Chu spaces as a semantic bridge between linear logic and mathematics

Vaughan Pratt [*]
Dept. of Computer Science
Stanford University
Stanford, CA 94305-2140
pratt@cs.stanford.edu

July 27, 2004

Abstract

The motivating role of linear logic is as a "logic behind logic." We propose a sibling role for it as a logic of transformational mathematics via the self-dual category of Chu spaces, a generalization of topological spaces. These create a bridge between linear logic and mathematics by soundly and fully completely interpreting linear logic while fully and concretely embedding a comprehensive range of concrete categories of mathematics. Our main goal is to treat each end of this bridge in expository detail. In addition we introduce the dialectic lambda-calculus, and show that dinaturality semantics is not fully complete for the Chu interpretation of linear logic.

Keywords: Chu spaces, linear logic, universal mathematics.

1 Introduction

Linear logic was introduced by J.-Y. Girard as a "logic behind logic." It separates logical reasoning into a core linear part in which formulas are merely moved around, and an auxiliary nonlinear part in which formulas may be deleted and copied. The core, multiplicative linear logic (MLL), is a substructural logic whose basic connectives are linear negation $A^\perp$, and linear conjunction or "*tensor*" $A \otimes B$ with unit 1. MLL's axiomatization resembles relevance logic in lacking *weakening*, from $\vdash B$ derive

[*]This work was supported by ONR under grant number N00014-92-J-1974

$A \vdash B$, but differs from it in also lacking *contraction*, from $A, A \vdash B$ derive $A \vdash B$. MLL obeys $A^{\perp\perp} \cong A$, associativity and commutativity of tensor, and *linear distributivity*, the transformability of positive occurrences of $(A \multimap B) \otimes C$ into $A \multimap (B \otimes C)$, where *linear implication* $A \multimap B$ abbreviates $(A \otimes B^{\perp})^{\perp}$. The auxiliary part adds the operations of ordinary product "*with*" $A \& B$ (synonymous with $A \times B$) with unit $\top$ and obeying $(A \multimap B)\&(A \multimap C) \cong A \multimap (B \& C)$, and *exponential* $!A$ serving to retract linear logic to intuitionistic logic via (inter alia) $!(A\&B) \cong !A \otimes !B$ and $!\top \cong 1$, licensing the derivation of $!A \vdash B$ from either $\vdash B$ or $!A, !A \vdash B$.

The question arises as to the denotational semantics of linear logic: what is it about? Is it only an analytical tool of proof theory, or can it be understood as the logic of some foundationally significant domain in the same sense that first order logic is the logic of relational structures, and modal logic that of Kripke structures?

Girard has considered various semantics for linear logic: phase semantics and coherent spaces [Gir87], Hilbert spaces, and more recently Banach spaces. Phase semantics has truth-valued entailment and resembles Birkhoff and von Neumann's quantum logic [BvN36], while the other three have the set-valued entailment characteristic of categorical logic: $A \vdash B$ as the set of morphisms from A to B.

A number of other models have also been proposed. Inspired by Blass [Bla92], Abramsky and Jagadeesan [AJ94] have interpreted linear logic over sequential games, further studied by Hyland and Ong [HO93]. Barr has proposed fuzzy relations as a model [Bar96], while Blute [Blu96] has taken Hopf algebras as an interpretation of noncommutative linear logic.

Chu spaces, the model we treat here, were first proposed by Barr [Bar91] and Lafont and Streicher [LS91]. Generalizing an idea of Mackey [Mac45], Barr defined general V-enriched Chu spaces, whose carrier, cocarrier, and alphabet k are objects of a symmetric monoidal closed category V, forming the category $\mathbf{Chu}(V, k)$ studied by Barr's student P. Chu [Bar79, appendix]. Lafont and Streicher treated ordinary Chu spaces, the case $V = Set$, under the rubric of games.

None of the above models, Chu spaces included, would appear to have been proposed with the goal in mind of foundational generality of the kind associated with relational structures in their role as the standard model of first order logic. Rather their intended purpose seems to be as "occasional models": each is presumed to have some intrinsic interest in its own right, and the main concerns revolve around the quality of the model as a denotational semantics for linear logic: how closely it matches the structure of LL proofs.

Furthermore none of them stands out as the standard model of linear logic. Coherent spaces have the distinction that Girard based his original axiomatization on their structure, and in that sense coherent spaces can be said to be the motivating model for linear logic. But that motivation would appear to be insight into proof theory rather than any independent foundational role for coherent spaces.

So the interest to date in denotational semantics of linear logic appears to be entirely as an analytical tool of proof theory. In this role it can serve for example to expose patterns in the operational or computational behavior of the rules of linear logic that are not obvious from direct consideration of that behavior. Indeed there has been considerable interest in the computational implications of linear logic, particularly for concurrency, and it is a good question to what extent the proof theoretic and computational aspects of linear logic can be separated, both being operational.

It is our thesis that the denotational semantics of linear logic in fact serves two dual purposes. On the one hand, in its role as an analytical tool of proof theory it expresses aspects of structure in mathematical proofs and computations. On the other, in the role we propose for it here, it expresses the transformational structure of *universal mathematics*, extending higher order categorical logic [LS86] from its previous narrow preoccupation with cartesian closed categories, where it has the form of intuitionistic logic, to the broader universe of "the rest of mathematics," where it takes on the shape of linear logic.

That this is an extension is testified to by the operator $!A$. As normally understood on the operational side $!A$ liberates the formula A to permit weakening and contraction. On the denotational side however the ! operation serves to retract the larger universe of mathematics to a smaller cartesian closed subuniverse constituting the domain of intuitionistic logic. From this perspective intuitionistic logic is the logic of set-like and poset-like structures, whereas linear logic is that of the larger class of all structures, ranging from the extreme discreteness of sets to the extreme coherence of complete atomic Boolean algebras [Pra95].

These two roles, proof theory and transformational mathematics, are not necessarily best served by the same denotational semantics. In particular Girard has argued the need for a denotational semantics of linear logic sufficiently concrete as to reflect the cut-elimination process itself [Gir89, §III]. Categorical logic as an abstraction of transformational mathematics on the other hand does not need this much detail at least in its basic form and is therefore better served by the more abstract cut-free semantics implicit in our choice of axiom systems for linear logic, and in the modeling of cut-free proofs by dinatural transformations.

The rest of this paper is laid out in three sections, introducing Chu spaces and treating each end of the bridge created by Chu spaces between linear logic and mathematics.

Section 2 gives an overview of Chu spaces, treating their intrinsic properties, their morphisms, and those operations on them relevant to linear logic.

Section 3 considers semantics for which soundness and completeness of linear logic, understood as a categorical logic, may be judged for the category of functors on $\mathbf{Chu}(\mathbf{Set}, \Sigma)$. The papers of Barr and Lafont and Streicher cited above have described the objects of that category, namely the functors interpreting the terms of linear logic. However they have not mentioned the morphisms between those functors, needed to interpret the proofs between terms that make the system a categorical logic.

Morphisms between functors are usually taken to be natural transformations. However linear logic contains functors of mixed variance[1] such as $A \multimap A$, for which mere naturality is not enough. Elsewhere [Pra97] we have shown that when the morphisms are taken to be ordinary dinatural transformations, as done by Blute and Scott [BS96a, BS96b] for their Lauchli semantics of linear logic, then Girard's MIX-free axiomatization of multiplicative linear logic is sound and fully complete for the fragment admitting at most two occurrences of each atom. In Section 3 we show that this result cannot be extended to four occurrences. This is not surprising in the light of the limitations of ordinary dinaturality in other settings such as higher order intuitionistic logic, where a strengthening of dinaturality is required for full completeness, e.g. logical transformations as defined in terms of certain kinds of logical relations [Plo80]. We have recently showed, with H. Devarajan, D. Hughes, and G. Plotkin [DHPP99], that strengthening dinaturality to binary logicality rescues this situation for Chu spaces. Contrasting with this situation, A. Tan has shown [Tan97] that MLL with MIX is fully complete for dinatural transformations in Girard's $*$-autonomous category **Coh** of coherence spaces.

We also introduce in this section the dialectic λ-calculus, as a novel way of introducing duality into the simply-typed λ-calculus, namely by supplementing the usual notion $a{:}A$ of evidence *a for* proposition A with $x{\cdot\cdot}A$, evidence *x against* A, for which Chu spaces provide a natural interpretation. We interpret a linear dialectic λ-calculus whose types are those expressible in a variant of MLL that while not fully expressive for all of MLL nevertheless

[1]One might call these sesquifunctors by analogy with sesquilinear functions.

is rich enough to express all MLL theorems.

Section 4 gives evidence for the universality of ordinary Chu spaces. A small hint of this universality may be found in Lafont and Streicher [*ibid.*], who observed that vector spaces over any field are representable as Chu spaces over the underlying set of that field, and that coherent spaces and topological spaces are both representable as Chu spaces over $2 = \{0, 1\}$.

As it turns out Chu spaces can represent far more. The main modes of structure for sets that are traditionally employed in mathematics are relational and topological, either separately or together, with algebraic structure being obtained from relational. Chu spaces subsume both kinds of structure, in their full generality, with the one mechanism. Rather than blending the two, as done with say topological groups, there is a single uniform construction. The homogeneous universe of Chu spaces appears to span the entire range of structures, having sets at the discrete end, complete atomic Boolean algebras at the "coherent" end, and finite-dimensional vector spaces, complete semilattices, etc. in the middle, constituting what we have called the *Stone gamut* [Pra95]. As we have previously shown [Pra93, Pra95], archivally documented here, this representation of relational and topological structures is formalized as a full, faithful, and concrete functor from the category of k-ary relational structures and their homomorphisms to the category of Chu spaces over an alphabet of cardinality 2^k, or 2^{k+1} when there is topology, being concrete in the sense of preserving the underlying sets.

In this paper we show also that every small category C embeds fully in $\mathbf{Chu}(\mathbf{Set}, |C|)$, and that every small concrete category embeds fully and concretely in $\mathbf{Chu}(\mathbf{Set}, \Sigma)$ where Σ is the disjoint union of the underlying sets of the category, i.e. the totality of elements.

In competition with Chu spaces for this role as universal structured objects are such structures as directed graphs and semigroups, whose universal nature was observed in the late sixties in work of Trnková, Hedrlín, Lambek and others [Trn66, HL69, PT80] showing that the respective categories of such fully embed all small and all algebraic categories. However those embeddings are not concrete, typically representing even finite objects as uncountable graphs or semigroups contrary to the intuitive understanding of the represented objects.

In contrast, the above representations by Chu spaces are all concrete. That is, the Chu representation of a concrete structure given by our embeddings has *the same underlying set*, and the licensed transformations of those structures, whether by homomorphisms, continuous functions, or whatever, remain the same functions, but now with a *uniform* licensing criterion, that of being a morphism of Chu spaces. All that changes is the representation of the structure associated to those sets. Our representation of classes of sets and classes of functions between them is thus with the *same* sets and functions, differing only in how those functions are specified and not in which ones the specification selects.

One would think that the greater generality of V-enriched spaces would embrace more mathematics, but counterintuitively the opposite seems to be the case: the closer V is to the "center" of the "Stone gamut" coordinatizing mathematics [Pra95] the smaller is the universe encompassed by $\mathbf{Chu}(V,k)$. Taking $V = \mathbf{Set}$, located at the apparent edge of the universe, appears to subsume all other V's while having the additional advantage of accessibility, sets being simpler and more familiar than the objects of pretty much any other choice for V.

For those interested in more hands-on experience with Chu spaces than can be had from this static paper, `http://boole.stanford.edu/live` is the URL for Chu Spaces Live, an interactive menu-driven Chu space calculator written by Larry Yogman. It permits the user to create and operate on Chu spaces with the operations described in Section 2.3, and includes a substantial tutorial.

2 Chu Spaces

2.1 Objects

A *Chu space* $\mathcal{A} = (A, r, X)$ *over* an alphabet Σ consists of a set A of *points*, a set X of dual points or *states*, and a function $r : A \times X \to \Sigma$ relating points and states. We refer to A as the *carrier*, X as the *cocarrier*, and r as the *interaction matrix*.

It is convenient to view Chu spaces as organized either by rows or by columns. For the former, we define $\hat{r} : A \to (X \to \Sigma)$ as $\hat{r}(a)(x) = r(a,x)$, and refer to the function $\hat{r}(a) : X \to \Sigma$ as *row* a of $\mathcal{A}$. Dually we define $\check{r} : X \to (A \to \Sigma)$ as $\check{r}(x)(a) = r(a,x)$ and call $\check{r}(x) : A \to \Sigma$ *column* x of $\mathcal{A}$.

Even before defining any notion of morphism between Chu spaces we can say that two Chu spaces are *isomorphic* when they are identical up to a renaming of their points and states, i.e. when there is a bijection between their points, and another between their states, such that they have the same matrix via that bijection.

We call $\mathcal{A}$ *separated* when its matrix has no repeated rows, that is, when $\hat{r}$ is injective ($\hat{r}(a) = \hat{r}(b)$ implies $a = b$). The *separated collapse* of $\mathcal{A}$ is the result of identifying equal rows. Formally it is defined as the Chu space $(\hat{r}(A), r', X)$ where $\hat{r}(A) = \{\hat{r}(a) \mid a \in A\}$ (so $\hat{r}(A) \subseteq \Sigma^X$) and $r'(a, x) = a(x)$ for $a \in \hat{r}(A)$ and $x \in X$.

Dually $\mathcal{A}$ is *extensional* when it has no repeated columns. The *extensional collapse* of $\mathcal{A}$ is defined as $(A, r', \check{r}(X))$ where $\check{r}(X) = \{\check{r}(x) \mid x \in X\}$ (so $\check{r}(X) \subseteq \Sigma^A$) and $r'(a, x) = x(a)$ for $a \in A$ and $x \in \check{r}(X)$. A *normal* Chu space is one which is its own extensional collapse. Normal spaces may be written as (A, X), r being understood to be application.

When $\mathcal{A}$ is both separated and extensional we call it *biextensional*. The *biextensional collapse* of a Chu space is the result of identifying equal rows, and equal columns, to produce a biextensional Chu space. We could define the biextensional collapse of (A, r, X) to be the extensional collapse of the separated collapse or vice versa, but here we have a dilemma: for the former the points are of type Σ^X and the states of type Σ^{Σ^X} while for the latter the respective types are Σ^{Σ^A} and Σ^A. We can resolve this democratically by taking the biextensional collapse to be $(\hat{r}(A), r', \check{r}(X))$ where $r'(\hat{r}(a), \check{r}(x)) = r(a, x)$, making the respective types Σ^X and Σ^A. This differs from both the separated and extensional collapses in that it retains both A and X.

Two Chu spaces are called *point-equivalent* (resp. state-equivalent) when they have isomorphic separated (resp. extensional) collapses. Equivalent Chu spaces are simply those that are both point-equivalent and state-equivalent.

Every operation defined on general Chu spaces, e.g. those of Section 2.3, has its counterpart for biextensional Chu spaces obtained by taking the biextensional collapse as needed.

A *discrete* Chu space is a normal Chu space (A, X) for which $X = \Sigma^A$.

2.2 Chu Transforms

A Chu transform is a pair (f, g) consisting of functions $f : A \to B$ and $g : Y \to X$ such that $s(f(a), y) = r(a, g(y))$ for all a in A and y in Y. This equation is a primitive form of adjointness, which we therefore call the *adjointness condition*. Such an adjoint pair (f, g) is called a *Chu transform* from $\mathcal{A}$ to $\mathcal{B}$.

Adjoint pairs $(f, g) : \mathcal{A} \to \mathcal{B}$ and $(f', g') : \mathcal{B} \to \mathcal{C}$, where $\mathcal{C} = (C, t, Z)$, compose via $(f', g')(f, g) = (f'f, gg')$. This composite is itself an adjoint pair because for all a in A and z in Z we have $t(f'f(a), z) = s(f(a), g'(z)) = r(a, gg'(z))$. The associativity of this composition is inherited from that of composition in **Set**, while the pair $(1_A, 1_X)$ of identity maps on respectively A and X is an adjoint pair and is the identity Chu transform on $\mathcal{A}$.

The notion of isomorphism of Chu spaces can now be defined more formally as a Chu transform whose functions are bijections.

The category whose objects are Chu spaces over Σ and whose morphisms are Chu transforms composing as above is denoted $\mathbf{Chu}(\mathbf{Set}, \Sigma)$. The full subcategory consisting of the biextensional Chu spaces is denoted $\mathbf{chu}(\mathbf{Set}, \Sigma)$, so-called "little **chu**," with parent category "big **Chu**."

Yet another definition of isomorphism of Chu spaces is that it is an isomorphism of $\mathbf{Chu}(\mathbf{Set}, \Sigma)$.

By the usual abuse of notation we permit a function $f : A \to B$ between sets to be referred to as a function $f : \mathcal{A} \to \mathcal{B}$ between Chu spaces, whence a function from $\mathcal{B}^\perp$ to $\mathcal{A}^\perp$ means a function from Y to X. We call a function $f : \mathcal{A} \to \mathcal{B}$ *continuous* when it has an adjoint from $\mathcal{B}^\perp$ to $\mathcal{A}^\perp$, i.e. when there exists a function $g : Y \to X$ making (f, g) a Chu transform. When $\mathcal{A}$ is extensional g is determined uniquely by f.

Like discrete topological spaces, discrete Chu spaces transform like sets. More generally, if $\mathcal{A}$ is discrete and $\mathcal{B}$ is any Chu space then the Chu transforms from $\mathcal{A}$ to $\mathcal{B}$ are exactly the functions from $\mathcal{A}$ to $\mathcal{B}$.

2.3 Multiplicative Operations

We now define a number of operations on Chu spaces. These operations are of independent interest but our rationale for them in this paper will be as interpretations of the connectives of linear logic, which we treat in the next section. The main operations for this paper are the multiplicatives, which we therefore treat first.

Negation. The dual or *linear negation* $\mathcal{A}^\perp$ of $\mathcal{A} = (A, r, X)$ is defined as $(X, r^\smile, A)$ where $r^\smile : X \times A \to \Sigma$ satisfies $r^\smile(x, a) = r(a, x)$.

When $\mathcal{A}$ is separable, $\mathcal{A}^\perp$ is extensional; likewise when $\mathcal{A}$ is extensional, $\mathcal{A}^\perp$ is separable. Thus negation preserves biextensionality, whence both big **Chu** and little **chu** are biextensional.

The definition of $A^\perp$ makes $\mathcal{A}^{\perp\perp}$ not merely isomorphic to $\mathcal{A}$ but equal to it.

Tensor. The tensor product $\mathcal{A} \otimes \mathcal{B}$ of $\mathcal{A} = (A, r, X)$ and $\mathcal{B} = (B, s, Y)$ is defined as $(A \times B, t, \mathcal{F})$ where $\mathcal{F} \subset Y^A \times X^B$ is the set of all pairs (f, g) of functions $f : A \to Y$, $g : B \to X$ for which $s(b, f(a)) = r(a, g(b))$ for all $a \in A$ and $b \in B$, and $t : (A \times B) \times \mathcal{F}$ is given by $t((a, b), f) = s(b, f(a))$ $(= r(a, g(b)))$.

Associated with tensor product is the *tensor unit* 1, namely the space $(\{0\}, r, \Sigma)$ where $r(0, k) = k$.

$\mathcal{A} \otimes \mathcal{B}$ can be understood as *interacting conjunction* as follows. Its points (a, b) are to be understood as the possible interactions of the points $a \in A$ with the points $b \in B$. Each state (f, g) indicates which state $\mathcal{B}$ appears to be in when viewed from point a of A, namely $f(a)$, and symmetrically which state $\mathcal{A}$ appears to be in when viewed from point b of B, namely $g(b)$.

Tensor is conjunction in the sense that the total constraint on the states of $\mathcal{A} \otimes \mathcal{B}$ is representable as the conjunction of the constraints imposed separately by each of $\mathcal{A}$ and $\mathcal{B}$, respectively (i) and (ii) in the following.

Elsewhere we have proposed tensor product as an operation of process algebra, called orthocurrence or flow-through [Pra86, CCMP91].

When $\mathcal{A}$ and $\mathcal{B}$ are extensional it suffices to take $\mathcal{F}$ to consist instead of all functions $m : A \times B \to \Sigma$ such that (i) for every $a \in A$ there exists $y \in Y$ such that for all $b \in B$, $m(a, b) = s(b, y)$; and (ii) for every $b \in B$ there exists $x \in X$ such that for all $a \in A$, $m(a, b) = r(a, x)$. Extensionality of $\mathcal{A}$ ensures that the y promised in (i) is the unique y for which $\check{s}(y) = \lambda b.m(a, b)$, whence m uniquely determines the $f : A \to Y$ of the first definition of $\mathcal{F}$ above, with the view from point a of A being the state $\lambda b.m(a, b)$ of $\mathcal{B}$. Likewise extensionality of $\mathcal{B}$ ensures that the same m uniquely determines g, with the view from point b of B being the state $\lambda a.m(a, b)$ of $\mathcal{A}$. This puts the m's in this definition of $\mathcal{F}$ in bijection with the (f, g) pairs in the first definition of $\mathcal{F}$, via $m(a, b) = s(b, f(a)) = r(a, g(b))$.

A state of $\mathcal{A} \otimes \mathcal{B}$ may be visualized as a solution to an $A \times B$ crossword puzzle having no blacked-out squares. To this end we construe a state of $\mathcal{A}$ as a word of "length" A over the alphabet Σ, with each point a of A constituting a "position" in the word. It is unimportant what order if any we attach to these positions. The set of states of $\mathcal{A}$ constitutes the dictionary of vertical words for the puzzle, and that of $\mathcal{B}$ the horizontal dictionary. The states of $\mathcal{A} \otimes \mathcal{B}$ are then all possible solutions to the puzzle, namely all possible $A \times B$ matrices whose columns are words of $\mathcal{A}$ and whose rows are words of $\mathcal{B}$.

When $\mathcal{A}$ and $\mathcal{B}$ are extensional, $\mathcal{A} \otimes \mathcal{B}$ is extensional by definition: two distinct functions in $\mathcal{F}$ must differ at a particular point (a, b). It need not however be separable, witness $\mathcal{A} \otimes \mathcal{A}$ where $\mathcal{A}$ is the 2-point 2-state space with matrix $\substack{00 \\ 01}$. The two "crossword solutions" here are $\substack{00 \\ 00}$ and $\substack{00 \\ 01}$. Hence $\mathcal{A} \otimes \mathcal{A}$ is a 4-point 2-state space three of whose points are represented by the row 00 and one by the row 01. The biextensional collapse of this identifies the three 00 rows to yield the biextensional space $\substack{00 \\ 01}$.

Functoriality. We have defined $\mathcal{A}^{\perp}$ and $\mathcal{A} \otimes \mathcal{B}$ only for Chu spaces. We now make these operations functors on $\mathbf{Chu}(\mathbf{Set}, \Sigma)$ by extending their respective domains to include morphisms.

Given $(f, g) : \mathcal{A} \to \mathcal{B}$, $(f, g)^\perp : \mathcal{B}^\perp \to \mathcal{A}^\perp$ is defined to be (g, f). This suggests writing the adjoint g as $f^\perp$, which we do henceforth.

Given functions $f : \mathcal{A} \to \mathcal{A}'$ and $g : \mathcal{B} \to \mathcal{B}'$, define $f \otimes g : \mathcal{A} \otimes \mathcal{B} \to \mathcal{A}' \otimes \mathcal{B}'$ to be the function $(f \otimes g)(a, b) = (f(a), g(b))$. When f and g are continuous, so is $f \otimes g$, whose adjoint $(f \otimes g)^\perp$ from $\mathcal{G}$ to $\mathcal{F}$ (where $\mathcal{G}$ and $\mathcal{F}$ consist respectively of pairs $(h' : A' \to Y', k' : B' \to X')$ and $(h : A \to Y, k : B \to X)$) sends $h' : A' \to Y'$ to $g^\perp h' f : A \to Y$.

Laws. Tensor is commutative and associative, albeit only up to a natural isomorphism: $\mathcal{A} \otimes \mathcal{B} \cong \mathcal{B} \otimes \mathcal{A}$ and $\mathcal{A} \otimes (\mathcal{B} \otimes \mathcal{C}) \cong (\mathcal{A} \otimes \mathcal{B}) \otimes \mathcal{C}$. The naturality of these isomorphisms follows immediately from that of the corresponding isomorphisms in **Set**. We show their continuity separately for each law.

For commutativity, the isomorphism is $(\gamma, \delta) : (\mathcal{A} \otimes \mathcal{B}) \to (\mathcal{B} \otimes \mathcal{A})$ where $\mathcal{A} \otimes \mathcal{B} = (A \times B, t, \mathcal{F})$, $\mathcal{B} \otimes \mathcal{A} = (B \times A, u, \mathcal{G})$, $\gamma : A \otimes B \to B \otimes A$ satisfies $\gamma(a, b) = (b, a)$, and $\delta : \mathcal{G} \to \mathcal{F}$ satisfies $\delta(g, f) = (f, g)$. The continuity of (γ, δ) then follows from $u(\gamma(a, b), (g, f)) = u((b, a), (g, f)) = r(a, g(b)) = s(b, f(a)) = t((a, b), (f, g)) = t((a, b), \delta^\perp(g, f))$, for all (a, b) in $A \times B$ and (g, f) in $\mathcal{G}$.

For associativity, let $\mathcal{A} = (A, r, X)$, $\mathcal{B} = (B, s, Y)$, and $\mathcal{C} = (C, t, Z)$. Observe that both $(\mathcal{A} \otimes \mathcal{B}) \otimes \mathcal{C}$ and $\mathcal{A} \otimes (\mathcal{B} \otimes \mathcal{C})$ can be understood as $(A \times B \times C, u, \mathcal{F})$ where $\mathcal{F}$ consists of all functions $m : A \times B \times C \to \Sigma$ satisfying the conjunction of three conditions: (i) for all b, c there exists x such that for all a, $m(a, b, c) = r(a, x)$; (ii) for all c, a there exists y such that for all b, $m(a, b, c) = s(b, y)$; and (iii) for all a, b there exists z such that for all c, $m(a, b, c) = t(c, z)$. In the imagery of crosswords, $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ supply the dictionaries for the respective axes of a three-dimensional crossword puzzle, with $\lambda a.m(a, b, c)$ denoting the word in m at point (b, c) of $B \times C$ parallel to the A axis and similarly for the word $\lambda b.m(a, b, c)$ at (c, a) parallel to the B axis and $\lambda c.m(a, b, c)$ at (a, b) parallel to the C axis.

With this observation we can now describe the isomorphism between $(\mathcal{A} \otimes \mathcal{B}) \otimes \mathcal{C}$ and $\mathcal{A} \otimes (\mathcal{B} \otimes \mathcal{C})$: on the points it is the usual set-theoretic isomorphism of $(A \times B) \times C$ and $A \times (B \times C)$, while on the states it is the correspondence pairing each map $m : (A \times B) \times C \to \Sigma$ with the map $m' : A \times (B \times C) \to \Sigma$ satisfying $m'(a, (b, c)) = m((a, b), c)$. It is immediate that this pair of bijections is a Chu transform. Hence tensor is associative up to this isomorphism.[2]

The tensor unit behaves as such, i.e. $\mathcal{A} \otimes 1 \cong \mathcal{A}$, via the evident isomor-

[2] If **Set** is organized to make cartesian product associative "on the nose", i.e. up to identity, possible assuming the Axiom of Choice though not if **Set** is not skeletal [Mac71, p.161], then tensor product in **Chu**(**Set**, Σ) is also associative on the nose.

phism pairing $(a, 0)$ with a.

Linear Implication. We define linear implication, namely $\mathcal{A}\multimap\mathcal{B}$, as $(\mathcal{A}\otimes\mathcal{B}^\perp)^\perp$. It follows that $\mathcal{A}\multimap\mathcal{B} = (\mathcal{F}, t, A \times Y)$ where $\mathcal{F}$ is the set of all pairs (f, g), $f : A \to B$ and $g : Y \to X$, satisfying the adjointness condition for Chu transforms.

For little **chu**, biextensional spaces, the crossword puzzle metaphor applies. A function $f : \mathcal{A} \to \mathcal{B}$ may then be represented as an $A \times Y$ matrix m over Σ, namely $m(a, y) = s(f(a), y)$, while a function $g : \mathcal{B}^\perp \to \mathcal{A}^\perp$ may be represented as a $Y \times A$ matrix over Σ, namely $m(y, a) = r(a, g(y))$. For such spaces we then have an alternative characterization of continuity: a function is continuous just when the converse (transpose) of its representation represents a function from $\mathcal{B}^\perp$ to $\mathcal{A}^\perp$.

With biextensional arguments $\mathcal{A}\multimap\mathcal{B}$ is separable but not necessarily extensional for the same reason $\mathcal{A} \otimes \mathcal{B}$ is extensional but not necessarily separable. Hence to make $\mathcal{A}\multimap\mathcal{B}$ biextensional, equal columns must be identified.

2.4 Other operations

Besides the multiplicative operations, linear logic has additives, which are generalizable to limits and colimits, and exponentials.

Additives The additive connectives of linear logic are *plus* $\mathcal{A} \oplus \mathcal{B}$ and *with* $\mathcal{A}\&\mathcal{B}$, with respective units 0 and $\top$.

$\mathcal{A} \oplus \mathcal{B}$ is defined as $(A + B, t, X \times Y)$ where $A + B$ is the disjoint union of A and B while $t(a,(x, y)) = r(a, x)$ and $t(b,(x, y)) = s(b, y)$. Its unit 0 is the discrete empty space having no points and one state. For morphisms $f : \mathcal{A} \to \mathcal{A}'$, $g : \mathcal{B} \to \mathcal{B}'$, $f \oplus g : \mathcal{A} \oplus \mathcal{B} \ \to \ \mathcal{A}' \oplus \mathcal{B}'$ sends $a \in \mathcal{A}$ to $f(a) \in \mathcal{A}'$ and $b \in \mathcal{B}$ to $g(b) \in \mathcal{B}'$. *With* as the De Morgan dual of *plus* is defined for both objects and morphisms by $\mathcal{A}\&\mathcal{B} = (\mathcal{A}^\perp \oplus \mathcal{B}^\perp)^\perp$, while $\top = 0^\perp$.

Limits and Colimits The additives furnish Chu spaces with only finite discrete limits and colimits. In particular $\mathcal{A} \oplus \mathcal{B}$ is a coproduct of $\mathcal{A}$ and $\mathcal{B}$ while $\mathcal{A}\&\mathcal{B}$ is a product of them.

In fact $\mathbf{Chu}(\mathbf{Set}, \Sigma)$ is bicomplete, having all small limits and colimits, which it inherits in the following straightforward way from **Set**. Given any diagram $D : J \to \mathbf{Chu}(\mathbf{Set}, \Sigma)$ where J is a small category, the limit of D is obtained independently for points and states in respectively **Set** and $\mathbf{Set}^\circ$.

Exponentials The *exponential* $!\mathcal{A}$ serves syntactically to "loosen up" the formula A so that it can be duplicated or deleted. It achieves this by adding enough states to $\mathcal{A}$. Semantically as an operation on $\mathbf{Chu}(\mathbf{Set}, \Sigma)$ it serves to retract that domain to a cartesian closed subcategory.

A candidate for this subcategory is that of the discrete Chu spaces (A, Σ^A), a subcategory equivalent to the category **Set** of sets and functions. This exponential adds all possible states to $\mathcal{A}$ to make it discrete, the ultimate in "loosening up." A larger subcategory that is also cartesian closed is that of the comonoids, which we now define.

A *comonoid in* **Chu**(**Set**, Σ) is a Chu space $\mathcal{A}$ for which the diagonal function $\delta : \mathcal{A} \to \mathcal{A} \otimes \mathcal{A}$ and the unique function $\epsilon : \mathcal{A} \to 1$ are continuous (where 1 is the tensor unit). These two functions constitute the interpretation of duplication and deletion respectively, in that $\delta(a) = (a, a)$ (a is duplicated) and $\epsilon(a) = ()$ (a is deleted).

An equivalent definition of comonoid is as a Chu space $\mathcal{A}$ such that (i) for each element of Σ there exists a column of $\mathcal{A}$ all of whose entries are that element (this makes ϵ continuous), and (ii) every state of $\mathcal{A} \otimes \mathcal{A}$ construed as an $A \times A$ matrix has for its main diagonal a column of $\mathcal{A}$ (this makes δ continuous).

It is immediate that discrete Chu spaces satisfy both (i) and (ii) and hence are comonoids.

A normal comonoid is a comonoid that is a normal Chu space (A, X), one whose state set X is a subset of Σ^A. The comonoid *generated by* a normal Chu space $\mathcal{A} = (A, X)$, denoted $!\mathcal{A}$, is defined as the normal comonoid (A, Y) having the least $Y \supseteq X$. That this Y exists is a corollary of the following lemma.

Lemma 1 *Given any family of comonoids (A, X_i) with fixed carrier A, $(A, \bigcap_i X_i)$ is a comonoid. (When the family is empty we define $\bigcap_i X_i$ to be Σ^A.)*

Proof: The case of the empty family is covered by the remark three paragraphs back.

The unique function from (A, X) to 1 is continuous just when X includes every constant function. All the X_i's must have this property and therefore so does their intersection.

Given $\mathcal{A} = (A, X)$, $\delta : \mathcal{A} \to \mathcal{A} \otimes \mathcal{A}$ is continuous just when every $A \times A$ crossword solution with dictionary X has for its leading diagonal a word from X. It follows that if a solution uses words found in every dictionary X_i, then the diagonal is found in every X_i. Hence the dictionary $\bigcap_i X_i$ also has this property.

Therefore $(A, \bigcap_i X_i)$ is a comonoid. ∎

The domain of the ! operation is easily extended to arbitrary Chu spaces by first taking the extensional collapse.

M. Barr has pointed out to us that $!\mathcal{A}$ is cofree for the category of extensional Chu spaces over Σ, but not for the larger category of all Chu spaces over Σ.

3 Relationship of Chu Spaces to Linear Logic

3.1 Language

We have already encountered the linear logic connectives in the form of functors on $\mathbf{Chu}(\mathbf{Set}, \Sigma)$, which we shall take as the Chu interpretations of the linear logic connectives. What makes linear logic a logic is its axiomatization and its relationship to those interpretations, the subject of this section. We give two axiomatizations, systems S1 and S2, the latter being more suited to interpretation over Chu spaces. In the process we give a tight correspondence between the two systems in terms of linkings and switchings.

Linear logic is usually axiomatized in terms of Gentzen sequents $\Gamma \vdash \Delta$. However it can also be axiomatized in Hilbert style, with $A \vdash B$ denoting not a Gentzen sequent but rather that B is derivable from A in the system, the approach we follow here. We confine our attention to the multiplicative fragment, MLL, further simplified by omitting the constants 1 and $1^\perp = \perp$, which is sufficient to illustrate the relationship between the Chu interpretation of the linear logic connectives and their axiomatization.

It will be convenient to assume a normal form for the language in which implications $A \multimap B$ have been expanded as $A^\perp \invamp B$ and all negations have been pushed down to the leaves. Accordingly we define a *formula* A (B, C,...) to be either a *literal* P (an atom or propositional variable or its negation), a *conjunction* $A \otimes B$ of two formulas, or a *disjunction* $A \invamp B$ of two formulas. When P is a literal of the form $Q^\perp$ then $P^\perp$ *denotes* the literal Q. This simplifies the axiom system by permitting double negation, the De Morgan laws, and all properties of implication to be omitted from the axiomatization. Call this *monotone MLL*.

3.2 Axiomatization

We axiomatize MLL with one axiom schema together with rules for associativity, commutativity, and linear or weak distributivity.

T	$(P_1^\perp \parr P_1) \otimes \ldots \otimes (P_n^\perp \parr P_n), \quad n \geq 1$		
$A1$	$(A \otimes B) \otimes C$	$\vdash$	$A \otimes (B \otimes C)$
$A2$	$(A \parr B) \parr C$	$\vdash$	$A \parr (B \parr C)$
$C1$	$A \otimes B$	$\vdash$	$B \otimes A$
$C2$	$A \parr B$	$\vdash$	$B \parr A$
D	$(A \parr B) \otimes C$	$\vdash$	$A \parr (B \otimes C)$
$E1$	$A \otimes B$	$\vdash$	$A' \otimes B'$
$E2$	$A \parr B$	$\vdash$	$A' \parr B'$

Table 1. System S1. ($A \vdash A'$ and $B \vdash B'$)

The rules have the interesting feature of all having exactly one premise.[3] This makes the system *cut-free*, lacking the cut rule either in the form "from $A \multimap B$ and $B \multimap C$ infer $A \multimap C$," or as *modus ponens*, "from A and $A \multimap B$ infer C." It also performs all collecting of theorems at the outset in a single axiom rather than later via a rule of the form $A, B \vdash A \otimes B$. We may then treat $\vdash$ as a binary relation on formulas, being defined as the reflexive transitive closure of the binary relation whose pairs are all substitution instances of the above rules. We read $A \vdash B$ as A derives B, or B is deducible from A.

Rules A1 and A2 express associativity of $\otimes$, while Rules C1 and C2 express its symmetry (commutativity). Rule D is linear distributivity. Rules E1 and E2 express functoriality; they assume $A \vdash A'$ and $B \vdash B'$, allowing the rules to be applied to subformulas. The $\vdash$ defined by this system is not altered by imposing the restriction that either $A \vdash A'$ be an instance of one of the rules A1-D, and $B = B'$, or vice versa, i.e. when only one side of $A \otimes B$ can be rewritten by E1 at a time.

An instance of T is determined by a choice of association for the $n-1$ $\otimes$'s and a choice of n literals (atoms P or negated atoms $P^\perp$). When $P_i^\perp$ is instantiated with $Q^\perp$ the resulting double negation is cancelled, as in the instance $(Q^\perp \parr Q) \otimes ((P^\perp \parr P) \otimes (Q \parr Q^\perp))$ which instantiates P_1 with Q and P_3 with $Q^\perp$.

An MLL theorem B is any formula deducible from an instance A of axiom schema T, i.e. one for which $A \vdash B$ holds. For example $(P \otimes Q) \multimap (P \otimes Q)$, which abbreviates $(P^\perp \parr Q^\perp) \parr (P \otimes Q)$, can be proved as follows from instance $(P^\perp \parr P) \otimes (Q^\perp \parr Q)$ of T.

[3] We view T purely as an axiom and not also as a rule with no premises.

$$
\begin{array}{rcll}
(P^{\perp} \parr P) \otimes (Q^{\perp} \parr Q) & \vdash & P^{\perp} \parr (P \otimes (Q^{\perp} \parr Q)) & (D) \\
& \vdash & P^{\perp} \parr ((Q^{\perp} \parr Q) \otimes P) & (C1, E2) \\
& \vdash & P^{\perp} \parr (Q^{\perp} \parr (Q \otimes P)) & (D, E2) \\
& \vdash & P^{\perp} \parr (Q^{\perp} \parr (P \otimes Q)) & (C1, E2) \\
& \vdash & (P^{\perp} \parr Q^{\perp}) \parr (P \otimes Q)) & (A2)
\end{array}
$$

3.3 Semantics

Multiplicative linear logic has essentially the same language as propositional Boolean logic, though only a proper subset of its theorems. But whereas the characteristic concern of Boolean logic is truth, *separating* the true from the false, that of linear logic is proof, *connecting* premises to consequents.

In Boolean logic proofs are considered syntactic entities. While MLL derivations in S1 are no less syntactic intrinsically, they admit an abstraction which can be understood as the underlying semantics of MLL, constituting its abstract proofs. These are cut-free proofs, S1 being a cut-free system.

Define a *linking* L of a formula A to be a matching of complementary pairs or *links* $P, P^{\perp}$ of literal occurrences. Call such a pair (A, L) a *cut-free proof structure* [Gir87], or just proof structure as we shall be working exclusively with cut-free proofs.

There exist both syntactic and semantic characterizations of theorems in terms of linkings, which Danos and Regnier have shown to be equivalent [DR89].

For the syntactic characterization, every MLL derivation of a theorem A determines a proof structure as follows. The proof structure determined by an instance of T matches $P_i^{\perp}$ and P_i in each conjunct. Since the rules neither delete nor create subformulas but simply move them around, the identities of the literals are preserved and hence so is their linking. Hence the rules can be understood as transforming not just formulas but proof structures. Call a proof structure *sound*, or a proof *net*, when it can be derived from that of an axiom instance by the rules of S1. It is immediate that A is a theorem if and only if it has a sound proof structure.

For the semantic characterization, define a *switching* σ for a formula A to be a marking of one disjunct in each disjunction occurring in A; since disjunctions are binary and there are n of them, there are 2^n possible switchings. A linking L of A and a switching σ for L together determine an undirected graph $G(A, L, \sigma)$ whose vertices are the $4n - 1$ subformulas of A, consisting of $2n$ literals, $n - 1$ conjunctions, and n disjunctions, and whose edges consist of:

(i) the n pairs $(P_i, P_i^\perp)$ of literals matched by the linking;

(ii) the $2n-2$ pairs (B, C) where B is a conjunction in A and C is either of B's two conjuncts; and

(ii) the n pairs (B, C) where B is a disjunction in A and C is the disjunct in B marked by σ (n disjunctions hence n such edges).

(The linear logic literature refers to $G(A, L, \sigma)$ itself as a switching.)

Call a proof structure (A, L) *valid* when for all switchings σ for L, $G(A, L, \sigma)$ is a tree (connected acyclic graph).[4]

The more usual term for this notion in the linear logic literature is "proof structure satisfying the Danos-Regnier criterion." However the criterion seems to us semantical in the same sense as validity in Boolean propositional calculus, which is defined as truth over all assignments of truth values to variables. If we regard linear logic as fundamentally a proof-oriented "connectionist" logic, in contradistinction to Boolean logic as a truth-oriented "separationist" logic, and if we view switching as the connectionist counterpart of truth assignment, then a condition that universally quantifies over all switchings is just as much a notion of validity as is one that universally quantifies over all truth assignments.

With this perspective we may then restate Danos and Regnier's celebrated theorem [DR89] as follows.

Theorem 2 *(Danos-Regnier) A proof structure is sound if and only if it is valid.*

This result constitutes a form of completeness result for MLL. However it is stronger than the usual notion of completeness in that it sets up a bijection between syntactic and semantic criteria for theoremhood called *full completeness*, the term coined by Abramsky and Jagadeesan for their game semantics of MLL [AJ92] but equally applicable to switching semantics. Here the bijection is identification: the valid linking that each sound linking is paired with is itself. The sound linkings of A constitute abstract proofs of A, semantically justified by their validity. For transformational semantics as treated in section 3.8, the corresponding bijection is between cut-free proofs (as sound linkings) and transformations meeting a suitable naturality condition such as dinaturality or binary logicality.

[4]There being $4n-1$ vertices and $4n-2$ edges, had $G(A, L, \sigma)$ failed to be a tree it would necessarily have done so by both being disconnected *and* containing a cycle.

3.4 Syntactic Expression of Linking

The boundary between syntax and semantics is not sharp, and semantical information can often be encoded syntactically. For example the satisfying assignments of a Boolean formula can be represented syntactically by putting the formula in disjunctive normal form, with each disjunct (conjunction of literals) then denoting those satisfying assignments for which the positive literals in the disjunct are assigned *true* and the negative *false*. When all the atoms occurring in a formula occur in every disjunct, either positively or negatively, the disjuncts are in bijection with the satisfying assignments.

The semantical notions of linking and switching can likewise be incorporated into MLL formulas. We begin with linking, the key idea for which is to label each atom with the name of the link it belongs to.

In general a formula A may have many linkings or no linking. But for a *binary* formula, one such that every atom occurring in A does so once positively and once negatively (e.g. when all P_i's of T are distinct), there exists a unique linking. Conversely a linking of an arbitrary formula A determines a binary formula A' obtained from A by assigning distinct names to the links and subscripting each atom with the name of the link it belongs to. It follows that the notions of a proof structure and a binary formula can be used interchangeably. It should be borne in mind that the theoremhood question for a formula is in general harder than for a proof structure or a binary formula.

Since we will be dealing only with proof structures (A, L) in this section, we may assume for the rest of this section that all formulas are binary. The links still exist but they are now uniquely determined by A alone, having been absorbed into the language. $G(A, L, \sigma)$ becomes just $G(A, \sigma)$, and instead of saying the linking L of A is sound or valid we can simply say that the binary formula A is provable or valid respectively. The Danos-Regnier theorem then says more simply that a binary formula A is provable if and only if it is valid.

3.5 Syntactic Expression of Switching

Switching semantics is well motivated in that it serves as a crucial stepping stone for all known completeness proofs of other MLL semantics. A more intrinsic motivation for it however is based on the notion of information flow in proofs. In the Chu interpretation this flow is realized by transformations. However the flow can be understood abstractly in its own right, which we treat prior to considering the transformational interpretation of such flows.

The key idea here is the choice of $A^\perp \multimap B$ or $A \circ\!\!- B^\perp$ as direction-encoding synonyms for the direction-neutral $A ⅋ B$. Marking ⅋ with each of two possible directions permits us to reconcile the commutativity of ⅋ with our λ-calculus interpretation of the axiom and rules of system S2.

The customary direction of flow in assigning a denotation to an expression is upwards in the tree, with the denotation of the expression flowing from leaves to root. But now consider the theorem $(P \otimes (P \multimap Q)) \multimap Q$. There is a natural direction of flow starting from P through $P \multimap Q$ and ending at Q. The flow at $P \multimap Q$ would seem to go from the P leaf up to the $\multimap$ thence down to the Q leaf.

Now this theorem is just an abbreviation of $(P^\perp ⅋ (P \otimes Q^\perp)) ⅋ Q$, whose connectives can be reassociated and permuted to yield $P^\perp ⅋ (Q ⅋ (P \otimes Q^\perp))$. The latter can be abbreviated as $P \multimap (Q^\perp \multimap (P \otimes Q^\perp))$, with the flow now taking on the form of a pair of flows from P to P and from $Q^\perp$ to $Q^\perp$, changing the apparent direction of one of the two ⅋'s.

This example suggests the possibility of correlating switchings with theorems stated using implications. In fact there exists a very good correlation taking the form of a bijection between the essential switchings of a binary theorem A and the set of bi-implicational expressions of A, terms that we now define.

Essential switchings. Given a binary theorem A, the tree $G(A, \sigma)$ induced by a switching σ is made a directed graph by orienting its edges towards the root. Non-link edges, namely those connecting a conjunction or disjunction to one of its operands, may be oriented either downwards or upwards (we follow the usual convention of putting the root of a parse tree at the top; the linear logic literature tends to invert this). Call a conjunction or disjunction *downwards* or *upwards* in $G(A, \sigma)$ according respectively to whether or not a downwards edge is incident on it. The root is necessarily upwards. For example in $(P^\perp ⅋ (P \otimes Q^\perp)) ⅋ Q$, $P \otimes Q$ is downwards for all switchings save that in which the first ⅋ is switched to the right and the second to the left, and for that switching the links are oriented $P^\perp$ to P and Q to $Q^\perp$.

We now analyze the topology of $G = G(A, \sigma)$ at any given ⅋. For any subformula $B = C ⅋ D$, if the edge from B (meaning the root of B) to whichever of C or D it is directly connected to is removed, G must separate into two trees. The tree containing vertex B cannot contain either C or D or there would be a cycle when the corresponding edge from B is put in. Hence the other tree must contain both C and D.

Now suppose B is an upward disjunction. Then whichever of C or D was directly connected to B in G must be the root of the tree containing it, and the other of C or D a leaf. This is interchanged by changing the switching at B, which has the side effect of reversing all edges along the path between C and D.

If however B is a downward disjunction, then both C and D are leaves of their common tree. Changing the switching at B does not change this fact, nor the orientation of any edge in either tree.

In the above example $(P^\perp \parr (P \otimes Q^\perp)) \parr Q$, when the second $\parr$ is switched to the right, the first $\parr$ becomes downwards. In that case the path to the root starts at $P^\perp$ and proceeds via P, $P \otimes Q^\perp$, $Q^\perp$, and Q ending at the root $(P^\perp \parr (P \otimes Q^\perp)) \parr Q$; the direction of the first $\parr$ connects the vertex $P^\perp \parr (P \otimes Q^\perp)$ to one of $P^\perp$ or $P \otimes Q^\perp$. Changing that connection does not reverse any edge but merely replaces one downwards edge from $P^\perp \parr (P \otimes Q^\perp)$ by the other.

An *essential switching* is one that records the direction only of the upward disjunctions for that switching. We can think of the downwards disjunctions as being recorded as X for don't-care. This has the effect of identifying those switchings differing only at their downward disjunctions. Thus $(P^\perp \parr (P \otimes Q^\perp)) \parr Q$ has only three essential switchings because when the second $\parr$ is switched to the right we ignore the now downward first $\parr$.

3.6 Bi-Implication

We would like to interpret the formulas $A \otimes B$ and $A \parr B$ as types. With the Chu interpretations of these connectives in mind, we regard entities of the former type as pairs, and of the latter as functions, either from $A^\perp$ to B or from $B^\perp$ to A.

Now the connectives appearing in the rules of S1 are just $\otimes$ and $\parr$, without any negations $A^\perp$. It would be a pity to have to introduce negations as a side effect of interpreting $A \parr B$ as consisting of functions. To avoid this we shall make $\multimap$ perform the role of $\parr$, allowing us to talk of functions of type $A \multimap B$.

This works fine except for rule C2, commutativity of $\parr$, which must rewrite $A \multimap B$ as $B^\perp \multimap A^\perp$. To avoid having negation appear in C2 we adopt $A \multimapinv B$ as a synonym for $B^\perp \multimap A^\perp$.

With this motivation we introduce the *language of bi-implicational MLL*. A formula in this language is one that is built up from literals using $\otimes$, $\multimap$, and $\multimapinv$.

We axiomatize bi-implicational MLL as follows. The axiom and rules are obtained from System S1 by rewriting each $A \parr B$ in T or on the left side of a rule by either $A^\perp \multimap B$ or $A \multimapinv B^\perp$ in all possible combinations, with the negations pushed down to the metavariables $(A, B, C, \ldots)$ and with any resulting negative metavariables then instantiated with their complement.

T	$(P_1 \multimapboth P_1) \otimes \ldots \otimes (P_n \multimapboth P_n), \quad n \geq 1$		
$A1$	$(A \otimes B) \otimes C$	$\vdash$	$A \otimes (B \otimes C)$
$A2$	$(A \otimes B) \multimap C$	$\vdash$	$A \multimap (B \multimap C)$
$A2'$	$(A \multimap B) \multimapinv C$	$\vdash$	$A \multimap (B \multimapinv C)$
$A2''$	$(A \multimapinv B) \multimapinv C$	$\vdash$	$A \multimapinv (B \otimes C)$
$C1$	$A \otimes B$	$\vdash$	$B \otimes A$
$C2$	$A \multimap B$	$\vdash$	$B \multimapinv A$
$C2'$	$A \multimapinv B$	$\vdash$	$B \multimap A$
D	$(A \multimap B) \otimes C$	$\vdash$	$A \multimap (B \otimes C)$
D'	$(A \multimapinv B) \otimes C$	$\vdash$	$A \multimapinv (B \multimapinv C)$
$E1$	$A \otimes B$	$\vdash$	$A' \otimes B'$
$E2$	$A' \multimap B$	$\vdash$	$A \multimap B'$
$E3$	$A \multimapinv B'$	$\vdash$	$A' \multimapinv B$

Table 2. System S2

By $P \multimapboth P$ we mean the choice of $P \multimap P$ or $P \multimapinv P$, where P is a literal as for system S1, with the choice made independently for each of the n implications of T. Thus T has 2^n instantiations for any given selection of n literals. As before we assume $A \vdash A'$ and $B \vdash B'$ for E1-E3.

As with system S1, the only negations are on literals and remain there, and the rules do not mention negation, catering solely for $\otimes$, $\multimap$, and $\multimapinv$.

Theorem 3 *The binary theorems of S2 are in bijection with the pairs* (A, σ) *where* A *is a binary theorem of S1 and* σ *is an essential switching for* A*.*

(Note that different linkings of the same nonbinary theorem can affect which switchings are essential. Hence we cannot strengthen this to a bijection between the theorems of S2 and pairs (A, σ) where A is a theorem of S1, since not all linkings of A need be compatible with the same σ.)

Proof: We exhibit a map in each direction and prove that they compose in either order to the respective identity. Neither map by itself requires induction on length of proofs to specify the map, but does require it in order to prove theoremhood of the result.

We translate theorems of S2 to formulas of S1 via *bi-implication expansion.* This is simply the result of rewriting each $A \multimap B$ as $A^\perp \parr B$ and $A \multimapinv B$

as $A \parr B^\perp$ and pushing the negations down to the literals via De Morgan's laws for $\otimes$ and $\parr$, canceling double negations.[5]

Applying this translation to S2 converts it to S1. It follows by induction on length of proofs that every theorem of S2 translates in this way to a theorem of S1.

For the other direction, we are given a binary theorem A of S1 together with a switching σ and want a theorem of S2. We can specify a formula without using induction on length of proofs by appealing to the Danos-Regnier theorem. The switching determines a graph $G(A, \sigma)$, oriented as in the description above of essential switchings.

Rewrite each downward disjunction $B = C \parr D$ as $(C^\perp \otimes D^\perp)^\perp$. Note that this rewriting ignores the direction of switching at this $\parr$. Rewrite each upward disjunction $B = C \parr D$ as either $C \circ\!\!- D^\perp$ or $C^\perp \multimap D$ according to whether C or D respectively is the marked disjunct. Lastly rewrite each downward conjunction $B \otimes C$ as either $(B^\perp \circ\!\!- C)^\perp$ or $(B \multimap C^\perp)^\perp$ according to whether the path from $B \otimes C$ goes to B or C respectively. Cancel any double negations that arise directly from this translation. Call the final result the σ-translation of A.

We now claim that the σ-translation of a binary theorem of S1 is a formula in the language of S2. To see this observe that the negations introduced by this rewriting appear only on downward edges. Morever every downward edge between two compound subformulas (i.e. not involving a literal) receives a negation at each end, whence all such negations may be cancelled directly without bothering to apply De Morgan's laws to push negations down. The only remaining negations are then those on nonlink edges involving literals. If the literal is $P^\perp$ then we have another pair of negations that may be cancelled. If it is P then leave the negation in place so that it becomes $P^\perp$, and observe that the literal to which P (now negated) is linked is $P^\perp$. It follows that the only remaining negations are at literals, and furthermore that links connect occurrences of the *same* literal (in S1 they connected complementary pairs). Such a formula is in the language of S2.

We further claim that this formula is a theorem of S2. To see this proceed by induction of the length of proofs in S1. For the basis case, translating an instance of T in S1 turns the i-th disjunctions into one of $P \multimap P$, $P \circ\!\!- P$, $P^\perp \multimap P^\perp$, or $P^\perp \circ\!\!- P^\perp$ depending on the sign of P_i in the S1 theorem and the direction determined by σ for that $\parr$ in the σ-translation.

[5]For noncommutative l inear logic [Abr90] the De Morgan laws also reverse order; here we leave the order unchanged so as to preserve the exact structure of all formulas.

For the inductive step, every way of rewriting the $\parr$'s on the left of a rule of S1 as either $\multimap$ or $\multimapinv$ is represented on the left of some rule of S2. (Associativity has only three such combinations rather than four for essentially the same reason that $(P^\perp \parr (P \otimes Q^\perp)) \parr Q$ has only three essential switchings: when the second $\parr$ is switched to the right the first $\parr$ becomes downwards, and translates to $\otimes$ which does not have separate notations for its two directions.) Hence every step of an S1 derivation can be mimicked by an S2 step, preserving the claimed bijection. This completes the proof of the claim.

It should now be clear that the two translations are mutually inverse, establishing the bijection claimed by the theorem. ■

What we have shown in effect is that S1 and S2 are equivalent axiomatizations of MLL, modulo the difference in language and the additional information in S2 about the switching. From the Danos-Regnier theorem we have that each binary theorem A of S1 in monotone MLL corresponds to a set of theorems of S2 in bi-implicational MLL, one for each essential switching of A.

Higher Order Theorems. Note that rule D' of S2 may increase order (depth of nesting of implications in the antecedent). This allows S2 to prove theorems of arbitrarily high order limited only by n. For example with $n = 3$ we can proceed using only D' as follows to obtain a theorem of order 5.

$$
\begin{aligned}
(P \multimap P) \otimes ((Q \multimap Q) \otimes (R \multimap R)) \ &\vdash \ (P \multimap P) \otimes (((R \multimap R) \multimap Q) \multimap Q) \\
&\vdash \ ((((R \multimap R) \multimap Q) \multimap Q) \multimap P) \multimap P
\end{aligned}
$$

Starting from an axiom of order one, each step adds two to the order. Thus if we had started with n implications, in $n-1$ steps we would by the above process prove a theorem of order $2n-1$.

Negative Literals. Having avoided negation everywhere else it seems a shame to have negative literals in formulas. This is unavoidable if S1 theorems such as $(P \parr P) \multimap (P \parr P)$ are to have S2 counterparts, since such theorems cannot be expressed in the bi-implicational language using only positive literals.

3.7 The Dialectic λ-Calculus

One popular formulation of constructive logic is based on the notion of evidence a for a proposition A, written $a : A$. The Curry-Howard isomorphism of types and propositions reads $a : A$ ambiguously as an element a of type

A, and as evidence a for proposition A. It further reads $A \times B$ ambiguously as the product of types A and B, and as conjunction of propositions; thus evidence a for A and b for B constitutes evidence (a, b) for the conjunction $A \times B$. Similarly $A \to B$ is read as the function space from A to B and the implication of B by A. Evidence for an implication $A \to B$ takes the form of a function $f : A \to B$ which given evidence a for A produces evidence $f(a)$ for B.

Proofs as evidence for theorems may in a suitable setting be identified with closed terms of the simply-typed λ-calculus. For example the closed term $\lambda(a, b) : A \times B \;.\; (b, a) : B \times A$ proves the theorem $A \times B \;\to\; B \times A$ while $\lambda a : A \;.\; \lambda b : B \;.\; a : A$ proves $A \to (B \to A)$.

From the viewpoint of System S2 above, the λ-calculus has the limitation that the direction of $f : A \to B$ is always from A to B. This is not compatible with switching semantics, which capriciously chooses a direction for every $\parr$. This is where Chu spaces enter the picture. A Chu space consists of not one but two sets A and X, both of which can be thought of evidence. But whereas points $a \in A$ serve as evidence for A, states x of X, the underlying set of $\mathcal{A}^{\perp}$, can be thought of as evidence against A, i.e. evidence for the negation $A^{\perp}$, an interpretation suggested by G. Plotkin [conversation].

Now we could write $x : A^{\perp}$ but this requires writing $A^{\perp}$ in the rules, which we would like to avoid as not matching up well to S2. Instead we shall introduce a new notation $x \cdot\cdot A$, dual to $a : A$, expressing that x is evidence *against* A, permitting us to avoid saying "evidence for $A^{\perp}$." In the Chu space interpretation of a proposition A as a Chu space $\mathcal{A} = (A, r, X)$, evidence a for A is a point of $\mathcal{A}$ while evidence x against A is a state of $\mathcal{A}$.

We realize evidence for $A \multimap B$ as an adjoint pair $(f; f')$ of functions, one mapping evidence for A to evidence for B, the other mapping evidence against B to evidence against A. (Abbreviating $(f; f')$ to f is permitted; the use of semicolon instead of comma avoids the ambiguity that would otherwise arise when say the pair $((f; f'), g)$ is abbreviated to (f, g).) Evidence for $B \multimapinv A$ is then $(f'; f)$, as an application of commutativity. Note this is not the same thing as evidence against $A \multimap B$, i.e. for $A \otimes B^{\perp}$, namely a pair (a, x) consisting of evidence for A and evidence against B.

With Gödel's Dialectica interpretation and the work of de Paiva [dP89a, dP89b] in mind, we call this variant of the simply-typed λ-calculus the *dialectic λ-calculus*. The two language features distinguishing it from the simply-typed λ-calculus, taken to have the usual exponentiation operator $\to$ and also $\times$ for convenience, are a second implication $\leftarrow$, and the notion $x \cdot\cdot A$ of *evidence against*, dual to evidence for, $a : A$.

The *linear* dialectic λ-calculus imposes the additional requirement that every λ-binding binds exactly one variable occurrence in the formula. We distinguish the linear case in the manner of linear logic by writing $\otimes$, $\multimap$, and $\circ\!-$ in place of $\times$, $\rightarrow$, and $\leftarrow$.

We now specify in full the language of the linear dialectic λ-calculus. Examples of all constructs can be found in Table 3 below. Terms are built up from variables $a, b, \ldots, x, y, \ldots$ and types $A, B, \ldots$ using λ-abstraction, application, and pairing. All terms are typed either positively or negatively.

A type is any bi-implicational MLL formula A all of whose literals are positive. The atoms $P, Q, \ldots$ of A constitute its ground types. In the terminology of context-free or BNF grammars, $P, Q, \ldots$ here play the role of terminal symbols or actual type variables while $A, B, \ldots$ serve as nonterminal symbols or type metavariables.

A variable $a, b, \ldots, x, y, \ldots$ of the λ-calculus is either positively typed as in $a : A$ or negatively typed as in $x \cdot\cdot A$. Both positively and negatively typed variables are drawn from the same set of variables, but by convention we will usually use $a, b, c, \ldots$ for positively typed variables and $x, y, z, \ldots$ for negative as an aid to keeping track of the sign of its type.

A positive application MN consists of a pair of terms positively typed respectively $A \multimap B$ and A,[6] and is positively typed B. A negative application MN consists of a pair of terms, with M positively typed $A \circ\!- B$ and N negatively typed A, and is negatively typed B. For linearity M and N must have no free variables in common, either as an occurrence or as λa.

A positive or consistent pair is a term $(M : A, N : B)$ positively typed by $A \otimes B$. A negative or conflicting pair is either a term $(M : A, N \cdot\cdot B)$ negatively typed by $A \multimap B$ or $(M \cdot\cdot A, N : B)$ negatively typed by $A \circ\!- B$. For linearity M and N must have no λ-variables in common, either as an occurrence or as λa. (In consequence of this and the corresponding rule for application, a λ-variable can appear just once in the form λa.)

A positive λ-abstraction is a term $\lambda a : A \,.\, M : B$ positively typed $A \multimap B$, and the variable a must occur in M with positive type A. A negative λ-abstraction is a term $\lambda x \cdot\cdot A \,.\, M \cdot\cdot B$ positively typed $B \circ\!- A$, and the variable x must occur in M with negative type A.

When the variable a of a λ-abstraction is positively typed by a conjunction or negatively typed by an implication (in which case we will have usually written x rather than a), a may be expanded as the pair (a_1, a_2) where the a_i's are variables of the appropriate type and sign depending on

[6] Were we trying to follow noncommutative linear logic more closely we would presumably write positive applications in the reverse order, NM, along with some other order reversals.

A and its sign. This expansion may be applied recursively to the a_i's, as for example in rule A1 of Table 3 below.

When a is positively typed by an implication, a may be written $(a_1; a_2)$ but the a_i's do not have any type of their own independent of that of a. Unlike λ-bound pairs (a, b), λ-bound functions $(a_1; a_2)$ cannot be split up, and the occurrence of the a_i's in M is restricted to either $(a_1; a_2)$ positively typed by the implication A (either $A_1 \multimap A_2$ or $A_1 \circ\!\!- A_2$) or $(a_2; a_1)$ positively typed by the reverse implication (respectively either $A_2 \circ\!\!- A_1$ or $A_2 \multimap A_1$).

This completes the specification of the language of the linear dialectic λ-calculus.

The usual syntactic approach to defining the meaning of any λ-calculus is in terms of reduction rules. To avoid getting too far afield here we shall instead view λ-terms as denoting Chu transforms parametrized by choice of Chu spaces over some fixed alphabet Σ interpreting the ground types. For example, given an interpretation of ground type P as a Chu space $\mathcal{A}$, $\lambda a : P \,.\, a : P$ is the identity function $1_{\mathcal{A}}$ on $\mathcal{A}$. Technically speaking such an interpretation of a λ-term is a natural transformation (more precisely dinatural), but we defer that point of view to the next section since the idea of a parametrized function is natural enough in its own right when represented as a typed λ-term.

We interpret System S2 in the linear dialectic λ-calculus as follows. The i-th atomic implication in an instantiation of axiom T has one of four forms interpreted as follows:

(i) $P_i \multimap P_i$ as $\lambda a_i : P_i \,.\, a_i : P_i$;
(ii) $P_i^\perp \circ\!\!- P_i^\perp$ as $\lambda a_i \cdot\!\cdot P_i^\perp \,.\, a_i \cdot\!\cdot P_i^\perp$;
(iii) $P_i \circ\!\!- P_i$ as $\lambda x_i \cdot\!\cdot P_i \,.\, x_i \cdot\!\cdot P_i$; and
(iv) $P_i^\perp \multimap P_i^\perp$ as $\lambda x_i : P_i^\perp \,.\, x_i : P_i^\perp$.

These constitute the four ways of typing the identity function 1_{P_i} on P_i, which we construe as either a ground type or if P_i is a negative literal then the negation of ground type. All four types are necessary if one wishes to be able to interpret every theorem of S2 in this way.

Interpret T itself as consisting of those n identity functions, associated into pairs of pairs however the conjunctions are associated. For example the instance $(P \multimap P) \otimes ((Q^\perp \circ\!\!- Q^\perp) \otimes (R^\perp \multimap R^\perp))$ of T is interpreted as $(1_P : P \multimap P, (1_Q : Q^\perp \circ\!\!- Q^\perp, 1_R : R^\perp \multimap R^\perp))$.

With this interpretation of the axiom instance as the starting point, interpret successive theorems in a proof by applying the following transformations, each associated with the correspondingly labeled inference rule of S2. In the derivation $A \vdash B$ via rule R, the transformation associated by Table 3 to rule R maps the λ-term interpreting A to that interpreting B.

$A1$	$\lambda((a,b),c) : (A \otimes B) \otimes C \ . \ (a,(b,c)) : A \otimes (B \otimes C)$
$A2$	$\lambda f : (A \otimes B) \multimap C \ . \ \lambda a : A \ . \ \lambda b : B \ . \ f(a,b) : C$
$A2'$	$\lambda f : (A \multimap B) \circ\!\!- C \ . \ \lambda a : A \ . \ \lambda y \cdot\cdot B \ . \ f(a,y) \cdot\cdot C$
$A2''$	$\lambda f : (A \circ\!\!- B) \circ\!\!- C \ . \ \lambda x \cdot\cdot A \ . \ \lambda b : B \ . \ f(x,b) \cdot\cdot C$
$C1$	$\lambda(a,b) : A \otimes B \ . \ (b,a) : B \otimes A$
$C2$	$\lambda(f;f') : A \multimap B \ . \ (f';f) : B \circ\!\!- A$
$C2'$	$\lambda(f;f') : A \circ\!\!- B \ . \ (f';f) : B \multimap A$
D	$\lambda(f,c) : (A \multimap B) \otimes C \ . \ \lambda a : A \ . \ (f(a),c) : B \otimes C$
D'	$\lambda(f,c) : (A \circ\!\!- B) \otimes C \ . \ \lambda x \cdot\cdot A \ . \ (f(x),c) \cdot\cdot B \circ\!\!- C$
$E1$	$\lambda(a,b) : A \otimes B \ . \ (f(a),g(b)) : A' \otimes B'$
$E2$	$\lambda h : A' \multimap B \ . \ ghf : A \multimap B'$
$E3$	$\lambda h : A \circ\!\!- B' \ . \ g'hf' : A' \circ\!\!- B$

Table 3. Transformations Associated to Rules of S2

Rules E1-E3 assume that $A \vdash A'$ is realized by $(f;f') : A \multimap A'$ and $B \vdash B'$ by $(g;g') : B \multimap B'$.

Rule A1 transforms evidence $((a,b),c)$ for $(A \otimes B) \otimes C$ to $(a,(b,c))$ as evidence for $A \otimes (B \otimes C)$. Rule A2 maps the function f witnessing $(A \otimes B) \multimap C$ to the function $\lambda a \ . \ \lambda b \ . \ f(a,b)$ witnessing $A \multimap (B \multimap C)$.

Rule A2' maps witness f for $(A \multimap B) \circ\!\!- C$ to $\lambda a \ . \ \lambda y \ . \ f(a,y)$ which given evidence a for A and y against B, constituting evidence against $A \multimap B$, produces evidence $f(a,y)$ against C.

The remaining rules are interpreted along the same lines.

Theorem 4 *Every theorem of S2 is interpreted by Table 3 as a transformation represented by a closed term of the linear dialectic λ-calculus.*

Proof: This is a straightforward consequence of the form of Table 3. The interpretations of the axiom instances and the rules are in the language, contain no free variables, λ-bind exactly one variable, and are typed compatibly with the rules. Free variables in A, B, C remain free after transformation, by the requirement that all λ-bound variables are distinct. The theorem then follows by induction on the length of Π. ∎

It is a nice question to characterize those terms of the linear dialectic λ-calculus for which the converse holds: every closed term of the linear dialectic λ-calculus meeting that characterization interprets some theorem. Taking this a step further, a calculus with reduction rules should permit a notion of normal form permitting a strenthening of the above theorem to a bijection between certain terms in normal form and cut-free proofs.

3.8 Transformational Semantics

The dialectic λ-calculus has provided a syntactic connection with Chu spaces by depending on its mixture of points and states as positive and negative evidence, and moreover has furnished us with a potentially useful library of transformations of Chu spaces, namely those defined by Table 3.

The trouble with such syntactically defined classes of transformations is that it is easy to imagine extending the class by extending the language with whatever operations we might have overlooked. The intrinsic interest in the class would be more compelling if it had a language-independent definition, such that the operations of our language constituted a complete basis for that class, as with conjunction, negation, and the constant 0 for the class of Boolean operations.

The appropriate semantic notion is *naturality*, in which the interpretation of a proof of A as an element of the interpretation of A is required to remain "constant" as the latter varies in response to variations in the atoms of A. For example $\lambda a : A \;.\; a : A$ is in an intuitive sense "constantly" the identity function, yet its domain and codomain must necessarily track A as it ranges over different sets, groups, or whatever category we are working in.

Naturality is formalized by interpreting terms A, B as functors $F_A, F_B : D \to C$ from a category D of values for variables appearing in terms to a category C of values for terms (where D will for us be C^n where n is the number of variables), and entailments $A \vdash B$ as natural transformations $\tau : F_A \to F_B$. A transformation is a "variable morphism" varying over D, defined as a family of morphisms $\tau_x : F_A(a) \to F_B(a)$ of C indexed by objects a of D.

A transformation is *natural* when for all morphisms $f : a \to b$ of D, the diagram on the left of the following figure commutes.

$$
\begin{array}{ccc}
F(a) & \xrightarrow{\tau_a} & G(a) \\
{\scriptstyle F(f)}\downarrow & & \downarrow{\scriptstyle G(f)} \\
F(b) & \xrightarrow{\tau_b} & G(b)
\end{array}
\qquad\qquad
\begin{array}{ccc}
F(a,a) & \xrightarrow{\tau_a} & G(a,a) \\
{\scriptstyle F(f,a)}\uparrow & & \downarrow{\scriptstyle G(a,f)} \\
F(b,a) & & G(a,b) \\
{\scriptstyle F(b,f)}\downarrow & & \uparrow{\scriptstyle G(f,b)} \\
F(b,b) & \xrightarrow{\tau_b} & G(b,b)
\end{array}
$$

This basic notion of naturality is only defined for terms that are (covariantly) functorial in their variables, i.e. for variables all of whose occurrences have the same sign. (In linear logic a variable P all of whose occurrences are negative may be replaced by a negated variable $Q^\perp$; the new variable Q

then occurs only positively and the resulting formula is no less general.) The diagram on the right generalizes naturality to the case of mixed variance, where the same variable may occur both positively and negatively, as in the entailment $A \multimap A \vdash A^\perp \multimap A^\perp$. Here $F(a', a)$ separates the negative occurrences a' from the positive occurrences a. When this diagram commutes for all morphisms $f : a \to b$ of D τ is called *dinatural.*

Elsewhere [Pra97] we have shown that the dinatural transformations between functors in Chu built up with tensor and perp soundly interpret MLL in that every cut-free MLL proof has a distinct interpretation as a dinatural transformation. We showed furthermore that for Binary MLL, that fragment of MLL having one positive and one negative occurrence of each variable, dinatural transformations are a complete interpretation in that every dinatural transformation interprets some cut-free proof. Here we show that the latter result does not extend to formulas containing four occurrences of a variable.

Theorem 5 *Both big* $\mathbf{Chu}(\mathbf{Set}, 2)$ *and little* $\mathbf{chu}(\mathbf{Set}, 2)$ *contain spurious dinaturals on* $A \multimap A$*, in the sense that such dinaturals correspond to no MLL proof of* $(A \multimap A) \multimap (A \multimap A)$.

Proof: Ideally there would be one proof for both big and little **Chu**, using biextensional spaces for the counterexample but without depending on biextensionality in the argument. Unfortunately we have not found such a proof and have different proofs for each case. We begin with big **Chu**.

Call a Chu space *inconsistent* when it has no states, and consistent otherwise. Let $\tau : I \multimap I$ be the transformation that, at each inconsistent space $\mathcal{A}$, is the functional on $\mathcal{A} \multimap \mathcal{A}$ taking $f : \mathcal{A} \multimap \mathcal{A}$ to the identity function, and to f otherwise. Such a τ is not among the transformations corresponding to MLL proof nets and is therefore spurious. Its dinaturality is easily verified for $\mathcal{A}, \mathcal{B}$ both consistent or both inconsistent. (When both are inconsistent, so is $\mathcal{A} \multimap \mathcal{A}$, and furthermore the Chu transforms constituting the points of $\mathcal{A} \multimap \mathcal{A}$ are all functions on the carrier A.)

When $\mathcal{A}$ is inconsistent and $\mathcal{B}$ is consistent there are no test morphisms from $\mathcal{A}$ to $\mathcal{B}$ to witness failure of dinaturality.

When $\mathcal{A}$ is consistent and $\mathcal{B}$ is inconsistent, $\mathcal{B} \multimap A$ has no points while $\mathcal{A} \multimap \mathcal{B}$ has no states, whence there is a unique Chu morphism from $\mathcal{B} \multimap \mathcal{A}$ to $\mathcal{A} \multimap \mathcal{B}$, forcing the dinaturality diagram to commute. Hence τ is dinatural.

We now prove the corresponding theorem for little **chu**. The analogue of the inconsistent spaces for this case will be those biextensional Chu spaces which contain both a row of all zeros and a column of all zeros. Call these the Type I spaces, and the rest the Type II spaces.

Claim: $\mathcal{A} \multimap \mathcal{A}$ is Type I if and only if $\mathcal{A}$ is.

(If) We need to show that $\mathcal{A} \multimap \mathcal{A}$ has both a zero row, namely a constantly zero function, and zero column, namely a pair (a, x) in $A \times X$ at which every function is zero. The former is representable because $\mathcal{A}$ (in its role as target of $\mathcal{A} \multimap \mathcal{A}$) has a constantly zero row, and is continuous because $\mathcal{A}$ in the role of source has a constantly zero column. The latter follows by taking a to be the point indexing $\mathcal{A}$'s zero row, and x the state indexing its zero column.

(Only if) If $\mathcal{A} \multimap \mathcal{A}$ is of Type I then it has a zero row, i.e. a zero function, possible only if $\mathcal{A}$ as source has a zero column and as target a zero row. Hence $\mathcal{A}$ must also be Type I.

We now define the transformation $\tau_{\mathcal{A}}$ at Type I spaces $\mathcal{A}$ to be the constantly zero function on $\mathcal{A} \to \mathcal{A}$, and at all other Chu spaces the identity function on $\mathcal{A} \to \mathcal{A}$, both easily seen to be continuous by the above claim. This τ is spurious, again because it corresponds to no MLL proof net.

For dinaturality, observe that the dinaturality hexagon commutes for any pair $\mathcal{A}, \mathcal{B}$ of Chu spaces of the same type and for any Chu transform $f : \mathcal{A} \to \mathcal{B}$. For $\mathcal{A}, \mathcal{B}$ of Type I this is because going round the hexagon either way yields the zero row. For Type II it is because f commutes with identities.

When $\mathcal{A}$ and $\mathcal{B}$ are of opposite types, one of the two homsets $\mathrm{Hom}(\mathcal{A}, \mathcal{B})$ or $\mathrm{Hom}(\mathcal{B}, \mathcal{A})$ must be empty, the former when $\mathcal{A}$ lacks a zero column or $\mathcal{B}$ lacks a zero row, the latter when it is $\mathcal{B}$ that lacks the zero column or $\mathcal{A}$ that lacks the zero row. If $\mathrm{Hom}(\mathcal{A}, \mathcal{B})$ is empty then there can be no hexagon because there is no $f : \mathcal{A} \to \mathcal{B}$. If $\mathrm{Hom}(\mathcal{B}, \mathcal{A})$ is empty then the hexagon commutes vacuously, its starting object being empty. We have thus shown that τ is dinatural.

∎

We conclude that dinaturality semantics is too weak for full completeness of Chu semantics in MLL. This weakness of dinaturality has already been observed in other contexts, e.g. **Set** as a model of intuitionistic logic where $A \to A \vdash A \to A$ also has spurious dinaturals as pointed out by Paré and Román [PR98]. Constructive intuitionistic logic has as the proofs from $A \to A$ (the set of all functions on the set A) to itself just the Church numerals, those functions sending $f : A \to A$ to $f^n : A \to A$ for some fixed natural number n independent of A. However the spurious transformation sending f in $A \to A$ to $f^{|A|!!}$ can be seen to be dinatural in the category **FinSet** of finite sets, though not in the category **Pos** of posets.

A stronger criterion than dinaturality is invariance under logical relations [Plo80] instead of just morphisms. In collaboration with H. Devarajan, D.

Hughes, and G. Plotkin we have shown full completeness of MLL for binary logical transformations over Chu spaces.

4 Relationship of Chu Spaces to Mathematics

4.1 Relational Structures

A relational structure of a given similarity type or *signature* (m, n, σ) is an m-tuple of sets $A_1, A_2, \ldots, A_m$ together with an n-tuple of relations $R_1, \ldots, R_n$ where each $R_j \subseteq A_{\sigma(j,1)} \times A_{\sigma(j,2)} \times \ldots A_{\sigma(j,\alpha(j))}$ is a subset of $\alpha(j)$ A_i's determined by σ. These are the models standardly used in first order logic, typically with $m = 1$, the homogeneous case. A *homomorphism* between two structures $\mathcal{A}, \mathcal{A}'$ with the same signature is an m-tuple of functions $f_i : A_i \to A'_i$ such that for each $1 \leq j \leq n$ and for each $(a_1, \ldots, a_{\alpha(j)}) \in R_j$ of $\mathcal{A}$, $(f(a_1), \ldots, f(a_{\alpha(j)})) \in R'_j$. The class of relational structures having a given signature together with the class of homomorphisms between them form a category.

There is no loss of generality in restricting to homogeneous (singlesorted) structures because the carriers of a heterogeneous structure may be combined with disjoint union to form a single carrier. The original sorts can be kept track of by adding a new unary predicate for each sort which is true just of the members of that sort. This ensures that homomorphisms remain type-respecting.

There is also no loss of generality in restricting to a single relation since the structural effect of any family of nonempty relations can be realized by the natural join of those relations, of arity the sum of the arities of the constituent relations. A tuple of the composite relation can then be viewed as the concatenation of tuples of the constituent relations. The composite relation consists of those tuples each subtuple of which is a tuple of the corresponding constituent relation.

This reduces our representation problem for relational structures to that of finding a Chu space to represent the structure (A, R) where $R \subseteq A^n$ for some ordinal n. The class $\mathbf{Str}_n$ of all such n-ary relational structures (A, R) is made a category by taking as its morphisms all homomorphisms between pairs $(A, R), (A', R')$ of such structures, defined as those functions $f : A \to A'$ such that for all $(a_1, \ldots, a_n) \in R$, $(f(a_1), \ldots, f(a_n)) \in R'$. Every category whose objects are (representable as) n-ary relational structures and whose morphisms are all homomorphisms between them is a full subcategory of $\mathbf{Str}_n$. For example the category of groups and group homomorphisms is a full subcategory of $\mathbf{Str}_3$, since groups are fully and faithfully represented by the ternary relation $ab = c$.

We represent (A, R) as the Chu space (A, r, X) over 2^n (subsets of $n = \{0, 1, \ldots, n-1\}$) where (i) $X \subseteq (2^A)^n \cong (2^n)^A$ consists of those n-tuples $(x_1, \ldots, x_n)$ of subsets of A such that every $(a_1, \ldots, a_n) \in R$ is *incident on* $(x_1, \ldots, x_n)$ in the sense that there exists i for which $a_i \in x_i$; and (ii) $r(a, x) = \{i | a \in x_i\}$.

This representation is *concrete* in the sense that the representing Chu space has the same carrier as the structure it represents.

Theorem 6 *A function $f : A \to B$ is a homomorphism between (A, R) and (B, S) if and only if it is a continuous function between the respective representing Chu spaces (A, r, X) to (B, s, Y).*

Proof: $(\to)$ For a contradiction let $f : A \to B$ be a homomorphism which is not continuous. Then there must exist a state $(y_1, \ldots, y_n)$ of $\mathcal{B}$ for which $f^{-1}(y_1), \ldots, f^{-1}(y_n))$ is not a state of $\mathcal{A}$. Hence there exists $(a_1, \ldots, a_n) \in R$ for which $a_i \notin f^{-1}(y_i)$ for every i. But then $f(a_i) \notin y_i$ for every i, whence $(f(a_1), \ldots, f(a_n)) \notin S$, impossible because f is a homomorphism.

$(\leftarrow)$ Suppose f is continuous. Given $(a_1, \ldots, a_n) \in R$ we shall show that $(f(a_1), \ldots, f(a_n)) \in S$. For if not then $(\overline{\{f(a_1)\}}, \ldots, \overline{\{f(a_n)\}})$ is a state of $\mathcal{B}$. Then by continuity, $(f^{-1}(\overline{\{f(a_1)\}}), \ldots, f^{-1}(\overline{\{f(a_n)\}}))$ is a state of $\mathcal{A}$. Hence for some i, $a_i \in f^{-1}(\overline{\{f(a_i)\}})$, i.e. $f(a_i) \in \overline{\{f(a_i)\}}$, which is impossible. ∎

As an example, groups as algebraic structures determined by a carrier and a binary operation can also be understood as ternary relational structures. Hence groups can be represented as Chu spaces over 8 (subsets of $\{0, 1, 2\}$) as above, with the continuous functions between the representing Chu spaces being exactly the group homomorphisms between the groups they represent.

The above theorem can be restated in categorical language as follows. Any full subcategory C of the category of n-ary relational structures and their homomorphisms embeds fully and concretely in $\mathbf{Chu}(\mathbf{Set}, 2^n)$. That is, there exists a full and faithful functor $F : C \to \mathbf{Chu}(\mathbf{Set}, 2^n)$ such that $FU = U'F$ where $U : C \to \mathbf{Set}$ and $U' : \mathbf{Chu}(\mathbf{Set}, 2^n)$ are the respective forgetful functors.

4.2 Topological Relational Structures

A natural generalization of this representation is to topological relational structures (A, R, O), where $R \subseteq A^n$ and $O \subseteq 2^A$ is a set of subsets of A constituting the open sets of a topology on A. (R itself may or may not be continuous with respect to O in some sense, but this is immaterial here.)

Such a structure has a straightforward representation as a Chu space over 2^{n+1}, as follows. Take $X = X' \times O$ where $X' \subseteq (2^A)^n$ is the set of states on A determined by R as in the previous subsection. Hence $X \subseteq (2^A)^{n+1}$. With this new representation the continuous functions will remain homomorphisms with respect to R, but in addition they will be continuous in the ordinary sense of topology with respect to the topology O.

For example topological groups can be represented as Chu spaces over 16.

This is an instance of a more general technique for combining two structures on a given set A. Let (A, r, X_1) and (A, s, X_2) be Chu spaces over Σ_1, Σ_2 respectively, having carrier A in common. Then $(A, t, X_1 \times X_2)$ is a Chu space over $\Sigma_1 \times \Sigma_2$, where $t(a, (x_1, x_2)) = (r(a, x_1), s(a, x_2))$.

If now $(A', t', X'_1 \times X'_2)$ is formed from (A', r', X'_1) over Σ_1 and (A', s', X'_2) over Σ_2, then $f : A \to A'$ is a continuous function from $(A, t, X_1 \times X_2)$ to $(A', t', X'_1 \times X'_2)$ if and only if it is continuous from (A, r, X_1) to (A', r', X'_1) and also from (A, s, X_2) to (A', s', X'_2). For if (f, g_1) and (f, g_2) are the latter two Chu transforms, with $g_1 : X'_1 \to X_1$ and $g_2 : X'_2 \to X_2$, then the requisite $g : X'_1 \times X'_2 \to X_1 \times X_2$ making (f, g) an adjoint pair is simply $g(x_1, x_2) = (g_1(x_1), g_2(x_2))$. The adjointness condition is then immediate.

4.3 Concretely Embedding Small Categories

The category of n-ary relational structures and their homomorphisms is a very uniformly defined concrete category. It is reasonable to ask whether the objects of less uniformly defined concrete categories can be represented as Chu spaces. The surprising answer is that $\mathbf{Chu}(\mathbf{Set}, \Sigma)$ fully and concretely embeds *every* concrete category C of cardinality (total number of elements of all objects, which are assumed disjoint) at most that of Σ, no matter how arbitrary its construction, save for one small requirement, that objects with empty underlying set be initial in C.

We begin with a weaker embedding theorem not involving concreteness which in effect combines the two Yoneda embeddings, namely the embedding of C into $\mathbf{Set}^{C^\circ}$ and of C° into $\mathbf{Set}^C$. The difference is that whereas these two targets of the Yoneda embeddings depend nontrivially on the structure of C, that of our embedding depends only on the cardinality $|C|$, the number of arrows of C.

This theorem shows off to best effect the relationship between Chu structure and category structure, being symmetric with respect to points and states. The stronger concrete embedding that follows modifies this proof only slightly but enough to break the appealing symmetry.

Theorem 7 *Every small category C embeds fully in* $\mathbf{Chu}(\mathbf{Set}, |C|)$.

Proof:

Define the functor $F : C \to \mathbf{Chu}(\mathbf{Set}, |C|)$ as $F(b) = (A, r, X)$ where $A = \{f : a \to b \mid a \in ob(C)\}$, $X = \{h : b \to c \mid c \in ob(C)\}$, and $r(f, h) = hf = f; h$, the converse of composition. That is, the points of this space are all arrows into b, its states are all arrows out of b, and the matrix entries $f; h$ are all composites $a \xrightarrow{f} b \xrightarrow{h} c$ of inbound arrows with outbound.

(A, r, X) is separable because X includes the identity morphism 1_b, for which we have $s(f, 1_b) = f; 1_b = f$, whence $f \neq f'$ implies $s(f, 1_b) \neq s(f', 1_b)$. Likewise A includes 1_b and the dual argument shows that (A, r, X) is extensional.

For morphisms take $F(g : b \to b')$ to be the pair (φ, ψ) of functions $\varphi : A \to A'$, $\psi : X' \to X$ defined by $\varphi(f) = f; g$, $\psi(h) = g; h$. This is a Chu transform because the adjointness condition $\varphi(f); h = f; \psi(h)$ for all $f \in A$, $h \in X'$ has $f; g; h$ on both sides. In fact the condition expresses associativity and no more.

To see that F is faithful consider $g, g' : b \to b'$. Let $F(g) = (\varphi, \psi)$, $F(g') = (\varphi', \psi')$. If $F(g) = F(g')$ then $g = 1_b; g = \varphi(1_b) = \varphi'(1_b) = 1_b; g' = g'$.

For fullness, let (φ, ψ) be any Chu transform from $F(b)$ to $F(b')$. We claim that (φ, ψ) is the image under F of $\varphi(1_b)$. For let $F(\varphi(1_b)) = (\varphi', \psi')$. Then $\varphi'(f) = f; \varphi(1_b) = f; \varphi(1_b); 1_{b'} = f; 1_b; \psi(1_{b'}) = f; \psi(1_{b'}) = \varphi(f)$, whence $\varphi' = \varphi$. Dually $\psi' = \psi$. ∎

Comparing this embedding with the covariant Yoneda embedding of C in $\mathbf{Set}^{C^\circ}$, we observe that the latter realizes φ_g directly while deferring ψ_g via the machinery of natural transformations. The contravariant embedding, of C in $(Set^C)^\circ$ (i.e. of C° in $\mathbf{Set}^C$) is just the dual of this, realizing ψ_g directly and deferring φ_g. Our embedding in Chu avoids functor categories altogether by realizing both simultaneously.

The adjointness condition can be more succinctly expressed as the dinaturality in b of composition $m_{abc} : C(a, b) \times C(b, c) \to C(a, c)$. The absence of b from $C(a, c)$ collapses the three nodes of the right half of the dinaturality hexagon to one, shrinking it to the square

$$\begin{array}{ccc}
C(a,b) \times C(b',c) & \xrightarrow{1\times\psi_g} & C(a,b) \times C(b,c) \\
\varphi_g \times 1 \downarrow & & \downarrow m_{abc} \\
C(a,b') \times C(b',c) & \xrightarrow{m_{ab'c}} & C(a,c)
\end{array}$$

Here $1 \times \psi_g$ abbreviates $C(a,b) \times C(g,c)$ and $\varphi_g \times 1$ abbreviates $C(a,g) \times C(b',c)$. Commutativity of the square asserts $\varphi_g(f); h = f; \psi_g(h)$ for all $f : a \to b$ and $h : b' \to c$. By letting a and c range over all objects of C we extend this equation to the full force of the adjointness condition for the Chu transform representing g.

This embedding is concrete with respect to the forgetful functor which takes the underlying set of b to consist of the arrows to b. From that perspective it is a special case of the following, which allows the forgetful functor to be almost arbitrary. The one restriction we impose is that objects of C with empty underlying set be initial. When this condition is met we say that C is *honestly concrete*.

Theorem 8 *Every small honestly concrete category* (C, U) *embeds fully and concretely in* $\mathbf{Chu}(\mathbf{Set}, \sum_{b \in ob(C)} U(b))$.

Here the alphabet Σ is the disjoint union of the underlying sets of the objects of C. In the previous theorem the underlying sets were disjoint by construction and their union consisted simply of all the arrows of C. Now it consists of all the elements of C marked by object of origin.

Proof: Without loss of generality assume that the underlying sets of distinct objects of C are disjoint. Then we can view Σ as simply the set of all elements of objects of C. Modify $F(b) = (A, r, X)$ in the proof of the preceding theorem by taking $A = U(b)$ instead of the set of arrows to b. When $U(b) \neq \emptyset$ take X as before, otherwise take it to be just $\{1_b\}$. Lastly take $r(f,h) = U(h)(f)$ where $f \in U(b)$, i.e. application of concrete $U(h)$ to f instead of composition of abstract h with f. (We stick to the name f, even though it is no longer a function but an element, to reduce the differences from the previous proof to a minimum.)

(A, r, X) is separable for the same reason as before. For extensionality there are three cases. When $U(b) = \emptyset$ we forced extensionality by taking X to be a singleton. Otherwise, for $h \neq h' : b \to c$, i.e. having the same codomain, U faithful implies that $U(h)$ and $U(h')$ differ at some $f \in U(b)$. Finally, for $h : b \to c$, $h' : b \to c'$ where $c \neq c'$, *any* $f \in U(b)$ suffices to distinguish $U(h)(f)$ from $U(h')(f)$ since $U(c)$ and $U(c')$ are disjoint.

For morphisms take $F(g : b \to b')$ to be the pair (φ, ψ) of functions $\varphi : A \to A'$, $\psi : X' \to X$ defined by $\varphi(f) = U(g)(f)$, $\psi(h) = U(hg)$. This is a Chu transform because the adjointness condition $U(h)(\varphi(f)) = U(\psi(h))(f)$ for all $f \in A$, $h \in X'$ has $U(hg)(f)$ on both sides.

This choice of φ makes $\varphi = U(g)$, whence F is faithful simply because U is.

For fullness, let (φ, ψ) be any Chu transform from $F(b)$ to $F(b')$. We break this into two cases.

(i) $U(b)$ empty. In this case there is only one Chu transform from $F(b)$ to $F(b')$, and by honesty there is one from b to b', ensuring fullness.

(ii) $U(b)$ nonempty. We claim that $\psi(1_{b'})$ is a morphism $g : b \to b'$, and that $F(g) = (\varphi, \psi)$. For the former, $\psi(1_{b'})$ is a state of $F(b)$ and hence a map from b. Let $f \in U(b)$. By adjointness $U(1_{b'})(\varphi(f)) = U(\psi(1_{b'}))(f)$ but the left hand side is an element of $U(b')$ whence $\psi(1_{b'})$ must be a morphism to b'.

Now let $F(\psi(1_{b'})) = (\varphi', \psi')$. Then for all $f \in U(b)$, $U(\psi'(h))(f) = U(h \circ \psi(1_{b'}))(f) = U(h)(U(\psi(1_{b'}))(f)) = U(h)(U(1_{b'})(\varphi(f))) = U(h)(\varphi(f))$ $= U(\psi(h))(f)$. Hence $U(\psi'(h)) = U(\psi(h))$. Since U is faithful, $\psi'(h) = \psi(h)$. Hence $\psi' = \psi$. Since $F(b)$ is separable, $\varphi' = \varphi$. ∎

4.4 Homomorphism = Continuous, generalized

Theorem 6 identified homomorphisms between relational structures with continuous functions between their respective representations as Chu spaces. Here we extend this result to arbitrary extensional Chu spaces by defining a homomorphism of Chu spaces to be a property-preserving function for a suitably abstract notion of property of a Chu space.

Fix an alphabet Σ. For any set A we define a Σ-*preproperty* of A, or just preproperty when context determines Σ, to be a subset of Σ^A. In particular Σ^A constitutes the identically true preproperty of A, while the empty set is its identically false preproperty. The Σ-preproperties of A form a power set, namely 2^{Σ^A}, and as such a complete atomic Boolean algebra, with intersection and union as respectively conjunction and disjunction.

We refer to the 2-preproperties of A, those for which $\Sigma = 2 = \{0, 1\}$, as its *logical* preproperties. These are just the ordinary Boolean propositions over A construed as a set of variables, of which there are $2^{2^{|A|}}$. Each proposition is represented by the set of its satisfying assignments, those assignments of truth values to variables that make the proposition true, with each assignment being represented by the set of variables assigned *true*. Thus if $A = \{a, b, c\}$ then the preproperty $\{\{a, b\}, \{a, b, c\}\}$ represents the proposition $a \wedge b$, and can be transliterated directly as the disjunctive normal form (DNF) proposition $(a \wedge b \wedge \overline{c}) \vee (a \wedge b \wedge c)$, having one disjunct per satisfying assignment in which every variable in A occurs exactly once. The logical preproperties of A constitute the free Boolean algebra 2^{2^A} on A.

All this extends to infinite A we provided we qualify "Boolean algebra" with "complete atomic" and allow infinite DNF propositions. This qualification was not needed for finite A because it is vacuous in that case.

Given an arbitrary Chu space $\mathcal{A} = (A, r, X)$ we define a *property* of $\mathcal{A}$ to be a preproperty of A containing every column of $\mathcal{A}$. As such the properties of $\mathcal{A}$ form the principal filter generated by the set $\check{r}(X)$ of columns of $\mathcal{A}$, which itself is the strongest property of $\mathcal{A}$, implying all other properties of $\mathcal{A}$ and also being the conjunction of those properties. This filter is the power set $2^{\Sigma^A - X}$, and is a sublattice of the Boolean algebra 2^{Σ^A} though not a subBoolean algebra of it.

To every function $f : A \to B$ we associate a function $\acute{f}$: $2^{\Sigma^A} \to 2^{\Sigma^B}$ defined as $\acute{f}(Y) = \{g : B \to \Sigma \mid gf \in Y\}$ for $Y \subseteq \Sigma^A$. To avoid notational clutter we think of $\acute{f}$ as merely an extension of f and say that f sends Y to $\acute{f}(Y)$. A *homomorphism* of Chu spaces $\mathcal{A} = (A, r, X)$, $\mathcal{B} = (B, s, Y)$ is a function $f : A \to B$ such that $\acute{f}(\check{r}(X)) \supseteq \check{s}(Y)$. This is equivalent to requiring that f send properties of $\mathcal{A}$ to properties of $\mathcal{B}$, justifying the term "homomorphism." As further clutter control, without loss of generality we restrict attention in this section to normal Chu spaces, which simplifies the above condition to $\acute{f}(X) \subseteq Y$.

Theorem 9 *A function $f : A \to B$ is a homomorphism from $\mathcal{A}$ to $\mathcal{B}$ if and only if it is a continuous function from $\mathcal{A}$ to $\mathcal{B}$.*

Proof: The function $f : \mathcal{A} \to \mathcal{B}$ is a homomorphism if and only if $f(X) \supset Y$, if and only if every $g : B \to \Sigma$ in Y satisfies $gf \in X$, if and only if f is continuous. ∎

We may relate this result to Theorem 6 by identifying an n-tuple with the property whose states are those n-tuples $(x_1, \ldots, x_n)$ on which that n-tuple is incident.

5 Conclusion

We have exhibited embeddings in **Chu** of two quite different notions of "general" category. One is that of relational structures and their homomorphisms, possibly with topological structure and the requirement that the homomorphisms be continuous. The other is that of an arbitrary small category, possibly concrete. Both embeddings are concrete. The first is concrete in the ordinary sense of the representing object (A, r, X) having as its underlying set A the carrier of the represented relational structure. The second is concrete with respect to arrows-to as elements, or to the underlying sets of the objects when these are provided.

Quite a few categories are known that are universal to the extent of fully embedding all small categories, as well as all algebraic categories. However those embeddings are highly artificial, relying on the ability of such objects as graphs and semigroups to code the compositional structure of morphisms that compose at an object to be so represented. Any representation based on clever coding introduces irrelevant complexity into the mathematics of objects so represented. Furthermore the coding obscures the ordinary elements of concrete objects, further undermining our intuitions about concrete objects.

These embeddings provide a sense in which the denotational semantics of linear logic can be understood to be at least as general as that of first-order logic. This is not to say that the generality is achieved at the same level. A model of first order logic is a relational structure, and the models of a given theory form a category. A model of linear logic on the other hand is the category itself, whose objects are the denotations of mere formulas.

This is the basic difference between first order or elementary logic and linear logic. First order logic reasons about the interior of a single object, the domain of discourse being the elements or individuals that exist in that object together with the relationships that hold between them. Linear logic reasons instead about how things appear on the outside, understanding the structure of objects externally in terms of how they interact rather than internally in terms of what they might contain. The fundamental interaction is taken to be that of transformation of one object into another. Elements and their relationships are not discussed explicitly, but their existence and nature is inferred from how the objects containing them interact.

This being the essence of the categorical way of doing mathematics, linear logic so construed must therefore be the categorical logic of general mathematics. As such it is sibling to intuitionistic categorical logic, whose domain of discourse is confined to cartesian closed mathematics, having as its exemplar category **Set**. The thesis we have presented here is that the examplar category of general mathematics is $\mathbf{Chu}(\mathbf{Set}, -)$.

Acknowledgments. The program realized in Section 4 was suggested to us by Michael Barr. The interpretation of states of a Chu space as evidence against a proposition was suggested to us by Gordon Plotkin.

References

[Abr90] V. Michele Abrusci. Non-commutative intuitionistic linear propositional logic. *Zeitschrift für Mathematische Logik und*

Grundlagen der Mathematik, 36:297–318, 1990.

[AJ92] S. Abramsky and R. Jagadeesan. Games and full completeness for multiplicative linear logic. In *Foundations of Software Technology and Theoretical Computer Science*, Lecture Notes in Computer Science, New Delhi, 1992. Springer-Verlag. Also Imperial College Report DoC 92/24.

[AJ94] Samson Abramsky and Radha Jagadeesan. Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic*, 59(2):543–574, 1994.

[Bar79] M. Barr. *∗-Autonomous categories*, volume 752 of *Lecture Notes in Mathematics*. Springer-Verlag, 1979.

[Bar91] M. Barr. ∗-Autonomous categories and linear logic. *Math Structures in Comp. Sci.*, 1(2):159–178, 1991.

[Bar96] M. Barr. Fuzzy models of linear logic. *Mathematical Structures in Computer Science*, 6(3):301–312, 1996.

[Bla92] Andreas Blass. A game semantics for linear logic. *Annals of Pure and Applied Logic*, 56:183–220, 1992.

[Blu96] R. F. Blute. Hopf algebras and linear logic (with appendix by M. Barr). *Mathematical Structures in Computer Science*, 6(2):189–217, April 1996.

[BS96a] R.F. Blute and P.J. Scott. Linear Läuchli semantics. *Annals of Pure and Applied Logic*, 77:101–142, 1996.

[BS96b] R.F. Blute and P.J. Scott. The shuffle Hopf algebra and noncommutative full completeness. *Submitted to J. Symbolic Logic*, 1996.

[BvN36] G. Birkhoff and J. von Neumann. The logic of quantum mechanics. *Annals of Mathematics*, 37:823–843, 1936.

[CCMP91] R.T Casley, R.F. Crew, J. Meseguer, and V.R. Pratt. Temporal structures. *Math. Structures in Comp. Sci.*, 1(2):179–213, July 1991.

[DHPP99] H. Devarajan, D. Hughes, G. Plotkin, and V. Pratt. Full completeness of the multiplicative linear logic of chu spaces. In *Proc.*

14th Annual IEEE Symp. on Logic in Computer Science, pages 234–243, Trento, Italy, July 1999.

[dP89a] V. de Paiva. The dialectica categories. In *Categories in Computer Science and Logic*, volume 92 of *Contemporary Mathematics*, pages 47–62, held June 1987, Boulder, Colorado, 1989.

[dP89b] V. de Paiva. A dialectica-like model of linear logic. In *Proc. Conf. on Category Theory and Computer Science*, volume 389 of *Lecture Notes in Computer Science*, pages 341–356, Manchester, September 1989. Springer-Verlag.

[DR89] V. Danos and L. Regnier. The structure of multiplicatives. *Archive for Mathematical Logic*, 28:181–203, 1989.

[Gir87] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[Gir89] J.-Y. Girard. Towards a geometry of interaction. In *Categories in Computer Science and Logic*, volume 92 of *Contemporary Mathematics*, pages 69–108, held June 1987, Boulder, Colorado, 1989.

[HL69] Z. Hedrlín and J. Lambek. How comprehensive is the category of semigroups. *J. Algebra*, 11:195–212, 1969.

[HO93] J.M.E. Hyland and C.-H.L. Ong. Fair games and full completeness for multiplicative linear logic without the mix-rule. Available by ftp from ftp.comlab.ox.ac.uk as fcomplete.ps.gz in /pub/Documents/techpapers/Luke.Ong, 1993.

[LS86] J. Lambek and P. Scott. *Introduction to Higher-Order Categorical Logic.* Cambridge University Press, 1986.

[LS91] Y. Lafont and T. Streicher. Games semantics for linear logic. In *Proc. 6th Annual IEEE Symp. on Logic in Computer Science*, pages 43–49, Amsterdam, July 1991.

[Mac45] G. Mackey. On infinite dimensional vector spaces. *Trans. Amer. Math. Soc.*, 57:155–207, 1945.

[Mac71] S. Mac Lane. *Categories for the Working Mathematician.* Springer-Verlag, 1971.

[Plo80] G.D. Plotkin. Lambda definability in the full type hierarchy. In *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 363–373. Academic Press, 1980.

[PR98] R. Paré and L. Román. Dinatural numbers. *J. Pure and Applied Algebra*, 128:33–92, 1998.

[Pra86] V.R. Pratt. Modeling concurrency with partial orders. *Int. J. of Parallel Programming*, 15(1):33–71, February 1986.

[Pra93] V.R. Pratt. The second calculus of binary relations. In *Proceedings of MFCS'93*, volume 711 of *Lecture Notes in Computer Science*, pages 142–155, Gdańsk, Poland, 1993. Springer-Verlag.

[Pra95] V.R. Pratt. The Stone gamut: A coordinatization of mathematics. In *Logic in Computer Science*, pages 444–454. IEEE Computer Society, June 1995.

[Pra97] V.R. Pratt. Towards full completeness of the linear logic of chu spaces. In *Electronic Notes in Theoretical Computer Science*, volume 6, Pittsburgh, 1997. URL: http://www.elsevier.nl/locate/entcs/volume6.html, 18 pages.

[PT80] A. Pultr and V. Trnková. *Combinatorial, Algebraic and Topological Representations of Groups, Semigroups, and Categories.* North-Holland, 1980.

[Tan97] A. Tan. *Full completeness for models of linear logic*. PhD thesis, King's College, University of Cambridge, October 1997.

[Trn66] V. Trnková. Universal categories. *Comment. Math. Univ.Carolinae*, 7:143–206, 1966.