be a Hopf algebra with bijective antipode. Then with the actions defined above,* $\mathcal{MOD}(\mathsf{H})$ *is a biautonomous category. The adjoint relation:*

$$HOM(A \otimes B, C) \cong HOM(B, A \multimap C)$$

holds whether or not the antipode is bijective. In the case of a cocommutative Hopf algebra, the two internal HOM*'s are equal.*

So the representation theory of Hopf algebras provides us access to a wide variety of models of noncommutative (intuitionistic) linear logic. There are several ways to extend this to obtain classical models. One could restrict to finite-dimensional representations, or again use the topological category **RTVec** for representations. The expository paper [Bl96] considers these ideas. In particular, we note the following as an example of the usefulness of Hopf algebras.

Theorem 5.9 *If* H *is a Hopf algebra with involutive antipode, then its category of finite-dimensional representations or representations in* **RTvec** *is a model of cyclic linear logic.*

As a corollary, we obtain from Proposition 5.4 that the *shuffle Hopf algebra* $\mathcal{A} = \mathbf{k}[X^*]$ models cyclic linear logic. Such Hopf algebras were used to obtain a *full completeness theorem* for cyclic multiplicative linear logic in [BS2]. This will be discussed in the next section.

6 Full Completeness and Representation Theorems

The most basic representation theorem of all is the Yoneda embedding:

Theorem 6.1 (Yoneda) *If* $\mathcal{A}$ *is locally small, the Yoneda functor* $\mathcal{Y} : \mathcal{A} \to \mathbf{Set}^{\mathcal{A}^{op}}$*, where* $\mathcal{Y}(A) = Hom_{\mathcal{A}}(-, A)$*, is a fully faithful embedding.*

Indeed, Yoneda preserves limits as well as cartesian closedness. This theorem, and its many variants, is critical to the development of category theory and categorical model theory.

However we seek mathematical models which fully and faithfully represent proofs. From the viewpoint of a logician, these are completeness theorems, but now at the level of proofs rather than provability. The results are known as *full completeness theorems.* The terminology arose in the work of Abramsky and Jagadeesan on full completeness for MLL + Mix in $*$-autonomous categories of games [AJ].

Definition 6.2 Let $\mathcal{F}$ be a free category. We say that a categorical model $\mathcal{M}$ is *fully complete for* $\mathcal{F}$ or that we have *full completeness of* $\mathcal{F}$ *with respect to* $\mathcal{M}$ if, with respect to some interpretation of the generators, the unique free functor $[\![-]\!] : \mathcal{F} \to \mathcal{M}$ is full. It is even better to demand that $[\![-]\!]$ is a fully faithful representation.

For example, suppose $\mathcal{F} = F(\mathcal{G}_0)$ is a free structured category (e.g. free ccc, $*$-autonomous, etc.) generated by the appropriate deductive system on a discrete graph $\mathcal{G}_0$. To say a categorical model $\mathcal{M}$ is fully complete for $\mathcal{F}$ means: any arrow $[\![A]\!] \to [\![B]\!] \in \mathcal{M}$ between definable objects is itself definable, i.e. it must be of the form $[\![f]\!]$ for some (equivalence class of a) proof $f : A \to B$ in $\mathcal{F}$. If the representation is fully faithful, then f is unique. Thus, by Curry-Howard-Lambek, any morphism in the model between definable objects is itself the image of a proof (or program); and this proof is unique if the representation is fully faithful.

Such results are mainly of interest when the models $\mathcal{M}$ are "genuine" mathematical models not a priori connected to the syntax. For example, an explicit use of the Yoneda embedding $\mathcal{Y} : \mathcal{F} \to \mathbf{Set}^{\mathcal{F}^{op}}$ is not what we want, since the target model $\mathbf{Set}^{\mathcal{F}^{op}}$ depends too much on $\mathcal{F}$.

Probably some of the earliest full completeness results were for free ccc's (i.e. for simply typed lambda-calculi). Plotkin in the 1970's and Statman in the 1980's studied lambda definability in terms of invariance under logical relations on set-theoretic Henkin models.

In the case of Linear Logic, the fundamental paper of Abramsky and Jagadeesan [AJ] proved full completeness for MLL + Mix, using categories of games with certain history-free winning strategies as morphisms. It is shown there that "uniform" history-free winning strategies are the denotations of unique proof nets. An alternate notion of game, developed by Hyland and Ong, permits eliminating the Mix rule in such game-theoretic full completeness theorems for the multiplicatives. These results paved the way for the most spectacular application of these game-theoretic methods: the solution of the full abstraction problem for PCF, by Abramsky, Jagadeesan, and Malacaria and by Hyland and Ong. See for example [HO2].

There have been a host of full completeness theorems for MLL + Mix, MLL, Yetter's CyLL, and recently for MALL. Very roughly speaking, we may distinguish two styles of fully-complete models in the literature:

- *Direct Models*: These are subcategories of some ambient $*$-autonomous category. The key idea is to impose an invariance or uniformity condition to restrict the class of arrows between definable types to those which are exactly the denotations of proofs. This is typical of the original game theoretic fully-complete models mentioned above (where proofs correspond to certain restricted kinds of winning strategies) as well as to Hamano's direct full completeness theorem for MLL + Mix, which uses a subcategory of **RTVec** restricted to certain $\mathbb{Z}$-invariant maps.

- *Functorial Models*: This approach, and similar ones using relational transformers and Reynolds' parametricity, uses techniques of functorial polymorphism in [BFSS]. The basic idea is to model formulas (i.e. types) as multivariant functors over some base monoidal category, and proofs as multivariant (dinatural) transformations. In the second case, one uses similar relational methods. In either approach, one often imposes additional uniformity requirements on (di)natural families to enable them to exactly correspond to proofs (e.g. in our previous work [BS1, BS2], we supposed dinatural families (discussed below) are invariant under continuous group or Hopf-algebra actions.).

The functorial models and their variants provide a powerful and increasingly popular framework for full completeness proofs. Uniformity is now imposed over (di)natural families, with a much wider range of examples than the direct models approach. But this flexibility comes at a price: the functorial approach only applies to cut-free systems of LL, since dinatural transformations (as well as logical relations) do not compose in general. Hence functorial models, unlike direct models, are not a priori categories.

Since the work of Ralph Loader in the early 1990's, later generalized in work of Hyland and Tan, it has become increasingly important to lift known full completeness theorems to larger base categories. The techniques for doing this involve using a Chu space or Double Gluing construction on top of the base. The point of the Chu or Double-gluing construction is to eliminate various unwanted maps, e.g. moving from a compact base category or one satisfying Mix to a more general $*$-autonomous setting, and then to rebuild the whole functorial framework at this level. Haghverdi [Hagh] introduces a new class of full completeness theorems for MLL by a 2-step process: first applying a GoI-model construction to certain traced monoidal categories, then applying modified Loader-Hyland-Tan techniques.

In the case of MALL, there are currently two full completeness theorems in the dinatural framework. Part of the difficulty here is that the associated notion of proof-nets for MALL is highly non-trivial. The first model, by Abramsky and Melliès [AM] uses dinaturals over a base category of so-called *concurrent games*, which are themselves related to a kind of double-gluing construction. The second, by the authors and Hamano [BHS] uses the dinatural framework on a double gluing category over Ehrhard's category of hypercoherence spaces.

As an example of how full completeness theorems work, we now give a brief picture of the dinatural results in the authors' papers [BS1, BS2].

Definition 6.3 Let $\mathcal{C}$ be a category, and $F, G : (\mathcal{C}^{op})^n \times \mathcal{C}^n \to \mathcal{C}$ functors. A *dinatural transformation* $\theta : F \to G$ is a family of $\mathcal{C}$-morphisms $\theta = \{\theta_A : FAA \to GAA \,|\, A \in \mathcal{C}^n\}$ satisfying (for any n-tuple $f : A \to B \in \mathcal{C}^n$):

$$\begin{tikzcd}
 & FAA \arrow[r, "\theta_A"] & GAA \arrow[dr, "GAf"] & \\
FBA \arrow[ur, "FfA"] \arrow[dr, "FBf"'] & & & GAB \\
 & FBB \arrow[r, "\theta_B"] & GBB \arrow[ur, "GfB"'] &
\end{tikzcd}$$

For a history of this notion, see [BFSS].

Let $\mathcal{C}$ be a $*$-autonomous category. Given an MLL formula $\varphi(\alpha_1, \ldots, \alpha_n)$ built from $\otimes, \multimap, (\ )^\perp$, with type variables $\alpha_1, \ldots, \alpha_n$, we inductively define its *functorial interpretation* $[\![\varphi(\alpha_1, \ldots, \alpha_n)]\!] : (\mathcal{C}^{op})^n \times \mathcal{C}^n \to \mathcal{C}$ as follows (boldface letters are vectors of objects):

- $[\![\varphi]\!](\mathbf{AB}) = \begin{cases} B_i & \text{if } \varphi(\alpha_1, \ldots, \alpha_n) \equiv \alpha_i \\ A_i^\perp & \text{if } \varphi(\alpha_1, \ldots, \alpha_n) \equiv \alpha_i^\perp \end{cases}$
- $[\![\varphi_1 \otimes \varphi_2]\!](\mathbf{AB}) = [\![\varphi_1]\!](\mathbf{AB}) \otimes [\![\varphi_2]\!](\mathbf{AB})$.

It is readily verified that $\llbracket \varphi^{\perp} \rrbracket = \llbracket \varphi \rrbracket^{\perp}$ and $\llbracket \varphi_1 \multimap \varphi_2 \rrbracket(\mathbf{AB}) = \llbracket \varphi_1 \rrbracket(\mathbf{BA}) \multimap \llbracket \varphi_2 \rrbracket(\mathbf{AB})$, where $A \multimap B$ is defined as $(A \otimes B^{\perp})^{\perp}$.

From now on, let $\mathcal{C} = \mathrm{RTVec}$. The set $Dinat(F, G)$ of dinatural transformations from F to G is a vector space, under pointwise operations. We call it the *proof space* associated to the sequent $F \vdash G$ (where we identify formulas with definable functors.) If $\vdash \Gamma$ is a one-sided sequent, then $Dinat(\Gamma)$ denotes the set of dinaturals from $\mathbf{k}$ to $\llbracket \invamp \Gamma \rrbracket$.

The following is proved in [BS1, BS2]. A binary sequent is one where each atom appears exactly twice, with opposite variances. A *diadditive dinatural transformation* is one which is a linear combination of substitution instances of binary dinaturals.

Theorem 6.4 (Full Completeness for MLL + Mix) *Let F and G be formulas in MLL + Mix, interpreted as definable multivariant functors on* **RTVec**. *Then the proof space $Dinat(F, G)$ of diadditive dinatural transformations has as basis the denotations of cut-free proofs in the theory MLL + Mix.*

Example 6.5 The proof space of the sequent

$$\alpha_1, \alpha_1 \multimap \alpha_2, \alpha_2 \multimap \alpha_3, \ldots, \alpha_{n-1} \multimap \alpha_n \vdash \alpha_n$$

has dimension 1, generated by the evaluation dinatural. Thus any proof of this sequent must be a scalar multiple of the evaluation dinatural.

The proofs of the above results actually yield a fully faithful representation theorem for a free $*$-autonomous category with Mix, whose homsets are canonically enriched over vector spaces ([BS1]).

In the same paper we proved a similar Full Completeness Theorem and fully faithful representation theorem for Yetter's Cyclic Linear Logic. In this case one employs the category $\mathcal{RTMOD}(\mathsf{H})$ for a Hopf algebra H. The particular Hopf algebra used is the shuffle Hopf algebra. Once again we consider formulas as multivariant functors on **RTVec**, but restrict the dinaturals to so-called H-*uniform dinaturals* $\theta_{|V_1|,\cdots,|V_n|}$, i.e. those which are equivariant with respect to the H-action induced from the atoms, for H-modules $V_i \in \mathcal{RTMOD}(\mathsf{H})$. This is completely analogous to the techniques used in logical relations.

Theorem 6.6 (Full Completeness for CyLL + Mix) *Let F and G be formulas in MLL + Mix, interpreted as definable multivariant functors on* **RTVec**. *Let H be the shuffle Hopf algebra. Then the proof space of H-uniform diadditive dinatural transformations has as basis the denotations of cut-free proofs in the theory CyLL + Mix.*

From the large literature on MLL full completeness theorems, we end by discussing an interesting line of research stemming from seminal work of Ralph Loader [Loa], who early on proved full completeness theorems using a linear version of logical predicates. His work led M. Hyland and A. Tan to introduce the method of *double gluing* [HS, Tan] as a new categorical technique for generating fully complete functorial models.

Definition 6.7 Let $\mathcal{C} = (\mathcal{C}, \otimes, I, (-)^{\perp})$ be a compact closed category. We define a new category, $\mathbf{G}\mathcal{C}$, the *double gluing category* of $\mathcal{C}$, whose objects are triples $\mathcal{A} = (A, \mathcal{A}_p, \mathcal{A}_{cp})$ where A is an object of $\mathcal{C}$, where $\mathcal{A}_p \subseteq \mathcal{C}(I, A)$ is called a set of *points* of A and where $\mathcal{A}_{cp} \subseteq \mathcal{C}(A, I) \cong \mathcal{C}(I, A^{\perp})$ is called a set of *copoints* of A.

A morphism $f : \mathcal{A} \longrightarrow \mathcal{B}$ in $\mathbf{G}\mathcal{C}$ is a morphism $f : A \longrightarrow B$ in $\mathcal{C}$ such that $f(\mathcal{A}_p) \subseteq \mathcal{B}_p$ and $f^{\perp}(\mathcal{B}_{cp}) \subseteq \mathcal{A}_{cp}$. Composition and identities are induced from the underlying composition and identities in $\mathcal{C}$.

Proposition 6.8 *For any compact closed category $\mathcal{C}$, $\mathbf{G}\mathcal{C}$ is a $*$-autonomous category, in which*

$$\begin{array}{rcl} \mathcal{A}^{\perp} & = & (A^{\perp}, \mathcal{A}_{cp}, \mathcal{A}_p) \\ \mathcal{A} \otimes \mathcal{B} & = & (A \otimes B, (\mathcal{A} \otimes \mathcal{B})_p, (\mathcal{A} \otimes \mathcal{B})_{cp}) \\ I_{\mathbf{G}\mathcal{C}} & = & (I, \{id_I\}, \mathcal{C}(I, I)) \end{array}$$

where $(\mathcal{A} \otimes \mathcal{B})_p = \{\alpha \otimes \beta | \alpha \in \mathcal{A}_p, \beta \in \mathcal{B}_p\}$ and $(\mathcal{A} \otimes \mathcal{B})_{cp} = \mathbf{G}\mathcal{C}(\mathcal{A}, \mathcal{B}^{\perp})$. The forgetful functor $U : \mathbf{G}\mathcal{C} \to \mathcal{C}$ preserves the $*$-autonomous structure.

We remark that in a logical setting one can think of an object $\mathcal{A} \in \mathbf{G}\mathcal{C}$ as a formula A in $\mathcal{C}$ together with a collection of proofs of A (the set $\mathcal{A}_p$) and a collection of refutations of A (the set $\mathcal{A}_{cp}$). Also, we remark that the double gluing construction works more generally for $*$-autonomous categories $\mathcal{C}$ (see [BHS, HS]). An important special case is:

Example 6.9 GRel denotes *the double gluing category over the category* $\mathbf{Rel}_{\times}$. Its objects are triples $\mathcal{A} = (A, \mathcal{A}_p, \mathcal{A}_{cp})$, where A is a set, $\mathcal{A}_p \subseteq \mathbf{Rel}(I, A) = \mathcal{P}(A)$ and $\mathcal{A}_{cp} \subseteq \mathbf{Rel}(A, I) = \mathcal{P}(A)$. A morphism $f : \mathcal{A} \to \mathcal{B}$ of **GRel** is a relation $R : A \to B$ of **Rel** such that:

(image condition:) $\forall \alpha \in \mathcal{A}_p\ [\alpha]R := \{b \in \beta \mid \exists\, a \in \alpha (a, b) \in R\} \in \mathcal{B}_p$
(co-image condition:) $\forall \beta \in \mathcal{B}_{cp}\ R[\beta] := \{a \in \alpha \mid \exists\, b \in \beta (a, b) \in R\} \in \mathcal{A}_{cp}$

There are many interesting full subcategories of **GRel**, e.g. Loader's *Linear Logical Predicates* and *Totality Spaces* [Loa], as well as **Coh**.

The Hyland-Tan approach to Loader's method is based on the following ideas. We start with a *compact closed* category $\mathcal{C}$ (e.g. $\mathcal{C} = \mathbf{Rel}$). We look at multivariant MLL-definable functors on the double gluing category $\mathbf{G}\mathcal{C}$ and dinatural transformations between them. Full completeness states that every such dinatural corresponds to a Danos-Regnier MLL proof net ρ_{θ}. The method is as follows:

1. Using the forgetful functor $U\colon \mathbf{G}\mathcal{C} \to \mathcal{C}$, the dinatural family θ on $\mathbf{G}\mathcal{C}$ is completely determined by arrows in $\mathcal{C}$. It thus suffices to prove a version of full completeness for compact categories $\mathcal{C}$. For **Rel** such a result holds, and it implies that every nontrivial such dinatural θ arises as a union of fixed-point-free involutions. Instantiating θ at appropriate subcategories of **GRel** determines axiom links of a proof structure ρ_{θ}.

2. One shows ρ_{θ} is a proof net, by showing it is acyclic and connected, using further instantiations of θ in **GRel**.

Haghverdi [Hagh] applied these techniques to compact closed categories arising from GoI, e.g. of the form $\mathbf{Int}(\mathcal{C})$, for certain traced monoidal categories $\mathcal{C}$.

Finally, we should remark that **GRel** has products and coproducts, so is a model of MALL. But neither **GRel** nor *Dinat*(**GRel**) is fully complete for MALL. Instead, it turns out that one must move to *Dinat*(**GHCoh**), where **HCoh** is Ehrhard's category of *hypercoherences* [Ehr1] in order to get a full completeness theorem for MALL (see [BHS]).

Acknowledgements

Both authors would like to thank the entire Èquipe de Logique de la Programmation (Luminy) and the TMR Network, along with the Directors Jean-Yves Girard and Laurent Regnier, for their kind hospitality and support. We also acknowledge support from operating grants from NSERC, Canada. Finally we thank Robert Seely and Mark Weber for helpful comments.

References

[Abr96] S. Abramsky , Retracing Some Paths in Process Algebra. In *CONCUR 96*, Springer Lecture Notes in Computer Science 1119, pp. 1-17 (1996).

[AJ1] S. Abramsky, R. Jagadeesan. New foundations for the geometry of interaction. *Information and Computation*, 111, pp. 53–119, (1994).

[AJ] S. Abramsky, R. Jagadeesan, Games and Full Completeness for Multiplicative Linear Logic, J. Symbolic Logic 59, pp. 543-574 (1994).

[AHS] S. Abramsky, E. Haghverdi, and P. Scott. Geometry of interaction and linear combinatory algebras. Mathematical Structures in Computer Science 12, pp. 625-665, (2002).

[AM] S. Abramsky, P.-A. Melliès. Concurrent games and full completeness. *Proceedings, Logic in Computer Science 1999*. IEEE Press, (1999).

[Abr97] S. Abramsky. Semantics of interaction: an introduction to game semantics. *in Semantics and logics of computation*, edited by A. Pitts and P. Dybjer, Cambridge University Press, (1997).

[AR] V.M. Abrusci, P. Ruet. Non-commutative logic I : the multiplicative fragment. Annals of Pure and Applied Logic 101 pp.29-64, (2000).

[AC] R. Amadio, P.L. Curien. *Domains and Lambda Calculi*. Cambridge University Press, (1998).

[BFSS] E. Bainbridge, P. Freyd, A. Scedrov, P. Scott, Functorial polymorphism, Theoretical Computer Science 70, pp. 1403-1456, (1990).

[Barr80] M. Barr. $*$-*autonomous categories*. Springer Lecture Notes in Mathematics 752, (1980).

[Bier95] G. Bierman. What is a categorical model of intuitionistic linear logic? *In* Proceedings of the Second International Conference on Typed Lambda Calculus and Applications. Lecture Notes in Computer Science 902, (1995).

[Bl93] R. Blute. Linear logic, coherence and dinaturality. Theoretical Computer Science, 115:3-41, 1993.

[Bl96] R. Blute, Hopf algebras and linear logic, Mathematical Structures in Computer Science 6, pp. 189-212, (1996).

[BCST] R. Blute, J. R. B. Cockett, R. A. G. Seely and T. Trimble. Natural deduction and coherence for weakly distributive categories. Journal of Pure and Applied Algebra 13, pp. 229–296, (1996)

[BCS1] R. Blute, J. R. B. Cockett, R. A. G. Seely. ! and ?: Storage as tensorial strength. Mathematical structures in Computer Science 6, pp. 313-351, (1996).

[BCS2] R. Blute, J. R. B. Cockett, R. A. G. Seely. The logic of linear functors. Mathematical structures in Computer Science 12, pp. 513-539, (2002).

[BLR] R. Blute, F. Lamarche, P. Ruet. Entropic Hopf algebras and models of noncommutative logic. Theory and Applications of Categories 10, pp. 424-460, (2002).

[BHS] R. Blute, M. Hamano, P. Scott. Softness of hypercoherences and MALL full completeness. In preparation, (2003).

[BS1] R. Blute, P. Scott. Linear Lauchli semantics, Annals of Pure and Applied Logic 77, pp. 101-142 (1996).

[BS2] R. Blute, P. Scott. The Shuffle Hopf algebra and noncommutative full completeness. Journal of Symbolic Logic 63, pp. 1413-1435, (1998).

[Borc] F. Borceux. *Handbook of Categorical Algebra* Cambridge University Press, (1993).

[BE] Antonio Bucciarelli and Thomas Ehrhard. On phase semantics and denotational semantics in multiplicative-additive linear logic. Annals of Pure and Applied Logic 102, pp. 247-282, 2000.

[CS1] J. R. B. Cockett, R. A. G. Seely. Weakly distributive categories. Journal of Pure and Applied Algebra 114, pp. 133-173, (1997).

[CS2] J. R. B. Cockett, R. A. G. Seely. Linearly distributive functors. Journal of Pure and Applied Algebra 143, pp. 155-203, (1999).

[DR] V. Danos, L. Regnier, The structure of multiplicatives, Arch. Math. Logic 28, pp.181-203, (1989).

[DR2] V. Danos, L. Regnier. Proof-nets and the Hilbert space, In *Advances in Linear Logic*, London Mathematical Society Lecture Notes Volume 222, (1995).

[DHPP] H. Devarajan, D. Hughes, G. Plotkin, and V. Pratt. Full completeness of the multiplicative linear logic of Chu spaces. *in* Proceedings 14^{th} Annual IEEE Symposium on Logic in Computer Science, LICS'99, Trento, Italy, July 1999.

[Ehr1] T. Ehrhard. Hypercoherences: a strongly stable model of linear logic. Mathematical Structures in Computer Science 3, pp. 365-385, (1993).

[Ehr2] T. Ehrhard. On Köthe sequence spaces and linear logic. Mathematical Structures in Computer Science 12, pp. 579-623, (2002).

[Ehr3] T. Ehrhard. Finiteness spaces, preprint, (2001).

[ER] T. Ehrhard and L. Regnier. The differential lambda-calculus. *To appear in* Theoretical Computer Science. 2003.

[FH] W. Fulton, J. Harris. Representation Theory: A First Course. Springer Verlag, (1991).

[G1] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[G2] J.Y. Girard. Geometry of interaction I: interpretation of system F. *Proceedings of the ASL Meeting, Padova, 1988.*

[G3] J.Y. Girard. Linear Logic, its syntax and semantics. In *Advances in Linear Logic*, London Mathematical Society Lecture Notes Volume 222, (1995).

[G4] J.Y. Girard. Geometry of Interaction III: accommodating the additives. *In* Advances in Linear Logic, London Mathematical Society Lecture Notes Volume 222, (1995).

[G5] J.Y. Girard. Proof-nets: the parallel syntax for proof-theory, Logic and Algebra, eds Ursini and Agliano, Marcel Dekker, New York 1996.

[G6] J.Y. Girard. Locus Solum. Mathematical Structures in Computer Science 11, pp. 301-506, (2001).

[GLT] J.Y. Girard, Y. Lafont, P. Taylor. *Proofs and Types.* Cambridge University Press, (1989)

[Hagh] E. Haghverdi. Unique decomposition categories, geometry of interaction and combinatory logic. Math. Structures Comput. Sci. 10, pp. 205–230, (2000).

[Ham1] M. Hamano. **Z**-modules and Full Completeness of Multiplicative Linear Logic, Annals of Pure Appl. Logic 107 , pp. 165-191 (2001).

[Ham2] M. Hamano. Pontrjagin Duality and Full Completeness for Multiplicative Linear Logic (Without Mix), Math. Struct. in Comp. Science 10, pp. 231-259 (2000).

[Hi] P. Hines. The Algebra of Self-Similarity and its Applications. Thesis. University of Wales, (1997).

[HO1] J. M. E. Hyland and C.-H. L. Ong. Fair Games and Full Completeness for Multiplicative Linear Logic without the Mix-Rule. Preprint, 1993.

[HO2] J. M. E. Hyland and C.-H. L. Ong. On Full Abstraction for PCF. Information and Computation, Volume 163, pp. 285-408, December 2000

[HS] J. M. E. Hyland and A. Schalk. Glueing and orthogonality for models of linear logic. Theoretical Computer Science 294, pp. 183-231, (2003).

[JSV] A. Joyal, R. Street and D. Verity. Traced monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society,* 119:425–446, 1996.

[K] C. Kassel. *Quantum Groups.* Springer-Verlag, (1995)

[KL] G. M. Kelly and M. Laplaza. Coherence for compact closed categories. Journal of Pure and Applied Algebra 19, pp. 193–213, (1980).

[L1] Y. Lafont. Interaction nets. *In* Principles of Programming Languages (POPL 1990), p. 95-108, (1990).

[LS] J. Lambek, P. Scott. *Introduction to Higher-Order Categorical Logic.* Cambridge University Press, (1988).

[Loa] R. Loader, Linear Logic, Totality and Full Completeness, *Symposium of Logic in Computer Science* (LICS), pp. 292-298 (1994)

[Mac] S. Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate texts in Mathematics.* Springer-Verlag, New York, 1971.

[Maj] S. Majid. *Foundations of Quantum Group Theory.* Cambridge University Press, (2000).

[Mog] E. Moggi. Notions of computation and monads. Information And Computation 93, 1991.

[PS] V. de Paiva, A. Schalk. Poset-valued sets, or, How to build models for linear logic, *to appear in* Theoretical Computer Science.

[RW] R. Rosebrugh, R. Wood. Constructive complete distributivity IV, Applied categorical structures 2, pp. 119-144, (1994).

[Ros] K. Rosenthal. *Quantales and Their Applications.* Pitman Research Notes in Mathematics, (1990).

[Sco] P. Scott. Some aspects of categories in computer science, in *Handbook of Algebra* , Volume 2, *edited by* M. Hazewinkel, North-Holland, pp. 1-77, (2000).

[See] R.A.G. Seely. Linear logic, *-autonomous categories and cofree coalgebras. *Contemporary Mathematics*, Volume 92. American Mathematical Society, (1989).

[Sel] P. Selinger. Control categories and duality: on the categorical semantics of the lambda-mu calculus. Math. Structures Comput. Sci. 11 , pp. 207–260 (2001).

[Tan] A. Tan. Full completeness for models of linear logic. Thesis, Cambridge University, (1997).

[Tr] A. Troelstra. Lectures on Linear Logic, Cambridge University Press, (1992).

[Yet] D. Yetter, Quantales and (noncommutative) linear logic, Journal of Symbolic Logic 55, p. 41-64, (1990)