Chu Spaces

Vaughan Pratt
Stanford University

Notes for the
School on Category Theory and Applications
University of Coimbra
July 13-17, 1999

Chapter 1

Introduction to Chu Spaces

Chapter 1 introduces the basic notions, gives examples of use of Chu spaces, points out some interference properties, and proves that functions between Chu spaces are continuous if and only if they are homomorphisms. Chapter 2 realizes a variety of mathematical objects as Chu spaces, including posets, topological spaces, semilattices, distributive lattices, and vector spaces. Chapter 3 gives several senses in which Chu spaces are universal objects of mathematics. Chapter 4 interprets operations of linear logic and process algebra over Chu spaces. Chapter 5 studies linear logic from an axiomatic viewpoint, with emphasis on the multiplicative fragment. Chapter 6 develops several notions of naturality as a semantic criterion for canonical transformations. Chapter 7 proves full completeness of the multiplicative linear logic of Chu spaces

For hands-on experience with Chu spaces, particularly in conjunction with Chapter 4, visit the Chu space calculator available on the World Wide Web at `http://boole.stanford.edu/live/`.

1.1 Definitions

A Chu space is simply a matrix over a set Σ, that is, a rectangular array whose entries are drawn from Σ. We formalize this as follows.

Definition 1. A Chu space $\mathcal{A} = (A, r, X)$ over a set Σ, called the ***alphabet***, consists of a set A of ***points*** constituting the ***carrier***, a set X of ***states*** constituting the ***cocarrier***, and a function $r : A \times X \to \Sigma$ constituting the ***matrix***. $\square$

The alphabet can be empty or a singleton, but starting with $\Sigma = \{0, 1\}$ it becomes possible to represent a rich variety of structured objects as Chu spaces. Only the underlying set of the alphabet participates in the actual definition of Chu space, and any structure on the alphabet, such as the order $0 \leq 1$ or the operations $\wedge$, $\vee$, $\neg$ of the two-element Boolean algebra, is purely in the eye of the beholder.

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 $\mathcal{A}$.

When $\hat{r}$ is injective, i.e. all rows distinct, we call $\mathcal{A}$ ***separable***. Similarly when $\check{r}$ is injective, we call $\mathcal{A}$ ***extensional***. When $\mathcal{A}$ is both separable and extensional we call it ***biextensional***.

We define the ***biextensional collapse*** of $\mathcal{A} = (A, r, X)$ to be $(\hat{r}(A), r', \check{r}(X))$ where $r'(\hat{r}(a), \check{r}(x)) = r(a,x)$. Intuitively the biextensional collapse simply identifies equal rows and equal columns. Its points however are no longer elements of A but functions from X to Σ, and its states are functions from A to Σ.

Matrices enjoy a similar duality principle to that of lattices (but not semilattices). Just as the order dual of a lattice is another lattice, the transpose of a matrix is another matrix. We denote the transpose or ***perp*** of $\mathcal{A} = (A, r, X)$ as $\mathcal{A}^\perp = (X, r^\smile, A)$ where $r^\smile(x,a) = r(a,x)$.

One popular application for lattices is logic, where there is a strong sense of "which way is up," with *true* at the top and *false* on the bottom. Our application for Chu spaces comes with a similar strong sense of orientation. Rows serve to represent individuals or events, which we think of as *coexisting entities* inhabiting a *conjunctive space*. Similarly columns represent predicates or states, which are to be understood as *alternative qualities* located in a *disjunctive space*.

Example 1. Taking $\Sigma = 2 = \{0,1\}$, the *set* $A = \{a,b,c\}$ may be represented as the Chu space $\begin{array}{c|c|} \mathrm{a} & 01010101 \\ \mathrm{b} & 00110011 \\ \mathrm{c} & 00001111 \end{array}$. The characteristic feature of a set is that its states permit each point independently to take on any value from Σ: all of the Σ^A (in this case $2^3 = 8$) possible states are permitted. Sets are the deserts of mathematics, not only barren of structure but having elements with the independent behavior of grains of dry sand.

Any 8-element set X could serve to index the columns of this example. However for extensional spaces like this one it is often convenient to treat the columns as self-identifying: each column is a function from A to Σ, i.e. $X \subseteq \Sigma^A$. We call Chu spaces organized in this way *normal*, and abbreviate (A, r, X) to (A, X) with r understood as application, i.e. $r(a,x)$ is taken to be $x(a)$, each $x \in X$ now being a function $x : A \to \Sigma$. For $\Sigma = 2$ this is equivalent to viewing columns as subsets of A, more precisely as the characteristic functions of those subsets, with the 1's in the column indicating the members of the subset and the 0's the nonmembers.

Example 2. Delete three columns from Example 1 to yield $\begin{array}{c|c|} \mathrm{a} & 01111 \\ \mathrm{b} & 00101 \\ \mathrm{c} & 00011 \end{array}$. If we define $a \leq b$ to hold just when it holds in every column (taking $0 \leq 1$ as usual), then we now have $b \leq a$ and $c \leq a$, still three distinct elements but now equipped with a nontrivial order relation. This Chu space represents not an unstructured set but rather a *poset* (partially ordered set) $(A, \leq)$, one with a reflexive transitive antisymmetric binary relation, meaning one satisfying $a \leq a$; $a \leq b$ and $b \leq c$ implies $a \leq c$; and $a \leq b \leq a$ implies $a = b$ for all $a, b, c \in A$. This last, antisymmetry, is the poset counterpart of separability for Chu spaces.

The columns of examples 1 and 2 are in each case closed under union and intersection (bitwise join and meet). This is not true (for either operation) for their rows.

Example 3. Delete one more column to yield $\begin{array}{c|c|} a & 0111 \\ b & 0101 \\ c & 0011 \end{array}$. The first row is now the join (bitwise disjunction) of the other two, which was not the case with the previous Chu space. This Chu space represents a *semilattice* $(A, \vee)$, a semigroup (set with an associative binary operation) that is commutative and idempotent $(a \vee a = a)$.

The rows of this space are now closed under binary union, the defining characteristic of a semilattice. But the very step we took to ensure this broke closure of the columns under intersection. This is no coincidence but an instance of a systematic phenomenon that we now examine briefly.

1.2 Interference

Events (rows) are made of the same bits as states (columns), so it stands to reason the two would interfere with other.

A simple case of interference is given by a Chu space having a constant row. If it also contains a constant column, then the two constants must be the same. Thus if A has a row of all 1's it cannot also have a column of all 0's. And if it has two or more different constant rows then it can have no constant columns at all.

This phenomenon formalizes a well-known paradox. Viewing points as objects, states as forces, and $r(a, x)$ as 1 just when object a can resist force x, an immovable object is a row of all 1's while an irresistible force is a column of all 0's.

This interference is the zeroary case of a more general phenomenon whose binary case is as follows. We say that the meet $a \wedge b$ or the join $a \vee b$ is ***proper*** just when the result is not one of the two arguments.

Proposition 1.2. *A proper meet in $\mathcal{A}$ precludes some proper join in $\mathcal{A}^{\perp}$.*

Proof. Let $a \vee b$ be proper. Then there must exist states x, y such that $r(a, x) = r(b, y) = 1$ and $r(a, y) = r(b, x) = 0$. Hence $r(a \vee b, x) = 1$ and $r(a \vee b, y) = 1$. Now suppose $x \wedge y$ exists. Then from the above, $r(a \vee b, x \wedge y) = 1$. But we also have $r(a, x \wedge y) = r(b, x \wedge y) = 0$, and hence $r(a \vee b, x \wedge y) = 0$, a contradiction. $\square$

Corollary 1.3. *If $\mathcal{A}$ has all meets then $\mathcal{A}^{\perp}$ has no proper joins.*

Proof. For if $\mathcal{A}^{\perp}$ had a proper join it would preclude some proper meet of $\mathcal{A}$. $\square$

Corollary 1.4. *If $\mathcal{A}$ has all binary joins and $\mathcal{A}^{\perp}$ has all binary meets then A and X are both linearly ordered.*

The irrestible-force-immovable-object scenario may be generalized to the binary case as follows. Treat points and states as players on two teams A and X, and treat matrix entries as expected outcomes of singles matches between two players on opposite teams, with a 1 or 0 indicating a win or loss respectively for the player from team A. Say that player a ***combines*** players b and c when the players defeated by a are exactly those defeated by either b or c, and likewise for players from team X. Now suppose that every pair of players has a single player combining them in this sense, on both teams. Then the preceding corollary says that both teams can be completely ranked by ability, namely by inclusion on the set of players that each player defeats.

Open problems. (i) Study this interference for infinite joins and meets of various cardinalities. (ii) Extend to larger Σ than 2. (iii) For Σ the complex numbers obtain the Heisenberg uncertainty principle as an instance of Chu interference.

1.3 Properties of Chu spaces

Why not represent both the poset and the semilattice more economically as $\begin{array}{l}\text{a}\\ \text{b}\\ \text{c}\end{array}\boxed{\begin{array}{l}11\\ 10\\ 01\end{array}}$, by deleting the two constant columns? The answer reveals a fundamental feature of Chu spaces differentiating them from conventional algebraic objects. The matrix serves not only to identify the points and states but also to furnish the Chu space with properties. The space $\begin{array}{l}\text{a}\\ \text{b}\\ \text{c}\end{array}\boxed{\begin{array}{l}11\\ 10\\ 01\end{array}}$ contains a constant, namely a, and a complementary pair b and c. Posets and semilattices contain neither constants nor complementary pairs. And semilattices differentiate themselves from posets by the solvability of $a \vee b = c$ in c for *all* for all pairs a, b, whereas for posets this equation is solvable, at least by our rules, if and only if $a \leq b$ or $b \leq a$.

We define the general notion of property of a normal Chu space as follows.

Definition 5. A ***property*** of a normal Chu space (A, X) is a superset of its columns, i.e. a set Y satisfying $X \subseteq Y \subseteq \Sigma^A$. □

The properties of (A, X) are in bijection with the power set of $\Sigma^A - X$. Each subset $Z \subseteq \Sigma^A - X$ corresponds to the property $Y = X \cup Z$. The properties are therefore closed under arbitrary intersection, which we interpret as *conjunction* of properties. And we interpret inclusion between properties as implication: if $Y \subseteq Y'$ we say that property Y implies property Y'. In particular property X, as the conjunction of all properties of (A, X), implies all those properties. In effect (A, X) is its own strongest or *defining* property.

Returning to our original 8-state example, since $X = 2^A$ there is only one property, which we can think of as the vacuous property *true*. Now define the property $b \leq a$ to consist of all states x in which $x(b) \leq x(a)$. The Chu space having this as its defining property is $\begin{array}{l}\text{a}\\ \text{b}\\ \text{c}\end{array}\boxed{\begin{array}{l}010111\\ 000101\\ 001011\end{array}}$. For $c \leq a$ we have similarly $\begin{array}{l}\text{a}\\ \text{b}\\ \text{c}\end{array}\boxed{\begin{array}{l}010111\\ 001101\\ 000011\end{array}}$. The conjunction of these two properties is obtained as the intersection of the two sets of columns, namely $\begin{array}{l}\text{a}\\ \text{b}\\ \text{c}\end{array}\boxed{\begin{array}{l}01111\\ 00101\\ 00011\end{array}}$. This was our second example, the poset defined by $b \leq a$ and $c \leq a$, which can be rolled into one formula as $b \vee c \leq a$.

Now consider $a \leq b \vee c$, which by itself eliminates only one state to yield the Chu space $\begin{array}{c|c|}\cline{2-2} \mathrm{a} & 0010101 \\ \mathrm{b} & 0110011 \\ \mathrm{c} & 0001111 \\ \cline{2-2}\end{array}$. In combination with example 2 however we obtain our previous example 3, corresponding to the property $a = b \vee c$, the only fact about $\vee$ that we need to know to define this particular semilattice $(A, \vee)$.

What is going on here is that the columns are acting as primitive models. In fact if we view the points of a Chu space over 2 as propositional variables then the columns are precisely those assignments of truth values to variables that satisfy the defining property of the space. In this way we can identify the n-point normal Chu spaces over 2 with the 2^{2^n} n-ary Boolean operations. For general Σ this number becomes $2^{|\Sigma|^n}$.

Further extending the connection with conventional logic, the set of properties of a Chu space can be understood as the theory of that space. A subset of a theory can serve to axiomatize that theory, i.e. form a basis for it. We therefore define an *axiomatization* of a Chu space to be a set of properties. The conjunction (intersection) of that set then denotes (the state set of) the Chu space so axiomatized.

One use of axiomatizations is to define a class of Chu spaces axiomatizable by axioms *of a particular form.* For example posets are those Chu spaces axiomatizable by atomic implications $a \to b$, that is, $a \leq b$.

1.4 Chu transforms

The class of all Chu spaces is made a category by defining a suitable notion of morphism. We can gain at least some insight into the properties of Chu spaces from a knowledge of how they transform into each other.

Given two Chu spaces $\mathcal{A} = (A, r, X)$ and $\mathcal{B} = (B, s, Y)$, a ***Chu transform*** from $\mathcal{A}$ to $\mathcal{B}$ 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 constitutes a primitive form of adjointness, and we therefore call it the ***adjointness condition***.

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 category whose objects are Chu spaces over Σ and whose morphisms are Chu transforms composing as above is denoted Chu_Σ.

We cite without proof the following facts about this category. Its isomorphisms are those Chu transforms (f, g) for which f and g are both bijections (isomorphisms in the category Set of sets). Its monics are those (f, g) for which f is an injection and g a surjection, and dually for its epis. Its initial objects are all Chu spaces with empty carrier and singleton cocarrier, while its final objects are those with singleton carrier and empty cocarrier.

We call two Chu spaces ***equivalent*** when their respective biextensional collapses are isomorphic. This is the Chu counterpart of equivalent categories as those having isomorphic skeletons.

Biextensional collapse is an idempotent operation, up to isomorphism.

1.5 Continuous = Homomorphism

The previous two sections give two ostensibly quite different understandings of the structure of Chu spaces, one in terms of properties of individual Chu spaces, the other in terms of the transformability of one space into another. In this section we completely reconcile these two understandings by giving a sense in which they are equivalent.

Homomorphisms. To every function $f : A \to B$ we associate a function $\tilde{f} : 2^{\Sigma^A} \to 2^{\Sigma^B}$ defined as $\tilde{f}(Y) = \{g : B \to \Sigma \mid gf \in Y\}$ for $Y \subseteq \Sigma^A$. A ***homomorphism*** of Chu spaces $\mathcal{A} = (A, r, X)$, $\mathcal{B} = (B, s, Y)$ is a function $f : A \to B$ such that $\tilde{f}(\check{r}(X)) \supseteq \check{s}(Y)$. This is equivalent to requiring that f send properties of $\mathcal{A}$ to properties of $\mathcal{B}$, justifying the term "homomorphism." For normal Chu spaces, the above condition simplifies to $\tilde{f}(X) \subseteq Y$.

Continuity. 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 $f : \mathcal{A} \to \mathcal{B}$ ***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.

Theorem 1.6. *A function $f : \mathcal{A} \to \mathcal{B}$ is a homomorphism if and only if it is continuous.*

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. $\square$

Corollary 1.7. *Under the interpretation of a normal Chu_2 space $\mathcal{A}$ as a Boolean formula $\varphi_\mathcal{A}$, with functions then understood as variable renamings, a function $f : \mathcal{A} \to \mathcal{B}$ is continuous if and only if the result of substituting variable $f(a)$ for each variable a in $\varphi_\mathcal{A}$ is a consequence of $\varphi_\mathcal{B}$.*

1.6 Historical Notes

The original Chu construction as described in Po-Hsiang (Peter) Chu's master's thesis took a symmetric monoidal closed category V with pullbacks and an object k of V and "completed" V to a self-dual category $\mathsf{Chu}(V, k)$. It appeared in print as the appendix to his advisor M. Barr's book introducing the notion of *-autonomous category [Bar79].

The intimate connection between linear logic and *-autonomous categories was first noticed by Seely [See89], furnishing Girard's linear logic [Gir87] with a natural constructive semantics. Barr then proposed the Chu construction as a source of constructive models of linear logic [Bar91].

The case $V = \mathsf{Set}$ is important for its combination of simplicity and generality. This case was first treated explicitly by Lafont and Streicher [LS91], where they treated its connections with von Neumann-Morgenstern games and linear logic, observing in passing that vector spaces, topological spaces, and coherent spaces were realizable as games, giving a small early hint of their universality.

Our own interest in Chu spaces was a consequence of attempts to formalize a suitable notion of partial distributive lattice as a generalization of Nielsen, Plotkin and Winskel's notion of event structure [NPW81] for modeling concurrent computation. After arriving at such a notion based on the dual interaction of ordered Stone spaces and distributive lattices, we found that the resulting category was equivalent to Chu_2.

The name "Chu space" was suggested to the author by Barr in 1993 as a suitable name for the objects of Chu_Σ reifying "Chu construction," which predated Lafont and Streicher's "game." (We had previously been considering calling them hyperspaces by analogy with hypergraphs, which transform by parallel rather than antiparallel functions.) An advantage of "Chu space" is that it requires no disambiguating qualification to uniquely identify it, unlike "game." By analogy with categories enriched in V [Kel82] one might refer to the objects of the general Chu construction $\mathsf{Chu}(V, k)$ as ***V-enriched*** Chu spaces, and indeed $\mathsf{Chu}(V, k)$ can be formulated as a V-category, one whose hom-objects are objects of V.

Chapter 2

Special Realizations

(This chapter is adapted from [Pra95].)

We confine our attention to Chu_2, which we abbreviate as Chu. We further restrict attention to normal Chu spaces (A, X), with $r(a, x)$, $x(a)$, and $a \in x$ used interchangeably according to whim.

The view of a normal Chu space as a set having properties gives one way of seeing at a glance that extensional Chu spaces can realize a great variety of ordered structures. Examples 1-3 at the beginning of the first chapter illustrated the realization of respectively a set, a poset, and a semilattice. The set was specified by giving no properties, the poset with properties of the form $a \leq b$, and the semilattice with properties of the form $a \vee b = c$ for all a, b in the carrier.

Besides these, Chu_2 spaces can also realize preordered sets, Stone spaces, ordered Stone spaces, topological spaces, locales, complete semilattices, distributive lattices (but not general lattices), algebraic lattices, frames, profinite (Stone) distributive lattices, Boolean algebras, and complete atomic Boolean algebras, to name some familiar "logical" structures.

Normally these structures "stick to their own kind" in that each forms its own category, with morphisms staying inside individual categories. Chu spaces bring all these objects into the one self-dual category Chu, permitting meaningful morphisms between say semilattices and algebraic lattices, while revealing various Stone dualities such as that between Boolean algebras and Stone spaces, frames and locales, sets and complete atomic Boolean algebras, etc. to be all fragments of one universal self-duality.

The notion of realization we intend here is the strong one defined by Pultr and Trnková [PT80]. Informally, one structure *represents* another when they transform in the same way, and *realizes* it when in addition they have the same carrier. Formally, a functor $F : C \to D$ is a *representation* of objects c of C by objects $F(c)$ of D when F is a full embedding[1]. A representation F is a *realization* when in addition $U_D F = U_C$, where $U_C : C \to \mathsf{Set}$, $U_D : D \to \mathsf{Set}$

[1] An embedding is a faithful functor $F : C_A \to C_B$, i.e. for distinct morphisms $f \neq g$ of C_A, $F(f) \neq F(g)$, and is *full* when for all pairs a, b of objects of C_A and all morphisms $g : F(a) \to F(b)$ of C_B, there exists $f : a \to b$ in C_A such that $g = F(f)$.

are the respective underlying-set functors. Pultr and Trnková give hardly any realizations in their book, concentrating on mere representations. In contrast all the representations of this chapter and the next will be realizations.

The self-duality of Chu, and of its biextensional subcategory, means that to every full subcategory C we may associate its dual as the image of C under the self-duality. This associates sets to complete atomic Boolean algebras, Boolean algebras to Stone spaces, distributive lattices to Stone-Priestley posets, semilattices to algebraic lattices, complete semilattices to themselves, and so on [Joh82].

We now illustrate the general idea with some examples.

Sets. We represent the set A as the normal Chu space with carrier A and no axioms. The absence of axioms permits all states, making the Chu space so represented $(A, 2^A)$.

Every function between these Chu spaces is continuous because there are no axioms to refute continuity. An equivalent way to see this is to observe that every function must have an adjoint because every state is available in the source to serve as the desired image of that adjoint. Hence this representation is full and faithful, and obviously concrete, and hence a realization.

Pointed Sets. A pointed set $(A, \star)$ is a set with a distinguished element $\star$. A homomorphism $h : (A, \star) \to (A', \star')$ is a function $f : A \to A'$ satisfying $f(\star) = \star'$.

We represent $(A, \star)$ as the normal Chu space (A, X) axiomatized by $\star = 0$. The effect of this axiom is to eliminate all columns x for which $r(\star, x) \neq 0$, making row $\star$ constantly 0 (quite literally a constant!).

An equivalent description is the result of adjoining the row of all 0's, representing $\star$, to the Chu space representing the set $A - \{\star\}$. For example $\begin{array}{c|c|} \star & 0000 \\ a & 0011 \\ b & 0101 \end{array}$ represents the pointed set $\{\star, a, b\}$. For finite A there are $2^{|A|-1}$ states.

Since $\star = 0$ is the only axiom, the only condition imposed on Chu transforms is $f(\star) = 0$, whose effect is $f(\star) = \star'$. Hence this representation is full and faithful, and clearly concrete, and hence a realization.

Bipointed sets $(A, \star, *)$ are also possible, axiomatized as $\star = 0$, $* = 1$. (For general Σ, up to $|\Sigma|$ distinguished elements are possible.)

Preorders. A preorder is a normal Chu space axiomatized by "atomic implications," namely propositions of the form $a \leq b$. When $a \leq b$ and $b \leq a$ their rows must be equal, whence a separable preorder represents a partial order. Example 2 of Chapter 1 illustrates this notion.

For f to preserve these axioms is to have $f(a) \leq f(b)$ hold in the target for every $a \leq b$ holding in the source. But this is just the condition for f to be monotonic, giving us a realization in Chu of the category of preordered sets, and also of partially ordered sets as a full subcategory.

Proposition 2.1. *A normal Chu space realizes a preorder if and only if the set of its columns is closed under arbitrary pointwise joins and meets.*

(An arbitrary pointwise join of rows takes a set of rows and "ors" them together, producing a 1 in a given column just when some row in that set has a 1 in that column, and likewise for meet, and for columns. It is convenient to refer to pointwise join and meet as respectively union and intersection, via the view of bit vectors as subsets.)

Proof. (Only if) Fix a set Γ of atomic implications defining the given preorder. Suppose that the intersection of some set Z of columns each satisfying all implications of Γ fails to satisfy some $a \to b$ in Γ. Then the intersection must assign 1 to a and 0 to b. But in that case every column in Z must assign 1 to a, whence every such column must also assign 1 to b, so the intersection cannot have assigned 0 to b after all.

Dually, if the union of Z assigns 1 to a and 0 to b, it must assign 0 to b in every column of Z and hence can assign 1 to a in no column of Z, whence the union cannot have assigned 1 to a after all.

So the satisfying columns of any set of atomic implications is closed under arbitrary union and disjunction.

(If) Assume the columns of $\mathcal{A}$ are closed under arbitrary union and intersection. It suffices to show that the set Γ of atomic implications holding in $\mathcal{A}$ axiomatizes $\mathcal{A}$, i.e. that $\mathcal{A}$ contains all columns satisfying Γ. So let $x \subseteq A$ be a column satisfying Γ. For each $a \in A$ form the intersection of all columns of $\mathcal{A}$ containing a, itself a column of $\mathcal{A}$ containing a, call it y_a. Now form the union $\bigcup_{a\in x} y_a$ to yield a column z of $\mathcal{A}$, itself a column of $\mathcal{A}$ which must be a superset of column x.

Claim: $z = x$. For if not then there exists $b \in z - x$. But then there exists $a \in x$ such that $b \in y_a$, whence b is in every column of $\mathcal{A}$ containing a, whence $a \to b$ is in Γ. But x contains a and not b, contradicting the assumption that x satisfies Γ.

To complete the argument that this is a realization we need the Chu transforms between posets realized in this way as Chu spaces to be exactly the monotone functions. Now monotonicity is the condition that if $a \leq b$ holds in (is a property of) the source then $f(a) \leq f(b)$ holds in the target. Since the only axioms are atomic implications, monotonicity is equivalent to being axiom-preserving, equivalently property-preserving. Hence monotonicity and continuity coincide. $\square$

The last paragraph of this argument is the same for any subcategory of Chu whose objects are defined by properties, and can be omitted once the principle is understood, as we will do in Proposition 2.2.

Topological spaces. A topological space is an extensional Chu space whose columns are closed under arbitrary union and *finite* (including empty) intersection. The Chu transforms between topological spaces are exactly the continuous functions. Lafont and Streicher [LS91] mention in passing this realization along with that of Girard's coherent spaces [Gir87], also in Chu_2, and the realization of vector spaces over the field k in $\mathsf{Chu}_{|k|}$.

For Chu spaces representing topological spaces, separability is synonymous with T_0.

Semilattices. A semilattice $(A, \vee)$ is a set A with an associative commutative idempotent binary operation $\vee$. We may realize it as a separable normal Chu space with carrier A and axiomatized by all equivalences $a \vee b \equiv c$ holding in $(A, \vee)$, one such equivalence for each pair a, b in A. This is illustrated by Example 3 of Chapter 1.

For f to preserve these axioms is to have $f(a) \vee f(b) \equiv f(c)$ hold in the target. But this is just the condition for f to be a semilattice homomorphism, giving us a realization in Chu of the category of semilattices.

Equivalently a semilattice is a separable normal Chu space whose rows are closed under binary union and which is column-maximal subject to the other conditions. Column-maximality merely ensures that all columns satisfying the axioms are put in.

The dual of a semilattice is an algebraic lattice [Joh82, p.252].

Problem. Formulate this in terms of closure properties on rows and columns.

Complete semilattices. A ***complete semilattice*** $(A, \bigvee)$ is a set A together with an operation $\bigvee : 2^A \to A$ such that (i) the binary relation $x \leq y$ satisfying $\bigvee\{x, y\} = y$ partially orders A, and (ii) for all subsets $B \subseteq A$, $\bigvee B$ is the least upper bound of B in $(A, \leq)$. A homomorphism of complete semilattices is a function $h : A \to B$ such that $h(\bigvee B) = \bigvee h(B)$ (where $h(B)$ denotes as usual the direct image $\{h(b) | b \in B\}$).

We realize the complete semilattice $(A, \bigvee)$ as the Chu space axiomatized by all equations holding in A of the form $\bigvee B = a$ for $B \subseteq A$ and $a \in A$. Reasoning analogously to the previous examples, this is a realization of complete semilattices as Chu spaces.

Proposition 2.2. *A normal Chu space realizes a complete join semilattice if and only if each of the set of its rows, and the set of its columns, is closed under arbitrary union.*

(This includes the empty union, mandating both a zero row and a zero column.)

Proof. (Only if) We are given that the rows are closed under union so it suffices to show the same for the columns. For any $B \subseteq A$, $(\bigvee Y)(\bigvee B) = \bigvee_{x \in Y} \bigvee_{a \in B} r(a, x)$, while $\bigvee(\bigvee Y)(B) = \bigvee_{a \in B} \bigvee_{x \in Y} r(a, x)$. These are equal since each expression is 0 just when the whole $B \times Y$ rectangle is zero, making it clear that the sups commute. Hence $\bigvee Y$ satisfies every axiom of $\mathcal{A}$ and therefore belongs to X. So the columns are closed under union.

(If) Given $\mathcal{A} = (A, X)$ with rows and columns closed under arbitrary union, it suffices to show that $\mathcal{A}$ is axiomatized by those of its properties of the form $\bigvee B = a$ for all $B \subseteq A$. The rows determine a complete semilattice, so it suffices to verify that every column satisfying the equations is included.

So let z be any column absent from $\mathcal{A}$; we shall show that z violates some property. Let $Y = \{x \in X | x \leq z\}$. Y consists of those columns that are 0 in every row where z is 0. Let $y = \bigvee Y$. Then $y \leq z$, and furthermore y meets the condition for membership in Y.

Since $y \in X$ and $z \notin X$, there must exist $b \in A$ such that $b \in z - y$, i.e. a row where $y = 0$ (whence b is 0 in every column of Y) and $z = 1$. Let $B = \{a \in A | a \leq b\}$. (So the whole $B \times Y$ rectangle is zero.) Let $C = \overline{z} = \{a \in A | r(a, z) = 0\}$, whence $C \subseteq B$. Now in the columns of Y, $\bigvee B$ is 0. But in all the remaining columns $\bigvee C$ is 1 since any counterexample would have been put in Y. So $\bigvee B = \bigvee C$ is a property of $\mathcal{A}$. But row b shows this property to be false for z, justifying its omission. $\square$

The symmetry of this second characterization of complete semilattices immediately entails:

Corollary 2.3. *The category of complete semilattices is self-dual.*

Distributive Lattices. The idea for semilattices $(A, \vee)$ is extended to distributive lattices $(A, \vee, \wedge)$ by adding to the semilattice equations for $\vee$ all equations $a \wedge b = c$ holding in $(A, \vee, \wedge)$ for each a, b in A. Distributivity being a Boolean tautology, it follows that all lattices so represented are distributive.

The converse, that every distributive lattice L is so represented, is a consequence of Birkhoff's theorem (assuming the Axiom of Choice) that every distributive lattice is representable as a "ring" of sets [Bir33, Thm 25.2], i.e. with join and meet realized respectively as union and intersection. For we can turn any such representation of L into a Chu realization by starting with A as the lattice and X as the set from which the representing subsets are drawn, and taking the matrix to be the membership relation. We then identify duplicate columns, and adjoin any additional columns that satisfy all equations of the form $a \vee b = c$ and $a \wedge b = c$ that are true of L.

With the help of Proposition 2.1 we then deduce Birkhoff's pairing of finite distributive lattices with posets, with the additional information that this pairing is in fact a duality: the category of finite posets is the opposite of the category of finite distributive lattices.

Boolean algebras. A Boolean algebra is a complemented distributive lattice, hence as a Chu space it suffices to add to the specifications of a distributive lattice the requirement that the set of rows be closed under complement. That is, a Boolean algebra is a biextensional Chu space whose rows form a Boolean algebra under pointwise Boolean combinations (complement and binary union suffice) and is column-maximal in the sense of containing every column satisfying the Boolean equations satisfied by the rows.

The dual of a Boolean algebra can be obtained as always by transposition. What we get however need not have its set of columns closed under arbitrary union, in which case this dual will not be a topological space. But M. Stone's theorem [Sto36] is that the dual of a Boolean algebra is a totally disconnected compact Hausdorff space. We therefore have to explain how the dual of a Boolean algebra may be taken to be either a topological space or an object which does not obviously behave like a topological space.

There is a straightforward explanation, which at the same time yields a slick proof of Stone's theorem stated as a categorical duality.

The transpose of a Boolean algebra may be made a topological space by closing its columns under arbitrary union. The remarkable fact is that when this adjustment is made to a pair of transposed Boolean algebras, *the set of Chu transforms between them does not change.* (Actually their adjoints may change, but since these spaces are extensional the adjoint is determined by the function, which is therefore all that we care about; the functions themselves do not change.)

We prove this fact by first closing the source, then the target, and observing that neither adjustment changes the set of Chu transforms.

Closing the columns of the source under arbitrary union can only add to the possible Chu transforms, since this makes it easier to find a counterpart for a target column in the source. Let $f : A \to B$ be a function that was not a Chu transform but became one after closing the source. Now the target is still a transposed Boolean algebra so its columns are closed under complement, whence so is the set of their compositions with f. But no new source column has a complement in the new source, whence no new source column can be responsible for making f a Chu transform, so f must have been a Chu transform before closing the source.

Closing the columns of the target under arbitary union can only delete Chu transforms, since we now have new target columns to find counterparts for. But since the new target columns are arbitrary unions of old ones, and all Boolean combinations of columns commute with composition with f (a simple way of seeing that f^{-1} is a CABA homomorphism), the necessary source columns will also be arbitrary unions of old ones, which exist because we previously so closed the source columns. Hence Chu transforms between transposed Boolean algebras are the same thing as Chu transforms, and hence continuous functions, between the topological spaces they generate.

We conclude that there exists a full subcategory of Top (topological spaces) dual to the category of Boolean algebras. Stone's theorem goes further than we attempt here by characterizing this subcategory as consisting of precisely the totally disconnected, compact, and Hausdorff spaces.

An interesting aspect of this proof of Stone's theorem is that usually a duality is defined as a contravariant *equivalence*. Here, all categorical equivalences appearing in the argument that are not actual isomorphisms are covariant. The one contravariant equivalence derives from the self-duality of Chu, which is an *isomorphism* of Chu with $\mathsf{Chu}^{\mathrm{op}}$. Those equivalences on either side of this duality that fail to be isomorphisms do so on account of variations in the choice of carrier and cocarrier. We pass through the duality with the aid of two independent sets A and X. But when defining Boolean algebras and Stone spaces, in each case we take X to consist of subsets of A, and it is on account of those conflicting representational details that we must settle for less than isomorphism on at least one side of the duality.

Vector spaces over $GF(2)$. An unexpected entry in this long list of full concrete subcategories of Chu is that of vector spaces over $GF(2)$. These are realizable as separable extensional Chu spaces whose rows are closed under binary exclusive-or, and column-maximal subject to this condition. This representation is the case $k = GF(2)$ of Lafont and Streicher's observation that the category of vector spaces over any field k is realizable in $\mathsf{Chu}_{U(k)}$.

Exercise. In the finite case (both finite dimension and finite cardinality since the field is finite), Proposition 2.2 for complete semilattices has its counterpart for finite dimensional vector spaces, whose Chu realizations are exactly those square Chu spaces whose rows and columns are closed under binary exclusive-or.

Self-duality of the category of finite dimensional vector spaces then follows by the same reasoning as for complete semilattices.

Problem. Does Chu_3 embed fully in Chu_2? (Conjecture: Yes.) More generally, for which $|\Sigma| > |\Sigma'|$ does Chu_Σ embed fully in $\mathsf{Chu}_{\Sigma'}$? (We showed at the beginning of the chapter that no such embedding could be concrete.)

Chapter 3

General Realizations

3.1 Universality of Chu_2

The previous chapter embedded a number of more or less well-known categories of mathematics fully and concretely in the category of Chu spaces over a suitable alphabet. A converse question would be, which categories do not so embed in some Chu category?

In fact we could ask a stronger question: which categories do not so embed in Chu_Σ for any given Σ?

Theorem 3.1. *Assuming the Generalized Continuum Hypothesis,* Chu_Σ *is realizable in* $Chu_{\Sigma'}$ *if and only if* $|\Sigma| \leq |\Sigma'|$.

Proof. If $|\Sigma| \leq |\Sigma'|$ then there exists an injection $h : \Sigma \to \Sigma'$. The Chu_Σ space (A, r, X) can then be realized in $\mathsf{Chu}_{\Sigma'}$ as (A, s, X) where $s(a, x) = h(r(a, x))$.

For the converse, choose any singleton 1 of Set. In each of the two categories consider those spaces $(1, X)$ having just one endomorphism (to get just the extensional spaces). The isomorphism classes of these are in bijection with 2^Σ and $2^{\Sigma'}$ respectively. Any realization of Chu_Σ in $Chu_{\Sigma'}$ must represent members of distinct such isomorphism classes from Chu_Σ with members of distinct isomorphism classes from $Chu_{\Sigma'}$. But if $|\Sigma| > |\Sigma'|$, then (assuming the generalized continuum hypothesis) this is impossible, □

(GCH asserts that if $\alpha < \beta$ then $2^\alpha \leq \beta$, which in combination with Cantor's theorem $\beta < 2^\beta$ gives $2^\alpha < 2^\beta$.)

Problem: Does Theorem 3.1 hold in the absence of GCH?

3.2 Relational Structures

In this section we show that n-ary relational structures are realizable as Chu spaces over (an alphabet of size) 2^n.

A relational structure of a given ***similarity type*** consists of an m-tuple of nonempty sets $A_1, A_2, \ldots, A_m$ and an n-tuple of nonempty[1] relations $R_1, \ldots, R_n$, such that each R_j is a subset of a product of $A_i's$, with the choice of the i's in that product depending only on j. (That is, if say $R_3 \subseteq A_2 \times A_1 \times A_2$ in one algebra then this is true in all algebras of the same similarity type.)

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 tuple $(a_1, \ldots, a_q)$ of $\mathcal{A}$ of the arity q appropriate for R_j, $(f(a_1), \ldots, f(a_q)) \in R'_j$. 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 (single-sorted) 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 or algebraic 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 (defined for our purposes as subsets of $n = \{1, 2, \ldots, n\}$) where

(i) $X \subseteq (2^A)^n$ 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, $1 \leq i \leq n$, for which $a_i \in x_i$; and

(ii) $r(a, x) = \{i | a \in x_i\}$.

[1]Nonemptiness is only needed here when $m > 1$ or $n > 1$, so that we can reliably join relations. If $m = n = 1$ to begin with, our main result here can cope with either A or R empty.

The bijection $(2^A)^n \cong (2^n)^A$ puts X in bijection with a subset of Σ^A when Σ is taken to be 2^n. Thus for $\mathbf{Str}_3$, ternary relations, $\Sigma = 2^3 = 8$. We may think of Chu spaces over 2^n as $A \times X \times n$ matrices over 2.

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

Theorem 3.2. *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. $(\rightarrow)$ 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. $\square$

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 Chu_{2^n}. That is, there exists a full and faithful functor $F : C \to \mathsf{Chu}_{2^n}$ such that $FU = U'F$ where $U : C \to \mathsf{Set}$ and $U' : \mathsf{Chu}_{2^n}$ are the respective forgetful functors.

In terms of properties, each tuple $\mathbf{a} = (a_1, \ldots, a_n)$ of R eliminates those states $(x_1, \ldots, x_n)$ for which $\mathbf{a} \in \prod_i \overline{x_i}$, thereby defining a property $\varphi_{\mathbf{a}} \subseteq (2^n)^A$ associated to tuple $\mathbf{a}$. The Chu space representing (A, R) can then be defined as the space satisfying $\varphi_{\mathbf{a}}$ for all $\mathbf{a}$ in R.

3.3 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 section. 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.

3.4 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 Chu_Σ 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. Readers familiar with the Yoneda embeddings of C into $\mathsf{Set}^{C^{\mathrm{op}}}$ and of C^{op} into Set^C will see a strong similarity. An important difference is that whereas both $\mathsf{Set}^{C^{\mathrm{op}}}$ and Set^C depend nontrivially on the structure of C, the target 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 3.3. *Every small category C embeds fully in $\mathsf{Chu}_{|C|}$.*

Proof. Define the functor $F : C \to \mathsf{Chu}_{|C|}$ 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 also 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 adjointness 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 $\mathsf{Set}^{C^{\mathrm{op}}}$, 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 $(Set^C)^{\mathrm{op}}$ (i.e. of C^{op} in 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.

Anticipating the treatment of dinaturality in Chapter 6, 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

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

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 the theorem (but not its witnessing realization) 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 3.4. *Every small honestly concrete category (C, U) embeds fully and concretely in $\mathsf{Chu}_{\sum_{b \in ob(C)} U(b)}$.*

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 $f \in U(b)$,

$$\begin{aligned}
U(\psi'(h))(f) &= U(h \circ \psi(1_{b'}))(f) \\
&= U(h)(U(\psi(1_{b'}))(f)) \\
&= U(h)(U(1_{b'})(\varphi(f))) \\
&= U(h)(\varphi(f)) \\
&= U(\psi(h))(f).
\end{aligned}$$

Hence $U(\psi'(h)) = U(\psi(h))$. Since U is faithful, $\psi'(h) = \psi(h)$. Hence $\psi' = \psi$. Since $F(b)$ is separable, $\varphi' = \varphi$. $\square$

Chapter 4

Operations on Chu spaces

We define a number of operations on Chu spaces. These operations come from linear logic [Gir87] and process algebra as interpreted over Chu spaces [GP93, Gup94]. There is some overlap: linear logic's *plus* operation $\mathcal{A} \oplus \mathcal{B}$ does double duty as independent (noninteracting) parallel composition, while *tensor* $\mathcal{A} \otimes \mathcal{B}$ serves also as *orthocurrence* [Pra86]. This last combines processes that "flow through" each other, as with a system of trains passing through a system of stations, or digital signals passing through logic gates.

For hands-on experience with all the finitary operations presented here except the nondiscrete limits and colimits, visit the web site `http://boole.stanford.edu/live/` and a menu-driven Java-based Chu space calculator will immediately be running on your computer accompanied by a detailed tutorial containing many exercises.

Perp. We have already defined the ***perp*** $\mathcal{A}^\perp$ of $\mathcal{A} = (A, r, X)$ as $(X, r^{\smile}, A)$. When $\mathcal{A}$ is separable (all rows distinct), $\mathcal{A}^\perp$ is extensional (all columns distinct). Likewise when $\mathcal{A}$ is extensional, $\mathcal{A}^\perp$ is separable. Thus perp preserves biextensionality.

The definition of $A^\perp$ makes $\mathcal{A}^{\perp\perp}$ not merely isomorphic to $\mathcal{A}$ but equal to it, giving us our first law: $\mathcal{A}^{\perp\perp} = \mathcal{A}$.

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 $t((a, b), f) = s(b, f(a))$ $(= r(a, g(b)))$.

When $\mathcal{A}$ and $\mathcal{B}$ are biextensional, $\mathcal{F}$ can be understood as all solutions to an $A \times B$ crossword puzzle such that the vertical words are columns of $\mathcal{A}$ and the horizontal words are rows of $\mathcal{B}^\perp$ (columns of $\mathcal{B}$). The columns of $\mathcal{A} \otimes \mathcal{B}$ are then the solutions themselves, with the rows of the solutions laid end to end and transposed to form a single column.

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

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 $\mathcal{A} = \boxed{\begin{smallmatrix}00\\01\end{smallmatrix}}$. The two "crossword solutions" here are $\boxed{\begin{smallmatrix}00\\00\end{smallmatrix}}$ and $\boxed{\begin{smallmatrix}00\\01\end{smallmatrix}}$. Hence $\mathcal{A}\otimes\mathcal{A} = \boxed{\begin{smallmatrix}00\\00\\00\\01\end{smallmatrix}}$, whose biextensional collapse is $\boxed{\begin{smallmatrix}00\\01\end{smallmatrix}}$.

Functoriality. We have defined $\mathcal{A}^\perp$ and $\mathcal{A}\otimes\mathcal{B}$ only for Chu spaces. We now make these operations functors on Chu_Σ 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 the occasional notation $f^\perp$ for g.

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.[1]

The tensor unit behaves as such, i.e. $\mathcal{A}\otimes 1 \cong \mathcal{A}$, via the evident isomorphism pairing $(a,\star)$ with a.

Linear Implication. We define linear implication, namely $\mathcal{A}\multimap\mathcal{B}$, as $(\mathcal{A}\otimes$

[1] 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 required to be skeletal [Mac71, p.161], then tensor product in Chu_Σ is also associative on the nose.

$\mathcal{B}^\perp)^\perp$. 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 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 $m(a, y) = s(f(a), y)$. Taking the row $s(f(a), -)$ as the representation in $\mathcal{B}$ of $f(a)$, this representation of f is simply a listing of the representations in $\mathcal{B}$ of the values of f at points of A. Dually 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 one additional perp the definition of linear implication can be turned into that of ***par***, $\mathcal{A}⅋\mathcal{B}$, defined as $(\mathcal{A}^\perp \otimes \mathcal{B}^\perp)^\perp$. Thus tensor and par are De Morgan duals of each other, analogously to the De Morgan duality of conjunction and disjunction in Boolean logic.

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

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 equip linear logic 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 their product.

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

Exercise. Determine the matrix associated with general limits and colimits. (Use the discrete case above as a guide.)

Exponentials The ***exponential*** $!A$ serves syntactically to "loosen up" the formula A so that it can be duplicated or deleted. Semantically as defined below it serves to retract Chu_Σ 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. A larger subcategory that is also cartesian closed is that of the comonoids, which we now define.

A *comonoid in* Chu_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.

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 4.1. *Given any family of comonoids (A, X_i) with fixed carrier A, $(A, \bigcap_i X_i)$ is a comonoid.*

Proof. 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. $\square$

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

Comonoids over 2 are of special interest. *Exercise:* (M. Barr) The state set of a comonoid over 2 is closed under finite union and finite intersection. It follows that finite comonoids over 2 are simply posets.

Define a comonoid to be T_1 (as in point-set topology) when it is over 2 and when for every distinct pair a, b of points there exist states x, y such that $r(a, x) = r(b, y) \neq r(a, y) = r(b, x)$ (equivalently, when no point is included in another as subsets of X). By the preceding exercise, finite T_1 comonoids are discrete ($X = 2^A$).

Exercise: Extend this to countable T_1 comonoids.

Open problem: Is every T_1 comonoid discrete?

The following operations are motivated by applications to process algebra. Each assumes some additional structure for Σ.

Concatenation. This operation assumes that Σ is partially ordered, for example $0 \leq 1$ for two-state events or $0 \leq 1 \leq 2$ for three-state events (respectively "not started," "ongoing," and "done"). This induces the usual pointwise partial ordering of Σ^A: given $f, g : A \to \Sigma$, $f \leq g$ just when $f(a) \leq g(a)$ for all $a \in A$, and similarly for Σ^X. The intent of the order is to inject a notion of time by decreeing that an event in state $s \in \Sigma$ can proceed to state s' only if $s \leq s'$.

The ***concatenation*** $\mathcal{A}; \mathcal{B}$ is that quotient of $\mathcal{A} + \mathcal{B}$ obtained by taking only those states (x, y) of $\mathcal{A} + \mathcal{B}$ such that either x is a maximal (= final) state of $\mathcal{A}$ or y is a minimal (= initial) state of $\mathcal{B}$ under the induced pointwise orderings. The idea is that $\mathcal{B}$ may not leave whatever initial state it is in until $\mathcal{A}$ enters some final state. We call this a quotient rather than a subobject because the subsetting is being done in the contravariant part.

Concatenation is clearly not commutative, but is associative, the common condition on $((x,y),z)$ and $(x,(y,z))$ being (x maximal $\vee$ y minimal) $\wedge$ (y maximal $\vee$ z minimal) $\wedge$ (x maximal $\vee$ z minimal).

Choice. This operation assumes that Σ contains a distinguised element 0 corresponding to nonoccurrence. The ***choice*** $\mathcal{A}\sqcup\mathcal{B}$ is defined as $(A+B,t,X+Y)$ where $t(a,x)=r(a,x)$, $t(b,y)=s(b,y)$, and $t(a,y)=t(b,x)=0$. Thus if $\mathcal{A}+\mathcal{B}$ finds itself in a state of $\mathcal{A}$, none of the events of $\mathcal{B}$ can have occurred in that state, and conversely for states $\mathcal{B}$.

Choice is both associative and commutative, up to isomorphism, but is not however idempotent, unlike its regular expression counterpart $a+b$ which satisfies $a+a=a$. This departs from process algebra which takes choice to be idempotent, in the tradition of regular expressions.

Chapter 5

Axiomatics of Multiplicative Linear Logic

(This chapter is adapted from a journal submission.)

We axiomatize Multiplicative Linear Logic (MLL) and propose the machinery of Danos-Regnier switching as its semantics. To develop intuition about the nature of switching we prove a bijection between the essential switchings of a cut-free proof in MLL and the interpretations of the implications in that proof as based on evidence either for or against.

5.1 Axiomatization

We have already encountered the linear logic connectives in the form of functors on Chu_Σ, 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 chapter. 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 the fact that B is derivable from A in the system, the approach we follow here. This continues an algebraic tradition dating back to Peirce and Schröder's relational algebra [Pei33, Sch95], updated for linear logic by Seely and Cockett [See89, CS97]. We confine our attention to the multiplicative fragment, MLL, further simplified by omitting the constants 1 and $1^\perp = \perp$.

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 $(B, C, \ldots)$ to be either a ***literal*** (an atom P or its negation $P^\perp$), a ***conjunction*** $A \otimes B$ of two formulas, or a ***disjunction*** $A \parr B$ of two formulas. 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***.

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

$$
\begin{array}{lrcl}
T & (P_1^\perp \parr P_1) \otimes \ldots \otimes (P_n^\perp \parr P_n), & & n \geq 1 \\
\mathsf{A} & (A \otimes B) \otimes C & \vdash & A \otimes (B \otimes C) \\
\overline{\mathsf{A}} & (A \parr B) \parr C & \vdash & A \parr (B \parr C) \\
\mathsf{C} & A \otimes B & \vdash & B \otimes A \\
\overline{\mathsf{C}} & A \parr B & \vdash & B \parr A \\
\mathsf{D} & (A \parr B) \otimes C & \vdash & A \parr (B \otimes C) \\
\mathsf{E} & A \otimes B & \vdash & A' \otimes B' \\
\overline{\mathsf{E}} & A \parr B & \vdash & A' \parr B'
\end{array}
$$

Table 1. System S1

The rules have the interesting feature of all having exactly one premise.[1] 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 E and $\overline{\mathsf{E}}$ assume $A \vdash A'$ and $B \vdash B'$, allowing the rules to be applied to subformulas.

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.

$$
\begin{array}{rcll}
(P^\perp \parr P) \otimes (Q^\perp \parr Q) & \vdash & P^\perp \parr (P \otimes (Q^\perp \parr Q)) & (\mathsf{D}) \\
& \vdash & P^\perp \parr ((Q^\perp \parr Q) \otimes P) & (\mathsf{C}, \overline{\mathsf{E}}) \\
& \vdash & P^\perp \parr (Q^\perp \parr (Q \otimes P)) & (\mathsf{D}, \overline{\mathsf{E}}) \\
& \vdash & P^\perp \parr (Q^\perp \parr (P \otimes Q)) & (\mathsf{C}, \overline{\mathsf{E}}) \\
& \vdash & (P^\perp \parr Q^\perp) \parr (P \otimes Q)) & (\overline{\mathsf{A}})
\end{array}
$$

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

Although this theorem has the form of an instance of $P_1 \multimap P_1$, it cannot be proved directly that way because the P_i's may only be instantiated with literals. Nevertheless all such more general instances of T can be proved.

5.2 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 ***linked formula***.

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 linking of A as follows. The linking determined for 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. Call a linking so determined by a derivation ***sound***. It is immediate that A is a theorem if and only if it has a sound linking.

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).

Call a linking L of formula A ***valid*** [2] when for all switchings σ for L, $G(A, L, \sigma)$ is a tree (connected acyclic graph).[3] This criterion is semantical in the same sense as validity in Boolean propositional calculus, defined as truth over all assignments of truth values to variables.

[2]Although it is not customary to refer to this notion as validity, the idea of linear logic as basically a connectionist logic makes it natural to think of switching as the connectionist counterpart of truth assignment, and hence a condition universally quantified over all switchings as a notion of validity.

[3]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.

Theorem 5.1. *(Danos-Regnier [DR89]) A linking of a formula 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 Chapter 6, the corresponding bijection is between cut-free proofs (as sound linkings) and transformations meeting a suitable naturality condition.

5.3 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 linked formula 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 linked or binary formula.

Since we will be dealing only with linked formulas (A, L) in this chapter, for simplicity we assume for the rest of this chapter 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 that a binary formula A is provable if and only if it is valid.

5.4 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 $A \parr B$. Marking $\parr$ with each of two possible directions permits us the reconcile the commutativity of $\parr$ 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 $(P^\perp \parr (P \otimes Q^\perp)) \parr Q$, whose connectives can be reassociated and permuted to yield $P^\perp \parr (Q \parr (P \otimes Q^\perp))$. 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 $\parr$'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. 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 $(P^\perp \parr (P \otimes Q^\perp)) \parr Q$, $P \otimes Q$ is downwards for all switchings save that in which the first $\parr$ 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 $\parr$. For any subformula $B = C \parr D$, if the edge from 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 $(P^\perp ⅋ (P \otimes Q^\perp)) ⅋ Q$, when the second ⅋ is switched to the right, the first ⅋ 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 $(P^\perp ⅋ (P \otimes Q^\perp)) ⅋ Q$; the direction of the first ⅋ connects the vertex $P^\perp ⅋ (P \otimes Q^\perp)$ 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 $P^\perp ⅋ (P \otimes Q^\perp)$ 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 $(P^\perp ⅋ (P \otimes Q^\perp)) ⅋ Q$ has only three essential switchings because when the second ⅋ is switched to the right we ignore the now downward first ⅋.

5.5 Bi-Implication

We would like to interpret the formulas $A \otimes B$ and $A ⅋ B$ 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 ⅋, without any negations $A^\perp$. It would be a pity to have to introduce negations as a side effect of interpreting $A ⅋ B$ as consisting of functions. To avoid this we shall make $\multimap$ perform the role of ⅋, allowing us to talk of functions of type $A \multimap B$.

This works fine except for Rule $\overline{\mathsf{C}}$, commutativity of ⅋, which must rewrite $A \multimap B$ as $B^\perp \multimap A^\perp$. To avoid having negation appear in $\overline{\mathsf{C}}$ 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 per Table 2 below. The axiom and rules are obtained from System S1 by rewriting each $A ⅋ B$ 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.

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 $\mathsf{E} - \overline{\mathsf{E}}'$.

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 $\multimapinv$.

T	$(P_1 \multimapboth P_1) \otimes \ldots \otimes (P_n \multimapboth P_n), \quad n \geq 1$		
A	$(A \otimes B) \otimes C$	$\vdash$	$A \otimes (B \otimes C)$
$\overline{\mathsf{A}}$	$(A \otimes B) \multimap C$	$\vdash$	$A \multimap (B \multimap C)$
$\overline{\mathsf{A}}'$	$(A \multimap B) \multimapinv C$	$\vdash$	$A \multimap (B \multimapinv C)$
$\overline{\mathsf{A}}''$	$(A \multimapinv B) \multimapinv C$	$\vdash$	$A \multimapinv (B \otimes C)$
C	$A \otimes B$	$\vdash$	$B \otimes A$
$\overline{\mathsf{C}}$	$A \multimap B$	$\vdash$	$B \multimapinv A$
$\overline{\mathsf{C}}'$	$A \multimapinv B$	$\vdash$	$B \multimap A$
D	$(A \multimap B) \otimes C$	$\vdash$	$A \multimap (B \otimes C)$
D'	$(A \multimapinv B) \otimes C$	$\vdash$	$A \multimapinv (B \multimapinv C)$
E	$A \otimes B$	$\vdash$	$A' \otimes B'$
$\overline{\mathsf{E}}$	$A' \multimap B$	$\vdash$	$A \multimap B'$
$\overline{\mathsf{E}}'$	$A \multimapinv B'$	$\vdash$	$A' \multimapinv B$

Table 2. System S2

Theorem 5.2. *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.[4]

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

[4]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.

disjunction $B = C \parr D$ as either $C \multimapinv D^\perp$ 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 $(B^\perp \multimapinv C)^\perp$ 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 $P \multimap P$, $P \multimapinv P$, $P^\perp \multimap P^\perp$, or $P^\perp \multimapinv P^\perp$ depending on the sign of P_i in the S1 theorem and the direction determined by σ for that $\parr$ in the σ-translation.

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. $\square$

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.

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

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.

5.6 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 to us 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 $B \circ\!\!- A$ is then $(f';f)$, as an application of commutativity. Note that 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 $\rightarrow$ 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,[5] 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

[5] 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.

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 A and its sign. This expansion may be applied recursively to the a_i's, as for example in Rule A 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 chapter 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.

A	$\lambda((a,b),c) : (A \otimes B) \otimes C \;.\; (a,(b,c)) : A \otimes (B \otimes C)$
$\overline{\mathsf{A}}$	$\lambda f : (A \otimes B) \multimap C \;.\; \lambda a : A \;.\; \lambda b : B \;.\; f(a,b) : C$
$\overline{\mathsf{A}}'$	$\lambda f : (A \multimap B) \multimapinv C \;.\; \lambda a : A \;.\; \lambda y \cdot\cdot B \;.\; f(a,y) \cdot\cdot C$
$\overline{\mathsf{A}}''$	$\lambda f : (A \multimapinv B) \multimapinv C \;.\; \lambda x \cdot\cdot A \;.\; \lambda b : B \;.\; f(x,b) \cdot\cdot C$
C	$\lambda(a,b) : A \otimes B \;.\; (b,a) : B \otimes A$
$\overline{\mathsf{C}}$	$\lambda(f;f') : A \multimap B \;.\; (f';f) : B \multimapinv A$
$\overline{\mathsf{C}}'$	$\lambda(f;f') : A \multimapinv B \;.\; (f';f) : B \multimap A$
D	$\lambda(f,c) : (A \multimap B) \otimes C \;.\; \lambda a : A \;.\; (f(a),c) : B \otimes C$
D'	$\lambda(f,c) : (A \multimapinv B) \otimes C \;.\; \lambda x \cdot\cdot A \;.\; (f(x),c) \cdot\cdot B \multimapinv C$
E	$\lambda(a,b) : A \otimes B \;.\; (f(a),g(b)) : A' \otimes B'$
$\overline{\mathsf{E}}$	$\lambda h : A' \multimap B \;.\; ghf : A \multimap B'$
$\overline{\mathsf{E}}'$	$\lambda h : A \multimapinv B' \;.\; g'hf' : A' \multimapinv B$

Table 3. Transformations Associated to Rules of S2

Rules $\mathsf{E} - \overline{\mathsf{E}}'$ assume that $A \vdash A'$ is realized by $(f;f') : A \multimap A'$ and $B \vdash B'$ by $(g;g') : B \multimap B'$.

Rule A transforms evidence $((a,b),c)$ for $(A \otimes B) \otimes C$ to $(a,(b,c))$ as evidence for $A \otimes (B \otimes C)$. Rule $\overline{\mathsf{A}}$ 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 $\overline{\mathsf{A}}'$ maps witness f for $(A \multimap B) \multimapinv C$ 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 5.3. *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. $\square$

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.

Chapter 6

Transformational Logic

6.1 Natural Transformations

Transformation is fundamental to logic, mathematics, and computer science. First order logic transforms premises into conclusions while equational logic transforms terms into terms. In mathematics algebra transforms structures homomorphically into other structures while geometry transforms spaces continuously, linearly, etc. into other spaces. In computer science, programs operate by transforming data structures into other data structures, while programmers develop software by transforming available programs, data specifications, and process specifications in order to match them up with software requirements.

Consider for example functions between two sets A and B. Now there are many such functions, in fact $|B|^{|A|}$ of them. When $A = B = \{0, 1\}$ we have names for all of them: the identity, the two constant functions, and the "twist map" or Boolean negation. But for much larger sets, most of the functions between them are as anonymous as ants in a colony. However we do have names for certain ants such as the queen. Is there a counterpart for functions between sets?

One approach to limiting to namable functions is by fiat: adopt a system of names such as the λ-calculus. The dialectic λ-calculus introduced in the previous chapter achieves this. It provides 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 by induction on length of derivation, starting with the n-tuple of identity transformations named by Axiom T.

The trouble with such syntactically defined classes of transformations is that it is easy to imagine extending the class by extending the λ-calculus to something richer. What is so special about the λ-calculus that we should stop there? The intrinsic interest in the class would be more compelling if it had some other characterization making no mention of the λ-calculus.

The functions from A to $2 = \{0, 1\}$ are a case in point, being understandable as bit vectors. In the absence of any information about the elements of A, we can nevertheless name and even reason about the two constant functions. We can write these as $\lambda x.0$ and $\lambda x.1$, these constants being available because of the involvement of the set containing them. As an example of such reasoning, any function from 2 to B when composed with either of these two constant functions is itself a constant function from A to B, an inference that required no knowledge of A or B.

Can we identify and reason about any other functions from A to 2? Well, if we knew that A were the set of integers then a great many predicates would immediately spring to mind: positive, even, square, prime, and so on, and for some of these predicates such as primality the literature contains a wealth of reasoning.

But if we are given no information at all about the set A then we are hard pressed to name any further functions. And what about other situations, such as functions from 2 to A?

Can our intuition of what is namable be formalized?

A reasonable first thought might be that a namable function $f : A \to 2$ should be *permutation invariant.* That is, for any bijection $\pi : A \to A$, we require $f = f\pi$. And it is easy to see that the constant functions to 2 are all and only those functions $A \to 2$ that are *invariant* in this sense.

But suppose now that we take $f : A \to 2$ to be the constantly one function in the case that A contains $\sqrt{17}$, and the constantly zero function otherwise. We have certainly named a function, and it is certainly a constant function, as we have just proved it must be. But it is also clear that we have exploited a loophole in our definition of "namable" that allows arbitrarily absurd names that depend on the choice of A.

We can dispose of that example by broadening the class of bijections to $\pi : A \to B$. This entails modifying the requirement $f = f\pi$ to $f_A = f_B\pi$ where $f_A : A \to 2$ and $f_B : B \to 2$ so that the types make sense. But now we have a *polymorphic* requirement, imposed not on single functions but on families f_A of functions indexed by the class of all sets A. We take "transformation" to denote such a family of functions, or "polymorphic function."

But this too fails: let f_A be the constantly one function when the number of elements of A is finite and prime, say, and zero otherwise.

So bijections aren't good enough either. But we can easily dispose of this example by further broadening our requirement to invariance under *arbitrary* functions $g : A \to B$.

Call a transformation from A to 2 meeting this criterion *natural.*

Proposition 6.1. *There are two natural transformations from A to 2, namely the two constant functions chosen independently of A.*

Proof. The two constant functions are easily verified to be natural. For the converse, let f_A be natural, let $a \in A$, and let $g_a : 1 \to A$ (where $1 = \{\star\}$) be the function satisfying $g_a(\star) = a$. Then $f_A(a) = f_A g_a(\star) = f_1(\star)$. Hence f_A is a constant function whose value depends only on the choice of f_1. $\square$

Earlier we mentioned the converse problem of naming functions from 2 to A. The corresponding naturality condition here is the requirement that, for all functions $g : A \to B$, $gf_A = f_B$. Clearly there can be no natural transformation from 2 to A as A could be empty.

Now we have two separate notions of naturality, one for each of $A \to 2$ and $2 \to A$, along with two different results for them. We want to put them on the same footing. A reasonable first step would be to consider natural transformations from A to B. What we have observed is that when A is unknown and B is known, there would appear to be $|B|$ natural transformations from A to B, namely the constant functions, while when A is known and B is unknown, there is one if A is empty and otherwise none. Going beyond this, it is reasonable to suppose that if both A and B are known then all functions from A to B ought to be natural, while if neither is known then there should be none.

The appropriate formalization of this intuitively clear notion of naturality is due to Eilenberg and Mac Lane [EML42b, II.12]. There they defined naturality just for group homomorphisms, but subsequently generalized the concept to families of morphisms in arbitrary categories [EML42a]. To define naturality they had first to define functor, but to define functor they had first to define category.

Given two categories C, D and two functors $F, G : C \to D$, a transformation $\tau : F \to G$ is a family τ_a of morphisms of D indexed by objects a of C. A transformation is ***natural*** when it satisfies $G(f)\tau_a = \tau_b F(f)$ for all objects a, b and morphisms $f : a \to b$ of D, i.e. when the following diagram commutes.

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

For the examples thus far, take $C = D = \mathsf{Set}$, the category whose objects are sets and whose morphisms are functions. Represent an unknown set by the identity functor $I : \mathsf{Set} \to \mathsf{Set}$, which sends all sets and functions to themselves, and a known set A by a constant functor $K_A : \mathsf{Set} \to \mathsf{Set}$ defined as $K_A(B) = A$ for all sets B and $K_A(f) = 1_A$, the identity function on A, for all functions f between any two sets. It is now a straightforward exercise to rephrase the above special-case definitions and their associated propositions and proofs for this formalization of "natural," and to verify the above conjectures for the cases, A and B both known, and both unknown.

So far we have only considered two extremes: no information and total information about a set. A more common situation is partial information. For example we may have an unknown square A^2. The fact that A^2 is a square gives us a little more information to go on.

Exercise. (i) There are two natural transformations from A^2 to A, namely the two projections. (Hint: set A to $2 = \{0, 1\}$, show that only two of the 16 functions from 2^2 to 2 are natural, then deduce that the choice for $A = 2$ determines f_A for all other A.)

(ii) There is only one natural transformation from A to A^2, namely the diagonal function $f(a) = (a, a)$.

(iii) There are four from A^2 to 2, namely the two constants, equality, and its negation. (How nice that equality is natural.)

Problem. How many from A^3 to 2? (There are 32 obvious ones, are there any more?)

Instead of the unknown square A^2 we may have the unknown power set 2^A. Here there are at least three functors we could associate with 2^A, taking $f : A \to B$ to the inverse image function $f^{-1}(X) = \{a \in A | f(a) \in X\}$, the direct image function $f(X) = \{f(a) | a \in X\}$, and its dual $f(X) = \overline{\{f(a) | a \notin X\}}$. For each we may ask the same questions as above with 2^A in place of A^2. Clearly there are no natural transformations from 2^A to A because of the case $A = \emptyset$, but the other cases are more fun to pursue.

Other kinds of object. Instead of functions from the set A to itself we could ask about the naturality of homomorphisms from a group G to itself.

Proposition 6.2. *The natural transformations on the identity functor on the category* **Grp** *of groups are in bijection with the integers.*

Rather than proving this directly we may prove both it and the fact that there is only one natural transformation from the set A to itself from the same master theorem about varieties. A ***variety*** is a category whose objects are the models of some equational theory and whose morphisms are the homomorphisms determined by the signature of that theory. The n-ary operations of a variety are the elements of the free algebra on n generators, i.e. the terms built from variables $x_1, \ldots, x_n$ up to provable equality. A variety is ***commutative*** when all its unary operations commute.

We say that a term $t(x)$ in one variable ***realizes*** the operation $t_A = \lambda x : A.t(x)$ on algebra A, where $x : A$ means that the values of x range over A. For every algebra A of a commutative variety, every term $t(x)$ realizes an endomorphism (self-homomorphism) $t_A : A \to A$.

Theorem 6.3. *The natural transformations on the identity functor on a commutative variety are the unary operations of the variety realized on its algebras.*

In particular for pointed sets there are two unary operations, namely the identity function and the constant function, and therefore two natural transformations on I_{Set^*}, the identity functor on the category Set^* of pointed sets. For monoids the unary operations are the nonnegative integer scalar multiples, while for groups they are all integer scalar multiples, proving Proposition 6.2.

Proof. We first verify naturality of the family of realized operations. For any unary operation $t(x)$ realized as $t_A : A \to A$ and any homomorphism $h : A \to B$, the naturality condition $h(t_A(a)) = t_B(h(a))$ holds because h is a homomorphism.

To see that there are no other natural transformations, let F be the free algebra on one generator x. The endomorphisms of F are exactly the unary operations t_F, since these correspond to where the generator of F is sent. (This generalizes the situation in Set where there is just one function on the singleton.) Now consider any homomorphism $h : F \to A$ where A is any algebra of the variety. There is one such for each element of A, determined by where h sends the generator $x \in F$, since F is free. Hence by naturality the choice of t_F determines t_A at every element $a \in A$: $t_A(a) = h(t_F(x))$. $\square$

6.2 Naturality in Chu

One difference between Chu spaces and sets is that Set is a cartesian closed category. The one product $A \times B$ does double duty as categorical product with projections and as tensor product left adjoint to the internal hom $A \to B$. Chu is not cartesian closed and has these as distinct products, respectively $\mathcal{A}\&\mathcal{B}$ and $\mathcal{A} \otimes \mathcal{B}$. This greatly increases the number of functors constructible with products.

As in the previous chapter we restrict ourselves to the tensor product $\mathcal{A} \otimes \mathcal{B}$ as being more fundamental to Chu by virtue of its intimate connection with the internal hom or linear implication $\mathcal{A} \multimap \mathcal{B}$. Since the carrier of $\mathcal{A} \otimes \mathcal{B}$ is just cartesian product of carriers, any natural transformation in Chu (for any Σ) between functors built with tensor products is also natural in Set. Hence the natural transformations in Set serve as an upper bound on those in Chu for such functors.

Recall that in Set there were two natural transformations from A^2 to A, and one from A to A^2. In Chu_2, there are no natural transformations between $\mathcal{A}^2$ $(= \mathcal{A} \otimes \mathcal{A})$ and $\mathcal{A}$, in either direction.

The problem is not with naturality but continuity. Naturality limits the functors from $\mathcal{A}^2$ to $\mathcal{A}$ to being just the two projections, which fail continuity at certain $\mathcal{A}$'s, for example $\mathcal{A} = \boxed{\begin{smallmatrix}00\\01\end{smallmatrix}}$. Here $\mathcal{A}^2 = \boxed{\begin{smallmatrix}00\\00\\00\\01\end{smallmatrix}}$ and the 4×2 crossword solutions constituting the two projections are $\boxed{\begin{smallmatrix}00\\01\\00\\01\end{smallmatrix}}$ and $\boxed{\begin{smallmatrix}00\\00\\01\\01\end{smallmatrix}}$, whose right-hand column does not appear in $\mathcal{A}^2$.

This choice of $\mathcal{A}$ does not however rule out the diagonal function $d : \mathcal{A} \to \mathcal{A}^2$, which is continuous there. However the diagonal is not continuous at $\mathcal{A} = \boxed{\begin{smallmatrix}01\\10\end{smallmatrix}}$, for which $\mathcal{A}^2 = \boxed{\begin{smallmatrix}01\\10\\10\\01\end{smallmatrix}}$. The diagonal map is represented by the crossword solution $\boxed{\begin{smallmatrix}01\\01\end{smallmatrix}}$, whose columns do not appear in $\mathcal{A}$. (This $\mathcal{A}$ also makes the projections discontinuous.)

There are however two natural transformations from $\mathcal{A}^2$ to itself, sending (a, b) to respectively (a, b) and (b, a). Set of course has these, as well as (a, a) and (b, b), namely the composites $A^2 \xrightarrow{\pi_i} A \xrightarrow{d} A^2$ where π_0, π_1 are the two projections. A similar argument to the above rules out the latter two for Chu.

6.3 Dinatural Transformations

For both Set and Chu, not to mention other closed categories, the internal hom or implication $A \to B$ ($\mathcal{A} \multimap \mathcal{B}$ in Chu) is a useful functor to which we would like to extend all of the above. In any closed category C the type of $\to$ is $C^{\mathrm{op}} \times C \to C$. That is, A occurs contravariantly or negatively in $A \to B$, while B occurs covariantly or positively.

A difficulty enters when we allow the same variable to appear both positively and negatively. While we can treat A^2 as a functor from C to C without difficulty, we cannot do so with $A \to A$ because there is no sensible extension of the functor to morphisms f. In $f \multimap f$, f does not know which way to turn.

Such cases of mixed variance can be treated by generalizing naturality to dinaturality, connoting "naturality in both directions." Given functors F, G : $C^{\mathrm{op}} \times C \to D$, a "diagonal transformation" $\tau : F \to G$, one whose family τ_{aa} of morphisms in D is indexed only by the diagonal of $C^{\mathrm{op}} \times C$, is called ***dinatural*** when for every $f : a \to b$ in C, the diagram on the left below commutes.

$$
\begin{tikzcd}
F(p;p) \arrow[r, "\tau_p"] & G(p;p) \arrow[d, "G(p;f)"] \\
F(q;p) \arrow[u, "F(f;p)"] \arrow[d, "F(q;f)"'] & G(p;q) \\
F(q;q) \arrow[r, "\tau_q"] & G(q;q) \arrow[u, "G(f;q)"']
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \arrow[r, "\tau_p"] \arrow[d, equal] & H(p;p) \arrow[d, "H(p;f)"] \\
1 \arrow[d, equal] & H(p;q) \\
1 \arrow[r, "\tau_q"] & H(q;q) \arrow[u, "H(f;q)"']
\end{tikzcd}
$$

The diagram on the right shows the simplification possible in MLL. We can transpose each morphism $\tau_p : F(p;p) \to G(p;p)$ to an element τ_p of $H(p;p)$, where $H(p;q)$ is defined as $(F \multimap G)(p;q)$. We refer to the resulting dinatural transformation as a *dinatural element*. Here $1 = \boxed{01}$ is Chu_2's tensor unit.

We may think of dinatural transformations as variable morphisms, and of dinatural elements as variable elements or points. Now dinaturals are defined only on the diagonal of $(C^{\mathrm{op}})^n \times C^n$, namely at pairs of n-tuples $(p_1, \dots, p_n; p_1, \dots, p_n)$, with the further requirement that the p_i's be objects. This restriction makes the $(C^{\mathrm{op}})^n$ portion of parameter space redundant for dinaturals. Hence in thinking of dinaturals as variable morphisms the parameter space over which those morphisms vary can be taken to be simply the objects of C^n, and the generic parameter can therefore be simplified to $(p_1, \dots, p_n)$.

In the next two sections we develop soundness and (to the extent possible) full completeness results for Chu_2 based on dinaturality semantics, with similar results to Blute and Scott [BS96] who studied the corresponding question for vector spaces perturbed by group actions. We follow this with an example showing the limitations of dinaturality for Chu spaces, motivating the treatment of (binary) logicality as a suitable strengthening of dinaturality, using methods analogous to those of Plotkin [Plo80].

6.4 Soundness of MLL for dinaturality semantics

The following enunciation of soundness holds more generally for any $*$-autonomous category [Bar79], of which Chu is just an instance. We state and prove it just for Chu to avoid the considerable work of defining a general $*$-autonomous category, of which Chu is a reasonably representative example.

Theorem 6.4. *(Soundness) Every n-variable theorem of MLL, interpreted as an n-ary operation on* Chu*, contains a dinatural element.*

Proof. We prove this by induction on the length of proofs in System S1, whose steps are interpreted according to Table 3 as transforming transformations starting with the identity transformation interpreting Axiom T at the polymorphic object $(p_1, \ldots, p_n)$ of Chu^n. Although Table 3 is given for System S2, the latter can be understood as simply System S1 with a specified orientation for each $⅋$, yielding the evident application of Table 3 to System S1.

To establish dinaturality of this element, observe that for any n-tuple $\langle f_i : p_i \to q_i \rangle$ of Chu transforms, $F(p_1, \ldots, p_n; f_1, \ldots, f_n) \circ (1_{p_1}, \ldots, 1_{p_n})$ and $F(f_1, \ldots, f_n; q_1, \ldots, q_n) \circ (1_{q_1}, \ldots, 1_{q_n})$ both simplify to $(f_1, \ldots, f_n)$.

The structural rules for associativity and commutativity correspond to isomorphisms, which therefore cannot harm dinaturality.

For linear distributivity, Rule D, suppose $\varphi((A \multimap B) \otimes C)$ has a dinatural element. We exhibit the corresponding dinatural element of $\varphi(A \multimap (B \otimes C))$. We prove this first for the case where $(A \multimap B) \otimes C$ is the whole formula, then generalize.

Let the formulas A, B, C denote the (variable) Chu spaces $\mathcal{A}, \mathcal{B}, \mathcal{C}$ respectively. A dinatural element of $(\mathcal{A} \multimap \mathcal{B}) \otimes \mathcal{C}$ is a pair (g, c) where $g : \mathcal{A} \to \mathcal{B}$ is a dinatural element of $\mathcal{A} \multimap \mathcal{B}$, equivalently a dinatural transformation from $\mathcal{A}$ to $\mathcal{B}$, and c is a point of $\mathcal{C}$. There is an evident choice of corresponding dinatural element of $\mathcal{A} \multimap (\mathcal{B} \otimes \mathcal{C})$, which we can write as $\lambda a.(g(a), c)$, viewable as a transformation from $\mathcal{A}$ to $\mathcal{B} \otimes \mathcal{C}$.

Claim: The following diagram commutes. (Here $f : p \to q$ is as before the generic test of (di)naturality while g, g' are the values of the given dinatural transformation from $\mathcal{A}$ to $\mathcal{B}$ at p, q respectively.)

$$
\begin{array}{ccc}
\mathcal{A}(p;p) & \xrightarrow{\lambda a.(g(a),c)} & \mathcal{B}(p;p) \otimes \mathcal{C}(p;p) \\
{\scriptstyle \mathcal{A}(f;p)} \uparrow & & \downarrow {\scriptstyle \mathcal{B}(p;f) \otimes \mathcal{C}(p;f)} \\
\mathcal{A}(q;p) & & \mathcal{B}(p;q) \otimes \mathcal{C}(p;q) \\
{\scriptstyle \mathcal{A}(q;f)} \downarrow & & \uparrow {\scriptstyle \mathcal{B}(f;q) \otimes \mathcal{C}(f;q)} \\
\mathcal{A}(q;q) & \xrightarrow{\lambda a.(g'(a),c)} & \mathcal{B}(q;q) \otimes \mathcal{C}(q;q)
\end{array}
$$

This follows from the fact that the maps associated with B and C commute separately, by dinaturality of g and c, whence so does their tensor product. We conclude that $\lambda a.(g(a), c)$ is dinatural.

When $(A\multimap B) \otimes C$ appears as a proper subformula of $\varphi((A\multimap B) \otimes C)$, we can view the latter as having the form $D_1 \multimap (E_1 \otimes (D_2 \multimap (E_2 \otimes \ldots (D_k \multimap (E_k \otimes ((A \otimes B)\multimap C))) \ldots)))$. This is the same situation as before, except that f now becomes a function of the Chu space $\mathcal{D}_1 \otimes \mathcal{D}_2 \otimes \ldots \otimes \mathcal{D}_k$, where each D_i denotes $\mathcal{D}_i$. $\square$

Like f and c, the $\mathcal{D}_i$'s are polymorphically dependent on the parameter space Chu^n; without the polymorphism convention this picture would be much messier.

6.5 Full completeness of BMLL for dinaturality

In this section we prove that System S1 of Chapter 5, restricted to the binary fragment of MLL, is fully complete for the dinaturality semantics of Chu_2 spaces.

Binary MLL (BMLL) is that fragment of MLL for which each variable occurs exactly once with each sign. The interest in this small fragment of MLL is that dinaturality in Chu_2 is well-behaved there, in particular all dinatural transformations of BMLL can be accounted for as abstract cut-free proofs of multiplicative linear logic. This type of result is called *full completeness.*

Full completeness may be defined formally in terms of representation of proofs by transformations. We are given a "syntactic" category C consisting of formulas as objects and proofs or entailments as morphisms between those formulas. We are further given a "semantic" category D . The formula c, as an object of C, is represented by the object $F(c)$ of D, while the proof $\pi : c \to c'$ is represented by the morphism $F(\pi) : F(c) \to F(c')$. This representation is faithful and/or full according to whether F is a faithful or full functor respectively. When it is both, we say that D together with F constitutes a fully complete semantics for C.

For the case at hand, BMLL interpreted over Chu, the objects of C are the formulas of BMLL and its morphisms are the cut-free proofs between BMLL formulas. The objects of D are functors built from projections using tensor and perp, while the morphisms are dinatural transformations between those functors, which in this restricted situation compose to form a category. (In general dinatural transformations fail to compose.)

When distinctions between morphisms of a homset of C are ignored, we ask only whether there exists a morphism (proof) from c to c'. The functor itself witnesses soundness of such a proof $\pi : c \to c'$ with an interpretation $F(\pi)$ of that proof. Fullness ensures only that if there exists a morphism from $F(c)$ to $F(c')$ then there exists a proof from c to c', which amounts to the usual notion of completeness in logic, that every valid sentence have a proof.

When the morphisms of a homset are distinguished, faithfulness and fullness together entail a bijection between syntactic proofs and their semantic interpretations.

We begin our treatment of the BMLL of Chu spaces with ordinary completeness, namely that every BMLL formula with a semantic proof has a syntactic proof.

Proposition 6.5. *Every BMLL formula A containing a dinatural is a theorem of MLL.*

Proof. We follow the line of argument initiated by Abramsky and Jagadeesan applicable in the presence of MIX, and generalized by Hyland and Ong in the absence of MIX. Call an MLL formula ***semisimple*** when it is a *par* of tensor products of literals. (With MIX one uses the stronger notion of simple in which the tensor products contain at most two literals.) A semisimple formula can equivalently be described as being in disjunctive normal form. Hyland and Ong show that if all the semisimple consequents of an MLL formula are theorems then so is that formula. Hence to show that a formula is not a theorem it suffices to show that none of its semisimple consequents are.

We first treat the case of balanced binary, each variable occurring twice, once positively and once negatively. This case uniquely determines a proof-net.

We use the Danos-Regnier criterion to show that every semisimple nontheorem A of BMLL contains no dinatural. We do this by giving a particular assignment of Chu spaces to variables for which A evaluates to the 0×1 Chu space 0.

A semisimple BMLL formula determines an undirected graph whose vertices are the tensor products and whose edges are complementary pairs of literals each in one of the tensor products. Since there is only one $\parr$, at the root of A, its switchings are immaterial. Incorrect proof-nets as such graphs are therefore either disconnected or contain a cycle.

Case 1. The graph of A is disconnected.

We use the fact that all occurrences of a variable must occur within a connected component, since the formulas are binary. Set the variables of one component to the 1×1 Chu space $\boxed{0}$, call this W for white. Set the remaining variables to the 1×1 Chu space $\boxed{1}$, call this B for black.

Easy calculations show $W^\perp = W \otimes W = W \parr W$, $B^\perp = B \otimes B = B \parr B$, $B \parr W = 0$ where 0 is the 0×1 (empty discrete) Chu space. It follows that each clause in the W component evaluates to W, as does the *par* of those clauses. Similarly each of the remaining clauses evaluates to B, as does their *par*. The *par* of the B clauses and the W clauses is then 0, whence no element can exist let alone a dinatural one.

We illustrate this case with the formula $P \parr P^\perp \parr Q \parr Q^\perp$, all clauses of which are singletons. The P link connects the first two clauses while the Q link connects the other two, but this leaves two connected components. Setting P to white and Q to black makes $P \parr P^\perp$ white, $Q \parr Q^\perp$ black, and their par 0.

Case 2. The graph of A is connected and contains a cycle.

Select a cycle and orient it, and hence its links. Orient the remaining links to point towards the cycle, in the sense that the target end of the link is closer to the cycle than the source end, with ties broken arbitrarily. Now assign 0 or $\top = 0^\perp$ to the variables in such a way that every link is oriented from 0 to $\top$. It is immediate that all clauses on the cycle contain a 0 literal, and an argument by induction on distance from the cycle extends this property of clauses to the remainder of the graph. Since $0 \otimes \mathcal{A} = 0$ for all Chu spaces $\mathcal{A}$, all clauses evaluate to 0 and hence so does their *par*.

We illustrate case 2 with the formula $(A \otimes B) \parr (A^\perp \otimes B^\perp \otimes C) \parr C^\perp$. The A and B links between the first two clauses create a cycle with two edges. To orient it properly set A to 0 and B to $\top$ (or vice versa). This makes the first two clauses 0. Setting C to $\top$ zeros both $C^\perp$ and the whole formula.

This disposes of the balanced binary case. In the unbalanced case, the two-occurrence limit means that at least one literal has no oppositely signed counterpart. Assign that literal 0, and all other literals W (the 1×1 "white" Chu space, but B would do as well). Then all clauses evaluate to either 0 or W, and there is at least one 0 clause. Since $W \parr W = W$ and $W \parr 0 = 0 \parr 0 = 0$, it follows that the formula evaluates to 0. Hence no dinatural is possible for the unbalanced binary case, or for that matter any formula in which all occurrences of some variable have the same sign. ☐

Full Completeness. We now prove that every dinatural element of the functor denoted by a binary MLL theorem A corresponds to a cut-free proof net of A. In the binary case this is the rather unexciting result that a theorem has only one dinatural element. We shall first show this for semisimple binary theorems and then extend to general binary formulas.

Proposition 6.6. *(Full completeness) Every semisimple theorem of BMLL has at most one dinatural.*

Proof. Let A be such a theorem. Being binary, it uniquely determines a graph whose edges are links $P - P^\perp$ as in the previous section. Being a theorem, its graph is an undirected tree. Make it a rooted oriented tree by choosing any singleton clause for the root and orienting the edges to point towards the root.

We shall show by induction on the number n of variables that a dinatural element of A is uniquely determined at every assignment (point of parameter space). We shall start from the assignment of all 1's and proceed to an arbitrary assignment in n steps, showing that at each intermediate assignment all dinaturals have the same value at that assignment.

For the basis of the induction we take the assignment assigning the Chu space $\perp$ to the distal end of each link (that is, to the literal associated with that end of that link) and hence 1 to its near end. Since there is one edge leaving each clause except the root, which has no edges leaving it, it follows that the root clause evaluates to 1 while the remaining clauses evaluate to $1 \otimes 1 \otimes \ldots \otimes \perp \otimes \ldots \otimes 1 = \perp$. Hence A evaluates to 1. But 1 has only one element, so every dinatural element must be the same here.

We now move from this assignment to an arbitrary assignment. For simplicity of exposition assume that the distal literal occurrence of each P is negative, substituting $P^\perp$ for P if necessary, so that all variables have value 1 before the move. Assign Chu spaces arbitrarily to variables. We shall transform the all-ones assignment into this arbitrary assignment one variable at a time, at each stage showing that there is still at most one dinatural element.

Enumerate the variables in order of increasing distance from the root, with ties broken arbitrarily. For each variable P we first change the value of its positive occurrence to the Chu space chosen above for P, call it $\mathcal{A}$, and then change the value of its negative occurrence $P^\perp$ from $\perp$ to $\mathcal{A}^\perp$.

The value of A during the processing of one such variable P can be expressed as $(\mathcal{B} \parr P^\perp) \parr (P \otimes \mathcal{C})$ where $\mathcal{B}$ and $\mathcal{C}$ remain fixed while P varies.

$\mathcal{C}$ is the value, at the start of this stage, of the clause containing the positive occurrence of P. $P^\perp$ comes from the clause containing $P^\perp$, all of whose literals other than $P^\perp$ are positive and have value 1 throughout this stage, being further from the root than P. $\mathcal{B}$ denotes the combined value of all the remaining clauses (as combined by $\parr$). It will be convenient to rename $\mathcal{B}$ to $\mathcal{B}^\perp$ allowing us to rewrite this expression as $(\mathcal{B} \otimes P) \multimap (\mathcal{C} \otimes P)$.

At the start of this stage $P = 1$, so $A = \mathcal{B} \multimap \mathcal{C}$. By the induction hypothesis all dinatural elements have the same value here, call it $f : \mathcal{B} \to \mathcal{C}$. Transform the positive (righthand) occurrence of P from 1 to $\mathcal{A}$ via an arbitrarily chosen map $a : 1 \to \mathcal{A}$ (an element of $\mathcal{A}$). This determines a map $\mathcal{B} \multimap (\mathcal{C} \otimes a) : (\mathcal{B} \multimap (\mathcal{C} \otimes 1)) \to (\mathcal{B} \multimap (\mathcal{C} \otimes \mathcal{A}))$. This map sends the dinatural f to $\lambda c.\langle f(c), a\rangle$, a function from $\mathcal{B}$ to $\mathcal{C} \otimes \mathcal{A}$.

Now consider $(\mathcal{B} \otimes \mathcal{A}) \multimap (\mathcal{C} \otimes \mathcal{A})$. The above map $a : 1 \to \mathcal{A}$ determines a map $(\mathcal{B} \otimes a) \multimap (\mathcal{C} \otimes \mathcal{A}) : ((\mathcal{B} \otimes \mathcal{A}) \multimap (\mathcal{C} \otimes \mathcal{A})) \to ((\mathcal{B} \otimes 1) \multimap (\mathcal{C} \otimes \mathcal{A})$. Any dinatural element g at $(\mathcal{B} \otimes \mathcal{A}) \to (\mathcal{C} \otimes \mathcal{A})$ is sent by this map to $\lambda c.g(c, a)$, which by dinaturality must equal $\lambda c.\langle f(c), a\rangle$ as defined above. This must hold for all elements $a \in \mathcal{A}$, whence g must satisfy $g(c, a) = \langle f(c), a\rangle$ for all $c \in \mathcal{B}$ and $a \in \mathcal{A}$. But this uniquely determines g, whence $(\mathcal{B} \otimes \mathcal{A}) \multimap (\mathcal{C} \otimes \mathcal{A})$ can contain at most one dinatural.

Proceeding in this way for all variables of A, we arrive at the arbitrarily chosen assignment to all variables, at which A still has at most one dinatural element. □

We conclude by extending the above theorem from the semisimple case to general formulas of BMLL.

Proposition 6.7. *(Full completeness) Every theorem of BMLL has at most one dinatural.*

Proof. It suffices to show that linear distributivity maps dinaturals injectively. For if our claim were false and some BMLL theorem had two or more dinaturals, each of its semisimple consequents would have two or more as well, contradicting the previous theorem.

Consider first the top-level case where $(A \multimap B) \otimes C$ derives $A \multimap (B \otimes C)$. Here (f, c) is mapped to $\lambda a.(f(a), c)$. If $(f, c) \neq (f', c')$ then either $f \neq f'$ or $c \neq c'$. In either case $\lambda a.(f(a), c) \neq \lambda a.(f'(a), c')$.

In the general case the dinatural element being so transformed becomes a function from $D_1 \otimes D_2 \otimes \ldots \otimes D_k$ to $A \multimap (B \otimes C)$. Having two distinct dinaturals means that for some tuple $(d_1, \ldots, d_k)$ we have $(f, c) \neq (f', c')$ as above. But then at the same tuple we have the corresponding $\lambda a.(f(a), c) \neq \lambda a.(f'(a), c')$. □

Combining this proposition with soundness, we have

Theorem 6.8. *BMLL is fully complete for the dinaturality semantics of* Chu_2.

6.6 Incompleteness of MLL for dinaturality semantics in Chu

The full completeness of MLL for dinaturality semantics, which in the previous section we showed for two occurrences of a variable, does not extend to formulas containing four occurrences of a variable.

Proposition 6.9. Chu_2 *contains exotic dinaturals on* $A \multimap A$, *in the sense that they correspond to no MLL proof of* $(A \multimap A) \multimap (A \multimap A)$.

Proof. We exhibit a specific exotic dinatural transformation τ that will depend on the following notion. Call those Chu spaces which contain both a row of all zeros and a column of all zeros the Type I spaces, and the rest the Type II spaces. We claim that $\mathcal{A} \multimap \mathcal{A}$ 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. 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 state, the latter when it is $\mathcal{B}$ that lacks the zero column or $\mathcal{A}$ that lacks the zero state. 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.

By soundness of dinaturality semantics, all cut-free proofs have already been paired up with dinaturals none of which are of the above form. Hence the latter are exotic. □

Chapter 7

Full completeness of MLL for Chu spaces

(This chapter is adapted from [DHPP99].)

7.1 Logical Transformations

The preceding chapter concluded on a negative note: dinaturality semantics is incomplete for Chu spaces. Is there some stronger criterion than dinaturality that works with mixed variance? In this chapter we show that binary logical transformations strengthen dinatural transformations sufficiently to achieve full completeness of MLL, at the cost of narrowing the scope of their applicability to functors built with the operations of MLL.[1]

We extend the MLL operations to act not on morphisms but on *binary relations* $R : \mathcal{A} \rightarrow\!\!\!\!\times \mathcal{B}$, yielding binary relations $F(\mathcal{R}) : F(\mathcal{A}) \rightarrow\!\!\!\!\times F(\mathcal{B})$. Unlike functions, binary relations are closed under converse, which neatly sidesteps the main difficulty with mixed variance.

Definition 1. A ***Chu relation*** $R = (R^+, R^-) : \mathcal{A} \rightarrow\!\!\!\!\times \mathcal{B}$ between Chu spaces $\mathcal{A} = (A, r, X)$ and $\mathcal{B} = (B, s, Y)$ is a pair of ordinary binary relations $R^+ \subseteq A \times B$ and $R^- \subseteq X \times Y$ meeting the following adjointness condition: for all a, b, x, y such that $aR^+ b$ and $xR^- y$, we have $a \cdot x = b \cdot y$. $\square$

In the special case when R^+ is a function $A \to B$ and R^- a function $Y \to X$, a Chu relation is exactly a Chu transform. Hence Chu relations generalize Chu transforms.

[1] Actually the definition of logical relations between Chu spaces via the Chu construction applied to the bicomplete cartesian closed category of logical relations between sets allows logical transformations of any arity to work also with all limits and colimits. We have not looked at whether they work with linear logic's exponentials, or with process algebra's concatenation and choice.

We need the following notion both to define the action of MLL operations on Chu relations, and to define logical relations.

Definition 2. Let (A, B, R) be a two-sorted relational structure with one binary relation $R \subset A \times B$. Take a second structure (A', B', R') of the same similarity type. We call a pair of functions $f : A \to A'$, $g : B \to B'$ a ***two-sorted homomorphism*** when for all aRb we have $f(a)R'g(b)$. $\square$

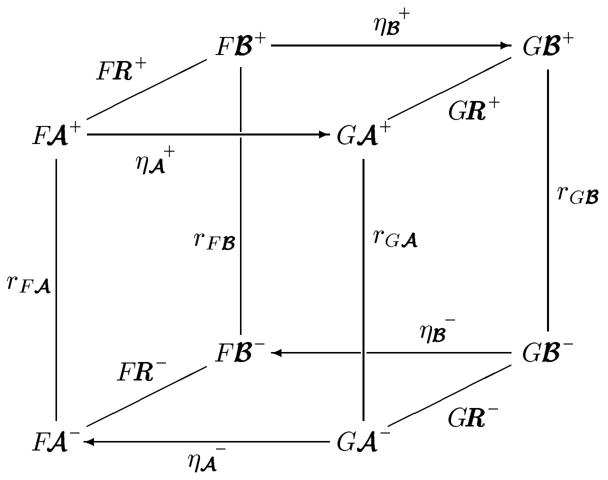
We now define how MLL operations act on Chu relations.

Perp. If $R = (R^+, R^-) : \mathcal{A} \nrightarrow \mathcal{B}$, so that $R^+ \subseteq \mathcal{A}^+ \times \mathcal{B}^+$ and $R^- \subseteq \mathcal{A}^- \times \mathcal{B}^-$ satisfy adjointness, then $R^\perp = (R^-, R^+) : \mathcal{A}^\perp \nrightarrow \mathcal{B}^\perp$, also satisfying adjointness. In the special case where R is a Chu transform from $\mathcal{A}$ to $\mathcal{B}$, $R^\perp$ will be a Chu transform from $\mathcal{B}^\perp$ to $\mathcal{A}^\perp$.

Tensor. Given $R : \mathcal{A} \nrightarrow \mathcal{B}$ and $R' : \mathcal{A}' \nrightarrow \mathcal{B}'$, we define $R \otimes R' : \mathcal{A} \otimes \mathcal{A}' \nrightarrow \mathcal{B} \otimes \mathcal{B}'$ by (i) $(a, a')(R \otimes R')^+(b, b')$ iff aR^+b and $a'R'^+b'$, and (ii) $(f_1, f_2)(R \otimes R')^-(g_1, g_2)$ iff (f_1, g_1) is a two-sorted homomorphism from R^+ to R'^- and (f_2, g_2) is a two-sorted homomorphism from R'^+ to R^-.

This brings us to the main concept for our result, that of logical transformation.

Let $\eta : F \to G$ be a transformation between n-ary MLL operations. Then each component $\eta_{\boldsymbol{\mathcal{A}}} : F\boldsymbol{\mathcal{A}} \to G\boldsymbol{\mathcal{A}}$ is a Chu transform, consisting of an adjoint pair of functions ${\eta_{\boldsymbol{\mathcal{A}}}}^+ : F\mathbf{A}^+ \to G\mathbf{A}^+$ and ${\eta_{\boldsymbol{\mathcal{A}}}}^- : G\boldsymbol{\mathcal{A}}^- \to F\boldsymbol{\mathcal{A}}^-$ (read $F\boldsymbol{\mathcal{A}}^+$ as $(F(\boldsymbol{\mathcal{A}}))^+$). The usual naturality commuting square becomes a pair of squares, one for points, and one for states, which we call the ***positive*** and ***negative logicality squares***, the top and bottom faces of the following cube.



In the cube $\boldsymbol{\mathcal{A}} = (\mathcal{A}_1, \ldots, \mathcal{A}_n) \in \mathsf{Chu}^n$, $\boldsymbol{\mathcal{B}} = (\mathcal{B}_1, \ldots, \mathcal{B}_n) \in \mathsf{Chu}^n$, $\boldsymbol{R} = R_1, \ldots, R_n$ is a tuple of Chu relations $R_i : \mathcal{A}_i \nrightarrow \mathcal{B}_i$, $F\boldsymbol{R}^+$ abbreviates $(F(\boldsymbol{R}))^+$, and $r_{\mathcal{C}}$ denotes the matrix of $\mathcal{C}$. The positive logicality square, namely the top face of the cube, consists of a two-sorted homomorphism between $F(\boldsymbol{R})^+ \subseteq$

$F(\boldsymbol{\mathcal{A}})^+ \times F(\boldsymbol{\mathcal{B}})^+$ and $G(\boldsymbol{R})^+ \subseteq G(\boldsymbol{\mathcal{A}})^+ \times G(\boldsymbol{\mathcal{B}})^+$, namely a pair of functions ${\eta_{\boldsymbol{\mathcal{A}}}}^+ : F(\boldsymbol{\mathcal{A}})^+ \to G(\boldsymbol{\mathcal{A}})^+$ and ${\eta_{\boldsymbol{\mathcal{B}}}}^+ : F(\boldsymbol{\mathcal{B}})^+ \to G(\boldsymbol{\mathcal{B}})^+$ such that $a\,F(\boldsymbol{R})\,b$ implies $\eta_{\boldsymbol{\mathcal{A}}}(a)\,G(\boldsymbol{R})\,\eta_{\boldsymbol{\mathcal{B}}}(b)$ for all $a \in F(\boldsymbol{\mathcal{A}})^+$ and $b \in F(\boldsymbol{\mathcal{B}})^+$.

Definition 3. A transformation η between n-ary MLL operations F and G is ***logical*** when all logicality squares in η are two-sorted homomorphisms. □

A less elementary but faster equivalent definition of both the action of MLL operations on relations and of logicality takes for these cubes the morphisms of **Chu**(Log,**2**). This is the result of applying the Chu construction [Bar79, App.] to Log, the category of binary relations as objects and two-sorted homomorphisms between them, with dualizer the identity relation $1_{\mathbf{2}}$. This approach requires a familiarity with the general categorical Chu construction that we have not presumed here.

7.2 Semisimple full completeness

An MLL formula is ***semisimple*** if it is of the form $⅋_{i \le m}\big(\bigotimes_{j \le k_i} L_{ij}\big)$ where L_{ij} are literals. Our route to full completeness for semisimple formulas passes through the category Coh of coherence spaces, via the map $(\widehat{-})$ constructed above. The primary attraction of Coh is the existence of full completeness results for MLL with MIX, for example [Tan97], which allow us to assign a MIX proof net to every logical transformation in Chu.

In the Lafont-Streicher image of Coh in Chu, the functional behaviour of the logical transformation corresponds to a tuple of lambda calculus terms associated canonically with its MIX proof net. We argue that the behaviour of the logical transformation outside the Coh image is also governed by the lambda terms, by asserting logical relations to hold between arbitrary Chu spaces and their "simulations" in the Coh image. Thus every logical transformation is characterized by a distinct tuple of lambda terms.

Finally we refute MIX, by showing that any such lambda term must use all of its arguments during computation. Hence the corresponding MIX proof net is connected under all switchings and therefore is a proof net, giving us the desired bijection between proofs of semisimple MLL theorems and logical transformations in Chu.

7.3 Coherence spaces

Coherence spaces were the first and indeed motivating model of linear logic [Gir87]. Define the $*$-autonomous category of coherence spaces and linear maps as follows.

Objects. A ***coherence space*** $U = (|U|, \overset{\frown}{\smile}_U)$ is a reflexive undirected graph: a set $|U|$ of ***tokens*** and a reflexive symmetric ***coherence*** relation $\overset{\frown}{\smile}_U \subseteq |K| \times |U|$ between tokens. Define ***strict coherence*** by $u \frown_U v$ iff $u \overset{\frown}{\smile}_U v$ and $u \neq v$, ***incoherence*** by $u \asymp_U v$ iff $\neg(u \frown_U v)$, and ***strict incoherence*** by $u \smile_U v$ iff $\neg(u \overset{\frown}{\smile}_U v)$.

A ***clique*** a in U is a subset a of $|U|$ made of pairwise coherent tokens; an ***anticlique*** x in U is a subset x of $|U|$ made of pairwise incoherent tokens. We write $U^\bullet$ and U° for the sets of cliques and anticliques of U respectively.

Morphisms. A ***linear map*** $l : U \to V$ is a binary relation between tokens $l \subseteq |U| \times |V|$ such that for all $(u,v),(u',v') \in l$, $u \frown_U u' \Rightarrow v \frown_V v'$ and $v \asymp_V v' \Rightarrow u \asymp_U u'$. Composition is usual relational composition, with the usual identities. A linear map l defines a function $[-]l : U^\bullet \to V^\bullet$ from cliques to cliques and a reverse function $l[-] : V^\circ \to U^\circ$ from anticliques to anticliques, with

$$\begin{aligned} [a]l &= \{\, v \in |V| : \exists u \in a\,((u,v) \in l)\,\} \\ l[x] &= \{\, u \in |U| : \exists v \in x\,((u,v) \in l)\,\} \end{aligned}$$

Either $[-]l$ or $l[-]$ determines l completely.

Linear negation: $U^\perp = (|U|, \asymp_U)$, the exchange of coherence and incoherence. On maps $l^\perp : V^\perp \to U^\perp$ is given by $(v,u) \in l^\perp$ iff $(u,v) \in l$.

Tensor product: $|U \otimes V| = |U| \times |V|$ with $(u,v) \frown_{U\otimes V} (u',v')$ iff $u \frown_U u'$ and $v \frown_V v$.

Tensor unit: $|I| = \{\star\}$, with (necessarily) $\star \frown_I \star$. Linear maps $I \to X$ correspond to cliques of X.

The interpretations of $\multimap$ and $\parr$ follow from definitions of $A \multimap B$ as $(A \otimes B^\perp)^\perp$ and $A \parr B$ as $(A^\perp \otimes B^\perp)^\perp$.

7.4 The Lafont-Streicher embedding

Lafont and Streicher [LS91] exhibit a full and faithful functor $\mathrm{LS} : \mathsf{Coh} \to \mathsf{Chu}$. Points are cliques, states are anticliques, and matrix entries are given by intersection: $\mathrm{LS}(U) = (U^\bullet, \sqcap, U^\circ)$, where $a \sqcap x = |a \cap x|$. (Note that a clique and an anticlique can intersect in at most one point.) On linear maps $l : U \to V$, $\mathrm{LS}(l) = ([-]l, l[-]) : \mathrm{LS}(U) \to \mathrm{LS}(V)$.

The embedding is weakly tensorial with tensorial strength $\tau_{UV} : \mathrm{LS}(U) \otimes \mathrm{LS}(V) \to \mathrm{LS}(U \otimes V)$ and $t : \mathbf{1} \to \mathrm{LS}(I)$ as follows:

$$\begin{array}{lcllcl} \tau^+ & : & (a,b) \mapsto a \times b & t^+ & : & \star \mapsto \{\star\} \\ \tau^- & : & z \mapsto ([-]z, z[-]) & t^- & : & \emptyset \mapsto 0,\ \ \{\star\} \mapsto 1 \end{array}$$

τ^- is an isomorphism because LS is full and faithful. The embedding commutes with involution: $\mathrm{LS}(U)^\perp = \mathrm{LS}(U^\perp)$.

7.5 Relating LS(Coh) to Coh

Both our semisimple full completeness result and the subsequent extension to higher types pivot on our ability to move freely between Chu and Coh. In this section we show that any semantic proof (logical element) η of a formula F in Chu can be pulled back to a semantic proof $\hat{\eta}$ of F in Coh. We first exhibit a map $(\hat{-})$, the ***hat*** map, taking points a of $F(\mathrm{LS}\mathbf{U})$ in Chu to cliques $\widehat{a}$ of $F(\mathbf{U})$ in Coh, then define $\widehat{\eta}$ componentwise by $\widehat{\eta}_{\mathbf{U}} = \widehat{\eta_{\mathrm{LS}\mathbf{U}}}$.

The Chu space $\mathrm{LS}U \otimes \mathrm{LS}V$ has the same states as $\mathrm{LS}(U \otimes V)$, namely all anticliques of $U \otimes V$. However $\mathrm{LS}U \otimes \mathrm{LS}V$ does not have all cliques of $U \otimes V$, only the "rectangular" ones formed as $a \times b$, where a, b are cliques of U, V respectively. The par $\mathrm{LS}U \parr \mathrm{LS}V$ has the dual defect. The value in Chu of an arbitrary MLL formula F on LSCoh will then be missing both cliques and anticliques of the value of F in Coh. However, sufficiently many cliques and anticliques remain that the latter can be reconstructed as a certain closure of the former, denoted $\overline{\mathcal{A}}$.

Let $\mathcal{A} = (A, r, X)$ be a biextensional Chu space, in other words, a Chu space with no repeated rows or columns. Then we can identify every point a of A with the subset $\{x \in X : r(a, x) = 1\}$ of X, and every state x with the subset $\{a \in A : r(a, x) = 1\}$ of A. This allows us to treat A as a set of subsets of X, treat X as a set of subsets of A, and to form unions of points and states, as in the following construction.

Define a set of sets to be ***consistent*** when it is non-empty and pairwise disjoint (stronger than necessary, but convenient and sufficient for our purposes).

Definition 4. The ***consistent closure*** of a biextensional Chu space $\mathcal{A}$ is the biextensional Chu space $\overline{\mathcal{A}} = (\overline{A}, \overline{r}, \overline{X})$ given by

$$\begin{aligned} \overline{A} &= \{ \bigcup A' : A' \subseteq A \text{ and } A' \text{ is consistent} \} \\ \overline{X} &= \{ \bigcup X' : X' \subseteq X \text{ and } X' \text{ is consistent} \} \end{aligned}$$

and for all consistent $A' \subseteq A$ and consistent $X' \subseteq X$

$$\overline{r}\,(\bigcup A', \bigcup X') = \bigvee_{a \in A',\, x \in X'} r(a, x)$$

For a non-biextensional Chu space $\mathcal{A}$ define $\overline{\mathcal{A}}$ to be the consistent closure of the biextensional collapse of $\mathcal{A}$. □

Note that $\overline{\overline{\mathcal{A}}} = \overline{\mathcal{A}}$ and $\overline{\mathcal{A}^\perp} = \overline{\mathcal{A}}^\perp$. If $\overline{\mathcal{A}} \cong \mathrm{LS}U$ for some coherence space U then we say that $\mathcal{A}$ ***generates*** U. The following Lemma states that if $\mathcal{A}$ generates U and $\mathcal{B}$ generates V, then $\mathcal{A} \otimes \mathcal{B}$ (tensor in Chu) generates $U \otimes V$ (tensor in Coh).

Lemma 7.5. *Let $\mathcal{A}, \mathcal{B}$ be Chu spaces such that $\overline{\mathcal{A}} \cong \mathrm{LS}U$ and $\overline{\mathcal{B}} \cong \mathrm{LS}V$ for coherence spaces U, V. Then*

$$\overline{\mathcal{A} \otimes \mathcal{B}} \cong \mathrm{LS}(U \otimes V)$$

Proof. Write $\mathcal{A} = (A, r, X)$, $\mathcal{B} = (B, s, Y)$, $\mathcal{A} \otimes \mathcal{B} = (A \times B, t, \mathcal{F})$, $\mathrm{LS}U = (U^\bullet, \sqcap, U^\circ)$ and $\mathrm{LS}V = (V^\bullet, \sqcap, V^\circ)$. Thus $\mathrm{LS}(U \otimes V) = ((U \otimes V)^\bullet, \sqcap, (U \otimes V)^\circ)$. We define a Chu isomorphism $(\theta, \phi) : \overline{\mathcal{A} \otimes \mathcal{B}} \to \mathrm{LS}(U \otimes V)$, i.e. an adjoint pair of isomorphisms $\theta : \overline{A \times B} \to (U \otimes V)^\bullet$ and $\phi : (U \otimes V)^\circ \to \overline{\mathcal{F}}$ between sets:

$$\begin{array}{rcl} \theta\Big(\bigcup\{\,(a_i, b_i) : i \in I\,\}\Big) & = & \bigcup_{i\in I}(a_i \times b_i) \\ \phi(\emptyset) & = & \emptyset \\ \phi(z) & = & \bigcup\{f_{\alpha,\beta} : \langle\alpha,\beta\rangle \in z\} \\ & & (\text{ for } z \neq \emptyset) \quad (z \neq \emptyset) \end{array}$$

where $f_{\alpha,\beta} = (f^1_{\alpha,\beta} : A \to Y, f^2_{\alpha,\beta} : B \to X)$ is given by

$$f^1_{\alpha,\beta}(a) = \begin{cases} \{\beta\} & \text{if } \alpha \in a \\ \emptyset & \text{if } \alpha \notin a \end{cases} \qquad f^2_{\alpha,\beta}(b) = \begin{cases} \{\alpha\} & \text{if } \beta \in b \\ \emptyset & \text{if } \beta \notin b \end{cases}$$

The definition of $\phi(z)$ requires working with tokens α, β. These are available since for a Chu space (A, r, X) to generate a coherence space U, both A and X must contain $\emptyset$ (as a row and column of 0s respectively) and all singletons (i.e. tokens) of U. $\square$

Proposition 7.6. *Let F be an n-ary MLL formula, $\mathbf{U} = U_1, \ldots, U_n$ for coherence spaces $U_i \in$ Coh, and $\mathrm{LS}\mathbf{U} = \mathrm{LS}U_1, \ldots, \mathrm{LS}U_n$. Then*

$$\overline{F(\mathrm{LS}\mathbf{U})} \;\cong\; \mathrm{LS}(F\mathbf{U}),$$

where F is interpreted on the left as an MLL operation in Chu*, and on the right as an MLL operation in* Coh*.*

Proof. Since $\overline{\mathcal{A}^\perp} = \overline{\mathcal{A}}^\perp$ for any Chu space $\mathcal{A}$, so we can write $A ⅋ B$ in F as $(A^\perp \otimes B^\perp)^\perp$, then apply Lemma 7.5 recursively. $\square$

For the following definition, note that points of $F(\mathrm{LS}U)$ are a subset of the points of $\overline{F(\mathrm{LS}U)}$.

Definition 7. Let F be an n-ary MLL formula, $\mathbf{U} = U_1, \ldots, U_n$ for coherence spaces $U_i \in$ Coh, and $\mathrm{LS}\mathbf{U} = \mathrm{LS}U_1, \ldots, \mathrm{LS}U_n$. Given any point a of $F(\mathrm{LS}\mathbf{U})$ in Chu, define the clique $\widehat{a}$ of $F(U)$ in Coh, the ***clique associated with*** a, as the image of a under the isomorphism of Proposition 7.6, acting from left to right. $\square$

7.6 MIX proof nets via Coh

In this section we a associate MIX proof net π_η with every Chu logical transformation η, by passing into Coh and appealing to full completeness for MLL with MIX [Tan97].

Lemma 7.8. *Let F be a semisimple formula. Every logical element $\eta : \mathbf{1} \to F$ in* Chu *gives rise to a dinatural transformation $\widehat{\eta} : \mathbf{1} \to F$ in* Coh*.*

Proof. Define $\widehat{\eta}_{\mathbf{U}} = \widehat{\eta_{\text{LS}\mathbf{U}}}$. We omit the proof that $\widehat{\eta}$ is dinatural. $\square$

Tan [Tan97] has shown that every dinatural transformation of a formula in Coh is the denotation of a unique MIX proof net, a proof structure acyclic under all switchings, though not necessarily connected. Define the ***MIX proof net*** π_η ***defined by*** η to be the MIX proof net denoting $\widehat{\eta}$ in Coh.

7.7 λ term characterization in LS(Coh)

We show that every logical transformation η in Chu, when restricted to the Coh-image, is determined by a tuple of lambda terms associated cananocally with the MIX proof net π_η that assigned to it via Coh.

Lemma 7.9. *Every MIX proof net of a semisimple formula $F = \bigparr_{i\le m}(\bigotimes_{j\le k_i} L_{ij})$ is characterized uniquely by an m-tuple $(t^1,\dots,t^m)$ of λ-terms. Each t^i is of the form $\lambda a_1 \dots a_{m-1}.M^i$ where the body M^i is a tuple $(M^i_1,\dots,M^i_{k_i})$ containing no λ abstractions and at most one occurrence of each variable a_j, $1 \le j \le m-1$.*

Proof. Let $\mathcal{G}_\pi$ be the undirected graph with vertices the clauses $F_i = \bigotimes_{j\le k_i} L_{ij}$ of F and edges the links e of π, with e connecting F_i and $F_{i'}$ in $\mathcal{G}_\pi$ just when e matches a literal of F_i with a literal of $F_{i'}$ in π. Since F is semisimple and π is acyclic under all switchings, $\mathcal{G}_\pi$ is acyclic.

We first construct t^m. Orient the edges of the connected component of the vertex F_m so as to point towards F_m. This defines a tree which we interpret as the applicative structure of M^m. Variable a_i corresponds to vertex F_i and serves as the function symbol at that vertex, taking as arguments the subtrees below it, with the leaves thus constituting ordinary variables. The k_m incoming edges of F^m give rise to the k_m components M^m_j of M^m.

The remaining t^i are obtained similarly, modulo matching a_l with F_{l+1} for $i \le l \le m-1$. $\square$

Our next Lemma allows us to describe every Chu transformation $\eta : \mathbf{1} \to \bigparr_{i\le m}(\bigotimes_{j\le k_i} L_{ij})$ into a semisimple formula of n variables as an m-tuple of families of functions $(\eta^1_{\vec{\mathcal{A}}},\dots,\eta^m_{\vec{\mathcal{A}}})$ indexed by $\vec{\mathcal{A}} \in \mathsf{Chu}^n$.

Lemma 7.10. *Let $\mathcal{A}_1,\dots,\mathcal{A}_m$ be Chu spaces. Every Chu transform $f : 1 \to \mathcal{A}_1 \parr \dots \parr \mathcal{A}_m$ is characterized by an m-tuple of functions $(f^1,\dots,f^m)$, where $f^i : \mathcal{A}_1^- \times \dots \times \mathcal{A}_{i-1}^- \times \mathcal{A}_{i+1}^- \times \dots \times \mathcal{A}_m^- \to \mathcal{A}_i{}^+$.*

Proof. Omitted. $\square$

Proposition 7.11. *Let $\eta : \mathbf{1} \to F$ be a logical element into a semisimple MLL formula of n variables in Chu, and let $t = (t^1,\dots,t^m)$ be the tuple of lambda terms representing the MIX proof net π_η associated with η. Then $\eta = t$ in $\text{LS}(\mathsf{Coh})^n \subset \mathsf{Chu}^n$. In other words, $\eta^i_{\text{LS}\mathbf{U}} = t^i$ for all $\mathbf{U} = U_1,\dots,U_n \in \mathsf{Coh}^n$ and $1 \le i \le m$, where $\text{LS}\mathbf{U} = \text{LS}(U_1),\dots,\text{LS}(U_n) \in \mathsf{Chu}^n$.*

Proof. Omitted. $\square$

7.8 λ-term characterization beyond $\mathrm{LS}(\mathsf{Coh})$

Having characterized logical transformations in the Coh-image by tuples of lambda terms, we show that this uniform behaviour extends to the whole of Chu.

Proposition 7.12. *Let F be a semisimple MLL formula of n propositional variables interpreted in Chu. Then every logical tranformation $\eta : \mathbf{1} \to F$ is determined by its restriction to the Coh-image, namely the sub-family $\eta_{\boldsymbol{\mathcal{A}}}$ indexed by $\boldsymbol{\mathcal{A}} \in \mathrm{LS}(\mathsf{Coh})^n$.*

Proof. To determine η at arbitrary Chu spaces $\boldsymbol{\mathcal{A}} \in \mathsf{Chu}^n$ we "simulate" each $\mathcal{A} = (A, r, X) \in \boldsymbol{\mathcal{A}}$ by a coherence space $\overline{\mathcal{A}}$, then use logical relations between $\mathrm{LS}(\overline{\mathcal{A}})$ in $\mathrm{LS}(\mathsf{Coh})$ and $\mathcal{A}$ in Chu to pin down the behaviour of $\eta_{\boldsymbol{\mathcal{A}}}$.

The set of tokens of $\overline{\mathcal{A}}$ is $A + X + r$, the disjoint union of the points, the states, and the coordinates of the 1s in the matrix. Coherence is "coherence along rows, incoherence within columns": $a \frown (a, x)$ and $(a, x) \frown (a, y)$ for all $a \in A$ and $x, y \in X$, together with the requisite loops $\alpha \frown \alpha$. The "row"-clique $\overline{a} = \{a\} \cup \{(a, x) \in r : x \in X\}$ "simulating" a intersects the "column"-anticlique $\overline{x} = \{x\} \cup \{(a, x) \in r : a \in A\}$ "simulating" x exactly when $r(a, x) = 1$. Hence the matrix $\sqcap$ of $\mathrm{LS}(\overline{\mathcal{A}})$ "simulates" the matrix of the original: $\overline{a} \sqcap \overline{x} = |\overline{a} \cap \overline{x}| = r(a, x)$. We establish this relationship formally as the logical relation $\mathcal{R}_{\mathcal{A}}$ between $\mathrm{LS}(\overline{\mathcal{A}})$ and $\mathcal{A}$ given on points by $\overline{a}\,\mathcal{R}_{\mathcal{A}}{}^{+} a$ for every $a \in A$ and on states by $\overline{x}\,\mathcal{R}_{\mathcal{A}}^{-} x$ for every $x \in X$. $\mathcal{R}_{\mathcal{A}}$ is logical because $\overline{a} \sqcap \overline{x} = r(a, x)$, the requisite adjointness condition.

Let $t = \lambda a_1 \ldots a_{m-1}.M$ be one of the tuple of λ-terms characterizing η in $\mathrm{LS}(\mathsf{Coh})$, and assume for simplicity, and without loss of generality, that the tuple M is a singleton. Recast the type of $\eta : \mathbf{1} \to F$ to parallel the natural typing of t, so that

$$\begin{array}{rcl} \eta_{\boldsymbol{\mathcal{A}}} & : & F_1 \otimes \ldots \otimes F_m \to \mathcal{B} \\ \eta_{\mathrm{LS}(\overline{\boldsymbol{\mathcal{A}}})} & : & G_1 \otimes \ldots \otimes G_m \to \mathrm{LS}(\overline{\mathcal{B}}) \\ F_i & = & \mathcal{B}_{i1} \otimes \ldots \otimes \mathcal{B}_{ik_i} \multimap \mathcal{B}_i \\ G_i & = & \mathrm{LS}(\overline{\mathcal{B}_{i1}}) \otimes \ldots \otimes \mathrm{LS}(\overline{\mathcal{B}_{ik_i}}) \multimap \mathrm{LS}(\overline{\mathcal{B}_i}) \end{array}$$

for $\mathcal{B}, \mathcal{B}_i, \mathcal{B}_{ij} \in \boldsymbol{\mathcal{A}}$ and $\mathrm{LS}(\overline{\boldsymbol{\mathcal{A}}}) = (\mathrm{LS}(\overline{\mathcal{A}_1}), \ldots, \mathrm{LS}(\overline{\mathcal{A}_n}))$. We shall determine

$$\eta_{\boldsymbol{\mathcal{A}}}{}^{+} : F_1{}^{+} \times \ldots \times F_m{}^{+} \to \mathcal{B}^{+},$$

a component of the tuple characterizing η as per Lemma 7.10, by showing that

$$\eta_{\boldsymbol{\mathcal{A}}}{}^{+}(f_1, \ldots, f_m) = t(f_1, \ldots, f_m)$$

for all inputs $f_i \in F_i{}^{+}$. For each f_i define the linear map

$$\widetilde{f_i} : \overline{\mathcal{B}_{i1}} \otimes \ldots \otimes \overline{\mathcal{B}_{ik_i}} \to \overline{\mathcal{B}_i}$$

between coherence spaces by

$$\{(b_{i1}, \ldots, b_{ik_i})\} \times \overline{f_i{}^{+}(b_{i1}, \ldots, b_{ik_i})} \subseteq \widetilde{f_i}$$

for all $b_{ij} \in \mathcal{B}_{ij}{}^{+} \subseteq |\overline{\mathcal{B}_{ij}}|$, where for a point a of a Chu space $\mathcal{A}$, $\overline{a}$ is the "row" clique of $\overline{\mathcal{A}}$ simulating a as defined above. Let τ_i be the tensorial strength

$$\mathrm{LS}(\overline{\mathcal{B}_{i1}}) \otimes \ldots \otimes \mathrm{LS}(\overline{\mathcal{B}_{ik_i}}) \to \mathrm{LS}(\overline{\mathcal{B}_{i1}} \otimes \ldots \otimes \overline{\mathcal{B}_{ik_i}})$$

and define $\overline{f_i} = \mathrm{LS}(\widetilde{f_i}) \circ \tau_i$ of type G_i, so that $\overline{f_i}^{+}$ is a function from (k_i-tuples of) "row"-cliques to "row"-cliques. In particular $\overline{f_i}$ "simulates" f_i, for example

$$\overline{f_i}^{+}(\overline{b_{i1}}, \ldots, \overline{b_{ik_i}}) = \overline{f_i{}^{+}(b_{i1}, \ldots, b_{ik_i})} \tag{7.1}$$

More formally, $\overline{f_i}\mathcal{R}_i{}^{+} f_i$ under the Chu logical relation $\mathcal{R}_i = \mathcal{R}_{\mathcal{B}_{i1}} \otimes \ldots \otimes \mathcal{R}_{\mathcal{B}_{ik_i}} \multimap \mathcal{R}_{\mathcal{B}_i}$ between G_i and F_i.

Since $\mathcal{R}_i$ is a Chu logical relation between G_i and F_i for each i, by the logicality of η we must have $\eta_{\mathrm{LS}(\overline{\boldsymbol{\mathcal{A}}})}{}^{+}(\overline{f_1}, \ldots, \overline{f_m})\, \mathcal{R}_{\mathcal{B}_i}{}^{+}\, \eta_{\boldsymbol{\mathcal{A}}}{}^{+}(f_1, \ldots, f_m)$. Furthermore since $\eta_{\mathrm{LS}(\overline{\boldsymbol{\mathcal{A}}})}{}^{+}$ is a λ-term t, by repeated application of (7.1) we have

$$\begin{aligned} \eta_{\mathrm{LS}(\overline{\boldsymbol{\mathcal{A}}})}{}^{+}(\overline{f_1}, \ldots, \overline{f_m}) &= t(\overline{f_1}, \ldots, \overline{f_m}) \\ &= \overline{t(f_1, \ldots, f_m)} \end{aligned}$$

Thus

$$\overline{t(f_1, \ldots, f_m)}\, \mathcal{R}_{\mathcal{B}_i}{}^{+}\, \eta_{\boldsymbol{\mathcal{A}}}{}^{+}(f_1, \ldots, f_m)$$

and since by construction $a\mathcal{R}_{\mathcal{B}_i}{}^{+}b$ if and only if $a = \overline{b}$, we conclude that $\eta_{\boldsymbol{\mathcal{A}}}{}^{+}(f_1, \ldots, f_m) = t(f_1, \ldots f_m)$.

Finally, repeat the argument for each of the other lambda terms of the tuple. □

7.9 MIX refutation

The final link in the chain to semisimple full completeness is to show that the MIX proof net assigned to a Chu logical element is connected under all switchings, and hence is a proof net. This occurs precisely when any (and hence all) of the lambda terms in the characterizing tuple use all their arguments.

Lemma 7.13. *, Let $t = \lambda a_1 \ldots a_{m-1}.M$ be a λ-term of the m-tuple characterizing a MIX proof net π of a semisimple formula $\invamp_{i \leq m}\left(\bigotimes_{j \leq k_i} L_{ij}\right)$. If each variable a_i occurs in M, $1 \leq i \leq m-1$, then π is a proof net.*

Proof. The graph $\mathcal{G}_\pi$ in the construction of t in Lemma 7.9 is connected if and only if π is a proof net. □

Proposition 7.14. *A λ-term of a tuple defining a Chu logical element of a semisimple MLL operation must use all its arguments.*

Proof. Without loss of generality assume that the operation has the form $A \multimap B$ where B is a literal and $A = A_1 \otimes \ldots \otimes A_n$ is a product of pars of literals. The λ-term then has the form $\lambda a_1 \ldots a_n.M$ for some applicative term M in λ-variables $a_1, \ldots, a_n$. Let η be the denotation of this λ-term. Set all variables of the formula to the Chu space $J = \begin{matrix} 00 \\ 01 \end{matrix}$, so η_J is a Chu transform from $A(J)$ to $B(J)$.

By inspection every MLL operation, in particular A, evaluates at J to a Chu space having just one nonzero entry. $B(J) = J$ since B is either a variable or its negation and $J^\perp = J$.

By continuity of η_J, every zero point (the index of an all-zero row) of $A(J)$ must be sent to 0 in $B(J)$. We now argue by logicality that the one nonzero point of $A(J)$ must be sent to 1 in $B(J)$. Let $\mathbf{N}$ be the 1×1 Chu space whose one entry is 1, and take R to be the Chu relation between $\mathbf{N}$ and J that relates the point of $\mathbf{N}$ to the nonzero point of J and the state of $\mathbf{N}$ to the nonzero state of J. By induction on height of MLL operations C, $C(R)$ is the Chu relation between $C(\mathbf{N}) = \mathbf{C}$ and $C(J)$ that relates the point of $C(\mathbf{N})$ to the nonzero point of $C(J)$ and the state of $C(\mathbf{N})$ to the nonzero state of $C(J)$. Applying this to $C = A$, we deduce that any two-sorted homomorphism from $A(J)$ to J must send the nonzero point of $A(J)$ to the nonzero point of $B(J)$, i.e. 1.

But the one nonzero point of $A(J)$ is indexed by the constantly-one n-tuple. So when all arguments to η_J are set to 1, and any one argument is then changed to 0, the result of η_J changes from 1 to 0. But then η, and hence the λ-term denoting it, depends on all n of its arguments. □

Theorem 7.15 (*Semisimple full completeness*). *Let F be a semisimple MLL formula interpreted in* Chu*. Then every logical tranformation $\eta : \mathbf{1} \to F$ is denoted by a unique proof of F.*

Proof. Proposition 7.11 characterizes η as a MIX proof net π_η in the Coh-image, and Proposition 7.12 extends this characterization to the whole of Chu. By Proposition 7.14, in conjunction with Lemma 7.13, π_η is a proof net. □

7.10 Full Completeness

We prove our main theorem by induction on level of formulas, a measure of distance of theorems to certain semisimple formulas. We first define the notion of level and state key supporting lemmas.

The rules of our axiomatization of MLL either rearrange the formula invertibly (A and C), cater trivially for context (E), or do some real work (D). Our argument hinges on the behavior of this last rule.

For convenience and to make more explicit the choice implicit in linear distributivity (LD), we replace Rule D by an equivalent pair of rules either one of which would suffice on its own.

$$\begin{array}{lccl} \mathsf{D1} & (A \parr B) \otimes C & \vdash & (A \otimes C) \parr B \\ \mathsf{D2} & (A \parr B) \otimes C & \vdash & A \parr (C \otimes B) \end{array}$$

A linking of the common antecedent of these rules is permuted by the rules to yield identical linkings of the two ***LD consequents***.

Definition 16. The ***level*** of an MLL formula is defined to be maximal subject to the following constraints. The level of a semisimple formula is zero. If a level l formula is derivable from A by associativity or commutativity then A has level l. If a pair of LD consequents of A have level at most l then A has level at most $l+1$. □

So the level of a formula A is the minimum, over all occurrences within A of a subformula matching Rule D, of one plus the maximum of the level of the corresponding pair of LD consequents A_1 and A_2. Every MLL formula reduces to semisimple formulas after finitely many applications of D, whence level is well-defined.

We shall need the following lemmas.

Lemma 7.17. *If the LD consequents of a transformation of an MLL formula A in* Chu *both represent proof nets, then those proof nets have the same Λ (linking of literals).*

We defer the proof of this lemma to after the main theorem.

Lemma 7.18. *If a linking Λ is a proof net for both Rule D consequents of a formula A, then Λ is a proof net for A.*

Proof. We omit the straightforward combinatorial argument. □

Theorem 7.19. *Every logical element η of an MLL formula A represents a proof.*

Proof. We proceed by induction on level. The previous section supplied the basis for the induction. We now assume as our induction hypothesis the case l and prove the case $l+1$.

Let A be a formula A of level $l+1$, and let η be a logical element of A. Apply commutativity and associativity as required to A so that when Rule D is applied to some subformula $(A \parr B) \otimes C$ in each of the two possible ways, both consequents A_1 and A_2 of A are of level l. The two applications of the rule map η to two transformations, call them η_1 and η_2.

By soundness each η_i is a logical element and therefore by the induction hypothesis has a proof net. By Lemma 7.17 the two proof nets must have the same linking Λ. By Lemma 7.18 (A, Λ) is a proof net. By soundness (A, Λ) denotes a logical element η' while (A_1, Λ) and (A_2, Λ) denote logical elements η'_1, η'_2 respectively. By the induction hypothesis $\eta'_1 = \eta_1$ and $\eta'_2 = \eta_2$. But D is injective, so $\eta = \eta'$. □

This completes the proof of our main theorem, leaving only Lemma 7.17 to prove.

Proof. (of Lemma 7.17) Let η in A have consequents η_1 and η_2 representing proof nets in Chu. Form the corresponding arrangement in Coh, consisting of a clique $\widehat{\eta}$ and consequents $\widehat{\eta}_1$ and $\widehat{\eta}_2$. By Lemma 7.21 η_i and $\widehat{\eta}_i$ represent the same proof nets for $i = 1, 2$. By Lemma 7.22 $\widehat{\eta}_1$ and $\widehat{\eta}_2$ are the LD consequents of a common clique $\widehat{\eta}$. By Lemma 7.20 they have the same linking, whence the same holds for η_1 and η_2. $\square$

We have discharged obligation 7.17 at the expense of three new obligations.

Lemma 7.20. *If the two LD consequents of a clique both realize proof nets then those nets have the same linking.*

Proof. Although the rules of System S1 nontrivially transform the coherence spaces they act on, their constituent tokens, as tuples of tokens of W, are not changed except to reflect permutations of variables. The linking information in a dinatural clique in Coh resides entirely in the individual tokens [Tan97] (as opposed to the coherence relations between the tokens). Under our reformulation of Rule D as D1 and D2, the two LD consequents of a clique undergo the same permutation of atomic tokens within each token of the clique and hence encode the same linking. $\square$

Lemma 7.21. *If η is the unique representation in* Chu *of a proof net (A, Λ), then $\widehat{\eta}$ represents (A, Λ) in* Coh.

Proof. Select any System S1 derivation of (A, Λ) and interpret it in both Coh and Chu. In the beginning the hat relationship holds between the respective proof representations. By commutativity of hat and derivation (Lemma 7.22) this relationship is maintained during the proof and hence still holds when A is reached. Since (A, Λ) only has one representation, the transformation in Chu we ended up with must be η, whence the transformation in Coh we arrived at must be $\widehat{\eta}$. $\square$

Lemma 7.22. *The hat map taking the points of the Chu space $F(\mathrm{LS}\boldsymbol{\mathcal{W}})$ to the coherence space $F(\boldsymbol{\mathcal{W}})$ commutes with the action of the rules of System S1.*

Proof. Since the rules act according to linear λ-terms it suffices to verify the commutativity for all linear λ-terms. If the correspondence between Chu spaces $F(\mathrm{LS}\boldsymbol{\mathcal{W}})$ and coherent spaces $F(\boldsymbol{\mathcal{W}})$ were an isomorphism this would be a triviality. However the points of the former embed as a subset of those of the latter, and likewise for states, both for the top level formula and for all subformulas. We therefore need to show that the correspondence is tight enough for the action of the rules to maintain the correspondence despite this difference.

We proceed by induction on the height of λ-terms. We take as our inductive hypothesis that for all MLL formulas F, for all assignments of Chu spaces $\mathrm{LS}\boldsymbol{\mathcal{W}}$, and corresponding assignments of coherent spaces $\boldsymbol{\mathcal{W}}$, to variables of F, and for all bindings of λ-variables to points of $F(\mathrm{LS}\boldsymbol{\mathcal{W}})$, and correspondingly to points of

$F(\mathcal{W})$, evaluating a λ-term of height h in each of the two environments produces a corresponding pair of points of respectively $F(\mathrm{LS}\mathcal{W})$ and $F(\mathcal{W})$.

The basis for the induction, λ-terms that are λ-variables, holds by choice of environment. We now assume the case of height h and proceed to height $h+1$.

For applications MN, the Chu point denoted by M is a function f between Chu spaces $\mathcal{A}$ and $\mathcal{B}$ (say) while the corresponding coherence space point denoted by M is a function $\widehat{f}$ from $\overline{\mathcal{A}}$ to $\overline{\mathcal{B}}$. Since the correspondence embeds the set A of points of $\mathcal{A}$ in $\overline{A}$, and since N evaluates to a point a in the image of that embedding by the induction hypothesis, $\widehat{f}(a)$ must be the corresponding point in $\overline{\mathcal{B}}$, which too will be in the image of the embedding of $\mathcal{B}$ in $\overline{\mathcal{B}}$.

A λ-abstraction $\lambda x.M$, as a point of say $A \multimap B$, will denote a Chu transform f of Chu spaces having corresponding coherence space map $\widehat{f}$. The induction hypothesis ensures that f and $\widehat{f}$ agree on A, while the fact that coherence space maps commute with consistent unions ensures that the coherence space map denoted by $\lambda x.M$ (as determined by evaluating M in each environment obtained by setting x to a point of $\overline{\mathcal{A}}$) agrees with $\widehat{f}$ on the whole of $\overline{A}$.

For pairs (M, N) the correspondence is immediate. $\square$

Open problem: Extend this full completeness result for MLL to the units, additives, and exponentials. More precisely, put the logical transformations between terms of linear logic in bijection with the cut-free proofs of those terms as axiomatized by Girard [Gir87], and note any exceptions.

Bibliography

[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.

[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.

[Bir33] G. Birkhoff. On the combination of subalgebras. *Proc. Cambridge Phil. Soc*, 29:441–464, 1933.

[BS96] R.F. Blute and P.J. Scott. Linear Läuchli semantics. *Annals of Pure and Applied Logic*, 77:101–142, 1996.

[CS97] J. R. B. Cockett and R. A. G. Seely. Proof theory for full intuitionistic linear logic, bilinear logic, and mix categories. *Theory and Applications of categories*, 3(5):85–131, 1997.

[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*, 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.

[EML42a] S. Eilenberg and S. Mac Lane. General theory of natural equivalences. *Trans. Am. Math. Soc.*, 58:231–294, 1942.

[EML42b] S. Eilenberg and S. Mac Lane. Natural isomorphism in group theory. *Proc. Nat. Acad. Soc.*, 28:537–543, 1942.

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

[GP93] V. Gupta and V.R. Pratt. Gates accept concurrent behavior. In *Proc. 34th Ann. IEEE Symp. on Foundations of Comp. Sci.*, pages 62–71, November 1993.

[Gup94] V. Gupta. *Chu Spaces: A Model of Concurrency*. PhD thesis, Stanford University, September 1994. Tech. Report, available as ftp://boole.stanford.edu/pub/gupthes.ps.Z.

[Joh82] P.T. Johnstone. *Stone Spaces*. Cambridge University Press, 1982.

[Kel82] G.M. Kelly. *Basic Concepts of Enriched Category Theory: London Math. Soc. Lecture Notes*. 64. Cambridge University Press, 1982.

[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.

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

[NPW81] M. Nielsen, G. Plotkin, and G. Winskel. Petri nets, event structures, and domains, part I. *Theoretical Computer Science*, 13:85–108, 1981.

[Pei33] C.S. Peirce. Description of a notation for the logic of relatives, resulting from an amplification of the conceptions of Boole's calculus of logic. In *Collected Papers of Charles Sanders Peirce. III. Exact Logic*. Harvard University Press, 1933.

[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.

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

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

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

[Sch95] E. Schröder. *Vorlesungen über die Algebra der Logik (Exakte Logik). Dritter Band: Algebra und Logik der Relative*. B.G. Teubner, Leipzig, 1895.

[See89] Robert A. G. Seely. Linear logic, $*$-autonomous categories and cofree coalgebras. In John W. Gray and Andre Scedrov, editors, *Categories in Computer Science and Logic*, volume 92 of *Contemporary Mathematics*, pages 371–382, Providence, Rhode Island, 1989. American Mathematical Society.

[Sto36] M. Stone. The theory of representations for Boolean algebras. *Trans. Amer. Math. Soc.*, 40:37–111, 1936.

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