A Fibrational Framework for Substructural and Modal Logics (extended version)

Daniel R. Licata*[1], Michael Shulman*[2], and Mitchell Riley*[1]

[1]Wesleyan University
[2]University of San Diego

April 23, 2017

Abstract

Many intuitionistic substructural and modal logics / type theories can be seen as a restriction on the allowed proofs in a standard structural logic / λ-calculus. For example, substructural logics remove properties such as associativity, weakening, exchange, and contraction, while modal logics place restrictions on the positions in which certain variables can be used. These restrictions are usually enforced by a specific context structure (trees,lists,multisets,sets,dual zones,...) that products, implications, and modalities manipulate. While the general technique is clear, it can be difficult to devise rules modeling a new situation, a problem we have recently run into while extending homotopy type theory to express additional mathematical applications.

In this paper, we define a general framework that abstracts the common features of many intuitionistic substructural and modal logics. The framework is a sequent calculus / normal-form type theory parametrized by a *mode theory*, which is used to describe the structure of contexts and the structural properties they obey. The framework makes use of resource annotations, where we pair the context itself, which obeys standard structural properties, with a term, drawn from the mode theory, that constrains how the context can be used. Product types, implications, and modalities are defined as instances of two general connectives, one positive and one negative, that manipulate these resource annotations. We show that specific mode theories can express non-associative, ordered, linear, affine, relevant, and cartesian products and implications; monoidal and non-monoidal comonads and adjunctions; strong and non-strong monads; n-linear variables; bunched implications; and the adjunctions that arose in our work on homotopy type theory. We prove cut (and identity) admissibility independently of the mode theory, obtaining it for all of the above logics at once. Further, we give a general equational theory on derivations / terms that, in addition to the usual $\beta\eta$-rules, characterizes when two derivations differ only by the placement of structural rules. Finally, we give an equivalent semantic presentation of these ideas, in which a mode theory corresponds to a 2-dimensional cartesian multicategory, and the framework corresponds to another such multicategory with a functor to the mode theory. The logical connectives have universal properties relative to this functor, making it into a bifibration. The sequent calculus rules and the equational theory on derivations are sound and complete for this. The resulting framework can be used both to understand existing logics / type theories and to design new ones.

1 Introduction

In ordinary intuitionistic logic or λ-calculus, assumptions or variables can go unused (weakening), be used in any order (exchange), be used more than once (contraction), and be used in any position in a term. *Substructural* logics, such as linear logic, ordered logic, relevant logic, and affine logic, omit some of these structural properties of

*This material is based on research sponsored by The United States Air Force Research Laboratory under agreement number FA9550-15-1-0053 and and FA9550-16-1-0292. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the United States Air Force Research Laboratory, the U.S. Government, or Carnegie Mellon University.

weakening, exchange, and contraction, while *modal logics* place restrictions on where variables may be used—e.g. a formula $\Box C$ can only be proved using assumptions of $\Box A$, while an assumption of $\Diamond A$ can only be used when the conclusion is $\Diamond C$. Substructural and modal logics have had many applications to both functional and logic programming, modeling concepts such state, staging, distribution, and concurrency. They are also used as *internal languages* of categories, where one uses an appropriate logical language to do constructions "inside" a particular mathematical setting, which often results in shorter statements than working "externally". For example, to define a function externally in domains, one must first define the underlying set-theoretic function, and then prove that it is continuous; when using untyped λ-calculus as an internal language of domains, one writes what looks like only the function part, and continuity follows from a general theorem about the language itself. Substructural logics extend this idea to various forms of monoidal categories, while modal logics describe monads and comonads. Recently, Schreiber and Shulman [2012], Shulman [2015] proposed using modal operators to add a notion of *cohesion* to homotopy type theory/univalent foundations [Univalent Foundations Program, 2013, Voevodsky, 2006]. Without going into the precise details of this application, the idea is to add a triple $\int \dashv \flat \dashv \sharp$ of type operators, where for example $\sharp$ is a monad (like a modal possibility $\Diamond$ or $\bigcirc$), $\flat$ is a comonad (like a modal necessity $\Box$), and there is an adjunction structure between them ($\flat A \to B$ is the same as $A \to \sharp B$). This raised the question of how to best add modalities with these properties to type theory.

Because other similar applications rely on functors with different properties, we would like general tools for going from a semantic situation of interest to a well-behaved logic/type theory for it—e.g. one with cut admissibility / normalization and identity admissibility / η-expansion. In previous work [Licata and Shulman, 2016], we considered the special case of a single-assumption logic, building most directly on the adjoint logics of Benton [1995], Benton and Wadler [1996], Reed [2009a]. Here we extend this previous work to the multi-assumption case. The resulting framework is quite general and covers many existing intuitionistic substructural and modal connectives: non-associative, ordered, linear, affine, relevant, and cartesian products and implications; combinations thereof such as bunched logic [O'Hearn and Pym, 1999] and resource separation [Atkey, 2004]; n-linear variables [Abel, 2015, McBride, 2016, Reed, 2008]; the comonadic $\Box$ and linear exponential ! and subexponentials [Danos et al., 1993, Nigam and Miller, 2009]; monadic $\Diamond$ and $\bigcirc$ modalities; and adjoint logic F and G [Benton, 1995, Benton and Wadler, 1996, Reed, 2009a], including the single-assumption 2-categorical version from our previous work [Licata and Shulman, 2016]. A central syntactic result is that cut and identity are admissible for our framework itself, and this implies cut admissibility for any logic that can be described in the framework, including all of the above, as well as any new logics that one designs using it. When we view the derivations in the framework as terms in a type theory, this gives an immediate normalization (and η-expansion) result.

At a high level, the framework makes use of the fact that all of the above logics / type theories are a restriction on how variables can be used in ordinary structural/cartesian proofs. We express these restrictions using a first layer, which is a simple type theory for what we will call *modes* and *context descriptors*. The modes are just a collection of base types, which we write as p, q, r, while a context descriptor α is a term built from variables and constants. The next layer is the main logic. Each proposition/type is assigned a mode, and the basic sequent is $x_1 : A_1, \ldots, x_n : A_n \vdash_\alpha C$, where if A_i has mode p_i, and C has mode q, then $x_1 : p_1, \ldots, x_n : p_n \vdash \alpha : q$ (we use a sequent calculus to concisely describe cut-free derivations/normal forms, but everything can be translated to natural deduction in the usual way). We write Γ for $x_1 : A_1, \ldots, x_n : A_n$, and Γ itself behaves like an ordinary structural/cartesian context, while the substructural and modal aspects are enforced by the *term* α, which constrains how the resources from Γ may be used. For example, in linear logic/ordered logic/BI, the context is usually taken to be a multiset/list/tree. We represent this by a pair of an ordinary structural context Γ, together with a term α that describes the multiset or list or tree structure, labeled with variables from the ordinary context at the leaves. We pronounce a sequent $\Gamma \vdash_\alpha A$ as "Γ proves A {along,over,using} α".

For example, if we have a mode n, together with a context descriptor constant $x : \mathsf{n}, y : \mathsf{n} \vdash x \odot y : \mathsf{n}$, then an example sequent $x : A, y : B, z : C, w : D \vdash_{(y \odot x) \odot z} E$ should be read as saying that we must prove E using the resources y and x and z (but not w) according to the particular tree structure $(y \odot x) \odot z$. If we say nothing else, the framework will treat $\odot$ as describing a non-associative, linear, ordered context [Lambek, 1958]: if we have a product-like type $A \odot B$ internalizing this context operation,[1] then we will *not* be able to prove associativity ($(A \odot B) \odot C \dashv\vdash A \odot (B \odot C)$) or

[1] We overload binary operations to refer both to context descriptors and propositional connectives, because it is clear from whether it is applied to variables x, y, z or propositions A, B, C which we mean.

exchange ($A \odot B \vdash B \odot A$) etc.

To get from this basic structure to a linear or affine or relevant or cartesian system, we provide a way to add structural properties governing the context descriptor term α. We analyze structural properties as *equations*, or more generally *directed transformations*, on such terms. For example, to specify linear logic, we will add a unit element $1 : \mathsf{n}$ together with equations making $(\odot, 1)$ into a commutative monoid ($x \odot (y \odot z) = (x \odot y) \odot z$ and $x \odot 1 = x = 1 \odot x$ and $x \odot y = y \odot x$) so that the context descriptors ignore associativity and order. To get BI, we add an additional commutative monoid $(\times, \top)$ (with weakening and contraction, as discussed below), so that a BI context tree $(x : A, y : B); (z : C, w : D)$ can be represented by the ordinary context $x : A, y : B, z : C, w : D$ with the term $(x \odot y) \times (z \odot w)$ describing the tree. Because the context descriptors are themselves ordinary structural/cartesian terms, the same variable can occur more than once or not at all. A descriptor such as $x \odot x$ captures the idea that we can use the *same* variable x twice, expressing n-linear types. Thus, we can express contraction for a particular context descriptor $\odot$ as a transformation $x \Rightarrow x \odot x$ (one use of x allows two). Weakening, on the other hand, is represented by a transformation $x \Rightarrow 1$, which is oriented to allow throwing away an allowed use of x, but not creating an allowed use from nothing. We refer to these as *structural transformations*, to evoke their use in representing the structural properties of object logics that are embedded in our framework. The main sequent $\Gamma \vdash_\alpha A$ respects the specified structural properties in the sense that when $\alpha = \beta$, we regard $\Gamma \vdash_\alpha A$ and $\Gamma \vdash_\beta A$ as the same sequent (so a derivation of one is a derivation of the other), while when $\alpha \Rightarrow \beta$, there will be an operation that takes a derivation of $\Gamma \vdash_\beta A$ to a derivation of $\Gamma \vdash_\alpha A$—i.e. uses of transformations are explicitly marked in the term.

Modal logics will generally involve a mode theory with more than one mode. For example, a context descriptor $x : \mathsf{c} \vdash \mathsf{f}(x) : \mathsf{l}$ will generate an adjoint pair of functors between the two modes, as in the adjoint syntax for linear logic's ! [Benton and Wadler, 1996] or other modal operators [Reed, 2009a]. Using this, a context descriptor $\mathsf{f}(x) \odot y$ expresses permission to use x in a cartesian way and y in a linear way. Structural transformations are used to describe how these modal operators interact with each other and with the products, and for some systems [Licata and Shulman, 2016] it is important that there can be more than one transformation between a given pair of context descriptors.

A guiding principle of the framework is a meta-level notion of *structurality over structurality*. For example, we always have *weakening over weakening*: if $\Gamma \vdash_\alpha A$ then $\Gamma, y : B \vdash_\alpha A$, where α itself is weakened with y. This does not prevent encodings of relevant logics: though we might weaken a derivation of $\Gamma \vdash_{x_1 \odot \ldots \odot x_n} A$ ("use x_1 through x_n") to a derivation of $\Gamma, y : B \vdash_{x_1 \odot \ldots \odot x_n} A$, the (weakened) context descriptor does not allow the use of y. Similarly, we have exchange over exchange and contraction over contraction. The *identity-over-identity* principle says that we should be able to prove A using exactly an assumption $x : A$ ($\Gamma, x : A \vdash_x A$). The cut principle says that from $\Gamma, x : A \vdash_\beta B$ and $\Gamma \vdash_\alpha A$ we get $\Gamma \vdash_{\beta[\alpha/x]} B$—the context descriptor for the result of the cut is the substitution of the context descriptor used to prove A into the one used to prove B. For example, together with weakening-over-weakening, this captures the usual cut principle of linear logic, which says that cutting $\Gamma, x : A \vdash B$ and $\Delta \vdash A$ yields $\Gamma, \Delta \vdash B$: if Γ binds $x_1, \ldots, x_n$ and Δ binds $y_1, \ldots, y_n$, then we will represent the two derivations to be cut together by sequents with $\beta = x_1 \odot \ldots \odot x_n \odot x$ and $\alpha = y_1 \odot \ldots \odot y_n$, so $\beta[\alpha/x] = x_1 \odot \ldots \odot x_n \odot y_1 \odot \ldots \odot y_n$ correctly deletes x and replaces it with the variables from Δ. In more subtle situations such as BI, the substitution will insert the resources used to prove the cut formula in the correct place in the tree.

The framework has two main logical connectives / type constructors. The first, $\mathsf{F}_\alpha(\Delta)$, generalizes the F of adjoint logic and the multiplicative products (e.g. $\otimes$ of linear logic). The second, $\mathsf{U}_{x.\alpha}(\Delta \mid A)$, generalizes the G/U of adjoint logic and implication (e.g. $A \multimap B$ in linear logic). Here Δ is a context of assumptions $x_i : A_i$, and trivializing the context descriptors (i.e. adding an equation $\alpha = \beta$ for all α and β) degenerates $\mathsf{F}_\alpha(\Delta)$ into the ordinary intuitionistic product $A_1 \times \ldots \times A_n$, while $\mathsf{U}_{x.\alpha}(\Delta \mid A)$ becomes $A_1 \to \ldots \to A_n \to A$. As one would expect, F is left-invertible and U is right-invertible. In linear logic terms, our F and U cover both the multiplicatives and exponentials; additives can be added separately by the usual rules. We discuss many examples of *logical adequacy* theorems, showing that a sequent can be proved in a standard sequent calculus for a logic iff its embedding using these connectives can be proved in the framework.

Being a very general theory, our framework treats the object-logic structural properties in a general but naïve way, allowing an arbitrary structural transformation to be applied at the non-invertible rules for F and U and at the leaves of a derivation. For specific embedded logics, there is often a more refined discipline that suffices—e.g. for cartesian logic, always contract all assumptions in all premises, and only weaken at the leaves. We view our framework as a tool for bridging the gap between an intended semantic situation (such as the cohesion example mentioned, "a comonad and a monad which are themselves adjoint") and a proof theory: the framework gives *some* proof theory for the semantics, and the placement of structural rules can then be optimized purely in syntax. To support this mode of use, we give an equational theory on derivations/terms that identifies different placements of the same structural rules. This can be used to prove correctness of such optimizations not just at the level of provability, but also identity of derivations—which matters for our intended applications to internal languages. We discuss some preliminary work on *equational adequacy*, which extends the logical correspondence to isomorphisms of definitional-equality-classes of derivations.

Semantically, the logic corresponds to a functor between *2-dimensional cartesian multicategories* which is a fibration in various senses. Multicategories are a generalization of categories which allow more than one object in the domain, and cartesianness means that the multiple domain objects are treated structurally. The 2-dimensionality supplies a notion of morphism between (multi)morphisms. A *mode theory* specifying context descriptors and structural properties is analyzed as a cartesian 2-multicategory, with the descriptors as 1-cells and the structural properties as 2-cells. The functor relates the sequent judgement to the mode theory, specifying the mode of each proposition and the context descriptor of a sequent. The fibration conditions (similar to [Hermida, 2002, Hörmann, 2015]) give respect for the structural transformations and the presence of F and U types. We prove that the sequent calculus and the equational theory are sound and complete for this semantics: the syntax can be interpreted in any bifibration, and itself determines one. This semantics shows that an interesting class of type theories can be identified with a class of more mathematical objects, fibrations of cartesian 2-multicategories, thus providing some progress towards characterizing substructural and modal type theories in mathematical terms.

Our framework builds on many approaches to substructural and modal logic in the literature. Logical rules that act at a leaf of a tree-structured context go back to the Lambek calculus [Lambek, 1958]. A rich collection of context structures that correspond to type constructors plays a central role in display logic [Belnap Jr., 1982]. Atkey [2004]'s λ-calculus for resource separation is similar to mode theories with one mode, where there is at most one 2-cell between a given pair of 1-cells; at the logical level, our calculus is a unification of this with the multimode adjoint logic of Reed [2009a]. Algebraic resource annotations on variables are used to track modalities in Agda's implementation [Abel, 2015] and in McBride [2016]'s approach to linear dependent types. LF representations of modal or substructural logics work by restricting the use of cartesian variables [Crary, 2010]. Relative to all of these approaches, we believe that the analysis of the context structures/resources as a *term* in a base type theory, and the fibrational structure of the derivations over them, is a new and useful observation. For example, rather than needing extra-logical conditions on proof rules to ensure cut admissibility, as in display logic, the conditions are encoded in the language of context descriptors and the definition of types from them. Moreover, none of these existing approaches allow for proof-relevant 2-cells/structural rules, and their presence (and the equational theory we give for them) is important for our applications to extensions of homotopy type theory. A point of contrast with substructural logical frameworks [Cervesato and Pfenning, 2002, Reed, 2009b, Watkins et al., 2002] is that logics are "embedded" in our calculus (giving a type translation such that provability in the object logic corresponds to provability in ours), rather than "encoding" the structure of derivations. This way, we obtain cut elimination for object languages as a corollary of framework cut elimination.

The remainder of this paper is organized as follows. In Section 2, we present the rules of the logic. In Section 3, we discuss how a number of logics are represented. In Section 4, we show how identity and cut are implemented. In Section 5, we give an equational theory on derivations. In Section 7, we return to the examples, proving logical adequacy of their representation. In Section 6, we discuss the logic's categorical semantics. In Section 8, we give an alternative presentation of equality of derivations, and in Section 9, we discuss one example of equational adequacy.

2 Sequent Calculus

2.1 Mode Theories

The first layer of our framework is a type theory whose types we will call *modes*, and whose terms we will call *context descriptors* or *mode morphisms*. The only modes are atomic/base types p. A term is either a variable (bound in a context ψ) or a typed n-ary constant (function symbol) c applied to terms of the appropriate types.

This is formalized in the notion of signature, or *mode theory*, defined in Figure 1. The judgement $\Sigma\,\mathsf{sig}$ means that Σ is a well-formed signature. The top line says that a signature is either empty, or a signature extended with a new

$$\boxed{\text{Signatures } \Sigma\,\mathsf{sig}} \qquad \frac{}{\cdot\,\mathsf{sig}} \qquad \frac{\Sigma\,\mathsf{sig}}{(\Sigma, p\,\mathsf{mode})\,\mathsf{sig}} \qquad \frac{\Sigma\,\mathsf{sig} \quad (p_1\,\mathsf{mode}, \ldots, p_n\,\mathsf{mode}, q\,\mathsf{mode}) \in \Sigma}{(\Sigma, c:\ p_1, \ldots, p_n \to q)\,\mathsf{sig}}$$

$$\frac{\Sigma\,\mathsf{sig} \quad \psi\mathsf{ctx}_\Sigma \quad p\,\mathsf{mode} \in \Sigma \quad \psi \vdash_\Sigma \alpha : p \quad \psi \vdash_\Sigma \alpha' : p}{(\Sigma, (\alpha \equiv \alpha' : \psi \to p))\,\mathsf{sig}} \qquad \frac{\Sigma\,\mathsf{sig} \quad \psi\mathsf{ctx}_\Sigma \quad p\,\mathsf{mode} \in \Sigma \quad \psi \vdash_\Sigma \alpha : p \quad \psi \vdash_\Sigma \alpha' : p}{(\Sigma, (\alpha \Rightarrow \alpha' : \psi \to p))\,\mathsf{sig}}$$

$$\boxed{\text{Mode contexts } \psi\,\mathsf{ctx}_\Sigma} \qquad \frac{}{\cdot\,\mathsf{ctx}_\Sigma} \qquad \frac{\psi\,\mathsf{ctx}_\Sigma \quad p\,\mathsf{mode} \in \Sigma}{(\psi, x : p)\,\mathsf{ctx}_\Sigma}$$

$$\boxed{\text{Context descriptors } \psi \vdash_\Sigma \alpha : p, \text{ where } \psi\,\mathsf{ctx}_\Sigma \text{ and } p\,\mathsf{mode} \in \Sigma} \qquad \frac{x : p \in \psi}{\psi \vdash_\Sigma x : p} \qquad \frac{(\mathsf{c} : p_1, \ldots, p_n \to q) \in \Sigma \quad \psi \vdash_\Sigma \alpha_i : p_i}{\psi \vdash_\Sigma \mathsf{c}(\alpha_1, \ldots, \alpha_n) : q}$$

$$\boxed{\text{Mode Substitutions } \psi \vdash_\Sigma \gamma : \psi', \text{ where } \psi\,\mathsf{ctx}_\Sigma \text{ and } \psi'\,\mathsf{ctx}_\Sigma} \qquad \frac{}{\psi \vdash_\Sigma \cdot : \cdot} \qquad \frac{\psi \vdash_\Sigma \gamma : \psi' \quad \psi \vdash_\Sigma \alpha : p}{\psi \vdash_\Sigma \gamma, \alpha/x : \psi', x : p}$$

$$\boxed{\text{Equalities of mode morphisms } \psi \vdash_\Sigma \alpha \equiv \alpha' : p, \text{ where } \psi\,\mathsf{ctx}_\Sigma \text{ and } p\,\mathsf{mode} \in \Sigma \text{ and } \psi \vdash_\Sigma \alpha : p \text{ and } \psi \vdash_\Sigma \alpha' : p}$$

$$\frac{}{\psi \vdash_\Sigma \alpha \equiv \alpha : p} \qquad \frac{\psi \vdash_\Sigma \alpha_2 \equiv \alpha_1 : p}{\psi \vdash_\Sigma \alpha_1 \equiv \alpha_2 : p} \qquad \frac{\psi \vdash_\Sigma \alpha_1 \equiv \alpha_2 : p \quad \psi \vdash_\Sigma \alpha_2 \equiv \alpha_3 : p}{\psi \vdash_\Sigma \alpha_1 \equiv \alpha_3 : p}$$

$$\frac{\psi, x : p, \psi' \vdash_\Sigma \beta \equiv \beta' : q \quad \psi, \psi' \vdash_\Sigma \alpha \equiv \alpha' : p}{\psi, \psi' \vdash_\Sigma \beta[\alpha/x] \equiv \beta'[\alpha'/x] : q} \qquad \frac{(\alpha \equiv \alpha' : \psi \to p) \in \Sigma}{\psi \vdash_\Sigma \alpha \equiv \alpha' : p}$$

$$\boxed{\text{Structural transformations } \psi \vdash_\Sigma \alpha \Rightarrow_p \alpha', \text{ where } \psi \vdash_\Sigma \alpha : p \text{ and } \psi \vdash_\Sigma \alpha' : p} \qquad \frac{}{\psi \vdash_\Sigma \alpha \Rightarrow_p \alpha}$$

$$\frac{\psi \vdash_\Sigma \alpha_1 \Rightarrow_p \alpha_2 \quad \psi \vdash_\Sigma \alpha_2 \Rightarrow_p \alpha_3}{\psi \vdash_\Sigma \alpha_1 \Rightarrow_p \alpha_3} \qquad \frac{\psi, x : p, \psi' \vdash_\Sigma \beta \Rightarrow_q \beta' \quad \psi, \psi' \vdash_\Sigma \alpha \Rightarrow_p \alpha'}{\psi, \psi' \vdash_\Sigma \beta[\alpha/x] \Rightarrow_q \beta'[\alpha'/x]} \qquad \frac{(\alpha \Rightarrow \alpha' : \psi \to p) \in \Sigma}{\psi \vdash_\Sigma \alpha \Rightarrow_p \alpha'}$$

Figure 1: Syntax for mode theories

mode declaration, or a signature extended with a typed constant/function symbol, all of whose modes are declared previously in the signature. The notation $p_1,\ldots,p_n \to q$ is not itself a mode, but notation for declaring a function symbol in the signature (it cannot occur on the right-hand side of a typing judgement). For example, the type and term constructors for a monoid $(\odot, 1)$ are represented by a signature $\mathsf{p}\,\mathsf{mode}, \odot : (\mathsf{p},\mathsf{p} \to \mathsf{p}), 1 : (\to \mathsf{p})$.

The judgement $\psi\,\mathsf{ctx}_\Sigma$ defines well-formedness of a context of variable declarations relative to a signature Σ: each mode in the context must be declared in the signature. The judgement $\psi \vdash_\Sigma \alpha : p$ defines well-typedness of context descriptor terms, which are either a variable declared in the context, or a constant declared in the signature applied to arguments of the correct types. The judgement $\psi \vdash_\Sigma \gamma : \psi'$ defines a substitution as a tuple of terms in the standard way. The context ψ in these judgements enjoys the cartesian structural properties (associativity, unit, weakening, exchange, contraction). Simultaneous substitution into terms and substitutions is defined as usual (e.g. $x[\gamma, \alpha/x] := \alpha$ and $\mathsf{c}(\vec{\alpha}_i)[\gamma] := \mathsf{c}(\alpha_i[\gamma])$).

Returning to the top of the figure, the final two rules of the judgement $\Sigma\,\mathsf{sig}$ permit two additional forms of signature declaration. The first of these extends a signature with an equational axiom between two terms α and α' that have the same mode p, in the same context ψ, relative to the prior signature Σ. These equational axioms will be used to encode reversible object language structural properties, such as associativity, commutativity, and unit laws. For example, to specify the right unit law for the above monoid $(\odot, 1)$, we add an axiom $(x \odot 1 \equiv x : (x : \mathsf{p}) \to \mathsf{p})$ to the signature, which can be read as "$x \odot 1$ is equal to x as a morphism from $(x : \mathsf{p})$ to p". The judgement $\psi \vdash_\Sigma \alpha \equiv \alpha' : p$ is the least congruence closed under these axioms.

The second of these extends a signature with a directed structural transformation axiom between two terms α and α' that have the same mode p, in the same context ψ, relative to the prior signature Σ. As discussed above, these structural transformations will be used to represent object language structural properties such as weakening and contraction that are not invertible. The judgement $\psi \vdash_\Sigma \alpha \Rightarrow_p \alpha'$ defines these transformations: it is the least precongruence (preorder compatible with the term formers) closed under the axioms specified in the signature Σ. For example, to say that the above monoid $(\odot, 1)$ is affine, we add in Σ a transformation axiom $(x \Rightarrow 1 : (x : \mathsf{p}) \to \mathsf{p})$. An alternative to including the judgement $\alpha \equiv \alpha'$ would be to present a desired equation $\alpha \equiv \alpha'$ as an isomorphism, with transformation axioms $s : \alpha \Rightarrow \alpha'$ and $s' : \alpha' \Rightarrow \alpha$. While this is conceptually and technically sufficient, we have found it helpful in examples to use "strict" equality of context descriptors. This simplifies the description of some situations, though the difference is important mainly at the level of identity of derivations rather than provability—for example, we can make a binary operation $\odot$ into a strict monoid, rather than adding associator and unitor isomorphisms.

Because context descriptors α and their equality $\alpha_1 \equiv \alpha_2$ are defined prior to the subsequent judgements, we suppress this equality by using α to refer to a term-modulo-$\equiv$—that is, we assume a metatheory with quotient sets/types, and use meta-level equality for object-level equality, as recently advocated by Altenkirch and Kaposi [2016]. For example, because the judgement $\psi \vdash \alpha \Rightarrow_p \beta$ is indexed by equivalence classes of context descriptions, the reflexivity rule above implicitly means $\alpha \equiv \beta$ implies $\alpha \Rightarrow \beta$. As discussed in Section 5, we will eventually need an equational theory between two structural property derivations $s \equiv s' :: \psi \vdash \alpha \Rightarrow_q \alpha'$. Because this equational theory does not influence provability in the sequent calculus, only identity of proofs, we defer the details to that section.

In examples, we will notate a signature declaration introducing a term constant/function symbol by showing the function symbol applied to variables, rather than writing the formal $\mathsf{c} : p_1,\ldots,p_n \to q$. For example, we write $x : \mathsf{p}, y : \mathsf{p} \vdash x \odot y : \mathsf{p}$ for $\odot : \mathsf{p},\mathsf{p} \to \mathsf{p}$. We also suppress the signature Σ.

2.2 Sequent Calculus Rules

For a fixed mode theory Σ, we define a second layer of judgements in Figure 2. The first judgement assigns each proposition/type A a mode p. Encodings of non-modal logics will generally only make use of one mode, while modal logics use different modes to represent different notions of truth, such as the linear and cartesian categories in the adjoint decomposition of linear logic [Benton, 1995, Benton and Wadler, 1996] and the true/valid/lax judgements in modal logic [Pfenning and Davies, 2001]. The next judgement assigns each context Γ a mode context ψ. Formally, we think of contexts as ordered: we do not regard $x : A, y : B$ and $y : B, x : A$ as the same context, though we will have an admissible exchange rule that passes between derivations in one and the other.

The sequent judgement $\Gamma \vdash_\alpha A$ relates a context $\Gamma\,\mathsf{ctx}_\psi$ and a type $A\,\mathsf{type}_p$ and context descriptor $\psi \vdash \alpha : p$, while the substitution judgement $\Gamma \vdash_\gamma \Delta$ relates $\Gamma\,\mathsf{ctx}_\psi$ and $\Delta\,\mathsf{ctx}_{\psi'}$ and $\psi \vdash \gamma : \psi'$. Because $\Gamma\,\mathsf{ctx}_\psi$ means that each variable

$$\boxed{\text{Types } A, B, C \quad A\ \mathsf{type}_p} \qquad \frac{}{P\ \mathsf{type}_p} \qquad \frac{\psi \vdash \alpha : q \quad \Delta\ \mathsf{ctx}_\psi}{\mathsf{F}_\alpha(\Delta)\ \mathsf{type}_q} \qquad \frac{\psi, x : q \vdash \alpha : p \quad \Delta\ \mathsf{ctx}_\psi \quad A\ \mathsf{type}_p}{\mathsf{U}_{x.\alpha}(\Delta \mid A)\ \mathsf{type}_q}$$

$$\boxed{\text{Contexts } \Gamma, \Delta \quad \Gamma\ \mathsf{ctx}_\psi} \qquad \frac{}{\cdot\ \mathsf{ctx}_\cdot} \qquad \frac{\Gamma\ \mathsf{ctx}_\psi \quad A\ \mathsf{type}_p}{\Gamma, x : A\ \mathsf{ctx}_{\psi, x:p}}$$

$$\boxed{\Gamma \vdash_\alpha A \text{ where } \Gamma\ \mathsf{ctx}_\psi \text{ and } A\ \mathsf{type}_q \text{ and } \psi \vdash \alpha : q} \qquad \frac{\Gamma, \Gamma', \Delta \vdash_{\beta[\alpha/x]} C}{\Gamma, x : \mathsf{F}_\alpha(\Delta), \Gamma' \vdash_\beta C}\ \mathsf{FL} \qquad \frac{\beta \Rightarrow \alpha[\gamma] \quad \Gamma \vdash_\gamma \Delta}{\Gamma \vdash_\beta \mathsf{F}_\alpha(\Delta)}\ \mathsf{FR}$$

$$\frac{x : \mathsf{U}_{x.\alpha}(\Delta \mid A) \in \Gamma \quad \beta \Rightarrow \beta'[\alpha[\gamma]/z] \quad \Gamma \vdash_\gamma \Delta \quad \Gamma, z{:}A \vdash_{\beta'} C}{\Gamma \vdash_\beta C}\ \mathsf{UL} \qquad \frac{\Gamma, \Delta \vdash_{\alpha[\beta/x]} A}{\Gamma \vdash_\beta \mathsf{U}_{x.\alpha}(\Delta \mid A)}\ \mathsf{UR} \qquad \frac{x : P \in \Gamma \quad \beta \Rightarrow x}{\Gamma \vdash_\beta P}\ \mathsf{v}$$

$$\boxed{\Gamma \vdash_\gamma \Delta \text{ where } \Gamma\ \mathsf{ctx}_\psi \text{ and } \Delta\ \mathsf{ctx}_{\psi'} \text{ and } \psi \vdash \gamma : \psi'} \qquad \frac{}{\Gamma \vdash_\cdot \cdot}\ \cdot \qquad \frac{\Gamma \vdash_\gamma \Delta \quad \Gamma \vdash_\alpha A}{\Gamma \vdash_{\gamma, \alpha/x} \Delta, x : A}\ {-,-}$$

Figure 2: Sequent Calculus

in Γ is in ψ, where $x : A_i \in \Gamma$ implies $x : p_i$ in ψ with A_i type_{p_i}, we think of Γ as binding variable names both in α and for use in the derivation.

As discussed in the introduction, a guiding principle is to make the following rules admissible (see Section 4 for details), which express respect for structural transformations and structurality-over-structurality:

$$\frac{\alpha \Rightarrow \beta \quad \Gamma \vdash_\beta A}{\Gamma \vdash_\alpha A}\ \textit{Lem}\ 4.1 \qquad \frac{}{\Gamma, x : A \vdash_x A}\ \textit{Thm}\ 4.4 \qquad \frac{\Gamma, x : A \vdash_\beta B \quad \Gamma \vdash_\alpha A}{\Gamma \vdash_{\beta[\alpha/x]} B}\ \textit{Thm}\ 4.6$$

$$\frac{\Gamma \vdash_\alpha C}{\Gamma, y : A \vdash_\alpha C}\ \textit{Lem}\ 4.2 \qquad \frac{\Gamma, x : A, y : B \vdash_\alpha C}{\Gamma, y : B, x : A \vdash_\alpha C}\ \textit{Lem}\ 4.3 \qquad \frac{\Gamma, x : A, y : A \vdash_\alpha C}{\Gamma, x : A \vdash_{\alpha[x/y]} C}\ \textit{Cor}\ 4.7$$

We now explain the rules for the sequent calculus; the reader may wish to refer to the examples in Section 3 in parallel with this abstract description. We assume atomic propositions P are given a specified mode p, and state identity as a primitive rule only for them with the v rule. This says that $\Gamma, x : P \vdash_x P$, and additionally composes with a structural transformation $\beta \Rightarrow x$. Using a structural property at a leaf of a derivation is common in e.g. affine logic, where the derivation of $\beta \Rightarrow x$ would use weakening to forget any additional resources besides x.

Next, we consider the $\mathsf{F}_\alpha(\Delta)$ type, which "internalizes" the context operation α as a type/proposition. Syntactically, we view the context $\Delta = x_1 : A_1, \ldots, x_n : A_n$ where A_i type_{p_i} as binding the variables $x_i : p_i$ in α, so for example $\mathsf{F}_\alpha(x : A, y : B)$ and $\mathsf{F}_{\alpha[x \leftrightarrow x']}(x' : A, y : B)$ are α-equivalent types (in de Bruijn form we would write $\mathsf{F}_\alpha(A_1, \ldots, A_n)$ and use indices in α). The type formation rule says that F moves covariantly along a mode morphism α, representing a "product" (in a loose sense) of the types in Δ structured according to the context descriptor α. A typical binary instance of F is a multiplicative product ($A \otimes B$ in linear logic), which, given a binary context descriptor $\odot$ as in the introduction, is written $\mathsf{F}_{x \odot y}(x : A, y : B)$. A typical nullary instance is a unit (1 in linear logic), written $\mathsf{F}_1()$. A typical unary instance is the F connective of adjoint logic, which for a unary context descriptor constant $\mathsf{f} : \mathsf{p} \to \mathsf{q}$ is written $\mathsf{F}_{\mathsf{f}(x)}(x : A)$. We sometimes write $\mathsf{F}_\mathsf{f}(A)$ in this case, eliding the variable name, and similarly for a unary U.

The rules for our F connective capture a pattern common to all of these examples. The left FL rule says that $\mathsf{F}_\alpha(\Delta)$ "decays" into Δ, but *structuring the uses of resources in* Δ *with* α *by the substitution* $\beta[\alpha/x]$. We assume that Δ is α-renamed to avoid collision with Γ (the proof term here is a "split" that binds variables for each position in Δ). The placement of Δ at the right of the context is arbitrary (because we have exchange-over-exchange), but we follow the convention that new variables go on the right to emphasize that Γ behaves mostly as in ordinary cartesian logic. The right FR rule says that you must rewrite (using structural transformations) the context descriptor to have an α at the outside, with a mode substitution γ that divides the existing resources up between the positions in Δ, and then prove each formula in Δ using the specified resources. We leave the typing of γ implicit, though there is officially a requirement $\psi \vdash \gamma : \psi'$ where $\Gamma\ \mathsf{ctx}_\psi$ and $\Delta\ \mathsf{ctx}_{\psi'}$, as required for the second premise to be a well-formed sequent. Another way to understand this rule is to begin with the "axiomatic FR" instance $\mathsf{FR}^* :: \Delta \vdash_\alpha \mathsf{F}_\alpha(\Delta)$ which says that there is a map from Δ to $\mathsf{F}_\alpha(\Delta)$ along α. Then, in the same way that a typical right rule for coproducts builds a precomposition into an "axiomatic injection" such as $\mathsf{inl} :: A \vdash A+B$, the FR rule builds a precomposition with $\Gamma \vdash_\gamma \Delta$ and then an application of a structural rule $\beta \Rightarrow \alpha[\gamma]$ into the "axiomatic" version, in order to make cut and respect for transformations admissible.

Next, we turn to $\mathsf{U}_{x.\alpha}(\Delta \mid A)$. As a first approximation, if we ignore the context descriptors and structural properties, $\mathsf{U}_{-}(\Delta \mid A)$ behaves like $\Delta \to A$, and the UL and UR rules are an annotation of the usual structural/cartesian rules for implication. In a formula $\mathsf{U}_{x.\alpha}(\Delta \mid A)$, the context descriptor α has access to the variables from Δ as well as an extra variable x, whose mode is the same as the *overall mode of* $\mathsf{U}_{x.\alpha}(\Delta \mid A)$, while the mode of A itself is the mode of the conclusion of α—in terms of typing, U is contravariant where F is covariant. It is helpful to think of x as standing for the context that will be used to prove $\mathsf{U}_{x.\alpha}(\Delta \mid A)$. For example, a typical function type $A \multimap B$ is represented by $\mathsf{U}_{x.x\otimes y}(y : A \mid B)$, which says to extend the "current context" x with a resource y. In UR, the context descriptor β being used to prove the U is substituted *for* x in α (dual to FL, which substituted α into β). The "axiomatic" UL instance $\mathsf{UL}^* :: \Delta, x : \mathsf{U}_{x.\alpha}(\Delta \mid A) \vdash_\alpha A$ says that $\mathsf{U}_{x.\alpha}(\Delta \mid A)$ together with Δ has a map to A along α. (The bound x in $x.\alpha$ subscript is tacitly renamed to match the name of the assumption in the context, in the same way that the typing rule for $\lambda x.e : \Pi x : A.B$ requires coordination between two variables in different scopes). The full rule builds in precomposition with $\Gamma \vdash_\gamma \Delta$, postcomposition with $\Gamma, z : A \vdash_{\beta'} C$, and precomposition with $\beta \Rightarrow \beta'[\alpha[\gamma]/z]$.

Finally, the rules for substitutions are pointwise. In examples, we will write the components of a substitution directly as multiple premises of FR and UL, rather than packaging them with $\_,\_$ and $\cdot$.

One subtle point about the FL rule is that there are two competing principles: making the rules "obviously" structural-over-structural, and reducing inessential non-determinism. Here, we choose the later, and treat the assumption of $\mathsf{F}_\alpha(\Delta)$ affinely, removing it from the context when it is used. It will turn out that the judgement nonetheless enjoys contraction-over-contraction (Corollary 4.7), because contraction for negatives is built into the UL-rule, and contraction for positives follows from this and the fact that we can always reconstruct a positive from what it decays to on the left (c.f. how purely positive formulas have contraction in linear logic).

Additives can be added to this sequent calculus; e.g. a mode p has sums $A_p + B_p\ \mathsf{type}_p$ if

$$\frac{\Gamma \vdash_\alpha A}{\Gamma \vdash_\alpha A+B} \quad \frac{\Gamma \vdash_\alpha B}{\Gamma \vdash_\alpha A+B} \quad \frac{\Gamma, \Gamma', y : A \vdash_{\beta[y/x]} C \quad \Gamma, \Gamma', z : B \vdash_{\beta[z/x]} C}{\Gamma, x : A+B, \Gamma' \vdash_\beta C}$$

3 Examples

In this section, we give some examples of logical connectives that can be represented by mode theories in this framework, and explain informally why they have the desired behavior with respect to provability. We give some formal adequacy (soundness and completeness of provability) proofs in Section 7.

3.1 Non-associative products

A mode theory with one mode m and a constant

$$x : \mathsf{m}, y : \mathsf{m} \vdash x \odot y : \mathsf{m}$$

specifies a completely astructural context (no weakening, exchange, contraction, associativity), as in non-associative Lambek calculus [Lambek, 1958].

If we write $A \odot B$ for $\mathsf{F}_{x \odot y}(x : A, y : B)$ we *cannot*, for example, derive associativity $A \odot (B \odot C) \vdash (A \odot B) \odot C$. To attempt a derivation, we can (without loss of generality) begin by applying the invertible (Lemma 4.5) FL rule twice, at which point no further left rules are possible, so we must apply FR:

$$\dfrac{\dfrac{\dfrac{\begin{array}{c} x \odot (y \odot z) \Rightarrow (q \odot z)[\alpha_1/q, \alpha_2/z] \\ x : A, y : B, z : C \vdash_{\alpha_1} \mathsf{F}_{x \odot y}(x : A, y : B) \\ x : A, y : B, z : C \vdash_{\alpha_2} C \end{array}}{x : A, y : B, z : C \vdash_{x \times (y \odot z)} \mathsf{F}_{q \odot z}(q : \mathsf{F}_{x \odot y}(x : A, y : B), z : C)} \;\mathsf{FR}}{x : A, p : \mathsf{F}_{y \odot z}(y : B, z : C) \vdash_{x \times p} \mathsf{F}_{q \odot z}(q : \mathsf{F}_{x \odot y}(x : A, y : B), z : C)} \;\mathsf{FL}}{a : \mathsf{F}_{x \odot p}(x : A, p : \mathsf{F}_{y \odot z}(y : B, z : C)) \vdash_a \mathsf{F}_{q \odot z}(q : \mathsf{F}_{x \odot y}(x : A, y : B), z : C)} \;\mathsf{FL}$$

To apply FR, we need to find a substitution for α_1/q and α_2/z with a structural transformations as above. In the absence of any equational or transformation axioms, the only possible choice is reflexivity, which requires $x/q, (y \odot z)/z$. Thus we need to show

$$x : A, y : B, z : C \vdash_x A \odot B \qquad x : A, y : B, z : C \vdash_{y \odot z} C$$

This is not possible because the context is not divided correctly.

3.2 Ordered Products and Implications

We extend the above mode theory with a constant $1 : \mathsf{m}$ and equations

$$\begin{array}{c} x \odot (y \odot z) \equiv (x \odot y) \odot z \\ x \odot 1 \equiv x \equiv 1 \odot x \end{array}$$

making $(\odot, 1)$ into a monoid. This makes the context behave like ordered logic, which has associativity but none of exchange, weakening, and contraction—a monoidal product that is not symmetric monoidal.

We can complete the above proof of associativity of $\odot$: where we need to find a substitution such that $x \odot (y \odot z) \Rightarrow (q \odot z)[\alpha_1/q, \alpha_2/z]$, we can now choose $(x \odot y)/q, z/z$ because

$$x \odot (y \odot z) \equiv (x \odot y) \odot z = (q \odot z)[x \odot y/q, z/z]$$

Thus, the subgoals are

$$x : A, y : B, z : C \vdash_{x \odot y} A \odot B \qquad x : A, y : B, z : C \vdash_z C$$

The latter is identity-over-identity (Theorem 4.4), and the former is a further FR and then identities:

$$\dfrac{\begin{array}{c} x \otimes y \Rightarrow (x' \odot y')[x/x', y/y'] \\ x : A, y : B, z : C \vdash_x A \\ x : A, y : B, z : C \vdash_y B \end{array}}{x : A, y : B, z : C \vdash_{x \odot y} \mathsf{F}_{x' \odot y'}(x' : A, y' : B)}$$

However, we cannot prove commutativity:

$$\dfrac{\dfrac{x \odot y \Rightarrow (z \odot w)[\alpha_1/z, \alpha_2/w] \quad x : A, y : B \vdash_{\alpha_1} B \quad x : A, y : B \vdash_{\alpha_2} A}{x : A, y : B \vdash_{x \odot y} \mathsf{F}_{z \odot w}(z : B, w : A)} \;\mathsf{FR}}{p : \mathsf{F}_{x \odot y}(x : A, y : B) \vdash_p \mathsf{F}_{z \odot w}(z : B, w : A)} \;\mathsf{FL}$$

because the only choice is $\alpha_1 = x$ and $\alpha_2 = y$, which sends the wrong resource to each branch.

Ordered logic has two different implications, one that adds to the left of the context, and one that adds to the right; the expected rules are

$$\dfrac{\Gamma, A \vdash^{\circ} B}{\Gamma \vdash^{\circ} A \rightharpoonup B} \quad \dfrac{\Delta \vdash^{\circ} A \quad \Gamma, B, \Gamma' \vdash^{\circ} C}{\Gamma, A \rightharpoonup B, \Delta, \Gamma' \vdash^{\circ} C} \quad \dfrac{A, \Gamma \vdash^{\circ} B}{\Gamma \vdash^{\circ} A \leftharpoondown B} \quad \dfrac{\Delta \vdash^{\circ} A \quad \Gamma, B, \Gamma' \vdash^{\circ} C}{\Gamma, \Delta, A \leftharpoondown B, \Gamma' \vdash^{\circ} C}$$

We represent these by

$$A \rightharpoonup B := \mathsf{U}_{c.c \odot x}(x : A \mid B) \quad A \leftharpoondown B := \mathsf{U}_{c.x \odot c}(x : A \mid B)$$

These have the expected right rules, putting x on the left or right of the current context descriptor, by the substitution β/c in UR:

$$\frac{\Gamma, x : A \vdash_{\beta \odot x} B}{\Gamma \vdash_\beta \mathsf{U}_{c.c \odot x}(x : A \mid B)} \qquad \frac{\Gamma, x : A \vdash_{x \odot \beta} B}{\Gamma \vdash_\beta \mathsf{U}_{c.x \odot c}(x : A \mid B)}$$

The instances of UL are

$$\frac{\begin{array}{l} c : \mathsf{U}_{c.c \odot x}(x : A \mid B) \in \Gamma \\ \beta \Rightarrow \beta'[c \odot \alpha/z] \\ \Gamma \vdash_\alpha A \\ \Gamma, z : A \vdash_{\beta'} C \end{array}}{\Gamma \vdash_\beta C} \qquad \frac{\begin{array}{l} c : \mathsf{U}_{c.x \odot c}(x : A \mid B) \in \Gamma \\ \beta \Rightarrow \beta'[\alpha \odot c/z] \\ \Gamma \vdash_\alpha A \\ \Gamma, z : A \vdash_{\beta'} C \end{array}}{\Gamma \vdash_\beta C}$$

Suppose that β is of the form $x_1 \odot \ldots c \ldots \odot x_n$ for distinct variables x_i, and consider the rule on the left, for $\rightharpoonup$. Because the only structural transformations are the associativity and unit equations, the transformation must reassociate β as $\beta_1 \odot (c \odot \alpha) \odot \beta_2$, with $\beta' = \beta_1 \odot z \odot \beta_2$, for some β_1 and β_2. Here α plays the role of Δ in the ordered logic rule—the resources used to prove A, which occur to the right of the implication being eliminated. Reading the substitution backwards, the resources β' used for the continuation are "β with $c \odot \alpha$ replaced by the result of the implication," as desired. While c and any variables used in α are still in Γ, permission to use them has been removed from β'—and there is no way to restore such permissions in this mode theory. The rule for $\leftharpoonup$ is the same, but with α on the opposite side of c.

More formally, for an ordered logic formula built from $\odot$ $\leftharpoonup$ $\rightharpoonup$ and atoms, write A^* for the translation to the above encodings, and extend this pointwise to Γ^* for an ordered logic context Γ. Further, define $\overline{x_1 : A_1, \ldots, x_n : A_n} = x_1 \odot \ldots \odot x_n$. Then the encoding of ordered logic is adequate in the sense that $\Gamma \vdash^{\circ} A$ iff $\Gamma^* \vdash_{\overline{\Gamma}} A^*$ (see Section 7). The analogous translation of types and judgements and adequacy statement is used for Examples 3.3,3.5,3.6.

3.3 Linear products and implication

Linear logic is ordered logic with exchange, so to model this we add a commutativity equation

$$x \otimes y \equiv y \otimes x$$

(and switch notation from $\odot$ to $\otimes$). For example, we can derive $p : A \otimes B \vdash_p B \otimes A$:

$$\cfrac{\cfrac{x \otimes y \Rightarrow (z \otimes w)[y/z, x/w] \quad x : A, y : B \vdash_y B \quad x : A, y : B \vdash_x A}{x : A, y : B \vdash_{x \otimes y} \mathsf{F}_{z \otimes w}(z : B, w : A)}\ \mathsf{FR}}{p : \mathsf{F}_{x \otimes y}(x : A, y : B) \vdash_p \mathsf{F}_{z \otimes w}(z : B, w : A)}\ \mathsf{FL}$$

where the first premise is exactly $x \otimes y = y \otimes x$.

For this mode theory, $\mathsf{U}_{c.c \odot x}(x : A \mid B)$ and $\mathsf{U}_{c.x \odot c}(x : A \mid B)$ are equal types (because commutativity is an equation, and types are parametrized by equivalence-classes of context descriptors), and both represent $A \multimap B$.

3.4 Multi-use variables

An n-use variable (see [Reed, 2008] for example) is like a linear variable, but instead of being used "exactly once" (modulo additives), it is used "exactly n times." In the above work, 0-use variables were used in an encoding of nominal techniques; another application of n-use variables is static analysis of functional programs (e.g. counting how many times a variable occurs to decide whether it will be efficient to unfold a substitution).

We use the following sequent calculus rules for n-linear functions

$$\frac{}{0 \cdot \Gamma, x :^1 P \vdash P} \qquad \frac{\Gamma, x :^n A \vdash B}{\Gamma \vdash A \to^n B} \qquad \frac{\Delta \vdash A \quad \Gamma, z :^k B \vdash C}{\Gamma + f :^k A \to^n B + (nk \cdot \Delta) \vdash C}$$

where $\Gamma + \Delta$ acts pointwise by $x :^n A + x :^m A = x :^{n+m} A$ and $n \cdot \Delta$ acts pointwise by $n \cdot x^m A = x :^{nm} A$. In the left rule, Γ and Δ have the same underlying variables and types (but potentially different counts), and $f :^k A \to^n B$ abbreviates a context with the same variables and types but 0's for all counts besides f's. The left rule says that if you spend k "uses" of a function that takes n uses of an argument, then you need nk uses of whatever you use to construct the argument, in order to get k uses of the result.

We can model this in the linear mode theory by using context descriptors that are themselves non-linear:

$$\begin{array}{rcl} x^0 & := & 1 \\ x^{n+1} & := & x^n \otimes x \\ A \to^n B & := & \mathsf{U}_{c.c\otimes(x^n)}(x : A \mid B) \end{array}$$

This has the following instances of UL and UR:

$$\frac{\Gamma, x : A \vdash_{\beta \otimes x^n} B}{\Gamma \vdash_\beta A \to^n B} \qquad \frac{\begin{array}{l} f : \mathsf{U}_{f.f\otimes x^n}(x : A \mid B) \in \Gamma \\ \beta \Rightarrow \beta'[f \otimes (\alpha)^n/z] \\ \Gamma \vdash_\alpha A \\ \Gamma, z : B \vdash_{\beta'} C \end{array}}{\Gamma \vdash_\beta C}$$

For this mode theory, the only transformations are the commutative monoid equations, and we can commute β' to the form $\beta'' \otimes z^k$ for some k and β'' not mentioning z because any context descriptor is a polynomial of variables. Thus the premise is really of form $\beta \equiv (\beta'' \otimes z^k)[f \otimes (\alpha)^n/z]$, which is equal to $\beta'' \otimes f^k \otimes (\alpha)^{nk}$. Here β'' corresponds to the Γ in the above left rule (the resources used in the continuation, besides z^k) and α corresponds to Δ. Overall, we have $x_1 :^{k_1} A_1, \ldots, x_n :^{k_n} A_n \vdash C$ iff $x_1 : A_1^*, \ldots, x_n : A_n^* \vdash_{x_1^{k_1} \otimes \ldots \otimes x_n^{k_n}} C^*$ (where A^* translates atoms to themselves and each $A \to^n B$ as indicated above).

We can also consider an n-use product $A^n := \mathsf{F}_{x^n}(x : A)$ as a positive type, which will decompose $A \to^n B$ as $A^n \multimap B$ (by Lemma 4.9). This has a map $p : A^n \vdash_p A \otimes \ldots \otimes A$ but not a converse map $p : A \otimes \ldots \otimes A \vdash_p A^n$. For example, we have

$$\cfrac{\cfrac{x \otimes x \equiv (y \otimes z)[x/y, x/z] \quad x : A \vdash_x A \quad x : A \vdash_x A}{x : A \vdash_{x\otimes x} \mathsf{F}_{x\otimes y}(x : A, y : A)}}{p : \mathsf{F}_{x\otimes x}(x : A) \vdash_p \mathsf{F}_{x\otimes y}(x : A, y : A)}$$

the essence of which is the contraction in the substitution $[x/y, x/z]$. However,

$$\cfrac{\cfrac{y \otimes z \equiv (x \otimes x)[?/x] \quad \ldots}{y : A, z : A \vdash_{y\otimes z} \mathsf{F}_{x\otimes x}(x : A)}}{p : \mathsf{F}_{y\otimes z}(y : A, z : A) \vdash_p \mathsf{F}_{x\otimes x}(x : A)}$$

is not derivable, because there is no substitution into $x \otimes x$ that makes it equal to $y \otimes z$ for distinct y and z. Conceptually, we think of A^2 as expressing a notion of identity: it is a *single* A that can be used twice, which is stronger than having two potentially different A's.

3.5 Affine products and implications

If we extend the linear logic mode theory with our first directed structural transformation w :: $x \Rightarrow 1$ then we get weakening. For example, we can define a projection

$$\cfrac{\cfrac{\cfrac{y \Rightarrow 1}{x \otimes y \Rightarrow x} \quad \cfrac{}{x : A, y : B \vdash_x A}\ \textit{Theorem } 4.4}{x : A, y : B \vdash_{x\otimes y} A}\ \textit{Lemma } 4.1}{p : A \otimes B \vdash_p A}\ \mathsf{FL}$$

3.6 Relevant and Cartesian products and implications

Next, we consider a logic with contraction, e.g. a map $A \vdash (A \otimes A)$. We always have the left and right components of the chain

$$A \quad \cong \quad \mathsf{F}_x(x : A) \quad \vdash^? \quad \mathsf{F}_{x \otimes x}(x : A) \quad \vdash \quad \mathsf{F}_{x \otimes y}(x : A, y : A)$$

The left isomorphism is just FL/FR, while the right map was given in Example 3.4. To give the middle map $\mathsf{F}_x(x : A) \vdash \mathsf{F}_{x \otimes x}(x : A)$, it suffices to add a structural transformation $\mathsf{c} :: x \Rightarrow x \otimes x$. because F is covariant on structural transformations (Lemma 4.10). Then we have $A \vdash A^2 \vdash (A \otimes A)$ but neither of the converses.

Moreover, if we have both $\mathsf{w} :: x \Rightarrow 1$ and $\mathsf{c} :: x \Rightarrow x \otimes x$, then $x \otimes y$ will behave like a cartesian product in the mode theory (with projections $x \otimes y \Rightarrow x$ and $x \otimes y \Rightarrow y$ and pairing of $z \Rightarrow x$ and $z \Rightarrow y$ to $z \Rightarrow x \otimes y$), and consequently $A \otimes B$ will behave like a cartesian product type, and $\mathsf{U}_{c.c \otimes x}(x : A \mid B)$ like the usual structural $A \to B$. We refer to this mode theory as an *cartesian monoid* and write $(\times, \top)$ for it.

3.7 Bunched Implication (BI)

Bunched implication [O'Hearn and Pym, 1999] has two context-forming operations Γ, Γ' and $\Gamma; \Gamma'$, along with corresponding products and implications. Both are associative, unital, and commutative, but ; has weakening and contraction while , does not. A context is represented by a tree such as $(x : A, y : B); (z : C, w : D)$ (considered modulo the laws), and the notation $\Gamma[\Delta]$ is used to refer to a tree with a hole $\Gamma[-]$ that has Δ as a subtree at the hole. In sequent calculus style, the rules for the product and implication corresponding to , are

$$\frac{\Gamma[A, B] \vdash C}{\Gamma[A * B] \vdash C} \quad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A * B} \quad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \mathrel{-\!\!*} B} \quad \frac{\Delta \vdash A \quad \Gamma[B] \vdash C}{\Gamma[A \mathrel{-\!\!*} B, \Delta] \vdash C}$$

There are similar rules for a product and implication for ; as well as structural rules of weakening and contraction for it.

We can model BI by a mode m with a commutative monoid $(*, I)$ and a cartesian monoid $(\times, \top)$. We define the BI products and implications using the monoids:

$$\begin{array}{ll} A * B := \mathsf{F}_{x * y}(x : A, y : B) & A \mathrel{-\!\!*} B := \mathsf{U}_{c.c * x}(x : A \mid B) \\ A \times B := \mathsf{F}_{x \times y}(x : A, y : B) & A \to B := \mathsf{U}_{c.c \times x}(x : A \mid B) \end{array}$$

A context descriptor such as $(x \times y) * (z \times w)$ captures the "bunched" structure of a BI context, and substitution for a variable models the hole-filling operation $\Gamma[\Delta]$. The left rule for $*$ (and similarly $\times$) acts on a leaf

$$\frac{\Gamma, \Gamma', x : A, y : B \vdash_{\beta[x * y / z]} C}{\Gamma, z : A * B, \Gamma' \vdash_\beta C}$$

and replaces the leaf where z occurs in the tree β with the correct bunch $x * y$, The left rule for $\mathrel{-\!\!*}$ (and similarly for $\to$)

$$\frac{c : A \mathrel{-\!\!*} B \in \Gamma \quad \beta \Rightarrow \beta'[c * \alpha / z] \quad \Gamma \vdash_\alpha A \quad \Gamma, z : B \vdash_{\beta'} C}{\Gamma \vdash_\beta C}$$

isolates a subtree containing the implication c and resources $*$'ed with it, uses those resources to prove A, and then replaces the subtree with the variable z standing for the result of the implication.

We assume the BI sequent is given as a judgement $\Gamma \vdash A$ where Γ is a tree and there are explicit equality premises for the algebraic laws on bunches. Then we define Γ^* as an in-order flattening of the tree into one of our contexts (e.g. $(x : A)^* = x : A^*$ and $(\Gamma, \Delta)^* = (\Gamma; \Delta)^* = \Gamma^*, \Delta^*$), while we define $\overline{\Gamma}$ as a context descriptor that preserves the tree structure (e.g. $\overline{x : A} = x$ and $\overline{(\Gamma, \Delta)} = \overline{\Gamma} * \overline{\Delta}$ and $\overline{\Gamma; \Delta} = \overline{\Gamma} \times \overline{\Delta}$). Then we have the usual adequacy statement $\Gamma \vdash A$ iff $\Gamma^* \vdash_{\overline{\Gamma}} A^*$.

3.8 Adjoint decomposition of !

Following Benton [1995], Benton and Wadler [1996], we decompose the ! exponential of intuitionistic linear logic as the comonad of an adjunction between "linear" and "cartesian" categories. We start with two modes l (linear) and c (cartesian), along with a commutative monoid $(\otimes, 1)$ on l and a cartesian monoid $(\times, \top)$ on c. Next, we add a context descriptor from c to l ($x : \mathsf{c} \vdash \mathsf{f}(x) : \mathsf{l}$) that we think of as including a cartesian context in a linear context. This generates types

$$\mathsf{F}_{\mathsf{f}(x)}(x : A_{\mathsf{c}})\ \mathsf{type}_{\mathsf{l}} \qquad \mathsf{U}_{x.\mathsf{f}(x)}(\cdot \mid A_{\mathsf{l}})\ \mathsf{type}_{\mathsf{c}}$$

which are adjoint $\mathsf{F}_{\mathsf{f}(x)}(x : -) \dashv \mathsf{U}_{x.\mathsf{f}(x)}(\cdot \mid -)$. The bijection on hom-sets is defined using FL and FR and their invertibility (Corollary 4.8, Lemma 4.5):

$$\dfrac{\dfrac{x : A \vdash_x \mathsf{U}_{x.\mathsf{f}(x)}(\cdot \mid B)}{x : A \vdash_{\mathsf{f}(x)} B}}{p : \mathsf{F}_{\mathsf{f}(x)}(x : A) \vdash_p B}$$

The comonad of the adjunction $\mathsf{F}_{\mathsf{f}(x)}(x : \mathsf{U}_{c.\mathsf{f}(c)}(\cdot \mid A))$ is the linear logic $!A$.

In the LNL models and sequent calculus [Benton, 1995], $F(A \times B) \cong F(A) \otimes F(B)$ and $F(\top) \cong 1$, which we can add to the mode theory by equations

$$\mathsf{f}(x \times y) \equiv \mathsf{f}(x) \otimes \mathsf{f}(y) \qquad \mathsf{f}(\top) \equiv 1$$

These equations then extend to isomorphisms using Lemma 4.9 because all of $F, \otimes, \times$ are represented by F-types in our framework. These properties of f are necessary to prove that $!A$ has weakening and contraction (with respect to $\otimes$) and $!A \otimes !B \vdash !(A \otimes B)$, for example. Omitting these equations allows us to describe non-monoidal (or lax monoidal, if we add only one direction) left adjoints.

In general, we translate $F(A)^* = \mathsf{F}_{\mathsf{f}(x)}(x : C^*)$ and $G(A)^* = \mathsf{U}_{x.\mathsf{f}(x)}(\cdot \mid A)$ and products and functions as usual. Then a sequent $x_1 : C_1, \ldots, x_n : C_n \vdash C$ in the cartesian category is represented by a sequent $x_1 : C_1^*, \ldots, x_n : C_n^* \vdash_{x_1 \times \ldots \times x_n} C^*$, and a mixed sequent with cartesian and linear assumptions and a linear conclusion $x_1 : C_1, \ldots, x_n : C_n; y_1 : A_1, \ldots, y_m : A_m \vdash A$ by $x_1 : C_1^*, \ldots, y_1 : A_1^*, \ldots \vdash_{\mathsf{f}(x_1) \otimes \ldots \otimes \mathsf{f}(x_n) \otimes y_1 \otimes \ldots \otimes y_n} A^*$.

3.9 Adjoint decomposition of $\Box$

The modal S4 $\Box$ as in Pfenning and Davies [2001] is similar to !. We call the two modes truth and $\mathsf{validity}$ and have cartesian monoids on both (we write $(\times, \top)$ for the t one and $(\times_v, \top_v)$ for the v one) along with $x : \mathsf{v} \vdash \mathsf{f}(x) : \mathsf{t}$. Here, following the analysis of $\Box$ as a monoidal comonad [Alechina et al., 2001], we have only lax monoid-preservation axioms

$$\mathsf{f}(x) \times \mathsf{f}(y) \Rightarrow \mathsf{f}(x \times_v y) \qquad \top \Rightarrow \mathsf{f}(\top_v)$$

though the difference is only at the level of equality of derivations.[2] We represent a sequent

$$x_1 : A_1\ \mathsf{valid}, \ldots, x_n : A_n\ \mathsf{valid}; y_1 : B_1\ \mathsf{true}, \ldots \vdash C\ \mathsf{true}$$

by

$$x_1 : \mathsf{U}_{\mathsf{f}}(A_1^*), \ldots, x_n : \mathsf{U}_{\mathsf{f}}(A_n^*); y_1 : B_1^*, \ldots \vdash_{\mathsf{f}(x_1) \times \ldots \times \mathsf{f}(x_n) \times y_1 \times \ldots \times y_n} C^*$$

[2] Because the context monoids are cartesian products, there are always converse maps, e.g. $\mathsf{f}(x \times_v y) \Rightarrow \mathsf{f}(x) \times \mathsf{f}(y)$ defined by pairing, projection, and congruence. However, in the equational theory of proofs in S4 [Pfenning and Davies, 2001], there is a section-retraction $(\Box A \times \Box B) \rightarrowtail \Box(A \times B) \twoheadrightarrow (\Box A \times \Box B)$ but not an isomorphism. If we had equalities above, they would generate type isomorphisms $\mathsf{F}(A \times_v B) \cong \mathsf{F}(A) \times \mathsf{F}(B)$, and because the right-adjoint U preserves products, we would have $\mathsf{FU}(A \times B) \cong \mathsf{F}(UA \times_v UB) \cong (\mathsf{FU}(A) \times \mathsf{FU}(B))$, which does not match the existing theory—though it is a reasonable alternative to consider.

3.10 Subexponentials

Subexponentials [Danos et al., 1993, Nigam and Miller, 2009] extend linear logic with a family of comonads $!_a A$. All of the comonads are monoidal ($!_a A \otimes !_a B \vdash !_a (A \otimes B)$ and $1 \vdash !_a A$), and there is a preorder $a \leq b$ such that $!_b A \vdash !_a A$. Each $!_a$ is allowed to have weakening and/or contraction subject to the constraint that when $a \leq b$, b must be at least as structural as a.

We illustrate the embedding on a specific example of the diamond preorder generated by $\mathsf{i} < \mathsf{j}, \mathsf{k} < \mathsf{m}$. Following [Reed, 2009a, Example 4.3], we identify each subexponential a with a mode, and have an additional mode l for basic linear truth, all with commutative monoids $(\otimes_a, 1_a)$. We add context descriptor constants $x : b \vdash ba(x) : a$ for each $a < b$ (so, in this example, mk, mj, ji, ki), with an additional $x : \mathsf{i} \vdash \mathsf{il}(x) : \mathsf{l}$. These include each "higher" mode into the immediately "lower" ones, and the lowest ones into l. We add an equation $\mathsf{ji}(\mathsf{mj}(x)) \equiv \mathsf{ki}(\mathsf{mk}(x))$ that the diamond commutes. Then $!_b A$ is the comonad $\mathsf{F}_{b\mathsf{l}(x)}(x : \mathsf{U}_{x.b\mathsf{l}(x)}(\cdot \mid A))$ for the unique $x : b \vdash b\mathsf{l} : \mathsf{l}$ generated by these constants. For example, $!_k$ is the comonad of $x : \mathsf{k} \vdash \mathsf{il}(\mathsf{ki}(x)) : \mathsf{l}$.

This mode theory is constructed so that every mode has a unique map to l. When $a \leq b$, we have a morphism $x : b \vdash ab(x) : a$, so the morphism $x : b \vdash b\mathsf{l}(x) : \mathsf{l}$ is equal to $x : b \vdash a\mathsf{l}(ba(x)) : \mathsf{l}$. Thus, by Lemma 4.9, we have

$$!_b A = \mathsf{F}_{b\mathsf{l}}(\mathsf{U}_{b\mathsf{l}} A) \cong \mathsf{F}_{a\mathsf{l}} \mathsf{F}_{ba} \mathsf{U}_{ba} \mathsf{U}_{a\mathsf{l}} A$$

The map $!_b A \vdash !_a A$ can thus be defined as the counit $\mathsf{F}_{ba}\mathsf{U}_{ba} A \vdash A$ for the comonad in the middle.

We add equations $ba(x \otimes_b y) \equiv ba(x) \otimes_a ba(y)$ and $ba(1_b) \equiv 1_a$ making each generator strictly monoidal. This ensures that each $!_b$ is monoidal and that $!_b A$ can be weakened or contracted if $(\otimes_b, 1_b)$ has weakening or contraction (and more generally that $\mathsf{F}_{ba}(B)$ can be weakened or contracted for any B, not just $\mathsf{U}_{ba}(A)$). Thus, we add weakening or contraction to a particular subexponential a by adding them to $(\otimes_a, 1_a)$.

When $a \leq b$, it does not seem that we need a condition that $(\otimes_b, 1_b)$ has whatever structural properties $(\otimes_a, 1_a)$ has in order to get that $!_b A$ is at least as structural as $!_a A$. As argued above $!_b A$ factors into the form $\mathsf{F}_{a\mathsf{l}}(C)$, which has whatever structural properties mode a has.

An interesting extension of this example would be to encode distributive laws between these modalities, following Jacobs [1994].

3.11 Monads

Consider a $\Diamond A$ modality with rules in the style of Pfenning and Davies [2001]:

$$\frac{\Gamma \vdash A\ \mathsf{true}}{\Gamma \vdash A\ \mathsf{poss}} \qquad \frac{\Gamma \vdash A\ \mathsf{poss}}{\Gamma \vdash \Diamond A\ \mathsf{true}} \qquad \frac{A\ \mathsf{true} \vdash C\ \mathsf{poss}}{\Gamma, \Diamond A\ \mathsf{true} \vdash C\ \mathsf{poss}}$$

We can model this using a mode theory with two modes t and p and context descriptor $x : \mathsf{t} \vdash \mathsf{g}(x) : \mathsf{p}$, defining the type $\Diamond A := \mathsf{U}_{c.\mathsf{g}(c)}(\cdot \mid \mathsf{F}_{\mathsf{g}(x)}(x : A))$. This is always a monad, but it does not automatically have a tensorial strength, which corresponds to the context-clearing in the left rule.

For example, if we have a monoid $(\otimes_\mathsf{t}, 1_\mathsf{t})$ on mode t and try to derive

$$\cfrac{\cfrac{\mathsf{g}(x \otimes_\mathsf{t} y) \Rightarrow \beta'[\mathsf{g}(y)/z] \quad x : A, y : \Diamond_\mathsf{g} B, z : \mathsf{F}_\mathsf{g}(B) \vdash_{\beta'} \mathsf{F}_\mathsf{g}(A \otimes_\mathsf{t} B)}{x : A, y : \Diamond_\mathsf{g} B \vdash_{\mathsf{g}(x \otimes_\mathsf{t} y)} \mathsf{F}_\mathsf{g}(A \otimes_\mathsf{t} B)}\ \mathsf{UL}}{x : A, y : \Diamond_\mathsf{g} B \vdash_{x \otimes_\mathsf{t} y} \Diamond_\mathsf{g} (A \otimes_\mathsf{t} B)}\ \mathsf{UR}$$

we are stuck, because there is no way to rewrite $\mathsf{g}(x \otimes_\mathsf{t} y)$ as a term containing $\mathsf{g}(y)$. If $(\otimes_t, 1_t)$ is affine, then we can weaken away x and take $\beta' = z$—the context-clearing in the left rule—but then in the right-hand premise we will only have access to z, not x, so we cannot complete the derivation.

In general, we translate all types at mode t, representing $\Diamond A$ as above. We translate $A_1\ \mathsf{true}, \ldots, A_1\ \mathsf{true} \vdash C\ \mathsf{true}$ by our sequent $x_1 : A_1^*, \ldots, x_1 : A_n^* \vdash_{x_1 \otimes \ldots \otimes x_n} C^*$, and the sequent $A_1\ \mathsf{true}, \ldots, A_b\ \mathsf{true} \vdash C\ \mathsf{poss}$ by the p-conclusioned sequent $x_1 : A_1^*, \ldots, x_1 : A_n^* \vdash_{\mathsf{g}(x_1 \otimes \ldots \otimes x_n)} \mathsf{F}_\mathsf{g}(C^*)$. Then the three "native" rules above are FR, UR, and a composite of UL followed by FL, respectively.

Some monads, such as the $\bigcirc A$ of [Pfenning and Davies, 2001] and those used to encapsulate effects in functional programming, do have a strength. One way to axiomatize the strength is via an asymmetric product of a t- and p-mode context:

$$\begin{array}{ll} x:\mathsf{t}, y:\mathsf{p} \vdash x \otimes_{\mathsf{tp}} y : \mathsf{p} & \mathsf{g}(x \otimes_{\mathsf{t}} y) \equiv x \otimes_{\mathsf{tp}} \mathsf{g}(y) \\ (x \otimes_{\mathsf{t}} y) \otimes_{\mathsf{tp}} z \equiv x \otimes_{\mathsf{tp}} (y \otimes_{\mathsf{tp}} z) & 1 \otimes_{\mathsf{tp}} y \equiv y \end{array}$$

The equations make this into a monoid action of the t-contexts on the p-contexts, and allow for "isolating" any one x_i in $\mathsf{g}(x_1 \otimes_{\mathsf{t}} \ldots \otimes_{\mathsf{t}} x_n)$ as the designated variable under a g. Using this (and switching notation from $\Diamond_{\mathsf{g}} A$ to $\bigcirc_{\mathsf{g}} A$), we can prove

$$\dfrac{\mathsf{g}(x \otimes_{\mathsf{t}} y) \Rightarrow (x \otimes_{\mathsf{tp}} z)[\mathsf{g}(y)/z] \qquad \dfrac{\dfrac{x \otimes_{\mathsf{tp}} \mathsf{g}(z') \Rightarrow \mathsf{g}(x \otimes_{\mathsf{t}} z') \qquad \dfrac{}{\ldots \vdash_{x \otimes_{\mathsf{t}} z'} A \otimes_{\mathsf{t}} B}\,\mathsf{FR}}{x : A, y : \bigcirc_{\mathsf{g}} B, z' : B \vdash_{x \otimes_{\mathsf{tp}} \mathsf{g}(z')} \mathsf{F}_{\mathsf{g}}(A \otimes_{\mathsf{t}} B)}\,\mathsf{UL}}{x : A, y : \bigcirc_{\mathsf{g}} B, z : \mathsf{F}_{\mathsf{g}}(B) \vdash_{x \otimes_{\mathsf{tp}} z} \mathsf{F}_{\mathsf{g}}(A \otimes_{\mathsf{t}} B)}\,\mathsf{FL}}{\dfrac{x : A, y : \bigcirc_{\mathsf{g}} B \vdash_{\mathsf{g}(x \otimes_{\mathsf{t}} y)} \mathsf{F}_{\mathsf{g}}(A \otimes_{\mathsf{t}} B)}{x : A, y : \bigcirc_{\mathsf{g}} B \vdash_{x \otimes_{\mathsf{t}} y} \bigcirc_{\mathsf{g}} (A \otimes_{\mathsf{t}} B)}\,\mathsf{UR}}\,\mathsf{UL}$$

An analogous description can be given for the "$\Box$-strong $\Diamond$" [Alechina et al., 2001, Pfenning and Davies, 2001], which has a strength only for boxed formulas ($\Box A \otimes \Diamond B \vdash \Diamond (\Box A \otimes B)$). We use 3 modes v,t,p and represent the $\Box$ as the comonad of a context descriptor $x : \mathsf{v} \vdash \mathsf{f}(x) : \mathsf{t}$ (with cartesian monoids on v and t and f laxly monoidal as above), and the $\Diamond$ as the monad of a $x : \mathsf{t} \vdash \mathsf{g}(x) : \mathsf{p}$. We have a mixed-mode product between v and p

$$\begin{array}{ll} x:\mathsf{v}, y:\mathsf{p} \vdash x \otimes_{\mathsf{vp}} y : \mathsf{p} & \mathsf{g}(\mathsf{f}(x) \times_{\mathsf{t}} y) \equiv x \otimes_{\mathsf{vp}} \mathsf{g}(y) \\ (x \times_{\mathsf{v}} y) \otimes_{\mathsf{tp}} z \equiv x \otimes_{\mathsf{tp}} y \otimes_{\mathsf{vp}} z & 1 \otimes_{\mathsf{vp}} y \equiv y \end{array}$$

We represent the truth-conclusioned sequent as in Example 3.9, and $x_1 : A_1 \,\mathsf{valid}, \ldots ; y_1 : B_1 \,\mathsf{true}, \ldots \vdash C \,\mathsf{poss}$ by

$$x_1 : \mathsf{U}_{\mathsf{f}}(A_1), \ldots, y_1 : B_1, \ldots \vdash_{\mathsf{g}(\mathsf{f}(x_1) \times_{\mathsf{t}} \mathsf{f}(x_2) \times_{\mathsf{t}} \ldots y_1 \times_{\mathsf{t}} \ldots)} \mathsf{F}_{\mathsf{g}}(C)$$

The left rule

$$\dfrac{\Delta; w' : A \;\mathsf{true} \vdash C \;\mathsf{poss}}{\Delta; \Gamma, z : \Diamond A \;\mathsf{true} \vdash C \;\mathsf{poss}}$$

that keeps the valid assumptions and discards the true ones is derivable by

$$\dfrac{\mathsf{g}(\mathsf{f}(x_i) \times y_i \times z) \Rightarrow (x_1 \times_v \ldots x_n) \otimes_{\mathsf{vp}} \mathsf{g}(z) \qquad \dfrac{\ldots, w' : A \vdash_{((x_1 \times_v \ldots \times_v x_n) \otimes_{\mathsf{vp}} \mathsf{g}(w'))} \mathsf{F}_{\mathsf{f}}(C)}{\ldots, w : \mathsf{F}_{\mathsf{g}}(A) \vdash_{((x_1 \times_v \ldots_v \times x_n) \otimes_{\mathsf{vp}} w)} \mathsf{F}_{\mathsf{f}}(C)}}{x_i : \mathsf{U}_{\mathsf{f}}(A_i), y_i : B_i, z : \Diamond_{\mathsf{g}} A \vdash_{\mathsf{g}(\mathsf{f}(x_i) \times y_i \times z)} \mathsf{F}_{\mathsf{f}}(C)}$$

The transformation is given by weakening away y_i and using the monoidalness of f and the isolation equation:

$$\begin{array}{ll} & \mathsf{g}(\mathsf{f}(x_1) \times \ldots \times \mathsf{f}(x_n) \times y_1 \times \ldots \times z) \\ \Rightarrow & \mathsf{g}(\mathsf{f}(x_1) \times \ldots \times \mathsf{f}(x_n) \times z) \\ \equiv & \mathsf{g}(\mathsf{f}(x_1 \times_v x_n) \times z) \\ \equiv & (x_1 \times_v x_n) \otimes_{\mathsf{vp}} z \end{array}$$

The right-hand premise is the encoding of the premise of the rule, using the isolation equation and monoidalness of f in the other direction. The restriction of the isolation equation to f prevents keeping any additional `true` variables in the premise.

3.12 Spatial Type Theory

The spatial type theory for cohesion [Shulman, 2015] (which motivated this work) has an adjoint pair $\flat \dashv \sharp$, where $\flat$ is a comonad and $\sharp$ is a monad, with some additional properties. In the one-variable case [Licata and Shulman, 2016], we analyzed this as arising from an idempotent comonad[3] in the mode theory: we have a mode c with a cartesian monoid $(\times, \top)$ and a context descriptor $x : \mathsf{c} \vdash \mathsf{r}(x) : \mathsf{c}$ such that $\mathsf{r}(\mathsf{r}(x)) \equiv \mathsf{r}(x)$ and there is a directed transformation $\mathsf{r}(x) \Rightarrow x$. Then we define $\flat A := \mathsf{F}_{\mathsf{r}}(A)$ and $\sharp A := \mathsf{U}_{\mathsf{r}}(A)$. These are adjoint as discussed in Example 3.8, and the transformation gives the counit $\mathsf{F}_{\mathsf{r}}(A) \vdash A$ and the unit $A \vdash \mathsf{U}_{\mathsf{r}}(A)$ by Lemma 4.10. Now that we have a multi-assumptioned logic, we can model the fact that $\flat A$ preserves products by the equational axiom $\mathsf{r}(x \times y) \equiv \mathsf{r}(x) \times \mathsf{r}(y)$. Overall, we encode a simply-typed spatial type theory judgement $x_1 : A_1\ \mathsf{crisp}, \ldots ; y_1 : B_1\ \mathsf{coh} \vdash C\ \mathsf{coh}$ as $x_1 : A_1, \ldots, y_1 : B_1, \ldots \vdash_{\mathsf{r}(x_1) \times \ldots \times y_1 \times \ldots} C$.

As a sequent calculus, the rules from [Shulman, 2015] are

$$\frac{A \in \Delta \quad \Delta;\Gamma,A \vdash C}{\Delta;\Gamma \vdash C} \quad \frac{\Delta;\cdot \vdash A}{\Delta;\Gamma \vdash \flat A} \quad \frac{\Delta,A;\Gamma \vdash C}{\Delta;\Gamma,\flat A \vdash C} \quad \frac{\Delta,\Gamma;\cdot \vdash C}{\Delta;\Gamma \vdash \sharp C} \quad \frac{\sharp A \in \Delta \quad \Delta;\Gamma,A \vdash C}{\Delta;\Gamma \vdash C}$$

In order, these correspond to (1) the action of the contraction and $\mathsf{r}(x) \Rightarrow x$ transformations; (2) FR with weakening, using monoidalness of r in one direction; (3) FL; (4) UR, using monoidalness of r in the other direction and idempotence; (5) UL, with contraction. This provides a satisfying explanation for the unusual features of these rules, such as promoting all cohesive variables to crisp in $\sharp$-right, and eliminating a crisp $\sharp$ in $\sharp$-left.

4 Syntactic Properties

4.1 Admissible Structural Rules

We show that identity, cut, weakening, exchange, contraction, and respect for transformations, are admissible. We give the cases for the rules in Figure 2, though the results readily extend to additive sums and products.

Define the *size* of a derivation of $\Gamma \vdash_\alpha A$ or $\Gamma \vdash_\gamma \Delta$ to be the number of inference rules for these judgements ($\mathsf{v}, \mathsf{FL}, \mathsf{FR}, \mathsf{UL}, \mathsf{UR}, \cdot, \_,\_$) used in it (i.e., the evidence that variables are in a context and the evidence for structural transformations do not contribute to the size). Sizes are necessary for the cut proof, where we sometimes weaken or invert a derivation before applying the inductive hypothesis.

LEMMA 4.1: RESPECT FOR TRANSFORMATIONS.

1. *If $\Gamma \vdash_\beta A$ and $\beta' \Rightarrow \beta$ then $\Gamma \vdash_{\beta'} A$, and the resulting derivation has the same size as the given one.*
2. *If $\Gamma \vdash_\gamma \Delta$ and $\gamma' \Rightarrow \gamma$ then $\Gamma \vdash_{\gamma'} \Delta$, and the resulting derivation has the same size as the given one.*

Proof. Mutual induction on the given derivation. The cases for v and FR and UL are immediate (with no use of the inductive hypothesis) by composing with the equality in the premise of the rule. This does not change the size of the derivation because the derivations of structural transformations are ignored by the size. The cases for FL and UR use the inductive hypothesis, along with congruence for structural transformations to show that $\beta[\alpha/x] \Rightarrow \beta'[\alpha/x]$ or $\alpha[\beta/x] \Rightarrow \alpha[\beta'/x]$. The cases for substitutions rely on the fact that no generating structural transformations for mode substitutions are allowed, so if $\gamma' \Rightarrow \cdot$ then γ' is literally $\cdot$, and $(-,-)$ is injective (if $\gamma' \Rightarrow (\gamma_1, \alpha_2/x)$, then γ' is $(\gamma_1', \alpha_2'/x)$ with $\gamma_1' \Rightarrow \gamma_1$ and $\alpha_2' \Rightarrow \alpha_2$); this is enough to use the inductive hypotheses in the cons case. ☐

LEMMA 4.2: WEAKENING OVER WEAKENING.

1. *If $\Gamma, \Gamma' \vdash_\alpha C$ then $\Gamma, z{:}A, \Gamma' \vdash_\alpha C$, and the resulting derivation has the same size as the given one.*
2. *If $\Gamma, \Gamma' \vdash_\gamma \Delta$ then $\Gamma, z{:}A, \Gamma' \vdash_\gamma \Delta$, and the resulting derivation has the same size as the given one.*
3. *If $\Gamma, \Gamma'' \vdash_\alpha C$ then $\Gamma, \Gamma', \Gamma'' \vdash_\alpha C$, and the resulting derivation has the same size as the given one.*

Proof. It is implicit that the mode morphism α is weakened with z in the conclusion. Intuitively, weakening holds because the contexts Γ are treated like ordinary structural contexts in all of the rules—they are fully general in every conclusion, and the premises check membership or extend them—and because weakening holds for mode morphisms and equalities of mode morphisms. Formally, the first two parts are proved by mutual induction; each case is either immediate or follows from weakening for the mode morphisms, weakening for transformations, and the inductive hypotheses. The third part is proved by induction over Γ', repeatedly applying the first part. ☐

[3] There it was an idempotent monad; the variance of F and U has been flipped in paper.

LEMMA 4.3: EXCHANGE OVER EXCHANGE. *If* $\Gamma, x : A, y : B, \Gamma' \vdash_\alpha C$ *then* $\Gamma, y : B, x : A, \Gamma' \vdash_\alpha C$*, and the resulting derivation has the same size as the given one. (And similarly for substitutions, and exchange can be iterated).*

Proof. Analogous to weakening. ☐

We sometimes write $\hat{\Gamma}$ for the ψ such that $\Gamma\ \mathsf{ctx}_\psi$ and similarly for $\hat{A}$.

THEOREM 4.4: IDENTITY.

1. *If* $x : A \in \Gamma$ *then* $\Gamma \vdash_x A$.
2. *If* $\hat{\Gamma} \vdash \rho : \hat{\Delta}$ *is a variable-for-variable mode substitution such that* $x : A \in \Delta$ *implies* $\rho(x) : A \in \Gamma$*, then* $\Gamma \vdash_\rho \Delta$.

Proof. The standard proof by induction on A (mutually with Δ) applies: the case for atomic propositions is a rule, and for the other connectives, apply the invertible and then non-invertible rule to reduce the problem to the inductive hypotheses. More specifically, identity for P is a rule. In the case for $\mathsf{F}_\alpha(\Delta)$, with $\Gamma = \Gamma_1, x : \mathsf{F}_\alpha(\Delta), \Gamma_2$, we reduce it to the inductive hypothesis as follows:

$$\dfrac{\dfrac{\alpha \Rightarrow \alpha[\vec{x}/x] \quad \Gamma_1, \Gamma_2, \Delta \vdash_{\vec{x}/x} \Delta}{\Gamma_1, \Gamma_2, \Delta \vdash_\alpha \mathsf{F}_\alpha(\Delta)}\ \mathsf{FR}}{\Gamma_1, x : \mathsf{F}_\alpha(\Delta), \Gamma_2 \vdash_x \mathsf{F}_\alpha(\Delta)}\ \mathsf{FL}$$

In the second premise, the $\vec{x}/x$ substitution for each $x \in \Delta$ is a variable-for-variable substitution, so the second part of the inductive hypothesis applies. The case for U is similar

$$\dfrac{\alpha \Rightarrow x[\alpha[\vec{x}/x]/x] \quad \Gamma, \Delta \vdash_{\vec{x}/x} \Delta \quad \Gamma, x : A \vdash_x A}{\dfrac{\Gamma, \Delta \vdash_\alpha A}{\Gamma \vdash_x \mathsf{U}_\alpha(\Delta \mid A)}\ \mathsf{UR}}\ \mathsf{UL}$$

For the second part, the hypothesis of the lemma asks that every variable in Δ is associated by ρ with a variable of the same type in Γ; this is enough to iterate the first part of the lemma for each position in Δ. Specifically, the case where Δ is the empty context $\cdot$ is a rule. In the case for a cons $\Delta, y : A$, we have $\hat{\Gamma} \vdash \rho : (\hat{\Delta}, y : \hat{A})$ which means ρ must be of the form $\rho', x/y$ where $x \in \hat{\Gamma}$ and ρ' is a variable-for-variable substitution. Because ρ was type-preserving, $x : A \in \Gamma$ and ρ' is type-preserving, so we obtain the result from the inductive hypotheses as follows:

$$\dfrac{\Gamma \vdash_\rho \Delta \quad \Gamma \vdash_x A}{\Gamma \vdash_{\rho, x/y} \Delta, y : A}$$

☐

LEMMA 4.5: LEFT-INVERTIBILITY OF F. *If* $d :: \Gamma_1, x_0 : \mathsf{F}_{\alpha_0}(\Delta_0), \Gamma_2 \vdash_\beta C$ *then there is a derivation* $d' :: \Gamma_1, \Gamma_2, \Delta_0 \vdash_{\beta[\alpha_0/x_0]} C$ *and size*$(d') \leq$ *size*(d) *(and analogously for substitutions).*

Proof. Intuitively, we find all of the places where d "splits" x_0, delete the FL used to do the split, and reroute the variables to the ones in the context of the result.

Formally, we proceed by induction on d. We write Γ for the whole context $\Gamma_1, x_0 : \mathsf{F}_{\alpha_0}(\Delta_0), \Gamma_2$.

In the case for v, $x : P \in \Gamma_1, x_0 : \mathsf{F}_{\alpha_0}(\Delta_0), \Gamma_2$ cannot be equal to $x_0 : \mathsf{F}_{\alpha_0}(\Delta_0)$ because the types conflict, so we can reapply the v rule in $\Gamma_1, \Gamma_2, \Delta$.

In the case for FR, we have

$$\dfrac{\beta \Rightarrow \alpha[\gamma] \quad \Gamma \vdash_\gamma \Delta}{\Gamma \vdash_\beta \mathsf{F}_\alpha(\Delta)}$$

with $x_0 : \mathsf{F}_{\alpha_0}(\Delta_0) \in \Gamma$. By the inductive hypothesis we get $\Gamma_1, \Gamma_2, \Delta_0 \vdash_{\gamma[\alpha_0/x]} \Delta$. Because x_0 is not free in α, $(\alpha[\gamma])[\alpha_0/x_0] = \alpha[\gamma[\alpha_0/x_0]]$, so we can reapply FR:

$$\dfrac{\beta[\alpha_0/x_0] \Rightarrow \alpha[\gamma[\alpha_0/x_0]] \quad \Gamma_1, \Gamma_2, \Delta_0 \vdash_{\gamma[\alpha_0/x]} \Delta}{\Gamma_1, \Gamma_2 \vdash_{\beta[\alpha_0/x_0]} \mathsf{F}_\alpha(\Delta)}$$

Both the input and the output have size 1 more than the size of their subderivations, and the output subderivation is no bigger than the input by the inductive hypothesis.

In the case for FL

$$\frac{\begin{array}{c} d \\ \Gamma_1', \Gamma_2', \Delta \vdash_{\beta[\alpha/x]} C \end{array}}{\Gamma_1', x : \mathsf{F}_\alpha(\Delta), \Gamma_2' \vdash_\beta C}\ \mathsf{FL}$$

with $\Gamma_1, x_0 : \mathsf{F}_{\alpha_0}(\Delta_0), \Gamma_2 = \Gamma_1', x : \mathsf{F}_\alpha(\Delta), \Gamma_2'$, we distinguish cases on whether $x = x_0$ or not. If they are the same (i.e. we have hit a left rule on x_0), then $\alpha_0 = \alpha$ and $\Delta_0 = \Delta$ and d is the result, and the size is 1 less than the size of the input. If they are different, then (because x_0 is somewhere in Γ_1', Γ_2') by the inductive hypothesis we have a derivation

$$d' :: (\Gamma_1', \Gamma_2') - x_0, \Delta, \Delta_0 \vdash_{\beta[\alpha/x][\alpha_0/x_0]} C$$

that is no bigger than d. Because x_0 is from Γ and not Δ, it does not occur in α, so

$$\beta[\alpha/x][\alpha_0/x_0] = \beta[\alpha_0/x_0][\alpha/x]$$

By (iterating) exchange, we get a derivation

$$d'' :: (\Gamma_1', \Gamma_2') - x_0, \Delta_0, \Delta \vdash_{\beta[\alpha_0/x_0][\alpha/x]} C$$

whose size is the same as d' and so no bigger than d. Applying FL to d'' (using the fact that $(\Gamma_1', x : \mathsf{F}_\alpha(\Delta), \Gamma_2') - x_0 = \Gamma_1, \Gamma_2$) derives $\Gamma_1, \Gamma_2 \vdash_{\beta[\alpha_0/x_0]} C$, and the size is no bigger than the size of the input.

In the case for UR,

$$\frac{\Gamma, \Delta \vdash_{\alpha[\beta/x]} A}{\Gamma \vdash_\beta \mathsf{U}_{x.\alpha}(\Delta \mid A)}$$

the inductive hypothesis gives a $d' :: \Gamma_1, \Gamma_2, \Delta, \Delta_0 \vdash_{\alpha[\beta/x][\alpha_0/x_0]} A$ and (iterated) exchange gives $d'' :: \Gamma_1, \Gamma_2, \Delta_0, \Delta \vdash_{\alpha[\beta/x][\alpha_0/x_0]} A$, both no bigger than d. Because x_0 is in Γ and not Δ, it is not free in α, so

$$\alpha[\beta/x][\alpha_0/x_0] = \alpha[\beta[\alpha_0/x_0]/x]$$

Thus, we can derive

$$\frac{\begin{array}{c} d'' \\ \Gamma_1, \Gamma_2, \Delta_0, \Delta \vdash_{\alpha[\beta[\alpha_0/x_0]/x]} A \end{array}}{\Gamma_1, \Gamma_2, \Delta_0 \vdash_{\beta[\alpha_0/x_0]} \mathsf{U}_{x.\alpha}(\Delta \mid A)}$$

In the case for UL,

$$\frac{x : \mathsf{U}_{x.\alpha}(\Delta \mid A) \in \Gamma \quad \beta \Rightarrow \beta'[\alpha[\gamma]/z] \quad \Gamma \vdash_\gamma \Delta \quad \Gamma, z{:}A \vdash_{\beta'} C}{\Gamma \vdash_\beta C}$$

we know that x is different than x_0 because the types conflict. The inductive hypotheses give no-bigger derivations of

$$\Gamma_1, \Gamma_2 \Delta_0 \vdash_{\gamma[\alpha_0/x_0]} \Delta \qquad \Gamma_1, \Gamma_2, z{:}A, \Delta_0 \vdash_{\beta'[\alpha_0/x_0]} C$$

and the latter can be exchanged to

$$\Gamma_1, \Gamma_2, \Delta_0, z{:}A \vdash_{\beta'[\alpha_0/x_0]} C$$

again without increasing the size. Thus, we can produce

$$\frac{\begin{array}{l} x : \mathsf{U}_{x.\alpha}(\Delta \mid A) \in \Gamma_1, \Gamma_2, \Delta_0 \\ \beta[\alpha_0/x] \Rightarrow \beta'[\alpha_0/x_0][\alpha[\gamma[\alpha_0/x_0]]/z] \\ \Gamma_1, \Gamma_2, \Delta_0 \vdash_{\gamma[\alpha_0/x_0]} \Delta \\ \Gamma_1, \Gamma_2, \Delta_0, z{:}A \vdash_{\beta'[\alpha_0/x_0]} C \end{array}}{\Gamma_1, \Gamma_2, \Delta_0 \vdash_{\beta[\alpha_0/x]} C}$$

where the transformation is the composition of the $-[\alpha_0/x_0]$ substitution into the given transformation, and rearranging the substitution (note that x_0 does not occur in α):

$$\begin{aligned}\beta[\alpha_0/x_0] &\Rightarrow \beta'[\alpha[\gamma]/z][\alpha_0/x_0] = \beta'[\alpha_0/x_0][\alpha[\gamma][\alpha_0/x_0]/z] \\ &= \beta'[\alpha_0/x_0][\alpha[\gamma[\alpha_0/x_0]]/z]\end{aligned}$$

The case for $\cdot$ is immediate. The case for $\_,\_$ follows from the two inductive hypotheses, because $(\gamma,\alpha/x)[\alpha_0/x_0] = (\gamma[\alpha_0/x_0],\alpha[\alpha_0/x_0]/x)$. ☐

THEOREM 4.6: CUT.

1. *If* $\Gamma,\Gamma' \vdash_{\alpha_0} A_0$ *and* $\Gamma, x_0 : A_0, \Gamma' \vdash_\beta B$ *then* $\Gamma,\Gamma' \vdash_{\beta[\alpha_0/x_0]} B$
2. *If* $\Gamma,\Gamma' \vdash_{\alpha_0} A_0$ *and* $\Gamma, x_0 : A_0, \Gamma' \vdash_\gamma \Delta$ *then* $\Gamma,\Gamma' \vdash_{\gamma[\alpha_0/x_0]} \Delta$
3. *If* $\Gamma \vdash_\gamma \Delta$ *and* $\Gamma,\Delta \vdash_\beta C$ *then* $\Gamma \vdash_{\beta[\gamma]} C$.

Proof. We write d for the derivation of A_0 and e for the derivation from A_0. The induction ordering is the usual one: First the cut formula, and then the sizes of size of the two derivations. More specifically, any part can call another with a smaller cut formula (A_0 for part 1 and part 2, Δ for part 3). Additionally, part 1 and part 2 call themselves and each other with the same cut formula and smaller d or e.

For part 1, there are 5 rules in the sequent calculus, so 25 pairs of final rules. We use the following case analysis on d/e to cover them all:

1. any rule and v
2. FR and FL^{x_0} (principal)
3. UR and UR^{x_0} (principal)
4. any rule and FR (right-commutative)
5. any rule and UR (right-commutative)
6. any rule and FL$^{x \neq x_0}$ (right-commutative)
7. any rule and UL$^{x \neq x_0}$ (right-commutative)
8. FL and any rule (left-commutative)
9. UL and any rule (left-commutative)

To see that this is exhaustive, cases 1 and 4-7 cover all pairs except when e is a left rule on the cut variable x_0. In these cases, d must be either a left rule or a right rule for the cut formula (the right rules for other types and identity do not have the appropriate conclusion formula). If d is a right rule, then it is a principal cut; if it is a left rule, then the left-commutative cases apply.

The left- and right-commutative cases overlap when d is a left-rule and e is not a left rule on the cut variable. We resolve this arbitrarily, prioritizing right-commutative over left-commutative.

1. Any rule and variable

$$\overset{d}{\Gamma,\Gamma' \vdash_{\alpha_0} A_0} \qquad \frac{z : Q \in (\Gamma, x_0 : A_0, \Gamma') \quad \beta \overset{s}{\Rightarrow} z}{\Gamma, x : A, \Gamma' \vdash_\beta Q}$$

There two subcases, depending on whether the cut variable is z or not. If z is x_0 and A_0 is Q, then d derives $\Gamma,\Gamma' \vdash_{\alpha_0} Q$ and we want a derivation of $\Gamma,\Gamma' \vdash_{\beta[\alpha_0/z]} Q$. By congruence on s, $\beta[\alpha_0/z] \Rightarrow z[\alpha_0/z]$, so Lemma 4.1 gives the result. If z is not x_0, then $z : Q \in \Gamma,\Gamma'$. We want a derivation of $\Gamma,\Gamma' \vdash_{\beta[\alpha_0/x_0]} Q$, and substituting into s gives $\beta[\alpha_0/x_0] \Rightarrow z$ (because $z \neq x_0$), so we can derive

$$\frac{z : Q \in (\Gamma,\Gamma') \quad \beta[\alpha_0/x_0] \Rightarrow z}{\Gamma,\Gamma' \vdash_{\beta[\alpha_0/x_0]} Q}$$

2. FR and FL (principal)

$$\frac{\overset{s}{\alpha_0 \Rightarrow \alpha[\gamma]} \quad \overset{d}{\Gamma,\Gamma' \vdash_\gamma \Delta}}{\Gamma,\Gamma' \vdash_{\alpha_0} \mathsf{F}_\alpha(\Delta)} \qquad \frac{\overset{e}{\Gamma,\Gamma',\Delta \vdash_{\beta[\alpha/x_0]} C}}{\Gamma, x_0 : \mathsf{F}_\alpha(\Delta),\Gamma' \vdash_\beta C}$$

Using the inductive hypothesis part 3 to cut d and e (Δ is a subformula of the original cut formula $\mathsf{F}_\alpha(\Delta)$) gives

$$\Gamma,\Gamma' \vdash_{\beta[\alpha/x_0][\gamma]} C$$

By congruence on s and because γ substitutes only for variables in $\hat{\Delta}$,

$$\beta[\alpha_0/x_0] \Rightarrow \beta[\alpha[\gamma]/x_0] = \beta[\alpha/x_0][\gamma]$$

So applying Lemma 4.1 gives $\Gamma,\Gamma' \vdash_{\beta[\alpha_0/x_0]} C$.

3. UR and UL (principal).

   We have

$$d \quad = \quad \frac{\overset{d'}{\Gamma,\Gamma',\Delta \vdash_{\alpha[\alpha_0/x_0]} A}}{\Gamma,\Gamma' \vdash_{\alpha_0} \mathsf{U}_{x_0.\alpha}(\Delta \mid A)}$$

$$\frac{\begin{array}{l} s :: \beta \Rightarrow \beta'[\alpha[\gamma]/z] \\ e_1 :: \Gamma, x_0 : \mathsf{U}_{x_0.\alpha}(\Delta \mid A),\Gamma' \vdash_\gamma \Delta \\ e_2 :: \Gamma, x_0 : \mathsf{U}_{x_0.\alpha}(\Delta \mid A),\Gamma', z{:}A \vdash_{\beta'} C \end{array}}{\Gamma, x_0 : \mathsf{U}_{x_0.\alpha}(\Delta \mid A),\Gamma' \vdash_\beta C}$$

First, cutting the original d and the smaller e_1 and e_2 gives

$$\overset{e_1'}{\Gamma,\Gamma' \vdash_{\gamma[\alpha_0/x_0]} \Delta} \qquad \overset{e_2'}{\Gamma,\Gamma', z{:}A \vdash_{\beta'[\alpha_0/x_0]} C}$$

Cutting e_1' *into* d' (the derivations have switched places, so are not necessarily smaller, but the cut formula Δ is a subformula of $\mathsf{U}_{x_0.\alpha}(\Delta \mid A)$) gives

$$\overset{d_1'}{\Gamma,\Gamma' \vdash_{\alpha[\gamma[\alpha_0/x]]} A}$$

Cutting d_1' into e_2' gives

$$\Gamma,\Gamma' \vdash_{\beta'[\alpha_0/x_0][\alpha[\alpha_0/x_0]/z]} A$$

But by using s and commuting substitutions we have

$$\beta[\alpha_0/x_0] \Rightarrow (\beta'[\alpha[\gamma]/z])[\alpha_0/x_0] = \beta'[\alpha_0/x_0][\alpha[\gamma[\alpha_0/x_0]]/z]$$

so Lemma 4.1 gives the result.

4. Any rule and FR (right-commutative)

$$\overset{d}{\Gamma,\Gamma' \vdash_{\alpha_0} A_0} \qquad \frac{\beta \Rightarrow \alpha[\gamma] \quad \overset{e}{\Gamma, x_0 : A_0,\Gamma' \vdash_\gamma \Delta}}{\Gamma, x_0 : A_0, \Gamma' \vdash_\beta \mathsf{F}_\alpha(\Delta)}$$

By the inductive hypothesis, cutting into d into e gives $\Gamma,\Gamma' \vdash_{\gamma[\alpha_0/x_0]} \Delta$. By congruence, $\beta[\alpha_0/x_0] \Rightarrow \alpha[\gamma][\alpha_0/x_0]$. Since γ is a total substitution for all variables in $\hat{\Delta}$, $\alpha[\gamma][\alpha_0/x_0] = \alpha[\gamma[\alpha_0/x_0]]$, so $\beta[\alpha_0/x_0] \Rightarrow \alpha[\gamma[\alpha_0/x_0]]$. Thus we can reapply the FR rule to get $\Gamma,\Gamma' \vdash_{\beta[\alpha_0/x_0]} \mathsf{F}_\alpha(\Delta)$.

5. Any rule and UR (right-commutative).

$$\overset{d}{\Gamma,\Gamma' \vdash_{\alpha_0} A_0} \qquad \frac{\overset{e}{\Gamma, x_0 : A_0, \Gamma', \Delta \vdash_{\alpha[\beta/x]} A}}{\Gamma, x_0 : A_0, \Gamma' \vdash_{\beta} \mathsf{U}_{x.\alpha}(\Delta \mid A)}$$

The inductive cut of d into e gives

$$\Gamma,\Gamma',\Delta \vdash_{\alpha[\beta/x][\alpha_0/x_0]} A$$

Because the variables from $\hat{\Gamma},\hat{\Gamma}'$ occur only in β, not in α, this substitution equals $\alpha[\beta[\alpha_0/x_0]/x]$ so reapplying the UR rule derives $\Gamma,\Gamma' \vdash_{\beta[\alpha_0/x_0]} \mathsf{U}_{x.\alpha}(\Delta \mid A)$.

6. Any rule and $\mathsf{FL}^{x \neq x_0}$ (right commutative)

If the left rule is not on the cut variable, then we have

$$\overset{d}{\Gamma,\Gamma' \vdash_{\alpha_0} A_0} \qquad \frac{x : \mathsf{F}_{\alpha}(\Delta) \in \Gamma,\Gamma' \quad \overset{e}{((\Gamma, x_0 : A_0, \Gamma') - x), \Delta \vdash_{\beta[\alpha/x]} C}}{\Gamma, x_0 : A_0, \Gamma' \vdash_{\beta} C}$$

We are going to commute d under FL on x, so need to reroute uses of x to the bottom by the left-inversion lemma, which gives

$$d' :: ((\Gamma,\Gamma') - x), \Delta \vdash_{\alpha_0[\alpha/x]} A_0$$

and d' is no bigger than d.

Cutting d' and e by the inductive hypothesis gives

$$((\Gamma,\Gamma') - x), \Delta \vdash_{\beta[\alpha/x][\alpha_0[\alpha/x]/x_0]} C$$

Because x_0 is not free in α,

$$\beta[\alpha/x][\alpha_0[\alpha/x]/x_0] = \beta[\alpha_0/x_0][\alpha/x]$$

so we can apply FL

$$\frac{(\Gamma,\Gamma' - x) \vdash_{\beta[\alpha_0/x_0][\alpha/x]} C}{\Gamma,\Gamma' \vdash_{\beta[\alpha_0/x_0]} C}$$

7. Any rule and $\mathsf{UL}^{x \neq x_0}$ (right commutative)

$$\overset{d}{\Gamma,\Gamma' \vdash_{\alpha_0} A_0} \qquad \frac{\begin{array}{l} x : \mathsf{U}_{x.\alpha}(\Delta \mid A) \in \Gamma,\Gamma' \\ \beta \Rightarrow \beta'[\alpha[\gamma]/z] \\ \Gamma, x_0 : A_0, \Gamma' \vdash_{\gamma} \Delta \\ \Gamma, x_0 : A_0, \Gamma', z{:}A \vdash_{\beta'} C \end{array}}{\Gamma, x_0 : A_0, \Gamma' \vdash_{\beta} C}$$

By the inductive hypotheses we get

$$\Gamma,\Gamma' \vdash_{\gamma[\alpha_0/x_0]} \Delta \qquad \Gamma,\Gamma', z : A \vdash_{\beta'[\alpha_0/x_0]} C$$

so we can derive

$$\frac{\begin{array}{l} x : \mathsf{U}_{x.\alpha}(\Delta \mid A) \in \Gamma,\Gamma' \\ \beta[\alpha_0/x_0] \Rightarrow \beta'[\alpha_0/x_0][\alpha[\gamma[\alpha_0/x_0]]/z] \\ \Gamma,\Gamma' \vdash_{\gamma[\alpha_0/x_0]} \Delta \\ \Gamma,\Gamma', z{:}A \vdash_{\beta'[\alpha_0/x_0]} C \end{array}}{\Gamma, x_0 : A_0, \Gamma' \vdash_{\beta[\alpha_0/x_0]} C}$$

For the second premise, we get

$$\beta[\alpha_0/x_0] \Rightarrow \beta'[\alpha[\gamma]/z][\alpha_0/x_0]$$

by congruence on the assumed transformation, and then commute substitutions.

8. FL and any rule (left commutative).

   There is one subtlety in this case. The usual strategy for a left rule against an arbitrary e is to push e into the "continuation" of the left rule. However, as discussed above, our left rule for F eagerly inverts *all* occurrences of x, while e itself also has x in scope. Thus, we use Lemma 4.5 to pull the left-inversion to the bottom of e, and then push that into d. On proof terms, this corresponds to making all references to x in e instead refer to the results of the "split" at the bottom of d.

   Formally, we have

$$\frac{x : \mathsf{F}_\alpha(\Delta) \in \Gamma, \Gamma' \quad \overset{d}{((\Gamma, \Gamma') - x), \Delta \vdash_{\alpha_0[\alpha/x]} A_0}}{\Gamma, \Gamma' \vdash_{\alpha_0} A_0}$$

$$\overset{e}{\Gamma, x_0 : A_0, \Gamma' \vdash_\beta C}$$

   By left invertibility on e, we obtain (note that $x \neq x_0$ because x_0 only in scope in e, not d) a derivation e' of $(\Gamma, x : A_0, \Gamma') - x, \Delta \vdash_{\beta[\alpha/x]} C$ that is no bigger than e. Because the cut formula is the same, and e' is no bigger than e, and d is smaller than the given derivation of A_0, we can apply the inductive hypothesis to cut d and e' to get

$$(\Gamma, \Gamma') - x, \Delta \vdash_{\beta[\alpha/x][\alpha_0[\alpha/x]/x_0]} C.$$

   Commuting substitutions gives

$$\beta[\alpha/x][\alpha_0[\alpha/x]/x_0] = \beta[\alpha_0/x_0][\alpha/x]$$

   so we can reapply FL

$$\frac{((\Gamma, \Gamma') - x), \Delta \vdash_{(\beta[\alpha_0/x_0])[\alpha/x]} C}{\Gamma, \Gamma' \vdash_{\beta[\alpha_0/x_0]} C}$$

9. UL and any rule (left commutative) In this case, $x : \mathsf{U}_\alpha(\Delta \mid A) \in \Gamma, \Gamma'$ and we have

$$\frac{\overset{s}{\alpha_0 \Rightarrow \alpha_0'[\alpha[\gamma]/z]} \quad \overset{d_1}{\Gamma, \Gamma' \vdash_\gamma \Delta} \quad \overset{d_2}{\Gamma, \Gamma', z : A \vdash_{\alpha_0'} A_0}}{\Gamma, \Gamma' \vdash_{\alpha_0} A_0}$$

$$\overset{e}{\Gamma, x_0 : A_0, \Gamma' \vdash_\beta B}$$

   Weakening e with z and then cutting d_2 and e by the inductive hypothesis (which applies because d_2 is smaller and weakening does not change the size) gives

$$\overset{d_2'}{\Gamma, \Gamma', z : A \vdash_{\beta[\alpha_0'/x_0]} B}$$

   Thus, we have the first, third, and fourth premises of

$$\frac{\begin{array}{l} x : \mathsf{U}_\alpha(\Delta \mid A) \in \Gamma, \Gamma' \\ \beta[\alpha_0/x_0] \Rightarrow \beta[\alpha_0'/x_0][\alpha[\gamma]/z] \\ \Gamma, \Gamma' \vdash_\gamma \Delta \\ \Gamma, \Gamma', z : A \vdash_{\beta[\alpha_0'/x_0]} B \end{array}}{\Gamma, \Gamma' \vdash_{\beta[\alpha_0/x_0]} A_0}$$

   The transformation premise is

$$\beta[\alpha_0/x_0] \Rightarrow \beta[\alpha_0'[\alpha[\gamma]/z]/x0] = \beta[\alpha_0'/x_0][\alpha[\gamma]/z]$$

   where the first step is by congruence with β on s, and the second is by properties of substitution (z is not free in β).

For part 2, there are just two right-commutative cases: For

$$\Gamma,\Gamma' \vdash_{\alpha_0} A_0 \qquad \Gamma, x_0 : A_0, \Gamma' \vdash_{\cdot} \cdot$$

we also have $\cdot[\alpha_0/x_0] = \cdot$ and $\Gamma,\Gamma' \vdash_{\cdot} \cdot$. For

$$\Gamma,\Gamma' \vdash_{\alpha_0} A_0 \qquad \frac{\Gamma, x_0 : A_0, \Gamma' \vdash_{\gamma} \Delta \quad \Gamma, x_0 : A_0, \Gamma' \vdash_{\alpha} A}{\Gamma, x_0 : A_0, \Gamma' \vdash_{\gamma,\alpha/x} \Delta, x : A}$$

we have $(\gamma,\alpha/x)[\alpha_0/x_0] = (\gamma[\alpha_0/x_0], \alpha[\alpha_0/x_0])$, and

$$\Gamma,\Gamma' \vdash_{\gamma[\alpha_0/x_0]} \Delta \quad \Gamma,\Gamma' \vdash_{\alpha[\alpha_0/x_0]} A$$

by the inductive hypotheses, so we can reapply the rule to conclude $\Gamma,\Gamma' \vdash_{(\gamma,\alpha/x)[\alpha_0/x_0]} \Delta, x : A$.

For part 3, we induct on Δ, reducing a simultaneous cut to iterated single-variable cuts. If Δ is empty, then we have

$$\Gamma \vdash_{\cdot} \cdot \qquad \overset{e}{\Gamma, \cdot \vdash_{\beta} C}$$

and we return e, noting that $\beta[\cdot] = \beta$. Otherwise we have

$$\frac{\overset{d_1}{\Gamma \vdash_{\gamma} \Delta} \quad \overset{d_2}{\Gamma \vdash_{\alpha} A}}{\Gamma \vdash_{\gamma,\alpha/x} \Delta, x : A} \qquad \overset{e}{\Gamma, \Delta, x : A \vdash_{\beta} C}$$

Using the inductive hypothesis to cut d_2 into e (A is smaller than $\Delta, x : A$) gives

$$\overset{e'}{\Gamma, \Delta \vdash_{\beta[\alpha/x]} C}$$

Using the inductive hypothesis to cut d_1 into e' (Δ is smaller than $\Delta, x : A$) gives

$$\Gamma \vdash_{\beta[\alpha/x][\gamma]} C$$

Because γ substitutes for $\hat{\Delta}$ (and not $\hat{\Gamma}$, the free variables of α),

$$\beta[\gamma, \alpha/x] = \beta[\alpha/x][\gamma]$$

□

Using this, we have

COROLLARY 4.7: CONTRACTION OVER CONTRACTION.
If $\Gamma, x : A, y : A, \Gamma' \vdash_{\alpha} C$ *then* $\Gamma, z : A, \Gamma' \vdash_{\alpha[z/x,z/y]} C$

Proof. Contraction can be shown by cutting with a renaming substitution. The mode substitution $z/x, z/y$ is a variable-for-variable substitution, and is type-preserving between $x : A, y : A$ and $\Gamma, z : A, \Gamma'$. Therefore, by identity (part 2), $\Gamma, z : A, \Gamma' \vdash_{z/x,z/y} x : A, y : A$. Thus, by cut (part 2), we obtain the result. □

COROLLARY 4.8: RIGHT-INVERTIBILITY OF U. *If* $\Gamma \vdash_{\beta} \mathsf{U}_{x.\alpha}(\Delta \mid A)$ *then* $\Gamma, \Delta \vdash_{\alpha[\beta/x]} A$.

Proof. UL with identities in both premises gives a derivation

$$\frac{\alpha = z[\alpha[\vec{x/x}]/z] \quad \Gamma, \Delta \vdash_{\vec{x/x}} \Delta \quad \Gamma, \Delta, x : \mathsf{U}_{x.\alpha}(\Delta \mid A), z : A \vdash_{z} A}{\Gamma, \Delta, x : \mathsf{U}_{x.\alpha}(\Delta \mid A) \vdash_{\alpha} A}$$

Weakening the assumed derivation to $\Gamma, \Delta \vdash_{\beta} \mathsf{U}_{x.\alpha}(\Delta \mid A)$ and then cutting for x in the above gives the result:

$$\frac{\Gamma, \Delta \vdash_{\beta} \mathsf{U}_{x.\alpha}(\Delta \mid A) \quad \Gamma, \Delta, x : \mathsf{U}_{x.\alpha}(\Delta \mid A) \vdash_{\alpha} A}{\Gamma, \Delta \vdash_{\alpha[\beta/x]} A}$$

□

4.2 General Properties

We give a couple of general constructions that were used in several examples above.

The following "fusion" lemmas (which are isomorphisms, not just interprovabilities) relate F and U. Special cases include: $A \times (B \times C)$ is isomorphic to a primitive triple product $\{x : A, y : B, z : C\}$; currying; and associativity of n-ary functions $(A_1,\ldots,A_n \to (B_1,\ldots,B_m \to C)$ is isomorphic to $A_1,\ldots,A_n,B_1,\ldots,B_m \to C)$. The derivations are in Figure 3. We adopt the convention that an unlabeled leaf $\alpha \Rightarrow \beta$ is proved by equality of context descriptors, and an unlabeled sequent leaf is proved by identity (Theorem 4.4).

LEMMA 4.9: FUSION.

1. $\mathsf{F}_\alpha(\Delta, x : \mathsf{F}_\beta(\Delta'), \Delta'') \dashv\vdash \mathsf{F}_{\alpha[\beta/x]}(\Delta, \Delta', \Delta'')$
2. $\mathsf{U}_{x.\alpha}(\Delta, y : \mathsf{F}_\beta(\Delta'), \Delta'' \mid A) \dashv\vdash \mathsf{U}_{x.\alpha[\beta/y]}(\Delta, \Delta', \Delta'' \mid A)$
3. $\mathsf{U}_{x.\alpha}(\Delta \mid \mathsf{U}_{y.\beta}(\Delta' \mid A)) \dashv\vdash \mathsf{U}_{x.\beta[\alpha/y]}(\Delta, \Delta' \mid A)$

The types respect the structural transformations, covariantly for F and contravariantly for U.

LEMMA 4.10: TYPES RESPECT STRUCTURAL TRANSFORMATIONS.

1. *If* $\alpha \Rightarrow \beta$ *then* $\mathsf{F}_\alpha(\Delta) \vdash \mathsf{F}_\beta(\Delta)$
2. *If* $\alpha \Rightarrow \beta$ *then* $\mathsf{U}_{x.\beta}(\Delta \mid A) \vdash \mathsf{U}_{x.\alpha}(\Delta \mid A)$

Proof.

$$\dfrac{\dfrac{\alpha \Rightarrow \beta[\vec{w}/w] \quad \Delta \vdash_{\vec{w}/w} \Delta}{\Delta \vdash_\alpha \mathsf{F}_\beta(\Delta)}\ \mathsf{FR}}{x : \mathsf{F}_\alpha(\Delta) \vdash_x \mathsf{F}_\beta(\Delta)}\ \mathsf{FL}$$

$$\dfrac{\dfrac{\alpha \Rightarrow z[\beta[\vec{w}/w]/z] \quad x : \mathsf{U}_{x.\beta}(\Delta \mid A), \Delta \vdash_{\vec{w}/w} \Delta \quad \ldots, z : A \vdash_z A}{x : \mathsf{U}_{x.\beta}(\Delta \mid A), \Delta \vdash_\alpha A}\ \mathsf{UL}}{x : \mathsf{U}_{x.\beta}(\Delta \mid A) \vdash \mathsf{U}_{x.\alpha}(\Delta \mid A)}\ \mathsf{UR}$$

□

5 Equational Theory

5.1 Equations on Structural Transformations

First, we need a notation for derivations of the $\alpha \Rightarrow \beta$ judgement in Figure 1. We assume names for constants are given in the signature Σ, and write 1_α for reflexivity, $s_1;s_2$ for transitivity (in diagramatic order), and $s_1[s_2/x]$ for congruence. We allow the signature Σ to provide some axioms for equality of transformations $s_1 \equiv s_2$ (for two derivations of the same judgement $s_1, s_2 :: \psi \vdash \alpha \Rightarrow_p \beta$), and define equality to be the least congruence closed under those axioms and the following associativity, unit, and interchange laws:

- $1_\alpha; s \equiv s \equiv s; 1_\beta$ for $s :: \alpha \Rightarrow \beta$
- $(s_1;s_2);s_3 \equiv s_1;(s_2;s_3)$
- $s[1_x/x] \equiv s \equiv 1_x[s/x]$ for $s :: \psi, x, \psi' \vdash \alpha \Rightarrow_r \beta$
- $s_1[s_2/x][s_3/y] \equiv s_1[s_3/y][s_2[s_3/y]/x]$ as transformations $\alpha_1[\alpha_2/x][\alpha_3/y] \Rightarrow \beta_1[\beta_2/x][\beta_3/y]$ for $s_1 :: (\psi, x : p, y : q \vdash \alpha_1 \Rightarrow_r \beta_1), s_2 :: (\psi, y : q \vdash \alpha_2 \Rightarrow_p \beta_2), s_3 :: (\psi \vdash \alpha_3 \Rightarrow_q \beta_3)$
- $s_1[t_1/x]; s_2[t_2/x] \equiv (s_1;s_2)[(t_1;t_2)/x]$ as morphisms $\alpha_1[\beta_1/x] \Rightarrow \alpha_3[\beta_3/x]$ for $s_1 :: (\psi, x : p, \psi' \vdash \alpha_1 \Rightarrow_r \alpha_2)$, $s_2 :: (\psi, x : p, \psi' \vdash \alpha_2 \Rightarrow_r \alpha_3)$ $t_1 :: (\psi, \psi' \vdash \beta_1 \Rightarrow_p \beta_2)$ $t_2 :: (\psi, \psi' \vdash \beta_2 \Rightarrow_p \beta_3)$

$$\dfrac{\dfrac{\dfrac{\alpha[\beta/x] \Rightarrow \alpha[\beta/x][z\vec{/}z] \quad \Delta,\Delta'',\Delta' \vdash_{z\vec{/}z} \Delta,\Delta',\Delta''}{\Delta,\Delta'',\Delta' \vdash_{\alpha[\beta/x]} \mathsf{F}_{\alpha[\beta/x]}(\Delta,\Delta',\Delta'')}\,\mathsf{FL}}{\Delta, x:\mathsf{F}_{\beta}(\Delta'),\Delta'' \vdash_{\alpha} \mathsf{F}_{\alpha[\beta/x]}(\Delta,\Delta',\Delta'')}\,\mathsf{FL}}{z:\mathsf{F}_{\alpha}(\Delta, x:\mathsf{F}_{\beta}(\Delta'),\Delta'') \vdash_{z} \mathsf{F}_{\alpha[\beta/x]}(\Delta,\Delta',\Delta'')}\,\mathsf{FL}$$

$$\dfrac{\dfrac{\alpha[\beta/x] \Rightarrow \alpha[y\vec{/}y, \beta/x, z\vec{/}z] \qquad \dfrac{\Delta,\Delta',\Delta'' \vdash_{y\vec{/}y} \Delta \qquad \dfrac{\beta \Rightarrow \beta[w\vec{/}w] \quad \Delta,\Delta',\Delta'' \vdash_{w\vec{/}w} \Delta'}{\Delta,\Delta',\Delta'' \vdash_{\beta} \mathsf{F}_{\beta}(\Delta')}\,\mathsf{FR} \qquad \Delta,\Delta',\Delta'' \vdash_{z\vec{/}z} \Delta''}{\Delta,\Delta',\Delta'' \vdash_{y\vec{/}y, \beta/x, z\vec{/}z} \Delta, x:\mathsf{F}_{\beta}(\Delta'),\Delta''}\,\mathsf{FR}}{\Delta,\Delta',\Delta'' \vdash_{\alpha[\beta/x]} \mathsf{F}_{\alpha}(\Delta, x:\mathsf{F}_{\beta}(\Delta'),\Delta'')}\,\mathsf{FL}}{z:\mathsf{F}_{\alpha[\beta/x]}(\Delta,\Delta',\Delta'') \vdash_{z} \mathsf{F}_{\alpha}(\Delta, x:\mathsf{F}_{\beta}(\Delta'),\Delta'')}$$

$$\dfrac{\dfrac{\alpha[\beta/y] \Rightarrow z[\alpha[w\vec{/}w, \beta/y]/z] \quad \Gamma \vdash_{w\vec{/}w} \Delta,\Delta'' \quad \dfrac{}{\Gamma \vdash_{\beta} \mathsf{F}_{\beta}(\Delta')}\,\mathsf{FR} \quad \Gamma, z:A \vdash_{z} A}{\Gamma := (x:\mathsf{U}_{x.\alpha}(\Delta, y:\mathsf{F}_{\beta}(\Delta'),\Delta'' \mid A), \Delta,\Delta',\Delta'') \vdash_{\alpha[\beta/y]} A}\,\mathsf{UL}}{x:\mathsf{U}_{x.\alpha}(\Delta, y:\mathsf{F}_{\beta}(\Delta'),\Delta'' \mid A) \vdash_{x} \mathsf{U}_{x.\alpha[\beta/y]}(\Delta,\Delta',\Delta'' \mid A)}\,\mathsf{UR}$$

$$\dfrac{\dfrac{\dfrac{\alpha[\beta/y] \Rightarrow z[\alpha[\beta/y][w\vec{/}w]] \quad \Delta,\Delta',\Delta'' \vdash_{w\vec{/}w} \Delta,\Delta',\Delta'' \quad \Delta,\Delta',\Delta'', z:A \vdash_{z} A}{x:\mathsf{U}_{x.\alpha[\beta/y]}(\Delta,\Delta',\Delta'' \mid A), \Delta,\Delta'',\Delta' \vdash_{\alpha[\beta/y]} A}\,\mathsf{UL}}{x:\mathsf{U}_{x.\alpha[\beta/y]}(\Delta,\Delta',\Delta'' \mid A), \Delta, y:\mathsf{F}_{\beta}(\Delta'),\Delta'' \vdash_{\alpha} A}\,\mathsf{FL}}{x:\mathsf{U}_{x.\alpha[\beta/y]}(\Delta,\Delta',\Delta'' \mid A) \vdash_{x} \mathsf{U}_{x.\alpha}(\Delta, y:\mathsf{F}_{\beta}(\Delta'),\Delta'' \mid A)}\,\mathsf{UR}$$

$$\dfrac{\dfrac{\beta[\alpha/y] \Rightarrow \beta[\alpha[w\vec{/}w]/y] \quad x:\mathsf{U}_{x.\alpha}(\Delta \mid \mathsf{U}_{y.\beta}(\Delta' \mid A)),\Delta,\Delta' \vdash_{w\vec{/}w} \Delta \quad \dfrac{}{x:\mathsf{U}_{x.\alpha}(\Delta \mid \mathsf{U}_{y.\beta}(\Delta' \mid A)),\Delta,\Delta', y:\mathsf{U}_{y.\beta}(\Delta' \mid A) \vdash_{\beta} A}\,\mathsf{UL}}{x:\mathsf{U}_{x.\alpha}(\Delta \mid \mathsf{U}_{y.\beta}(\Delta' \mid A)),\Delta,\Delta' \vdash_{\beta[\alpha/y]} A}\,\mathsf{UL}}{x:\mathsf{U}_{x.\alpha}(\Delta \mid \mathsf{U}_{y.\beta}(\Delta' \mid A)) \vdash_{x} \mathsf{U}_{x.\beta[\alpha/y]}(\Delta,\Delta' \mid A)}\,\mathsf{UR}$$

$$\dfrac{\dfrac{\dfrac{}{x:\mathsf{U}_{x.\beta[\alpha/y]}(\Delta,\Delta' \mid A),\Delta,\Delta' \vdash_{\beta[\alpha/y]} A}\,\mathsf{UL}}{x:\mathsf{U}_{x.\beta[\alpha/y]}(\Delta,\Delta' \mid A),\Delta \vdash_{\alpha} \mathsf{U}_{y.\beta}(\Delta' \mid A)}\,\mathsf{UR}}{x:\mathsf{U}_{x.\beta[\alpha/y]}(\Delta,\Delta' \mid A) \vdash_{x} \mathsf{U}_{x.\alpha}(\Delta \mid \mathsf{U}_{y.\beta}(\Delta' \mid A))}\,\mathsf{UR}$$

Figure 3: Derivations of fusion lemmas

$$
\begin{array}{rcll}
d[x/x] & \equiv & d & \\
x[d/x] & \equiv & d & \\
d_1[d_2/x] & \equiv & d_1 & \text{if } x\#d_1 \\
(d_1[d_2/x])[d_3/y] & \equiv & (d_1[d_3/y])[d_2[d_3/y]/x] & \\
\\
1_*(d) & \equiv & d & \\
(s_1;s_2)_*(d) & \equiv & s_{1*}(s_{2*}(d)) & \\
(s_2[s_1/x])_*(d_2[d_1/x]) & \equiv & s_{2*}(d_2)[s_{1*}(d_1)/x] & \\
\\
\mathsf{FL}^{x_0}(\Delta.d)[\mathsf{FR}(s,\vec{d_i/x_i})/x_0] & \equiv & (1_\beta[s/x_0])_*(d[\vec{d_i/x_i}]) & \\
\mathsf{UL}^{x_0}(s,\vec{d_i/x_i},z.d')[\mathsf{UR}(\Delta.d)/x_0] & \equiv & (s[1_\alpha/x_0])_*(d'[(d[\vec{d_i/x_i}])/z]) & \\
d :: \Gamma \vdash_\beta \mathsf{U}_{x.\alpha}(\Delta \mid A) & \equiv & \mathsf{UR}(\Delta.\mathsf{UL}^*_x[d/x]) & \\
d :: \Gamma, x : \mathsf{F}_\alpha(\Delta), \Gamma' \vdash_\beta C & \equiv & \mathsf{FL}^x(\Delta.d[\mathsf{FR}^*/x]) &
\end{array}
$$

Figure 4: Equality of Derivations

- $1_\alpha[1_\beta/x] \equiv 1_{\alpha[\beta/x]}$
- $1_\alpha[s/y] \equiv 1_\alpha$ if $y\#\alpha$

These are the 2-category axioms extended to the multicategorical case. The first two rules are associativity and unit for both kinds of compositions; the next two are interchange; the final is because terms with variables that do not occur are an implicit notation for product projections. The associativity and unit laws for congruence/horizontal composition $(s[s'/x])$ require the analogous associativity for composition $(\alpha[\alpha'/x])$ (which is true syntactically) to type check.

As we did with equality of context descriptors, we think of all definitions as being parametrized by $\equiv$-equivalence-classes of transformations, not raw syntax.

5.2 Equations on Derivations

To simplify the axiomatic description of equality, we give a notation for derivations where the admissible transformation, identity, and cut rules are internalized as explicit rules—so the calculus has the flavor of an explicit substitution one. We use the following notation for derivations:

$$
d \quad ::= \quad x \mid s_*(d) \mid d_1[d_2/x] \mid \mathsf{FL}^x(\Delta.d) \mid \mathsf{FR}_\gamma(s,\vec{d_i/x_i}) \mid \mathsf{UL}^x_\gamma(s,\vec{d_i/x_i},d) \mid \mathsf{UR}(\Delta.d)
$$

We omit the primitive hypothesis rule for atoms (it is derivable), write x for identity (Theorem 4.4), $s_*(d)$ for respect for transformations (Lemma 4.1—identity for atoms combines this and identity) and $d_1[d_2/x]$ for cut (Theorem 4.6). The next 4 terms correspond to the 4 U/Frules. from Figure 2. We do not notate weakenings or exchanges in these terms.

We write FR^* for $\mathsf{FR}_{\vec{x/x}}(1_\alpha,\vec{x/x}) :: \Gamma \vdash_\alpha \mathsf{F}_\alpha(\Delta)$ when $\Delta \subseteq \Gamma$ and we write and UL^*_x for $\mathsf{UL}^x_{\vec{x/x}}(1_\alpha,\vec{x/x},z.z) :: \Gamma \vdash_\alpha A$ when $x : \mathsf{U}_{x.\alpha}(\Delta \mid A) \in \Gamma$ and $\Delta \subseteq \Gamma$.

The equational theory of derivations is the least congruence containing the equations in Figure 4.

The first two equations say that identity is a unit for cut. The third says that non-occurence of a variable is a projection (with more explicit weakening, the notation $x\#d_1$ means that d_1 is the weakening of some derivation $\Gamma,\Gamma' \vdash_\alpha C$ to $\Gamma,x:A,\Gamma' \vdash_\alpha C$, and the equation says that we return that original derivation). The fourth is functoriality of cut (when phrased for single-variable substitutions, the equation $d[\theta][\theta'] \equiv d[\theta[\theta']]$ becomes a rule for commuting substitutions).

In the next group, the first two rules say that the action of a transformation is functorial, and commutes with cut. The typing in the third rule is $d_1 :: \Gamma \vdash_{\alpha'} A$ and $d_2 :: \Gamma, x : A \vdash_{\beta'} C$ and $s_1 :: \alpha \Rightarrow \alpha'$ and $s_2 :: \beta \Rightarrow \beta'$, so both sides are derivations of as derivations of $\Gamma \vdash_{\beta[\alpha/x]} C$.

In the next group, we have the $\beta\eta$-laws for F and U. The β laws are the principal cut cases given above. By the composition law for cut, the simultaneous substitution can be defined as iterated substitution in any order. The η law for U equates any derivation to

$$\dfrac{\dfrac{\dfrac{d}{\Gamma,\Delta \vdash_\beta \mathsf{U}_{x.\alpha}(\Delta \mid A)} \quad \dfrac{}{\Gamma,\Delta,x:\mathsf{U}_{x.\alpha}(\Delta \mid A) \vdash_\alpha A}\ \mathsf{UL}*}{\Gamma,\Delta \vdash_{\beta[\alpha/x]} \mathsf{U}_{x.\alpha}(\Delta \mid A)}\ \mathsf{cut}}{\Gamma \vdash_\alpha \mathsf{U}_{x.\alpha}(\Delta \mid A)}\ \mathsf{UR}$$

The η law for F equates any derivation to

$$\dfrac{\dfrac{\dfrac{}{\Gamma,\Gamma',\Delta \vdash_\alpha \mathsf{F}_\alpha(\Delta)}\ \mathsf{FR}^* \quad d}{\Gamma,\Gamma',\Delta \vdash_{\beta[\alpha/x]} C}\ \mathsf{cut}}{\Gamma,x:\mathsf{F}_\alpha(\Delta),\Gamma' \vdash_\beta C}\ \mathsf{FL}$$

6 Categorical Semantics

In this section, we give a category-theoretic structure corresponding to the above syntax. First, we define a cartesian 2-multicategory as a semantic analogue of the syntax in Figure 1.

DEFINITION 6.1. *A* ***(strict) cartesian 2-multicategory*** *consists of*

1. *A set $\mathscr{M}_0$ of* objects.

2. *For every object B and every finite list of objects $(A_1,\dots,A_n)$, a category $\mathscr{M}(A_1,\dots,A_n;B)$. The objects of this category are* 1-morphisms *and its morphisms are* 2-morphisms*; we write composition of 2-morphisms as $s_1 \cdot s_2$.*

3. *For each object A, an identity arrow $1_A \in \mathscr{M}(A;A)$.*

4. *For any object C and lists of objects $(B_1,\dots,B_m)$ and $(A_{i1},\dots,A_{in_i})$ for $1 \le i \le m$, a composition functor*

$$\mathscr{M}(B_1,\dots,B_m;C) \times \prod_{i=1}^{m} \mathscr{M}(A_{i1},\dots,A_{in_i};B_i) \longrightarrow \mathscr{M}(A_{11},\dots,A_{mn_m};C)$$
$$(g,(f_1,\dots,f_m)) \mapsto g \circ (f_1,\dots,f_m)$$

   We write the action of this functor on 2-cells as $d \circ_2 (e_1,\dots,e_m)$.

5. *For any $f \in \mathscr{M}(A_1,\dots,A_n;B)$ we have natural equalities (i.e. natural transformations whose components are identities)*

$$1_B \circ (f) = f \qquad f \circ (1_{A_1},\dots,1_{A_n}) = f.$$

6. *For any h, g_i, f_{ij} we have natural equalities*

$$(h \circ (g_1,\dots,g_m)) \circ (f_{11},\dots,f_{mn_m}) = \\ h \circ (g_1 \circ (f_{11},\dots,f_{1n_1}),\dots,g_m \circ (f_{m1},\dots,f_{mn_m}))$$

7. *For any function $\sigma : \{1,\dots,m\} \to \{1,\dots,n\}$ and objects $A_1,\dots,A_n,B$, a* renaming *functor*

$$\mathscr{M}(A_{\sigma 1},\dots,A_{\sigma m};B) \to \mathscr{M}(A_1,\dots,A_n;B)$$
$$f \mapsto f\sigma^*$$

8. *The functors σ^* satisfy the following natural equalities:*

$$f\sigma^*\tau^* = f(\tau\sigma)^*$$
$$f(1_n)^* = f$$
$$g \circ (f_1\sigma_1^*, \dots, f_n\sigma_n^*) = (g \circ (f_1, \dots, f_n))(\sigma_1 \sqcup \cdots \sqcup \sigma_n)^*$$
$$g\sigma^* \circ (f_1, \dots, f_n) = (g \circ (f_{\sigma 1}, \dots, f_{\sigma m}))(\sigma \wr (k_1, \dots, k_n))^*$$

In the last equation, k_i is the arity of f_i, and $\sigma \wr (k_1, \dots, k_n)$ denotes the composite function

$$\{1, \dots, \textstyle\sum_{i=1}^m k_{\sigma i}\} \xrightarrow{\sim} \bigsqcup_{i=1}^m \{1, \dots, k_{\sigma i}\} \xrightarrow{\widehat{\sigma}} \bigsqcup_{j=1}^n \{1, \dots, k_j\} \xrightarrow{\sim} \{1, \dots, \textstyle\sum_{j=1}^n k_j\}$$

where $\widehat{\sigma}$ acts as the identity from the i^{th} summand to the $(\sigma i)^{\text{th}}$ summand.

We will find it useful to work with the following alternative description of composition in a multicategory. If in the "multi-composition" $g \circ (f_1, \dots, f_m)$ all the f_j's for $j \neq i$ are identities, we write it as $g \circ_i f_i$. We may also write it as $g \circ_{B_i} f_i$ if there is no danger of ambiguity (e.g. if none of the other B_j's are equal to B_i). Thus we have **one-place composition** functors

$$\circ_i : \mathscr{M}(B_1, \dots, B_n; C) \times \mathscr{M}(A_1, \dots, A_m; B_i) \to \mathscr{M}(B_1, \dots, B_{i-1}, A_1, \dots, A_m, B_{i+1}, \dots, B_n; C)$$

that satisfy the following natural equalities:

- $1_B \circ_1 f = f$ (since 1_B is unary, $\circ_1$ is the only possible composition here).
- $f \circ_i 1_{B_i} = f$ for any i.
- If h is n-ary, g is m-ary, and f is k-ary, then

$$(h \circ_i g) \circ_j f = \begin{cases} (h \circ_j f) \circ_{i+k-1} g & \text{if } j < i \\ h \circ_i (g \circ_{j-i+1} f) & \text{if } i \le j < i+m \\ (h \circ_{j-m+1} f) \circ_i g & \text{if } j \ge i+m \end{cases}$$

The next three definitions will be used to describe the $\Gamma \vdash_\alpha A$ judgement.

DEFINITION 6.2. *A* ***functor of cartesian 2-multicategories*** *$F : \mathscr{D} \to \mathscr{M}$ consists of a function $F_0 : \mathscr{D}_0 \to \mathscr{M}_0$ and functors $\mathscr{D}(A_1, \dots, A_n; B) \to \mathscr{M}(F_0(A_1), \dots, F_0(A_n); F_0(B))$ such that the chosen identities, compositions, and renamings are preserved (strictly).*

DEFINITION 6.3. *A functor of cartesian 2-multicategories $\pi : \mathscr{D} \to \mathscr{M}$ is a* ***local discrete fibration*** *if each induced functor of ordinary categories $\mathscr{D}(A_1, \dots, A_n; B) \to \mathscr{M}(\pi A_1, \dots, \pi A_n; \pi B)$ is a discrete fibration.*

We write $\mathscr{D}_\alpha(A_1, \dots, A_n; B)$ for the fiber of this functor over $\alpha \in \mathscr{M}(\pi A_1, \dots, \pi A_n; \pi B)$; when π is a local discrete fibration, this fiber is a discrete set.

DEFINITION 6.4. *If $\pi : \mathscr{D} \to \mathscr{M}$ is a local discrete fibration, then a morphism $\xi \in \mathscr{D}(A_1, \dots, A_n; B)$ is* ***opcartesian*** *if all diagrams of the following form are pullbacks of categories:*

$$\begin{tikzcd}
\mathscr{D}(\vec{C}, B, \vec{D}; E) \arrow[r, "(-)\circ_B \xi"] \arrow[d, "\pi"'] & \mathscr{D}(\vec{C}, \vec{A}, \vec{D}; E) \arrow[d, "\pi"] \\
\mathscr{M}(\pi\vec{C}, \pi B, \pi\vec{D}; \pi E) \arrow[r, "(-)\circ_{\pi B} \pi\xi"'] & \mathscr{M}(\pi\vec{C}, \pi\vec{A}, \pi\vec{D}; \pi E)
\end{tikzcd}$$

Dually, a morphism $\xi \in \mathscr{D}(\vec{C}, B, \vec{D}; E)$ is ***cartesian at*** *B if all diagrams of the following form are pullbacks of categories:*

$$\begin{tikzcd}
\mathscr{D}(\vec{A}; B) \arrow[r, "\xi \circ_B (-)"] \arrow[d, "\pi"'] & \mathscr{D}(\vec{C}, \vec{A}, \vec{D}; E) \arrow[d, "\pi"] \\
\mathscr{M}(\pi\vec{A}; \pi B) \arrow[r, "\pi\xi \circ_{\pi B} (-)"'] & \mathscr{D}(\pi\vec{C}, \pi\vec{A}, \pi\vec{D}; \pi E)
\end{tikzcd}$$

*Given $\mu : (p_1,\ldots,p_n) \to q$ in $\mathscr{M}$, we say that π **has μ-products** if for any A_i with $\pi A_i = p_i$, there exists a B with $\pi B = q$ and an opcartesian morphism in $\mathscr{D}_\mu(A_1,\ldots,A_n;B)$. Dually, we say π **has μ-homs** if for any i, any B with $\pi B = q$, and any A_j with $\pi A_j = p_j$ for $j \neq i$, there exists an A_i with $\pi A_i = p_i$ and a cartesian morphism in $\mathscr{D}_\mu(A_1,\ldots,A_n;B)$.*

*We say that π is an **opfibration** if it has μ-products for all μ, a **fibration** if it has μ-homs for all μ, and a **bifibration** if it is both an opfibration and a fibration.*

Useful will be the following characterisation of pullbacks of categories:

LEMMA 6.5. *A diagram of categories*

$$\begin{tikzcd} \mathscr{A} \arrow[r, "H"] \arrow[d, "K"'] & \mathscr{B} \arrow[d, "F"] \\ \mathscr{C} \arrow[r, "G"'] & \mathscr{D} \end{tikzcd}$$

is a pullback diagram iff:

- *For every pair of objects $b \in \mathscr{B}$ and $c \in \mathscr{C}$ with $Fb = Gc$, there is a unique object $a \in \mathscr{A}$ such that $Ha = b$ and $Ka = c$; and,*
- *For every pair of morphisms $f \in \mathscr{B}(b,b')$ and $g \in \mathscr{C}(c,c')$ with $Fb = Gc$ and $Fb' = Gc'$ and $Ff = Gg$, there is a unique morphism $\theta \in \mathscr{A}$ such that $H\theta = f$ and $K\theta = g$. The domain and codomain of θ are fixed by the previous property.*

Proof. In the forward direction, any two objects $b \in \mathscr{B}$ and $c \in \mathscr{C}$ such that $Fb = Gc$ determine functors $J : \star \to \mathscr{B}$ and $L : \star \to \mathscr{C}$ from the terminal category, such that $FJ = GL$. The universal property of pullbacks gives a unique functor $\star \to \mathscr{A}$ making the whole diagram commute, and this functor picks out the unique object $a \in \mathscr{A}$ with the required property.

The property for arrows follows the same argument, instead using the fact that a morphism in a category determines a functor from the category given by the walking arrow $\star \to \star$.

For the reverse direction, suppose we have a category $\mathscr{E}$ and functors $J : \mathscr{E} \to \mathscr{B}$ and $L : \mathscr{E} \to \mathscr{C}$ such that $FJ = GL$. We construct a functor $P : \mathscr{E} \to \mathscr{A}$ as follows. On objects, set $P(e)$ to be the unique object $a \in \mathscr{A}$ such that $H(a) = J(e)$ and $K(a) = L(e)$. On morphisms, set $P(f)$ to be the unique morphism g such that $H(g) = J(f)$ and $K(g) = L(f)$. The uniqueness principles ensure that these assignments are functorial. This functor itself is unique as its definition is forced on both objects and morphisms, so the square is a pullback square. $\square$

We now describe how these definitions correspond to the syntax.

Before we begin, we will describe explicitly the process of simultaneous substitution. Suppose we are given a term $\psi' \vdash \alpha : q$ and a substitution $\psi \vdash \gamma : \psi'$. First weaken α to $\psi, \psi' \vdash \alpha : q$. Then for the empty substitution $\psi \vdash \cdot : \cdot$, set $\alpha[\cdot] := \alpha$. For a substitution $\psi \vdash (\beta/x, \gamma) : (x : p, \psi')$, we inductively define $\alpha[\beta/x, \gamma] := \alpha[\beta/x][\gamma]$, where β has been weakened to $\psi, \psi' \vdash \beta : p$. This type checks, because at each step we are invoking the substitution rule on derivations of the form $\psi, x : p, \psi' \vdash \alpha : q$ and $\psi, \psi' \vdash \beta : p$, yielding $\psi, \psi' \vdash \alpha[\beta/x] : p$, until ψ' is exhausted.

If we are given a term $x_1 : p_1, \ldots, x_n : p_n \vdash \alpha : q$ and terms $\psi_i \vdash \beta_i : p_i$ with differing contexts, we can construct a term $\psi_1, \ldots, \psi_n \vdash a[\beta_1/x_1, \ldots, \beta_n/x_n] : q$ first by weakening each β_i to all of $\psi_1, \ldots, \psi_n$ and then applying the above.

THEOREM 6.6: MODE THEORY PRESENTS A MUTILCATEGORY. *A mode theory Σ presents a cartesian 2-multicategory $\mathscr{M}$, where $\mathscr{M}_0$ is the set of modes, and an object of $\mathscr{M}(p_1,\ldots,p_n;q)$ is a term $x_1 : p_1, \ldots, x_n : p_n \vdash \alpha : q$ and a morphism of $\mathscr{M}(p_1,\ldots,p_n;q)$ is a structural transformation $s :: \psi \vdash \alpha \Rightarrow_q \beta$, both considered modulo $\equiv$ (see Section 5.1).*

Proof. First we need to check that, for any modes $p_1,\ldots,p_n,q$, the above definitions give a category $\mathscr{M}(p_1,\ldots,p_n;q)$. Both the identity 2-morphisms and composites of 2-morphisms are given directly by the first two rules for transformations. The first two $\equiv$-axioms for transformations are exactly the unit and associativity laws.

The identity 1-morphism is given is given by the derivation $x : p \vdash x : p$.

We define the composition functors

$$\circ : \mathscr{M}(q_1,\ldots,q_m;r) \times \prod_{i=1}^{m} \mathscr{M}(p_{i1},\ldots,p_{in_i};q_i) \to \mathscr{M}(p_{11},\ldots,p_{mn_m};r)$$

as follows. Given 1-morphisms $\alpha \in \mathscr{M}(q_1,\ldots,q_m;r)$ and $\beta_i \in \mathscr{M}(p_{i1},\ldots,p_{in_i};q_i)$, define $\alpha \circ (\beta_1,\ldots,\beta_m) = \alpha[\beta_1/x_1,\ldots,\beta_m/x_m]$, with weakening inserted as required, according to the discussion above.

These composition functors act on 2-morphisms in the following way: if we have 2-morphisms $s : \alpha \Rightarrow \alpha'$ and $t_i : \beta_i' \Rightarrow \beta_i'$, define $s \circ_2 (t_1,\ldots,t_m) = s[t_1/x_1,\ldots,t_m/x_m]$. This 2-morphism has the correct domain and codomain: $\alpha[\beta_1/x_1,\ldots,\beta_m/x_m]$ and $\alpha'[\beta_1'/x_1,\ldots,\beta_m'/x_m]$ respectively. One checks easily that this definition is functorial; by applying the rule $1_\alpha[1_\beta/x] \equiv 1_{\alpha[\beta/x]}$ iteratively we find

$$\begin{aligned} 1_\alpha \circ_2 (1_{\beta_1},\ldots,1_{\beta_m}) &= 1_\alpha[1_{\beta_1}/x_1][1_{\beta_2}/x_2]\ldots[1_{\beta_m}/x_m] \\ &\equiv (1_{\alpha[\beta_1/x_1]})[1_{\beta_2}/x_2]\ldots[1_{\beta_m}/x_m] \\ &\equiv \ldots \\ &\equiv 1_{\alpha[\beta_1/x_1]\ldots[\beta_m/x_m]} \end{aligned}$$

and applying $s_1[t_1/x];s_2[t_2/x] \equiv (s_1;s_2)[(t_1;t_2)/x]$ iteratively,

$$\begin{aligned} (s \cdot s') \circ_2 (t_1 \cdot t_1',\ldots,t_m \cdot t_m') &= (s;s')[t_1;t_1'/x_1][t_2;t_2'/x_2]\ldots[t_m;t_m'/x_m] \\ &= (s[t_1/x_1];s'[t_1'/x_1])[t_2;t_2'/x_2]\ldots[t_m;t_m'/x_m] \\ &= \ldots \\ &= (s[t_1/x_1]\ldots[t_m/x_m]);(s'[t_1'/x_1]\ldots[t_m'/x_m]) \\ &= (s \circ_2 (t_1,\ldots,t_m)) \cdot (s' \circ_2 (t_1',\ldots,t_m')) \end{aligned}$$

as required.

The unit and associativity laws follow for 1-morphisms from standard facts about building multi-variable substitutions from single variable substitutions. For 2-morphisms, they are enforced by the axioms in the equational theory.

The action of renaming is given by the admissible weakening, exchange and contraction rules. □

We fix a mode theory Σ and write $\mathscr{M}$ for the 2-multicategory it presents. The overall conjecture is that the syntax is the initial bifibration over $\mathscr{M}$. Together, the following soundness and completeness theorems give weak initiality; for every bifibration over $\mathscr{M}$ there exists a functor to it from a syntactic bifibration.

The syntactic bifibration $\pi : \mathscr{D} \to \mathscr{M}$ is constructed as follows.

THEOREM 6.7: COMPLETENESS/SYNTACTIC BIFIBRATION. *The syntax presents a bifibration $\pi : \mathscr{D} \to \mathscr{M}$, where:*

- *Objects of $\mathscr{D}$ are pairs $(p, A\ \mathsf{type}_p)$;*
- *1-morphisms $\Gamma \to B$, i.e., objects of $\mathscr{D}(\Gamma;B)$, are pairs $(\alpha, d :: \Gamma \vdash_\alpha B)$ (up to $\equiv$-equivalence of derivations); and,*
- *2-morphisms $(\alpha,d) \to (\alpha',d')$ are structural transformations $s :: \alpha \Rightarrow \alpha'$ such that $s_*(d') \equiv d$.*

The functor $\pi : \mathscr{D} \to \mathscr{M}$ is given by first projection on objects and 1-morphisms, and sends 2-morphisms to the underlying structural transformations.

Proof. To save space, we will simply write A and d for objects and derivations, when the underlying modes and mode morphisms are clear. We will also omit variable names from sequents when able.

Composition of 1-morphisms is defined analogously to the mode multicategory: for 1-morphisms $g :: (x_1 : B_1,\ldots,x_m : B_m \vdash_\alpha C)$ and $f_i :: (A_{i1},\ldots,A_{in_i} \vdash_{\beta_i} B_i)$ we set

$$g \circ (f_1,\ldots,f_n) := (\alpha[\beta_1/x_1,\ldots,\beta_m], g[f_1/x_1,\ldots,f_m/x_m])$$

where we have again implicitly used weakening in the same manner as in composition of mode morphisms. That the latter derivation lies over $\alpha[\beta_1/x_1,\ldots,\beta_m]$ follows from the cut rule and weakening-over-weakening.

For the action of these composition functors on 2-morphisms, suppose we are given 1-morphisms

$$\begin{aligned} d &:: x_1 : B_1,\ldots,x_m : B_m \vdash_\alpha C \\ d' &:: x_1 : B_1,\ldots,x_m : B_m \vdash_{\alpha'} C \\ e_i &:: A_{i1},\ldots,A_{in_i} \vdash_{\beta_i} B_i \\ e'_i &:: A_{i1},\ldots,A_{in_i} \vdash_{\beta'_i} B_i \end{aligned}$$

and 2-morphisms $S : (\alpha,d) \Rightarrow (\alpha',d')$ and $T_i(\beta_i,e_i) \to (\beta'_i,e'_i)$ such that S has underlying transformation $s :: \alpha \Rightarrow \alpha'$ and the T_i have underlying transformations $t_i :: \beta_i \Rightarrow \beta'_i$ respectively. This means that $d \equiv s_*(d')$ and $e_i \equiv (t_i)_*(e'_i)$ for all i. The composite $S \circ_2 (T_1,\ldots,T_m)$ is the 2-morphism given by the underlying transformation $s[t_1/x_1,\ldots,t_m/x_m]$. This is a valid 2-morphism $d[e_1/x_1,\ldots,e_m/x_m] \Rightarrow d'[e'_1/x_1,\ldots,e'_m/x_m]$ because

$$\begin{aligned} &(s[t_1/x_1,\ldots,t_m/x_m])_*(d'[e'_1/x_1,\ldots,e'_m/x_m]) \\ \equiv&(s[t_1/x_1,\ldots,t_{m-1}/x_{m-1}])_*(d'[e'_1/x_1,\ldots,e'_{m-1}m/x_{m-1}])[(t_m)_*(e'_m)/x_m] \\ &\vdots \\ \equiv&s_*(d')[(t_1)_*(e'_1)/x_1,\ldots,(t_m)_*(e'_m)/x_m] \\ \equiv&d[e_1/x_1,\ldots,e_m/x_m] \end{aligned}$$

as required, where we have repeatedly applied the rule $(s_2[s_1/x])_*(d_2[d_1/x]) \equiv s_{2*}(d_2)[s_{1*}(d_1)/x]$.

The unit and associativity laws for 1-morphisms follow from the first set of equations for derivations, and from the definition of multi-variable substitution as iterated single variable substitution. For 2-morphisms, they follow as composition of 2-morphisms is simply composition of the underlying transformations in the mode theory.

The cartesian structure is given by the admissible rules for weakening-over-weakening, exchange-over-exchange and contraction-over-contraction, from which all renamings can be made. These rules also all preserve the underlying mode morphisms in the correct way to make π functorial.

The next step is to show that π is a local discrete fibration. Suppose we have a context Γ and object B. We must show that the functor $\pi : \mathscr{D}(\Gamma;B) \to \mathscr{M}(\pi\Gamma;\pi B)$ is a discrete fibration. Let $\alpha,\alpha' \in \mathscr{M}(\pi\Gamma;\pi B)$ be mode morphisms and suppose we have a transformation $s :: \alpha \Rightarrow \alpha'$ between them. Any 2-morphism in $\mathscr{D}(\Gamma;B)$ lying over s must clearly have s as the underlying transformation. Given a lift $d' :: \Gamma \vdash_{\alpha'} B$ of α', then we can consider s as a 2-morphism $(\alpha,s_*(d')) \Rightarrow (\alpha',d')$ over s, whose domain is the action of s on d', $s_*(d')$, as expected. The equational condition $s_*(d) \equiv s_*(d)$ is trivially satisfied, and in fact forces $s_*(d)$ as the only possible choice of domain, so the lift is unique. So π is a local discrete fibration.

We now show that π is an opfibration, i.e., has α-homs for all mode morphisms $\psi \vdash \alpha : q$. Suppose we have lifts for the modes in ψ, i.e., a context Δ with $\pi\Delta = \psi$. We define the opcartesian lift of α to be $\mathsf{FR}^* :: \Delta \vdash_\alpha \mathsf{F}_\alpha(\Delta)$. To verify that this is an opcartesian morphism, we must show that all squares of the form

$$\begin{tikzcd} \mathscr{D}(\Gamma,\mathsf{F}_\alpha(\Delta),\Gamma';C) \arrow[r, "{-[\mathsf{FR}^*/x_0]}"] \arrow[d, "\pi"'] & \mathscr{D}(\Gamma,\Delta,\Gamma';C) \arrow[d, "\pi"] \\ \mathscr{M}(\pi\Gamma,q,\pi\Gamma';\pi C) \arrow[r, "{-[\alpha/x_0]}"'] & \mathscr{M}(\pi\Gamma,\psi,\pi\Gamma';\pi C) \end{tikzcd}$$

are pullbacks of categories. For this we will use the characterisation of pullbacks given in Lemma 6.5. First, the property for objects. Suppose we have an object $d \in \mathscr{D}(\Gamma,\Delta,\Gamma';C)$ and $\beta \in \mathscr{M}(\pi\Gamma,q,\pi\Gamma';\pi C)$ such that $\pi(d) = \beta[\alpha/x_0]$. This simply states that d is of the form $d :: \Gamma,\Delta,\Gamma' \vdash_{\beta[\alpha/x_0]} C$. We must produce a unique object $e \in \mathscr{D}(\Gamma,\mathsf{F}_\alpha(\Delta),\Gamma';C)$ such that $\pi(e) = \beta$ and $e[\mathsf{FR}^*/x_0] \equiv d$.

We take as our e the derivation $\mathsf{FL}^{x_0}(\Delta.d)$. This lies over β, and we calculate

$$\begin{aligned} e[\mathsf{FR}^*/x_0] &= \mathsf{FL}^{x_0}(\Delta.d)[\mathsf{FR}_{\vec{x/x}}(1_\alpha, x/x)/x_0] \\ &\equiv (1_\beta[1_\alpha/x_0])_*(d[\vec{x/x}]) \\ &\equiv (1_{\beta[\alpha/x_0]})_*(d) \\ &\equiv d \end{aligned}$$

by the β-law for F.

It remains to show uniqueness. Suppose we have some derivation e' such that $\pi(e') = \beta$ and $e'[\mathsf{FR}^*/x_0] \equiv d$. By the η-law for F, we have

$$\begin{aligned} e' &\equiv \mathsf{FL}^{x_0}(\Delta.e'[\mathsf{FR}^*/x_0]) \\ &\equiv \mathsf{FL}^{x_0}(\Delta.d) \\ &= e \end{aligned}$$

as required.

We now turn to the pullback property for morphisms. Let $\beta, \beta' \in \mathscr{M}(\pi\Gamma, q, \pi\Gamma'; \pi C)$ and let $s :: \beta \Rightarrow \beta'$ be a morphism. Further suppose that we have derivations $d :: \Gamma, \Delta, \Gamma' \vdash_{\beta[\alpha/x_0]} C$ and $d' :: \Gamma, \Delta, \Gamma' \vdash_{\beta'[\alpha/x_0]} C$ such that $(s[1_\alpha/x_0])_*(d') \equiv d$. This describes a morphism $T : d \Rightarrow d'$ in $\mathscr{D}(\Gamma, \mathsf{F}_\alpha(\Delta), \Gamma'; C)$ that lies over $s[1_\alpha/x_0]$. This latter transformation is the result of applying the functor $-[\alpha/x_0]$ to s.

We now must find a morphism S in $\mathscr{D}(\Gamma, \Delta, \Gamma'; C)$ that lies over s, and such that the functor $-[\mathsf{FR}^*/x_0]$ applied to the morphism S yields T. We know that for S to lie over s, its underlying structural transformation must be s. The action of $-[\mathsf{FR}^*/x_0]$ on S then takes s to $s[1_\alpha/x_0]$ as expected.

By the previous argument for objects, we know that S must have domain $\mathsf{FL}^{x_0}(\Delta.d)$ and codomain $\mathsf{FL}^{x_0}(\Delta.d')$. We can verify that choosing the underlying transformation s gives a well-defined morphism $S : \mathsf{FL}^{x_0}(\Delta.d) \Rightarrow \mathsf{FL}^{x_0}(\Delta.d')$:

$$\begin{aligned} s_*(\mathsf{FL}^{x_0}(\Delta.d')) &\equiv \mathsf{FL}^{x_0}(\Delta.s_*(\mathsf{FL}^{x_0}(\Delta.d'))[\mathsf{FR}^*/x_0]) \\ &\equiv \mathsf{FL}^{x_0}(\Delta.s_*(\mathsf{FL}^{x_0}(\Delta.d'))[(1_\alpha)_*(\mathsf{FR}^*)/x_0]) \\ &\equiv \mathsf{FL}^{x_0}(\Delta.(s[1_\alpha/x_0])_*(\Delta.\mathsf{FL}^{x_0}(d')[\mathsf{FR}^*/x_0]) \\ &\equiv \mathsf{FL}^{x_0}(\Delta.(s[1_\alpha/x_0])_*(d')) \\ &\equiv \mathsf{FL}^{x_0}(\Delta.d) \end{aligned}$$

where we have used the η-law followed by the β-law.

We conclude that all squares of the given form are pullback squares, and so every α has an opcartesian lift. Therefore π is an opfibration.

Finally, we show that π is also a fibration; the reasoning is almost identical. Suppose have a mode morphism $\psi, p \vdash \alpha : q$. We must show that π has α-homs. So suppose we have a context Δ over ψ and a type A over q. We define the cartesian lift of α to be $\mathsf{UL}^* :: (\Delta, \mathsf{U}_\alpha(\Delta \mid A) \vdash_\alpha A)$. We must now verify that all squares

$$\begin{tikzcd} \mathscr{D}(\Gamma; \mathsf{U}_{x.\alpha}(\Delta \mid A)) \arrow[r, "\mathsf{UL}^*[-/x]"] \arrow[d, "\pi"'] & \mathscr{D}(\Gamma, \Delta; A) \arrow[d, "\pi"] \\ \mathscr{M}(\pi\Gamma; p) \arrow[r, "\alpha[-/x]"'] & \mathscr{M}(\pi\Gamma, \psi; q) \end{tikzcd}$$

are pullback squares.

To check the pullback property for objects, suppose we have a $d \in \mathscr{D}(\Gamma, \Delta; A)$ and $\beta \in \mathscr{M}(\pi\Gamma; p)$ such that $\pi(d) = \alpha[\beta/x]$, i.e., a d of the form $d :: \Gamma, \Delta \vdash_{\alpha[\beta/x]} A$. We must produce a unique object $e :: (\Gamma \vdash_\beta \mathsf{U}_{x.\alpha}(\Delta \mid A))$ such that $\mathsf{UL}^*[e/x] \equiv d$.

We take as e the derivation $\mathsf{UR}(\Delta.d)$. We verify:

$$\begin{aligned}
\mathsf{UL}^*[e/x] &\equiv \mathsf{UL}^*[\mathsf{UR}(\Delta.d)/x] \\
&\equiv \mathsf{UL}^{x}_{\overrightarrow{x/x}}(1_\alpha, x/x, z.z)[\mathsf{UR}(\Delta.d)/x] \\
&\equiv (1_\alpha[1_\beta/x])_*(z[d/z]) \\
&\equiv (1_{\alpha[\beta/x]})_*(d) \\
&\equiv d
\end{aligned}$$

by the β-law for U. For uniqueness, suppose we have some other e' over β such that $\mathsf{UL}^*[e'/x] = d$. By the η-law we have

$$\begin{aligned}
e' &\equiv \mathsf{UR}(\Delta.\mathsf{UL}^*_x[e'/x]) \\
&\equiv \mathsf{UR}(\Delta.d) \\
&= e
\end{aligned}$$

So the pullback condition is satisfied for objects.

For the pullback condition for morphisms, suppose we have $\beta, \beta' \in \mathscr{M}(\pi\Gamma; p)$, a transformation $s :: \beta \Rightarrow \beta'$, and derivations $d :: \Gamma, \Delta \vdash_{\alpha[\beta/x]} A$ and $d' :: \Gamma, \Delta \vdash_{\alpha[\beta'/x]} A$ such that $(1_\alpha[s/x])_*(d') \equiv d$. This describes a morphism $T : d \Rightarrow d'$ in $\mathscr{D}(\Gamma, \Delta; A)$ that lies over $1_\alpha[s/x]$. As in the opfibration case, we must find a morphism S in $\mathscr{D}(\Gamma; \mathsf{U}_{x.\alpha}(\Delta \mid A))$ that lies over s, and that is sent to T by the functor $\alpha[-/x]$.

The morphism S must have domain $\mathsf{UR}(\Delta.d)$, codomain $\mathsf{UR}(\Delta.d')$, and underlying transformation s. This gives a well defined morphism $\mathsf{UR}(\Delta.d) \Rightarrow \mathsf{UR}(\Delta.d')$, because:

$$\begin{aligned}
s_*(\mathsf{UR}(\Delta.d')) &\equiv \mathsf{UR}(\Delta.\mathsf{UL}^*_x[s_*(\mathsf{UR}(\Delta.d'))/x]) \\
&\equiv \mathsf{UR}(\Delta.(1_\alpha)_*(\mathsf{UL}^*_x)[s_*(\mathsf{UR}(\Delta.d'))/x]) \\
&\equiv \mathsf{UR}(\Delta.(1_\alpha[s/x])_*(\mathsf{UL}^*_x[\mathsf{UR}(\Delta.d')/x])) \\
&\equiv \mathsf{UR}(\Delta.(1_\alpha[s/x])_*(d')) \\
&\equiv \mathsf{UR}(\Delta.d)
\end{aligned}$$

again by the η-law followed by the β-law.

Therefore the square is a pullback, so we conclude π is also an opfibration. □

THEOREM 6.8: SOUNDNESS/INTERPRETATION IN ANY BIFIBRATION. *Fix a bifibration $\pi : \mathscr{D} \to \mathscr{M}$. Then there is a function $[\![-]\!]$ from types A type_p to $[\![A]\!] \in \mathscr{D}_0$ with $\pi([\![A]\!]) = p$ and from derivations $x : A_1, \ldots, x_n : A_n \vdash_\alpha C$ to morphisms $d \in \mathscr{D}([\![A_1]\!], \ldots, [\![A_n]\!]) \to [\![C]\!]$, such that $\pi(d) = \alpha$.*

Proof. If π is a local discrete fibration, the 2-cells of $\mathscr{M}$ act on the fibers. Suppose $\alpha, \beta : \psi \to p$ and $s : \alpha \Rightarrow \beta$. We re-use the notation s_* for the induced function (of sets): $\mathscr{D}_\beta(\Gamma; A) \to \mathscr{D}_\alpha(\Gamma; A)$ that sends an object $d \in \mathscr{D}_\alpha(\Gamma; A)$ to the domain of the unique lift of s with codomain d.

The definition of an opfibration of 2-multicategories guarantees that, given a morphism in the mode category $\psi \vdash \alpha : q$ and a set of objects Δ that lies over ψ, there is an opcartesian morphism over α with domain Δ. For each α we choose one such lift and take the codomain of this morphism as our interpretation of $\mathsf{F}_\alpha(\Delta)$. Let us name this opcartesian lift $\zeta_{\alpha,\Delta} : \Delta \to \mathsf{F}_\alpha(\Delta)$. ζ corresponds to the axiomatic FR^*.

Similarly, the fibration structure ensures that, for every morphism $\psi, p \vdash \alpha : q$, context Δ over ψ and type A over q, there is cartesian morphism over α with codomain A, where the position in the domain over q as been filled by an object. We take this object as the interpretation of $\mathsf{U}_{x.\alpha}(\Delta \mid A)$. Let us name this cartesian lift $\xi_{\alpha,\Delta,A} : \Delta, \mathsf{U}_{x.\alpha}(\Delta \mid A) \to A$; it corresponds to the axiomatic UL^*.

We assume a given interpretation of each atomic proposition $[\![P\ \mathsf{type}_p]\!]$ as an object of $\mathscr{D}$ that lies over p.

The sequent calculus rules are then interpreted as follows (we elide the semantic brackets on objects):

- The identity derivation of a sequent $x :: \Gamma \vdash_x A$ is defined to be $[\![x]\!] = 1_A$.

- Given a derivation $d :: \Gamma \vdash_\beta A$ and transformation $s :: \beta' \Rightarrow \beta$, the respect-for-transformations derivation is interpreted as $[\![s_*(d)]\!] = s_*([\![d]\!])$.
- Given derivations $d_1 :: \Gamma, x : A, \Gamma' \vdash_\alpha B$ and $d_2 :: \Gamma, \Gamma' \vdash_\beta A$, cut is interpreted as $[\![d_1[d_2/x]]\!] = [\![d_1]\!] \circ_A [\![d_1]\!]$.
- For FL
$$\frac{\Gamma, \Gamma', \Delta \vdash_{\beta[\alpha/x]} C}{\Gamma, x : \mathsf{F}_\alpha(\Delta), \Gamma' \vdash_\beta M : C}\ \mathsf{FL}$$
the inductive hypothesis (after an exchange, which preserves the size of the derivations) gives a morphism $[\![d]\!] \in \mathscr{D}_{\beta[\alpha/x]}(\Gamma, \Delta, \Gamma'; C)$ and we must produce a morphism $\mathscr{D}_\beta(\Gamma, \mathsf{F}_\alpha(\Delta), \Gamma'; C)$. By the opcartesian-ness of $\zeta_{\alpha,\Delta}$, the following square is a pullback:
$$\begin{tikzcd}
\mathscr{D}(\Gamma, \mathsf{F}_\alpha(\Delta), \Gamma'; C) \arrow[r, "(-)\circ\zeta_{\alpha,\Delta}"] \arrow[d, "\pi"'] & \mathscr{D}(\Gamma, \Delta, \Gamma'; C) \arrow[d, "\pi"] \\
\mathscr{M}(\pi\Gamma, \pi\mathsf{F}_\alpha(\Delta), \pi\Gamma'; \pi C) \arrow[r, "(-)\circ\alpha"'] & \mathscr{M}(\pi\Gamma, \pi\Delta, \pi\Gamma'; \pi C)
\end{tikzcd}$$
We are given an object of the bottom left (β) and the top right ($[\![d]\!]$), with $\pi[\![d]\!] = \beta \circ_{\pi\mathsf{F}_\alpha(\Delta)} \alpha$. By Lemma 6.5, there is a unique object $[\![d]\!]^{\mathsf{F}}_{\alpha,\Delta} \in \mathscr{D}(\Gamma, \mathsf{F}_\alpha(\Delta), \Gamma'; C)$ so that $\pi([\![d]\!]^{\mathsf{F}}_{\alpha,\Delta}) = \beta$. We take this object to be our interpretation.
- For FR
$$\frac{s : \beta \Rightarrow \alpha[\gamma] \quad \Gamma \vdash_\gamma M : \Delta}{\Gamma \vdash_\beta \mathsf{F}_\alpha(\Delta)}\ \mathsf{FR}$$
where $\gamma = (\alpha_1, \ldots, \alpha_n)$ and $\Delta = (C_1, \ldots, C_n)$, the first premise is a 2-cell $s : \beta \Rightarrow \alpha \circ (\alpha_1, \ldots, \alpha_n)$, and the second is interpreted as a set of morphisms $[\![d_i]\!] \in \mathscr{D}_{\alpha_i}(\Gamma; C_i)$. We take the interpretation of the conclusion to be $s_*(\zeta_{\alpha,\Delta} \circ ([\![d_1]\!], \ldots, [\![d_n]\!]))$
- For UL
$$\frac{\begin{array}{l} x : \mathsf{U}_{x.\alpha}(\Delta \mid A) \in \Gamma \\ s : \beta \Rightarrow \beta'[\alpha[\gamma]/z] \\ \Gamma \vdash_\gamma M : \Delta \\ \Gamma, z{:}A \vdash_{\beta'} D : C \end{array}}{\Gamma \vdash_\beta C}\ \mathsf{UL}$$
let us again write $\gamma = (\gamma_1, \ldots, \gamma_n)$ and $\Delta = (C_1, \ldots, C_n)$, so that the interpretations of the premises are $[\![d_i]\!] \in \mathscr{D}_{\gamma_i}(\Gamma; C_i)$ and $[\![d]\!] \in \mathscr{D}_{\beta'}(\Gamma, A; C)$. Our interpretation is then $s_*([\![d]\!] \circ_A \xi_{\alpha,\Delta,A} \circ ([\![d_1]\!], \ldots, 1_{\mathsf{U}_\alpha(\Delta|A)}, \ldots, [\![d_n]\!]))$
- For UR
$$\frac{\Gamma, \Delta \vdash_{\alpha[\beta/x]} M : A}{\Gamma \vdash_\beta \mathsf{U}_{x.\alpha}(\Delta \mid A)}\ \mathsf{UR}$$
we are given a morphism $[\![d]\!] \in \mathscr{D}_{\alpha[\beta/x]}(\Gamma, \Delta; A)$ and must produce a morphism in $\mathscr{D}_\beta(\Gamma; \mathsf{U}_{x.\alpha}(\Delta \mid A))$. This time, by cartesian-ness of $\xi_{\alpha,\Delta,A}$, we have the pullback square
$$\begin{tikzcd}
\mathscr{D}(\Gamma; \mathsf{U}_{x.\alpha}(\Delta \mid A)) \arrow[r, "\xi_{\alpha,\Delta,A}\circ(-)"] \arrow[d, "\pi"'] & \mathscr{D}(\Gamma, \Delta; A) \arrow[d, "\pi"] \\
\mathscr{M}(\pi\Gamma; \pi\mathsf{U}_{x.\alpha}(\Delta \mid A)) \arrow[r, "\alpha\circ(-)"'] & \mathscr{M}(\pi\Gamma, \pi\Delta; \pi A)
\end{tikzcd}$$
Again we have objects β and $[\![d]\!]$ that agree in the bottom right, so an induced object $[\![d]\!]^{\mathsf{U}}_{\alpha,\Delta,A}$ in the top left which we take as our interpretation.

We now show that the above interpretation function respects the equational theory on derivations.

- The first set of equations all follow from properties of single-variable composition in a cartesian multicategory.
- The transport equations correspond to properties of discrete fibrations, where for each morphism in the base category there is a unique lift of that morphism with a chosen lift of the codomain. For the first, we must show $1_{\alpha*}(\llbracket d \rrbracket) \equiv \llbracket d \rrbracket$. The unique lift of 1_α with codomain $\llbracket d \rrbracket$ is simply $1_{\llbracket d \rrbracket}$, so has domain $\llbracket d \rrbracket$ as required.
- For $(s_1;s_2)_*(\llbracket d \rrbracket) \equiv s_{1*}(s_{2*}(\llbracket d \rrbracket))$, note that on the right-side, we lift s_2 with codomain $\llbracket d \rrbracket$, and then lift s_1 with codomain the domain of the first lift, so the two lifts can be composed. By functoriality of π, this composite must be the lift of $s_1;s_2$, so indeed the domains of the two lifts are equal.
- The equation $(s_2[s_1/x])_*(d_2[d_1/x]) \equiv s_{2*}(d_2)[s_{1*}(d_1)/x]$ holds for a similar reason: the horizontal composite of the two lifts must be the lift of the horizontal composite, by the fact that π preserves horizontal compositions and by uniqueness of lifts.
- *β-rule for* F:

$$\begin{aligned}
\llbracket \mathsf{FL}^{x_0}(d)[\mathsf{FR}(s,\vec{d_i/x_i})/x_0] \rrbracket &= \llbracket \mathsf{FL}^{x_0}(d) \rrbracket \circ_{F_\alpha(\Delta)} \llbracket \mathsf{FR}(s,\vec{d_i/x_i}) \rrbracket \\
&= \llbracket d \rrbracket^{\mathsf{F}}_{\alpha,\Delta} \circ_{F_\alpha(\Delta)} s_*(\zeta_{\alpha,\Delta} \circ (\llbracket d_1 \rrbracket, \ldots, \llbracket d_n \rrbracket)) \\
&= (1_\beta[s/x_0])_*((\llbracket d \rrbracket^{\mathsf{F}}_{\alpha,\Delta} \circ_{F_\alpha(\Delta)} \zeta_{\alpha,\Delta}) \circ (\llbracket d_1 \rrbracket, \ldots, \llbracket d_n \rrbracket)) \\
&= (1_\beta[s/x_0])_*(\llbracket d \rrbracket \circ (\llbracket d_1 \rrbracket, \ldots, \llbracket d_n \rrbracket)) \\
&= \llbracket (1_\beta[s/x_0])_*(d[\vec{d_i/x_i}]) \rrbracket
\end{aligned}$$

  where $\llbracket d \rrbracket^{\mathsf{F}}_{\alpha,\Delta}$ is the object induced by the pullback in the interpretation of FL. Here we have used that by the definition of $\llbracket d \rrbracket^{\mathsf{F}}_{\alpha,\Delta}$, the composite $\llbracket d \rrbracket^{\mathsf{F}}_{\alpha,\Delta} \circ_{F_\alpha(\Delta)} \zeta_{\alpha,\Delta}$ is equal to d.
- *η-rule for* F:

$$\llbracket \mathsf{FL}^x(\Delta.d[\mathsf{FR}^*/x]) \rrbracket = (\llbracket d \rrbracket \circ_{F_\alpha(\Delta)} \zeta_{\alpha,\Delta})^{\mathsf{F}}_{\alpha,\Delta}$$

  On the right, this is the unique object $f \in \mathscr{D}_\beta(\Gamma, \mathsf{F}_\alpha(\Delta), \Gamma'; C)$ such that $f \circ_{F_\alpha(\Delta)} \zeta_{\alpha,\Delta} = \llbracket d \rrbracket \circ_{F_\alpha(\Delta)} \zeta_{\alpha,\Delta}$. Clearly this must be $\llbracket d \rrbracket$ itself, so indeed $\llbracket \mathsf{FL}^x(\Delta.d[\mathsf{FR}^*/x]) \rrbracket = \llbracket d \rrbracket$.
- *β-rule for* U:

$$\begin{aligned}
\llbracket \mathsf{UL}^{x_0}(s,\vec{d_i}/x_i,z.d')[\mathsf{UR}(\Delta.d)/x_0] \rrbracket &= \llbracket \mathsf{UL}^{x_0}(s,\vec{d_i}/x_i,z.d') \rrbracket \circ_{\mathsf{U}_{x.\alpha}(\Delta|A)} \llbracket \mathsf{UR}(\Delta.d) \rrbracket \\
&= s_*(\llbracket d' \rrbracket \circ_A \xi_{\alpha,\Delta,A} \circ (\llbracket d_1 \rrbracket, \ldots, 1_{\mathsf{U}_\alpha(\Delta|A)}, \ldots, \llbracket d_n \rrbracket)) \circ_{\mathsf{U}_{x.\alpha}(\Delta|A)} \llbracket d \rrbracket^{\mathsf{U}}_{\alpha,\Delta,A} \\
&= (s[1_\alpha/x_0])_*(\llbracket d' \rrbracket \circ_A \xi_{\alpha,\Delta,A} \circ (\llbracket d_1 \rrbracket, \ldots, 1_{\mathsf{U}_\alpha(\Delta|A)}, \ldots, \llbracket d_n \rrbracket) \circ_{\mathsf{U}_{x.\alpha}(\Delta|A)} \llbracket d \rrbracket^{\mathsf{U}}_{\alpha,\Delta,A}) \\
&= (s[1_\alpha/x_0])_*(\llbracket d' \rrbracket \circ_A \xi_{\alpha,\Delta,A} \circ_{\mathsf{U}_{x.\alpha}(\Delta|A)} \llbracket d \rrbracket^{\mathsf{U}}_{\alpha,\Delta,A} \circ (\llbracket d_1 \rrbracket, \ldots, 1_\Gamma, \ldots, \llbracket d_n \rrbracket)) \\
&= (s[1_\alpha/x_0])_*(\llbracket d' \rrbracket \circ_A \llbracket d \rrbracket \circ (\llbracket d_1 \rrbracket, \ldots, 1_\Gamma, \ldots, \llbracket d_n \rrbracket)) \\
&= \llbracket (s[1_\alpha/x_0])_*(d'[(d[\vec{d_i}/x_i])/z]) \rrbracket
\end{aligned}$$

- *η-rule for* U:

$$\begin{aligned}
\llbracket \mathsf{UR}(\Delta.\mathsf{UL}^*_x[d/x]) \rrbracket &= \llbracket \mathsf{UL}^*_x[d/x] \rrbracket^{\mathsf{U}}_{\alpha,\Delta,A} \\
&= (\xi_{\alpha,\Delta,A} \circ_{\mathsf{U}_\alpha(\Delta|A)} \llbracket d \rrbracket)^{\mathsf{U}}_{\alpha,\Delta,A}
\end{aligned}$$

  Similarly to the η-rule for F, this is the unique object $f \in \mathscr{D}_\beta(\Gamma; \mathsf{U}_\alpha(\Delta \mid A))$ such that $f \circ_{\mathsf{U}_\alpha(\Delta|A)} \xi_{\alpha,\Delta,A} = \llbracket d \rrbracket \circ_{\mathsf{U}_\alpha(\Delta|A)} \xi_{\alpha,\Delta,A}$. Again this must be $\llbracket d \rrbracket$ itself.

$\square$

7 Logical Adequacy

Suppose we are representing some object logic, like the examples from Section 3, in our framework. In general, for a specific mode theory, the framework will have more types than the object logic. For example, if we represent a logic with a binary product by a product in the mode theory, then the framework will have not only $\mathsf{F}_{x\otimes y}(x:A,y:B)$, but also a primitive triple product $\mathsf{F}_{x\otimes y\otimes z}(x:A,y:B,z:C)$, and so on. If we represent a modal logic with a monad by its adjoint decomposition, then the framework will have not only the UF composite, but also the U and F types separately. Thus, in general we will define a translation from object-logic sequents J to framework sequents J^*, in such a way that J is provable in the object logic iff J^* is provable in the framework. No claims are made about framework derivations of sequents that are not in the image of the translation. We call this *logical adequacy*, because it says that entailment in the object logic is soundly and completely represented by entailment in the framework. We plan to consider stronger adequacy theorems, which extend this logical correspondence to an isomorphism on equivalence-classes of proofs.

We will often use the following lemma:

LEMMA 7.1: 0-USE STRENGTHING. *We say that a formula $\mathsf{F}_\alpha(\Delta)$ and $\mathsf{U}_{c.\alpha}(\Delta \mid A)$ is relevant if every variable from Δ (and c for U) occurs at least once in α.*

Suppose the mode theory has the property that for all x, α, β, if $\alpha \Rightarrow \beta$ and $x\#\alpha$ then $x\#\beta$ (in particular, equations must have the same variables on both sides). Suppose additionally a sequent $\Gamma \vdash_\alpha A$ such that every F/U subformula of Γ, A is relevant.

Then if $d :: \Gamma \vdash_\alpha A$ and $\vec{x}$ are variables such that $\vec{x}\#\alpha$ then there is a $d' :: \Gamma - \vec{x} \vdash_\alpha A$ and $size(d') \leq size(d)$.

Proof. The proof is by induction on d. In all cases, the assumption that every formula is relevant is preserved for all premises of a rule by the subformula property.

- In the case for a variable $x : P$ with transformation $\beta \Rightarrow x$, we need to show that the variable x being used is not one of the ones being strengthened away. But if x were in $\vec{x}$, then we would have $x\#\beta$, and therefore by the assumption, $x\#x$, a contradiction. Therefore $x \in \Gamma - \vec{x}$, and we can reapply the variable rule, which has the same size.

- In the case for FR, we have $\vec{x}\#\beta$, so by the reduction condition, $\vec{x}\#\alpha[\gamma]$. By the relevance condition, all variables that γ substitutes occur in α, which means each component of γ occurs in $\alpha[\gamma]$. Therefore $\vec{x}\#\gamma$. We can use the inductive hypothesis to obtain a no-bigger derivation of $\Gamma - \vec{x} \vdash_\gamma \Delta$, and then reapply FR.

- In the case for FL, we distinguish cases on whether $x \in \vec{x}$ or not.

  If it is, then this is an elimination on a 0-use variable that we would like to drop. Because $x\#\beta$, $\beta[\alpha/x] = \beta$, and note that $\Delta\#\beta$ because it occurs only in α. Thus, if we appeal to the inductive hypothesis on the premise with $\vec{x} - x, \Delta$, we get $(\Gamma, \Gamma', \Delta) - (\vec{x} - x, \Delta) \vdash_\beta C$, i.e. $\Gamma, x : \mathsf{F}_\alpha(\Delta), \Gamma', -\vec{x} \vdash_\beta C$ as desired. That is, we recursively drop all variables that came from the elimination, in addition to any others that we were trying to drop besides $\vec{x}$.

  If it is not, then $\vec{x}\#\beta$ and $\vec{x}\#\alpha$ (by scoping) implies $\vec{x}\#\beta[\alpha/x]$, so by the inductive hypothesis we get a no-bigger derivation of $\Gamma, \Gamma' - \vec{x}, \Delta \vdash_{\beta[\alpha/x]} C$, and we can reapply FL(because the principal variable x of the left rule is not removed).

- In the case for UR, we have $\vec{x}$ a collection of variables bound in Γ, so $\vec{x}\#\alpha$ (since the domain of α is not Γ) in addition to $\vec{x}\#\beta$. Thus $\vec{x}\#\alpha[\beta/x]$, so the inductive hypothesis gives a no-bigger derivation of $\Gamma - \vec{x}, \Delta \vdash_{\alpha[\beta/x]} A$, and we can reapply the rule.

- In the case for UL, we distinguish cases on whether z occurs in β'.

  If $z\#\beta'$, then this UL is generating a 0-use assumption z, so we can remove it and the UL along with $\vec{x}$. That is, we appeal to the inductive hypothesis on the continuation with $\vec{x}, z$, which gives $\Gamma, z : A - (\vec{x}, z) \vdash_{\beta'} C$, i.e. $\Gamma - \vec{x} \vdash_{\beta'} C$. We also have $\beta \Rightarrow \beta'$ because the $[\alpha[\gamma]/z]$ substitution cancels. So we have $\Gamma - \vec{x} \vdash_\beta C$ by Lemma 4.1.

  If z occurs in β', we further distinguish cases on whether $x \in \vec{x}$ or not.

  If it is not, then we know $\vec{x}\#\beta$, so pushing this along the transformation gives $\vec{x}\#\beta'[\alpha[\gamma]/z]$. Thus $\vec{x}\#\beta'$ (note that z cannot be in $\vec{x}$ because it is bound only in the continuation), and because z occurs in β', $\alpha[\gamma/z]$ occurs in the substitution, so $\vec{x}\#\alpha[\gamma]$. By the relevance assumption, each term in γ also occurs after the substitution, so $\vec{x}\#\gamma$ as well. Thus, by the inductive hypotheses we get no-bigger derivations of $\Gamma - \vec{x} \vdash_{\gamma} \Delta$ and $\Gamma - \vec{x}, z : A \vdash_{\beta'} C$, and the principal x survives in $\Gamma - \vec{x}$, so we can reapply UL.

Finally, if $x \in \vec{x}$ and $z \in \beta'$, then we have $x\#\beta$, so $x\#\beta'[\alpha[\gamma]/z]$ by moving along the transformation, and then $x\#\alpha[\gamma]$ by the fact that z occurs. However, this contradicts the relevance assumption on $\mathsf{U}_{x.\alpha}(\Delta \mid A)$, which says that x occurs in α.

□

LEMMA 7.2. *Suppose each axiom $c : \alpha \Rightarrow \beta$ has the property that $x\#\alpha$ implies $x\#\beta$. Then for any derivation of $\alpha \Rightarrow \beta$, $x\#\alpha$ implies $x\#\beta$.*

Proof. The cases for reflexivity is immediate, and the case for axioms is assumed. In the case for transitivity $\alpha \Rightarrow \beta_1 \Rightarrow \beta_2$, we get $\vec{x}\#\beta_1$ by the first inductive hypothesis and then $\vec{x}\#\beta_2$ by the second. In the case for congruence, we have $\vec{x}\#\alpha[\beta/y]$. This means that either $\vec{x}\#\alpha$ and $\vec{x}\#\beta$, or $\vec{x}\#\alpha$ and $y\#\alpha$ (in which case $\vec{x}$ might occur in β). In the first case, we get $\vec{x}\#\alpha'$ and $\vec{x}\#\beta'$ by the inductive hypotheses, so $\vec{x}\#\alpha'[\beta'/y]$. In the second, we get $\vec{x}, y\#\alpha'$ by the inductive hypothesis, so $\alpha'[\beta'/y] = \alpha'$, and $\vec{x}\#\alpha'$. □

7.1 Ordered Logic (Product Only)

As a first example of an adequacy proof, we consider ordered logic with only $A \odot B$:

$$\frac{}{A \vdash^{\circ} A} \qquad \frac{\Gamma, A, \Gamma' \vdash^{\circ} C \quad \Delta \vdash^{\circ} A}{\Gamma, \Delta, \Gamma' \vdash^{\circ} C} \qquad \frac{\Gamma, A, B, \Gamma' \vdash^{\circ} C}{\Gamma, A \odot B, \Gamma' \vdash^{\circ} C} \qquad \frac{\Gamma \vdash^{\circ} A \quad \Delta \vdash^{\circ} B}{\Gamma, \Delta \vdash^{\circ} A \odot B}$$

We use a mode theory with a monoid $(\odot, 1)$, so the only transformation axioms are equality axioms for associativity and unit.

The type translation is given by $P^* := P$ and $(A \odot B)^* := \mathsf{F}_{x \odot y}(x : A^*, y : B^*)$. A context $(x_1 : A_1, \ldots, x_n : A_n)^* := x_1 : A_1^*, \ldots, x_n : A_n^*$. Writing $\overline{x_1 : A_1, \ldots, x_n : A_n} := x_1 \odot \ldots \odot x_n$, a sequent $\Gamma \vdash^{\circ} A$ is translated to $\Gamma^* \vdash_{\overline{\Gamma}} A^*$.

We use the following properties of the mode theory:

- If $\overline{\Gamma^*} \equiv x$ then Γ is $x : Q$ for some Q.
- If $\overline{\Gamma} \equiv \alpha_1 \odot \alpha_2$, then there exist Γ_1, Γ_2 such that $\Gamma = \Gamma_1, \Gamma_2$ and $\overline{\Gamma_1} \equiv \alpha_1$ and $\overline{\Gamma_2} \equiv \alpha_2$.
- A^* and Γ^* are relevant propositions, and the monoid axioms preserve variables, so by Lemma 7.1 we can strengthen away any variables that are not in the context descriptor.

Using these definitions, we have

THEOREM 7.3: LOGICAL ADEQUACY FOR ORDERED PRODUCTS. $\Gamma \vdash^{\circ} A$ *iff* $\Gamma^* \vdash_{\overline{\Gamma}} A^*$

Proof. The forward direction is by induction on $\Gamma \vdash^{\circ} A$, where the inference rules for $\odot$ are derived as follows:

$$\dfrac{\dfrac{\Gamma^*, x : A, y : B, \Gamma'^* \vdash_{\overline{\Gamma} \odot x \odot y \odot \overline{\Gamma'}} C}{\Gamma^*, \Gamma'^*, x : A, y : B \vdash_{\overline{\Gamma} \odot x \odot y \odot \overline{\Gamma'}} C} \; \textit{Lemma } 4.3}{\Gamma^*, z : \mathsf{F}_{x \otimes y}(x : A^*, y : B^*), \Gamma'^* \vdash_{\overline{\Gamma} \odot z \odot \overline{\Gamma'}} C} \; \mathsf{FL}$$

$$\dfrac{\overline{\Gamma} \odot \overline{\Delta} \Rightarrow (x \odot y)[\overline{\Gamma}/x, \overline{\Delta}/y] \quad \dfrac{\Gamma^* \vdash_{\overline{\Gamma}} A}{\Gamma^*, \Delta^* \vdash_{\overline{\Gamma}} A} \; \textit{Lemma } 4.2 \quad \dfrac{\Delta^* \vdash_{\overline{\Delta}} B}{\Gamma^*, \Delta^* \vdash_{\overline{\Gamma}} A} \; \textit{Lemma } 4.2}{\Gamma^*, \Delta^* \vdash_{\overline{\Gamma} \odot \overline{\Delta}} \mathsf{F}_{x \odot y}(x : A, y : B)}$$

The backward direction is also by induction on the given derivation:

- For identity

$$\frac{\overline{\Gamma^*} \Rightarrow x \quad x : P \in \Gamma^*}{\Gamma^* \vdash_{\overline{\Gamma}} P}$$

Because the only structural transformation axioms are equalities for associativity and unit, we have $\overline{\Gamma^*} \equiv x$, which in turn implies that Γ is $x : Q$ for some Q (because if Γ is empty, does not contain x, or contains anything else, $\overline{\Gamma}$ will not equal x). By definition, this implies $Q = P$, so Γ is $x : P$. Therefore the identity rule applies.

- For FR, because the only type that encodes to $\mathsf{F}_\alpha(\Delta)$ is $A \odot B$, and in this case we have $(A_1 \odot A_2)^* = \mathsf{F}_{x \odot y}(x : A_1, y : A_2)$, we have

$$\frac{\overline{\Gamma} \equiv \alpha_1 \odot \alpha_2 \quad \Gamma^* \vdash_{\alpha_1} A_1^* \quad \Gamma^* \vdash_{\alpha_2} A_2^*}{\Gamma^* \vdash_{\overline{\Gamma}} \mathsf{F}_{x \odot y}(x : A_1^*, y : A_2^*)}$$

By properties of the mode theory, $\Gamma = \Gamma_1, \Gamma_2$ with $\overline{\Gamma_i} \equiv \alpha_i$, so we have derivations of $\Gamma^* \vdash_{\overline{\Gamma_i}} A_i^*$. Because 0-use strengthening applies, we can strengthen these to $\Gamma_i^* \vdash_{\overline{\Gamma_i}} A_i^*$. Then the inductive hypothesis gives $\Gamma_i \vdash^{\mathsf{o}} A_i$, so applying the $\odot$ right rule gives the result.

- For FL, because the only type encoding to F is $A \odot B$, we have

$$\frac{\Gamma^*, \Gamma'^*, x : A^*, y : B^* \vdash_{\overline{\Gamma} \odot (x \odot y) \odot \overline{\Gamma'}} C^*}{\Gamma^*, z : \mathsf{F}_{x \odot y}(x : A^*, y : B^*), \Gamma'^* \vdash_{\overline{\Gamma} \odot z \odot \overline{\Gamma'}} C^*}$$

By exchange (Lemma 4.3), we have a no-bigger derivation of $\Gamma^*, x : A^*, y : B^*, \Gamma'^* \vdash_{\overline{\Gamma} \odot (x \odot y) \odot \overline{\Gamma'}} C^*$ so applying the IH gives $\Gamma, x : A, y : B, \Gamma' \vdash^{\mathsf{o}} C$, and then $\odot$-left gives the result.

□

7.2 Affine Logic

Consider the following rules for affine logic, where the context is represented by a list of assumptions labeled with variables, and $\Gamma \rightrightarrows \Delta_1, \Delta_2$ means interleaving Δ_1 and Δ_2 in some order equals Γ.

$$\frac{P \in \Gamma}{\Gamma \vdash^{\mathsf{a}} P} \qquad \frac{\Gamma, \Gamma', x : A, y : B \vdash^{\mathsf{a}} C}{\Gamma, z : A \otimes B, \Gamma' \vdash^{\mathsf{a}} C} \qquad \frac{\Gamma \rightrightarrows \Delta_1; \Delta_2 \quad \Delta_1 \vdash^{\mathsf{a}} A \quad \Delta_2 \vdash^{\mathsf{a}} B}{\Gamma \vdash^{\mathsf{a}} A \otimes B}$$

$$\frac{\Gamma, x : A \vdash^{\mathsf{a}} B}{\Gamma \vdash^{\mathsf{a}} A \multimap B} \qquad \frac{\Gamma \rightrightarrows \Delta_1; \Delta_2; (f : A \multimap B) \quad \Delta_1 \vdash^{\mathsf{a}} A \quad \Delta_2, z : B \vdash^{\mathsf{a}} C}{\Gamma \vdash^{\mathsf{a}} C}$$

Weakening and exchange are admissible for these rules.

Using the mode theory from Section 3.5, we translate the propositions and contexts of adjoint logic as follows:

$$\begin{array}{rcl} P^* & = & P \\ (A \otimes B)^* & = & \mathsf{F}_{x \otimes y}(x : A^*, y : B^*) \\ (A \multimap B)^* & = & \mathsf{U}_{c.c \otimes x}(x : A^* \mid B^*) \\ \\ .^* & = & . \\ (\Gamma, x : A)^* & = & \Gamma^*, x : A^* \end{array}$$

We also define a function that collects the variables from Γ as a context descriptor:

$$\begin{array}{rcl} \overline{\cdot} & = & 1 \\ \overline{(\Gamma, x : A)} & = & \overline{\Gamma} \otimes x \end{array}$$

Overall, we have

THEOREM 7.4: LOGICAL ADEQUACY FOR AFFINE PRODUCTS AND FUNCTIONS.
$\Gamma \vdash^{\mathsf{a}} A$ *iff* $\Gamma^* \vdash_{\overline{\Gamma}} A^*$

Proof. The forward direction is by induction on $\Gamma \vdash^{\mathsf{a}} A$:

- For the hypothesis rule, we need to show $\Gamma^* \vdash_{\overline{\Gamma}} P$. Because x is in Γ, we can prove by induction on Γ that $x : P$ is in Γ^* and that $\overline{\Gamma} \equiv \alpha \otimes x$. Thus, the weakening transformation gives $\alpha \otimes x \Rightarrow 1 \otimes x \equiv x$. Therefore we can derive

$$\frac{x : P \in \Gamma \quad \overline{\Gamma} \Rightarrow x}{\Gamma^* \vdash_{\overline{\Gamma}} P}\ \mathsf{v}$$

- For $\otimes$-left, the inductive hypothesis gives $\Gamma^*, \Gamma'^*, x : A^*, y : B^* \vdash_{\overline{\Gamma} \otimes \overline{\Gamma'} \otimes x \otimes y} C^*$ and we want $\Gamma^*, z : \mathsf{F}_{x \otimes y}(x : A^*, y : B^*), \Gamma'^* \vdash_{\overline{\Gamma} \otimes z \otimes \overline{\Gamma'}} C^*$. This is FL on the inductive hypothesis, with using associativity and commutativity of $\otimes$ from the mode theory to move $x \otimes y$ to the end.

- For $\otimes$-right, we have $\Delta_1^* \vdash_{\overline{\Delta_1}} A^*$ and $\Delta_2^* \vdash_{\overline{\Delta_2}} B^*$ by the inductive hypotheses, which we can weaken to $\Delta_1^*, \Delta_2^* \vdash_{\overline{\Delta_1}} A^*$ and $\Delta_1^*, \Delta_2^* \vdash_{\overline{\Delta_2}} B^*$, and then exchange to $\Gamma^* \vdash_{\overline{\Delta_1}} A^*$ and $\Gamma^* \vdash_{\overline{\Delta_2}} B^*$ (using a lemma that when $\Gamma \rightrightarrows \Delta_1, \Delta_2$, Γ and (Δ_1, Δ_2) differ only in order, and that reordering is preserved by the mapped application of $*$). To apply FR to derive $\Gamma^* \vdash_{\overline{\Gamma}} \mathsf{F}_{x \otimes y}(x : A, y : B)$, it thus suffices to show that $\overline{\Gamma} \Rightarrow \overline{\Delta_1} \otimes \overline{\Delta_2}$. In fact they are $\equiv$, which we can prove by induction on $\Gamma \rightrightarrows \Delta_1, \Delta_2$ using the commutative monoid laws.

- For $\multimap$-right, we have $\Gamma^*, x : A^* \vdash_{\overline{\Gamma} \otimes x} B^*$ by the inductive hypothesis, which is exactly the premise of using UR to prove $\Gamma^* \vdash_{\overline{\Gamma}} \mathsf{U}_{c.c \otimes x}(x : A^* \mid B^*)$.

- For $\multimap$-left, we have $\Delta_1^* \vdash_{\overline{\Delta_1}} A$ and $\Delta_2^*, z : B^* \vdash_{\overline{\Delta_2} \otimes z} C^*$ by the inductive hypothesis, and by similar reasoning to the $\otimes R$ case, we can weaken and exchange to $\Gamma^*, \Gamma'^* \vdash_{\overline{\Delta_1}} A$ and $\Gamma^*, \Gamma'^*, z : B^* \vdash_{\overline{\Delta_2} \otimes z} C^*$ and then finally weaken to $\Gamma^*, f : (A \multimap B)^*, \Gamma'^* \vdash_{\overline{\Delta_1}} A$ and $\Gamma^*, f : (A \multimap B)^*, \Gamma'^*, z : B^* \vdash_{\overline{\Delta_2} \otimes z} C^*$. Thus, we only need to show the transformation premise of

$$\frac{\begin{array}{l} \overline{\Gamma} \otimes f \otimes \overline{\Gamma'} \Rightarrow (\overline{\Delta_2} \otimes z)[f \otimes \overline{\Delta_1}/z] \\ \Gamma^*, f : (A \multimap B)^*, \Gamma'^* \vdash_{\overline{\Delta_1}} A \\ \Gamma^*, f : (A \multimap B)^*, \Gamma'^*, z : B^* \vdash_{\overline{\Delta_2} \otimes z} C^* \end{array}}{\Gamma^*, f : (A \multimap B)^*, \Gamma'^* \vdash_{\overline{\Gamma} \otimes f \otimes \overline{\Gamma'}} C^*}\ \mathsf{UL}$$

  In fact $\overline{\Gamma} \otimes f \otimes \overline{\Gamma'} \equiv (\overline{\Delta_2} \otimes f \otimes \overline{\Delta_1})$, which again follows from $\Gamma \rightrightarrows \Delta_1, \Delta_2$, using associativity and commutativity of $\otimes$.

In terms of structural property placement, observe that the above proof uses only identity transformations on FR and UL, and uses the w axiom only at the leaves.

We need the following facts about the mode theory.

LEMMA 7.5. *If $\alpha \Rightarrow \beta$ then $\alpha \equiv \beta \otimes \beta'$ for some β'.*

Proof. In the case for weakening $\alpha \Rightarrow 1$ (the only axiom), take $\beta' = \beta$. In the case for reflexivity take $\beta' = 1$. In the case for transitivity, we have $\alpha \Rightarrow \beta_1 \Rightarrow \beta_2$. By the second inductive hypothesis, we have $\beta_1 \equiv \beta_2 \otimes \beta_2'$, and by the first we have $\alpha \equiv \beta_1 \otimes \beta_1'$, so $\alpha \equiv \beta_2 \otimes (\beta_2' \otimes \beta_1')$.

In the case for congruence, we have $\alpha_1[\alpha_2/x] \Rightarrow \beta_1[\beta_2/x]$ and the inductive hypotheses give $\alpha_1 \equiv \beta_1 \otimes \beta_1'$ and $\alpha_2 \equiv \beta_2 \otimes \beta_2'$. Thus, $\alpha_1[\alpha_2/x] \equiv \beta_1[\beta_2 \otimes \beta_2'/x] \otimes \beta_1'[\beta_2 \otimes \beta_2'/x]$, and then the right-hand side equals $\beta_1[\beta_2/x] \otimes \beta_2'^n \otimes \beta_1'[\beta_2 \otimes \beta_2'/x]$ for some n. This is because, for this mode theory, we can rewrite any α as $\alpha' \otimes (x \otimes \ldots \otimes x)$ where α' does not contain x (because any context descriptor is equal to a "polynomial" giving the multiplicity of each variable), so in general we have $\alpha[\beta_1 \otimes \beta_2/x] \equiv \alpha' \otimes (\beta_1 \otimes \beta_2)^n \equiv \alpha' \otimes \beta_1^n \otimes \beta_2^n \equiv \alpha[\beta_1/x] \otimes \beta_2^n$. ☐

LEMMA 7.6. *If $\overline{\Gamma} \equiv \alpha_1 \otimes \alpha_2$, then $\Gamma \rightrightarrows \Gamma_1, \Gamma_2$ with $\overline{\Gamma_1} \equiv \alpha_1$ and $\overline{\Gamma_2} \equiv \alpha_2$.*

Proof. We define the splitting $\Gamma \rightrightarrows \Gamma_1, \Gamma_2$ adding each variable in Γ to Γ_1 if it occurs in α_1, or Γ_2 if it occurs in α_2 and not α_1 (occurrence respects $\equiv$). Because $\equiv$ consists of associativity, commutativity, and unit, α_1 and α_2 have no duplicates and every variable from $\overline{\Gamma}$ occurs exactly once in one or the other, which gives $\overline{\Gamma_1} \equiv \alpha_1$ and $\overline{\Gamma_2} \equiv \alpha_1$. ☐

LEMMA 7.7. *If $\overline{\Gamma} \Rightarrow \alpha$, then there is a Γ' such that $\Gamma \geq \Gamma'$ and $\alpha \equiv \overline{\Gamma'}$.*

Proof. By Lemma 7.5, $\overline{\Gamma} \equiv \alpha \otimes \beta$ for some β. By Lemma 7.6, this means $\Gamma \rightrightarrows \Gamma_1, \Gamma_2$ with $\overline{\Gamma_1} \equiv \alpha$. Then the fact that $\Gamma \rightrightarrows \Gamma_1, \Gamma_2$ implies $\Gamma \geq \Gamma_1$ gives the result. ☐

LEMMA 7.8. *If ${\Gamma_0}^* \vdash_\alpha A^*$ and $\vec{x} \# \alpha$ then there is a no-bigger derivation of $\Gamma^* - \vec{x} \vdash_\alpha A^*$*

Proof. We will use Lemma 7.1. First, no variables are free in the range of weakening, so Lemma 7.2 gives that $\alpha \Rightarrow \beta$ and $x\#\alpha$ imply $x\#\beta$. Second, we prove by induction that every subformula of Γ^* and A^* is relevant, because the only context descriptors used are $\mathsf{F}_{x\otimes y}(x : A, y : B)$ and $\mathsf{U}_{c.c\otimes x}(x : A \mid B)$. ☐

We now prove that if $\Gamma^* \vdash_{\overline{\Gamma}} A^*$ then $\Gamma \vdash^{\mathsf{a}} A$. The proof is by induction on the size of the assumption, because we will sometimes use Lemma 7.8 before appealing to the inductive hypothesis.

- In the case for the assumption rule, we have

  $$\frac{x : P \in \Gamma^* \quad \overline{\Gamma} \Rightarrow x}{\Gamma^* \vdash_{\overline{\Gamma}} P}$$

  Since $x : P \in \Gamma^*$, $x : P \in \Gamma$, and we can apply the affine logic rule.

- In the case where UR was used to derive $\Gamma^* \vdash_{\overline{\Gamma}} A^*$, A must be $A_1 \multimap A_2$ (because no other types encode to U), and the premise is $\Gamma^*, x : A_1^* \vdash_{\overline{\Gamma}\otimes x} A_2^*$. The inductive hypothesis gives $\Gamma, x : A_1 \vdash^{\mathsf{a}} A_2$, and we can apply $\multimap$-right.

- In the case where FR was used, the conclusion must be $(A_1 \otimes A_2)$, and we have $\overline{\Gamma} \Rightarrow (\alpha_1 \otimes \alpha_2)$ with $\Gamma^* \vdash_{\alpha_1} A_1^*$ and $\Gamma^* \vdash_{\alpha_2} A_1^*$. By Lemma 7.7, this means there is a $\Gamma \geq \Gamma'$ with $\overline{\Gamma'} \equiv (\alpha_1 \otimes \alpha_2)$. By Lemma 7.6, we have $\Gamma' \rightrightarrows \Gamma_1, \Gamma_2$ with $\overline{\Gamma_1} \equiv \alpha_1$ and $\overline{\Gamma_2} \equiv \alpha_2$. So the premises are $\Gamma^* \vdash_{\overline{\Gamma_1}} A_1^*$ and $\Gamma^* \vdash_{\overline{\Gamma_2}} A_2^*$. By Lemma 7.8, we can modify the premises to no-bigger derivations of $\Gamma_1^* \vdash_{\overline{\Gamma_1}} A_1^*$ and $\Gamma_2^* \vdash_{\overline{\Gamma_2}} A_2^*$. Thus, by the inductive hypotheses we get $\Gamma_1 \vdash^{\mathsf{a}} A_1$ and $\Gamma_2 \vdash^{\mathsf{a}} A_2$, so $\Gamma' \vdash^{\mathsf{a}} A_1 \otimes A_2$. Then weakening and exchange on $\Gamma \geq \Gamma'$ gives the result.

- In the case where FL was used, the formula under elimination must be the translation of $z : (A_1 \otimes A_2) \in \Gamma$. The premise is $\Gamma^* - z, x : A_1^*, y : A_2^* \vdash_{\overline{\Gamma}[(x\otimes y)/z]} C^*$, and we want $\Gamma \vdash^{\mathsf{a}} C$. Since $\overline{\Gamma}$ has exactly one occurrence of z, $\overline{\Gamma}[(x \otimes y)/z] \equiv (\overline{\Gamma} - z) \otimes x \otimes y$, so by the inductive hypothesis we get $\Gamma - z, x : A_1, y : A_2 \vdash^{\mathsf{a}} C$ by the inductive hypothesis, and can apply $\otimes$-left.

- In the case for UL, the assumption f that is eliminated must be the translation of $f : (A_1 \multimap A_2) \in \Gamma$, so the premises are $\overline{\Gamma} \Rightarrow \beta'[f \otimes \alpha/z]$ with $\Gamma^* \vdash_\alpha A_1^*$ and $\Gamma^*, z : A_2^* \vdash_{\beta'} C$.

  By Lemma 7.7, there is a Γ' with $\Gamma \geq \Gamma'$ and $\overline{\Gamma'} \equiv \beta'[f \otimes \alpha/z]$. Since $\overline{\Gamma'}$ has no duplicates, z occurs at most once in β' (or else f would occur more than once in the substitution).

  If z occurs once in β', then because all context descriptors are products of variables, we can commute it to the end, writing $\beta' \equiv \beta'' \otimes z$, so $\overline{\Gamma'} \equiv \beta'' \otimes f \otimes \alpha$. Since f is in $\overline{\Gamma'}$, f must be declared with some type in Γ', and since $\Gamma \geq \Gamma'$ and $f : A_1 \multimap A_2 \in \Gamma$, we must have $f : A_1 \multimap A_2 \in \Gamma'$. So by Lemma 7.6, we can split $\Gamma' \rightrightarrows \Delta_1; \Delta_2; f : A_1 \multimap A_2$ where $\overline{\Delta_2} \equiv \beta''$ and $\overline{\Delta_1} \equiv \alpha$. Using these equalities, the premises derive $\Gamma^* \vdash_{\overline{\Delta_1}} A_1^*$ and $\Gamma^*, z : A_2^* \vdash_{\overline{\Delta_2}\otimes z} C$, so by Lemma 7.8 we can strengthen to no-bigger derivations of $\Delta_1^* \vdash_{\overline{\Delta_1}} A_1^*$ (removing $\Gamma - \Delta_1$) and $\Delta_2*, z : A_2^* \vdash_{\overline{\Delta_2}\otimes z} C$ (removing $\Gamma - \Delta_2$). Then the inductive hypotheses give $\Delta_1 \vdash_{A_1}$ and $\Delta_2, z : A_2 \vdash_C$, so we have the premises to use $\multimap$-left to conclude $\Gamma' \vdash_C$. Finally, we weaken/exchange with $\Gamma \geq \Gamma'$.

  If z occurs 0 times in β' (that is, we did a UL that introduced a 0-use variable in the continuation), then we have premises $\Gamma^*, z : A_2^* \vdash_{\beta'} C$ and $\overline{\Gamma} \Rightarrow \beta'$ (the substitution cancels). By Lemma 7.7, we get $\Gamma \geq \Gamma'$ with $\overline{\Gamma'} \equiv \beta'$. By Lemma 7.8, we can remove z and anything in Γ but not in Γ' to get a no-bigger derivation of $\Gamma'^* \vdash_{\overline{\Gamma'}} C$. Then the inductive hypothesis on this premise gives $\Gamma' \vdash^{\mathsf{a}} C$, and weakening/exchanging with $\Gamma \geq \Gamma'$ gives the result.

$\square$

Inspecting this proof, we can see that the translation from a "native" sequent proof in affine logic to our framework and back is the identity on cut-free derivations. The other round-trip is not the identity, because the framework allows two things that the native sequent calculus does not. First, the framework allows weakening at the non-invertible rules, rather than pushing it to the leaves. For example, we have the following two derivations of $P, Q, R \vdash P \otimes R$.

$$\dfrac{x \otimes y \otimes z \Rightarrow (x' \otimes y')[(x \otimes y)/x', z/y'] \quad \dfrac{(x \otimes y) \Rightarrow x}{x : P, y : Q, z : R \vdash_{x \otimes y} P}\,\mathsf{v} \quad \dfrac{z \Rightarrow z}{x : P, y : Q, z : R \vdash_{z} R}\,\mathsf{v}}{x : P, y : Q, z : R \vdash_{x \otimes y \otimes z} \mathsf{F}_{x' \otimes z'}(x' : P, z' : R)}\,\mathsf{FR}$$

$$\dfrac{x \otimes y \otimes z \Rightarrow (x' \otimes y')[x/x', z/y'] \quad \dfrac{x \Rightarrow x}{x : P, y : Q, z : R \vdash_{x} P}\,\mathsf{v} \quad \dfrac{z \Rightarrow z}{x : P, y : Q, z : R \vdash_{z} R}\,\mathsf{v}}{x : P, y : Q, z : R \vdash_{x \otimes y \otimes z} \mathsf{F}_{x' \otimes z'}(x' : P, z' : R)}\,\mathsf{FR}$$

The second is that a derivation may perform a left rule on a 0-linear (in the sense of Section 3.4) variable, i.e. one that does not occur in the context descriptor. Such variables arise because UL "removes a variable from the context" by marking it as 0-use, not by actually removing it. For this mode theory, these left rules produce only other 0-use variables, which ultimately cannot be used, and were strengthened away by Lemma 7.1.

The equational theory of derivations (Section 5) handles both of these issues, so we expect that the framework-native-framework composite of adequacy produces a derivation that is equal in this equational theory.

7.3 n-use Variables

Consider the rules and mode theory from Section 3.4. We use the following normal form theorem for the linear logic mode (commutative monoid) mode theory, which says that any mode morphism can be written as a "polynomial" of its variables:

LEMMA 7.9. *If $x_1 : \mathsf{l}, \ldots, x_n : \mathsf{l} \vdash \alpha : \mathsf{l}$ then there exist unique $k_1, \ldots, k_n$ such that $\alpha \equiv x_1^{k_1} \otimes \ldots \otimes x_n^{k_n}$.*

THEOREM 7.10: LOGICAL ADEQUACY FOR n-USE FUNCTIONS.
$x_1 :^{k_1} A_1, \ldots, x_n :^{k_n} A_n \vdash C$ *iff* $x_1 : A_1^*, \ldots, x_n : A_n^* \vdash_{x_1^{k_1} \otimes \ldots \otimes x_n^{k_n}} C^*$

Proof. When Γ is $x_1 :^{k_1} A_1, \ldots, x_n :^{k_n} A_n$, we write $\overline{\Gamma}$ for $x_1^{k_1} \otimes \ldots \otimes x_n^{k_n}$.

The native inference rules are derivable as follows:

- For the identity rule, we use the fact that $\overline{0 \cdot \Gamma}$ is equal to 1 by the unit laws for the monoid:

$$\dfrac{}{0 \cdot \Gamma + x :^1 P \vdash P} \qquad \dfrac{\overline{0 \cdot \Gamma} \otimes x \Rightarrow x}{\Gamma^*, x : P \vdash_{\overline{0 \cdot \Gamma} \otimes x} P}$$

- $$\dfrac{\Gamma, x :^n A \vdash B}{\Gamma \vdash A \to^n B} \qquad \dfrac{\Gamma^*, x : A^* \vdash_{\overline{\Gamma} \otimes x^n} B^*}{\Gamma^* \vdash_{\overline{\Gamma}} \mathsf{U}_{c.c \otimes x^n}(x : A^* \mid B^*)}$$

- Note that $\Gamma + \Delta$ is only defined on contexts that have the same variables and types, so $(\Gamma + \Delta)^* = \Gamma^* = \Delta^*$. Additionally, $\overline{\Gamma + \Delta} \equiv \overline{\Gamma} \otimes \overline{\Delta}$, and $\overline{n \cdot \Gamma} \equiv \overline{\Gamma}^n$.

$$\dfrac{\Delta \vdash A \quad \Gamma, z :^k B \vdash C}{\Gamma + f :^k A \to^n B + (nk \cdot \Delta) \vdash C} \qquad \dfrac{\begin{array}{l} f : \mathsf{U}_{f.f \otimes x^n}(x : A \mid B) \in \Gamma^* \\ \overline{\Gamma} \otimes f^k \otimes \overline{\Delta}^{nk} \equiv (\overline{\Gamma} \otimes z^k)[f \otimes (\alpha)^n / z] \\ \Delta^* = \Gamma^* \vdash_{\overline{\Delta}} A \\ \Gamma^*, z : B^* \vdash_{\overline{\Gamma} \otimes z^k} C^* \end{array}}{\Gamma^* \vdash_{\overline{\Gamma} \otimes f^k \otimes \overline{\Delta}^{nk}} C}$$

Here we use associativity and commutativity to show $(f \otimes (\alpha)^n)^k \equiv f^k \otimes (\alpha)^{nk}$.

Conversely, suppose we have $\Gamma^* \vdash_{\overline{\Gamma}} A^*$.

- Case for
$$\frac{x : P \in \Gamma^* \quad \overline{\Gamma} \Rightarrow x}{\Gamma^* \vdash_{\overline{\Gamma}} P}$$
Let Γ be $x_1 :^{k_1} A_1, \ldots, x :^k A, \ldots, x_n :^{k_n} A_n$. Since the only type that encodes to an atom is that atom, we have $A = P$. For the linear logic mode theory (commutative monoid), there are no transformation axioms besides equations, so $\overline{\Gamma} \Rightarrow x$ implies $\overline{\Gamma} \equiv x$, which in turn implies that $k_i = 0$ and $k = 1$ (anything else would encode to a monoid term with a non-zero coefficient for some variable besides x, or with a non-one coefficient for x). Thus
$$\Gamma = (x_1 :^0 A_1, \ldots, x :^1 P, \ldots, x_n :^0 A_n) = 0 \cdot (x_1 :^0 A_1, \ldots, x :^0 P, \ldots, x_n :^0 A_n) + x :^1 P$$
so the hypothesis rule applies:
$$\frac{}{0 \cdot (x_1 :^0 A_1, \ldots, x :^0 A, \ldots, x_n :^0 A_n) + x :^1 P \vdash P}$$

- Since the only type that encodes to a $\mathsf{U}_{c.\alpha}(\Delta \mid B)$ is $A \to^n B$, if the derivation was by UR, we have
$$\frac{\Gamma^*, x : A^* \vdash_{\overline{\Gamma} \otimes x^n} B^*}{\Gamma^* \vdash_{\overline{\Gamma}} \mathsf{U}_{c.c \otimes x^n}(x : A^* \mid B^*)}$$
Noting that the context of the premise is $(\Gamma, x :^n A)^*$ and the context descriptor of the premise is $\overline{\Gamma, x :^n A}$, the inductive hypothesis gives a derivation of $\Gamma, x :^n A \vdash B$, so we can derive
$$\frac{\Gamma, x :^n A \vdash B}{\Gamma \vdash A \to^n B}$$

- Since the only type that encodes to a $\mathsf{U}_{c.\alpha}(\Delta \mid B)$ is $A \to^n B$, if the derivation was by UL, we have
$$\frac{\begin{array}{l} f : \mathsf{U}_{f.f \otimes x^n}(x : A^* \mid B^*) \in \Gamma^* \\ \overline{\Gamma} \equiv \beta'[f \otimes (\alpha)^n / z] \\ \Gamma^* \vdash_\alpha A^* \\ \Gamma^*, z : B^* \vdash_{\beta'^*} C^* \end{array}}{\Gamma^* \vdash_{\overline{\Gamma}} C}$$
Suppose $\Gamma = x_1 :^{k_1} A_1, \ldots, x_n :^{k_n} A_n,$. By Lemma 7.9, we have $\alpha \equiv x_1^{a_1} \otimes \ldots \otimes x_n^{a_n}$ and $\beta' \equiv x_1^{b_1} \otimes \ldots \otimes x_n^{b_n} \otimes z^k$. The fact that
$$x_1^{k_1} \otimes \ldots \otimes x_n^{k_n} \equiv \beta'[f \otimes (\alpha)^n / z]$$
implies that $k_i = b_i + kna_i$ if $x_i \neq f$ and $k_i = b_i + kna_i + k$ if $x_i = f$, so $\Gamma = \Gamma' + f :^k (A \to^n B) + nk\Delta$. Writing
$$\Delta = x_1 :^{a_1} A_1, \ldots, x_n :^{a_n} A_n \Gamma' = x_1 :^{b_1} A_1, \ldots, x_n :^{b_n} A_n$$
We have $\Delta^* = \Gamma^*$ and $\Gamma'^* = \Gamma^*$ and $\overline{\Delta} = \alpha$ and $\overline{\Gamma', z :^k B} = \beta'$, so the inductive hypotheses give $\Delta \vdash A$ and $\Gamma', z :^k B \vdash C$, and we can apply the rule to get
$$\frac{\Delta \vdash A \quad \Gamma, z :^k B \vdash C}{\Gamma = (\Gamma' + f :^k A \to^n B + (nk \cdot \Delta)) \vdash C}$$

$\square$

7.4 Cartesian Logic

We compare the cartesian monoid mode theory from Section 3.6 with the following rules:

$$\frac{x : P \in \Gamma}{\Gamma \vdash^{\mathsf{c}} P} \quad \frac{\Gamma \vdash^{\mathsf{c}} A \quad \Gamma \vdash^{\mathsf{c}} B}{\Gamma \vdash^{\mathsf{c}} A \times B} \quad \frac{p : A \times B \in \Gamma \quad \Gamma, x : A, y : B \vdash^{\mathsf{c}} C}{\Gamma \vdash^{\mathsf{c}} C}$$

$$\frac{\Gamma, x : A \vdash^{\mathsf{c}} B}{\Gamma \vdash^{\mathsf{c}} A \to B} \quad \frac{f : A \to B \in \Gamma \quad \Gamma \vdash^{\mathsf{c}} A \quad \Gamma, z : B \vdash^{\mathsf{c}} C}{\Gamma \vdash^{\mathsf{c}} C}$$

In this case, neither round-trip will be the identity on raw derivations (as opposed to equivalence classes), though the one starting at these native rules will be the identity up to a (positive/left) η law for $\times$. The difference is that the above rules allow contraction for $A \times B$, whereas the framework reduces this to contraction at A and B separately. We could instead compare against a native sequent calculus that does not allow contraction for positives, but the above is more standard.

Overall, we have

THEOREM 7.11: LOGICAL ADEQUACY FOR CARTESIAN PRODUCTS AND FUNCTIONS. $\Gamma \vdash^{\mathsf{c}} A$ *iff* $\Gamma^* \vdash_{\overline{\Gamma}} A^*$

Proof. The proof of the forward direction is by induction on the given derivation, using the derivations of each rule:

$$\frac{x : P^* \in \Gamma^* \quad 1_{(\overline{\Gamma - x}) \times x}[\mathsf{w}/x] :: \overline{\Gamma - x} \times x \Rightarrow x}{\Gamma^* \vdash_{\overline{\Gamma}} P^*}\ \mathsf{v}$$

$$\frac{\mathsf{c} :: \overline{\Gamma} \Rightarrow \overline{\Gamma} \times \overline{\Gamma} \quad \Gamma^* \vdash_{\overline{\Gamma}} A^* \quad \Gamma^* \vdash_{\overline{\Gamma}} B^*}{\Gamma^* \vdash_{\overline{\Gamma}} \mathsf{F}_{x \times y}(x : A^*, y : B^*)}\ \mathsf{FR}$$

$$\frac{\overline{\Gamma} \Rightarrow \overline{\Gamma} \times p \qquad \dfrac{p : (A \times B)^* \in \Gamma^* \quad \dfrac{\Gamma^*, x : A^*, y : B^* \vdash_{\overline{\Gamma} \times x \times y} C^*}{\Gamma^*, q : (A \times B)^* \vdash_{\overline{\Gamma} \times q} C^*}\ \mathsf{FL}}{\Gamma^* \vdash_{\overline{\Gamma} \times p} C^*}\ \textit{Cor. } 4.7}{\Gamma^* \vdash_{\overline{\Gamma}} C^*}\ \textit{Lemma } 4.1$$

$$\frac{\Gamma^*, x : A^* \vdash_{\overline{\Gamma} \times x} B^*}{\Gamma^* \vdash_{\overline{\Gamma}} \mathsf{U}_{f.f \times x}(x : A^* \mid B^*)}\ \mathsf{UR}$$

$$\frac{f : (A \to B)^* \in \Gamma^* \quad \overline{\Gamma} \Rightarrow \overline{\Gamma} \times (f \times \overline{\Gamma}) \quad \Gamma^* \vdash_{\overline{\Gamma}} A^* \quad \Gamma^*, z : B^* \vdash_{\overline{\Gamma} \times z} B^*}{\Gamma^* \vdash_{\overline{\Gamma}} C}\ \mathsf{UL}$$

This shows that the above sequent calculus uses structural rules in the following places: The hypothesis rule weakens away all other variables. The $\times$ right rule contracts the entire context. The $\times$ left rule uses contraction for the mode theory ($p \Rightarrow p \times p$) and contraction-over-contraction to duplicate p to q—if we did not have a contraction here in the native rule, then this would just be FL, as the $\to$ right rule is just UR. The $\to$ left rule contracts everything in Γ for use in both the argument and the continuation, and contracts the function an additional time for use here.

Conversely, we show $\Gamma^* \vdash_{\overline{\Gamma}} A^*$ implies $\Gamma \vdash^{\mathsf{c}} A$ The proof is by induction on the size of the given derivation, to allow uses of Lemma 4.1 before applying the inductive hypothesis.

- The hypothesis rule is immediate because $x : P^* \in \Gamma^*$ implies $x : P \in \Gamma$.
- For a general use of FR, the conclusion must be $(A \times B)^*$ because this is the only type that encodes to an F:

$$\frac{\overline{\Gamma} \Rightarrow \alpha \times \beta \quad \Gamma^* \vdash_\alpha A \quad \Gamma^* \vdash_\beta B}{\Gamma^* \vdash_{\overline{\Gamma}} \mathsf{F}_{x \times y}(x : A, y : B)}$$

Because we have projections, we can compose $\overline{\Gamma} \Rightarrow \alpha \times \beta \Rightarrow \alpha$ and $\overline{\Gamma} \Rightarrow \alpha \times \beta \Rightarrow \beta$, and apply these to the premises by Lemma 4.1 to get no-bigger derivations of $\Gamma^* \vdash_{\overline{\Gamma}} A$ and $\Gamma^* \vdash_{\overline{\Gamma}} B$, and then the inductive hypotheses and $\times$-right give the result.

This corresponds to treating this derivation as if it were

$$\frac{\overline{\Gamma} \Rightarrow \overline{\Gamma} \times \overline{\Gamma} \quad \dfrac{\overline{\Gamma} \Rightarrow \alpha \quad \Gamma^* \vdash_\alpha A}{\Gamma^* \vdash_{\overline{\Gamma}} A} \quad \dfrac{\overline{\Gamma} \Rightarrow \beta \quad \Gamma^* \vdash_\beta B}{\Gamma^* \vdash_{\overline{\Gamma}} B}}{\Gamma^* \vdash_{\overline{\Gamma}} \mathsf{F}_{x \times y}(x : A, y : B)}$$

where we contract all variables and weaken the premises with any that did not already occur in α/β. In our equational theory on derivations these two are indeed equal, assuming equations on transformations giving the universal property of a cartesian product in the mode theory.

- For a general use of FL, which must be on the encoding of a $p : A \times B \in \Gamma$, we have

$$\frac{(\Gamma - p)^*, x : A^*, y : B^* \vdash_{\overline{\Gamma}[x \times y/p]} C}{\Gamma^* \vdash_{\overline{\Gamma}} C}$$

  Because $\Gamma \equiv (\Gamma - p) \times p$, the inductive hypothesis gives a derivation of $\Gamma - p, x : A, y : B \vdash^{\mathsf{c}} C$. Using the admissible weakening for cartesian logic, we have $\Gamma, x : A, y : B \vdash^{\mathsf{c}} C$, so $\times$-left gives the result. That is, our given derivation does not contract p, so we weaken with the extra occurrence of p.

- For UR, the inductive hypothesis applied to the premise gives exactly the premise of $\to$-right.

- For UL on $f : (A \to B)^*$, we have

$$\frac{\overline{\Gamma} \Rightarrow \beta'[(f \times \alpha)/z] \quad \Gamma^* \vdash_\alpha A^* \quad \Gamma^*, z : B^* \vdash_{\beta'} B^*}{\Gamma^* \vdash_{\overline{\Gamma}} C^*}$$

  Because all context descriptors are products of variables, we can rewrite $\beta' \equiv \beta'' \times z^k$ for some k and β'' not containing z. Thus, we have $\overline{\Gamma} \Rightarrow \beta'' \times (f \times \alpha)^k$ so using projections we have $\overline{\Gamma} \Rightarrow \beta''$ and $\overline{\Gamma} \Rightarrow \alpha$. Using contraction, we have $\overline{\Gamma} \times z \Rightarrow \beta'' \times z^k$. Applying these to the premises with Lemma 4.1 gives derivations of $\Gamma^* \vdash_{\overline{\Gamma}} A^*$ and $\Gamma^*, z : B^* \vdash_{\overline{\Gamma} \times z} B^*$. Thus, the inductive hypotheses give the premises of $\to$-left.

  Equationally, the only thing suspicious about this is projecting *one* of the α's from $(f \times \alpha)^k$. However, $\overline{\Gamma}$ has no duplicate variables, and for this mode theory any map $x \Rightarrow x^k$ is the k-fold contraction of x. Therefore, all projections are the same, and contracting-projecting-recontracting is the same as the original contraction.

□

7.5 Constructive S4 $\Box$

The native rules (writing $\Delta; \Gamma \vdash A$ for Δ valid; Γ $\mathsf{true} \vdash A$ true) are

$$\frac{P \in \Gamma}{\Delta; \Gamma \vdash P} \quad \frac{A \in \Delta \quad \Delta; \Gamma, A \vdash C}{\Delta; \Gamma \vdash C} \quad \frac{\Delta, A; \Gamma, \Gamma' \vdash C}{\Delta; \Gamma, \Box A, \Gamma' \vdash C} \quad \frac{\Delta; \cdot \vdash A}{\Delta; \Gamma \vdash \Box A}$$

For the mode theory in Section 3.9, we have

THEOREM 7.12: LOGICAL ADEQUACY FOR A COMONAD.

$$x_1 : A_1 \ \mathsf{valid}, \ldots; y_1 : B_1 \ \mathsf{true}, \ldots \vdash C \ \mathsf{true}$$
iff
$$x_1 : \mathsf{U_f}(A_1^*), \ldots; y_1 : B_1^*, \ldots \vdash_{\mathsf{f}(x_1) \times \ldots \times \mathsf{f}(x_n) \times y_1 \times \ldots \times y_n} C^*$$

Proof. We write Δ^* for $x_i : \vec{\mathsf{U}_\mathsf{f}}(A_i^*)$ for each assumption in $x : A_i \in \Delta$ and Γ^* as usual. We write $\overline{\Delta}$ for $\mathsf{f}(x_1) \times \ldots \times \mathsf{f}(x_n)$ for the variables in Δ and $\overline{\Gamma}$ as usual.

First we show that $\Delta; \Gamma \vdash C$ implies $\Delta^*, \Gamma^* \vdash_{\overline{\Delta} \times \overline{\Gamma}} C^*$ by induction, using the following encodings:

For the hyp rule:

$$\frac{x : P \in \Gamma^* \quad \overline{\Delta} \times \overline{\Gamma} \Rightarrow x}{\Delta^*, \Gamma^* \vdash_{\overline{\Delta} \times \overline{\Gamma}} P}$$

where the transformation weakens everything else.

For the copy rule:

$$\frac{\begin{array}{l} x : \mathsf{U}_\mathsf{f}(A^*) \in \Delta^* \\ \overline{\Delta} \times \overline{\Gamma} \Rightarrow (\overline{\Delta} \times \overline{\Gamma} \times z)[\mathsf{f}(x)/z] \\ \Delta^*, \Gamma^*, z : A^* \vdash_{\overline{\Delta} \times \overline{\Gamma} \times z} C^* \end{array}}{\Delta^*, \Gamma^* \vdash_{\overline{\Delta} \times \overline{\Gamma}} C^*}\ \mathsf{UL}$$

where the transformation contracts the $\mathsf{f}(x)$ that must be in $\overline{\Delta}$.

For $\Box$-left:

$$\dfrac{x : \mathsf{F}_\mathsf{f}(\mathsf{U}_\mathsf{f}(A^*)) \in \Gamma^* \quad \dfrac{\Delta^*, z : \mathsf{U}_\mathsf{f}(A^*), \Gamma^* - x \vdash_{\overline{\Delta} \times \overline{\Gamma - x} \times z} C^*}{\Delta^*, \Gamma^* - x, z : \mathsf{U}_\mathsf{f}(A^*) \vdash_{\overline{\Delta} \times \overline{\Gamma - x} \times z} C^*}\ 4.3}{\Delta^*, \Gamma^* \vdash_{\overline{\Delta} \times \overline{\Gamma}} C^*}\ \mathsf{FL}$$

We took the liberty of making $\Box$-left remove the $\Box$-assumption (which as usual for positives is a choice), or else we could do a contraction here to match it. We use commutativity of $\times$ and exchange to make the order match the native rule.

For $\Box$-right:

$$\dfrac{\overline{\Delta} \times \overline{\Gamma} \Rightarrow \mathsf{f}(x_1 \times_v \ldots) \quad \dfrac{\dfrac{\mathsf{f}(x_1 \times \ldots) \Rightarrow \mathsf{f}(x_1) \times \ldots \quad \dfrac{\Delta^* \vdash_{\overline{\Delta}} A^*}{\Delta^*, \Gamma^* \vdash_{\mathsf{f}(x_1) \times \ldots} A^*}\ 4.2}{\Delta^*, \Gamma^* \vdash_{\mathsf{f}(x_1 \times_v \ldots)} A^*}\ 4.1}{\Delta^*, \Gamma^* \vdash_{(x_1 \times_v \ldots)} \mathsf{U}_\mathsf{f}(A^*)}\ \mathsf{UR}}{\Delta^*, \Gamma^* \vdash_{\overline{\Delta} \times \overline{\Gamma}} \mathsf{F}_\mathsf{f}(\mathsf{U}_\mathsf{f}(A^*))}\ \mathsf{FR}$$

We write $x_1 \ldots$ for the variables from Δ. The first transformation weakens away Γ and uses the monoidalness transformation axioms for f to pull f outside the product. After the UR, we uses the converse $\mathsf{f}(\top_v) \Rightarrow \top$ and $\mathsf{f}(x \times_v y) \Rightarrow \mathsf{f}(x) \times \mathsf{f}(y)$ that follow from the intro forms for the cartesian $(\times, \top)$ and congruence of f on the projections. Finally, we weaken-over-weaken the encoding of the premise with Γ^*

Conversely, suppose we have $\Delta^*, \Gamma^* \vdash_{\overline{\Delta} \times \overline{\Gamma}} C^*$. For this mode theory, the context descriptors containing all variables are initial:

- If $\vec{x_i} : \mathsf{v}, \vec{y_i} : \mathsf{t} \vdash \alpha : \mathsf{v}$ then $x_1 \times_v \ldots \times_v x_n \Rightarrow \alpha$. α a variable, in which case we weaken all the others, 1, in which case we weaken everything, or $\alpha_1 \times_v \alpha_2$, in which case we get $x_1 \times_v \ldots \times_v x_n \Rightarrow \alpha_1$ and $x_1 \times_v \ldots \times_v x_n \Rightarrow \alpha_2$ by induction, apply congruence of $\times_v$, and then precompose with contraction.

- If $\vec{x_i} : \mathsf{v}, \vec{y_i} : \mathsf{t} \vdash \alpha : \mathsf{t}$ then $\mathsf{f}(x_1) \times \mathsf{f}(x_n) \times y_1 \times \ldots \times y_n \Rightarrow \alpha$. By induction on α. If it is y_i, then we weaken all other variables; if it is 1 then, we weaken everything. If it is $\alpha_1 \times \alpha_2$, then we get $f(\vec{x_i}) \otimes \vec{y_i} \Rightarrow \alpha_1$ $f(\vec{x_i}) \otimes \vec{y_i} \Rightarrow \alpha_2$ by induction, apply congruence of $\times$, and precompose with contraction. Finally, if it is $\mathsf{f}(\alpha)$, then we get $x_1 \times_v \ldots \times_v x_n \Rightarrow \alpha$ by the inductive hypothesis, apply congruence of f, and then weaken with $\vec{y_i}$ and precompose with $\mathsf{f}(x_1) \times \ldots \times \mathsf{f}(x_n) \Rightarrow \mathsf{f}(x_1 \times_v \ldots \times_v x_n)$.

We proceed by induction on the given derivation. Observe that Lemma 7.1 applies to any sequent $\Delta^*, \Gamma^* \vdash_{\overline{\Delta} \times \overline{\Gamma}} C^*$, because $\mathsf{F}_\mathsf{f}(A)$ and $\mathsf{U}_\mathsf{f}(A)$ are relevant, and the equations (associativity, unit, commutativity) have the same variables on both sides, and the transformations (weakening, contraction, distributing f) do not introduce any new variables on the right-hand side, so Lemma 7.2 applies.

- For the axiom rule, we know $x : P$ is in Γ^* not Δ^* because all Δ-formulas are prefixed with a U. Since the only formula that encodes to an atom is that atom, we have $x : P \in \Gamma$, so the hypothesis rule applies.

- If the last rule used was FR, then because the only formula that encodes to F is $\Box A$, we have

$$\frac{\Delta^* \times \Gamma^* \Rightarrow \mathsf{f}(\alpha) \quad \Delta^*, \Gamma^* \vdash_\alpha \mathsf{U_f}(A)}{\Delta^*, \Gamma^* \vdash_{\overline{\Delta} \times \overline{\Gamma}} \mathsf{F_f}(\mathsf{U_f}(A))}$$

  Writing the variables of Δ as $x_1, \ldots, x_n$, the first premise implies that $x_1 \times \ldots \times x_n \Rightarrow \alpha$, so by Lemma 4.1, we can make the second premise into a no-bigger derivation of $\Delta^*, \Gamma^* \vdash_{x_1 \times \ldots \times x_n} \mathsf{U_f}(A)$. Applying Lemma **??**, we get a no-bigger derivation of $\Delta^*, \Gamma^* \vdash_{\mathsf{f}(x_1 \times \ldots \times x_n)} A$. Again using Lemma 4.1, we get $\Delta^*, \Gamma^* \vdash_{\mathsf{f}(x_1) \times \ldots \times \mathsf{f}(x_n)} A$. Finally, we use Lemma 7.1 to remove Γ^*, getting a no-larger derivation of $\Delta^* \vdash_{\overline{\Delta}} A$. Then the inductive hypothesis gives $\Delta; \cdot \vdash A$, so we get $\Delta; \Gamma \vdash \Box A$ by rule.

- The last rule cannot have been UR, because no types encode to a formula beginning with U.

- For FL, because each variable in Δ^* begins with a U, the variable being eliminated must be in Γ. Because the only type that encodes to F is $\Box A$, we have $x : \Box A$ in Γ and

$$\frac{x : \mathsf{F_f}(\mathsf{U_f}(A^*)) \in \Gamma^* \quad \Delta^*, \Gamma^* - x, y : \mathsf{U_f}(A) \vdash_{\overline{\Delta} \times \overline{\Gamma}[\mathsf{f}(y)/x]} C^*}{\Delta^*, \Gamma^* \vdash_{\overline{\Delta} \times \overline{\Gamma}} C^*}$$

  Using Lemma 4.3, we can change the premise's context to $\Delta^*, y : \mathsf{U_f}(A^*), \Gamma^* - x$, which is $(\Delta, y : A)^*, \Gamma^* - x$. By associativity and commutativity, we have that $\overline{\Delta, y : A} \times \overline{\Gamma - x} \equiv \overline{\Delta} \times \overline{\Gamma}[\mathsf{f}(y)/x]$. So we can apply the inductive hypothesis to the premise to get $\Delta, y : A; \Gamma - x \vdash C$, and then we get $\Delta; \Gamma \vdash C$ by rule.

- For UL, because no types encode to U, the eliminated variable must be from Δ^*. This means $x : A \in \Delta$ and we have

$$\frac{\begin{array}{c} x : \mathsf{U_f}(A) \in \Delta^* \\ \overline{\Delta} \times \overline{\Gamma} \Rightarrow \beta'[\mathsf{f}(x)/z] \\ \Delta^*, \Gamma^*, z{:}A^* \vdash_{\beta'} C^* \end{array}}{\Delta^*, \Gamma^* \vdash_{\overline{\Delta} \times \overline{\Gamma}} C^*}$$

  We have $\overline{\Delta} \times \overline{\Gamma} \times z \Rightarrow \beta'$, so by Lemma 4.1, $\Delta^*, \Gamma^*, z{:}A^* \vdash_{\overline{\Delta} \times \overline{\Gamma} \times z} C^*$. Then the inductive hypothesis gives $\Delta; \Gamma, z : A \vdash C$, so the copy rule gives $\Delta; \Gamma \vdash C$.

□

7.6 Non-strong Monad ($\Diamond$)

We compare the mode theory for the non-strong $\Diamond$ (modes $\mathtt{t}$ and $\mathtt{p}$ with an affine (semicartesian) commutative monoid $(\otimes, 1, \mathsf{w} :: x \Rightarrow 1)$ on $\mathtt{t}$ and $x : \mathtt{t} \vdash \mathsf{g}(x) : \mathtt{p}$) against the rules at the beginning of Section 3.11.

Recall from above that $\Diamond A^* = \mathsf{U_g}(\mathsf{F_g}(A^*))$ and that the correspondence is:

THEOREM 7.13: LOGICAL ADEQUACY FOR A NON-STRONG MONAD.

$$A_1 \ \mathsf{true}, \ldots, A_1 \ \mathsf{true} \vdash C \ \mathsf{true}$$
$$\textit{iff}$$
$$x_1 : A_1^*, \ldots, x_1 : A_n^* \vdash_{x_1 \otimes \ldots \otimes x_n} C^*$$

and

$$A_1 \ \mathsf{true}, \ldots, A_n \ \mathsf{true} \vdash C \ \mathsf{poss}$$
$$\textit{iff}$$
$$x_1 : A_1^*, \ldots, x_1 : A_n^* \vdash_{\mathsf{g}(x_1 \otimes \ldots \otimes x_n)} \mathsf{F_g}(C^*).$$

Proof. We write Γ^* as usual and $\overline{\Gamma}$ for $x_1 \otimes \ldots \otimes x_n$.

The three rules are represented by

$$\frac{\mathsf{g}(\overline{\Gamma}) \Rightarrow \mathsf{g}(\overline{\Gamma}) \quad \Gamma^* \vdash_{\overline{\Gamma}} A^*}{\Gamma^* \vdash_{\mathsf{g}(\overline{\Gamma})} \mathsf{F_g}(C^*)} \ \mathsf{FR}$$

$$\frac{\Gamma^* \vdash_{\mathsf{g}(\overline{\Gamma})} \mathsf{F}_{\mathsf{g}}(C^*)}{\Gamma^* \vdash_{\overline{\Gamma}} \mathsf{U}_{\mathsf{g}}(\mathsf{F}_{\mathsf{g}}(C^*))}\ \mathsf{UR}$$

$$\frac{x : \mathsf{U}_{\mathsf{g}}(\mathsf{F}_{\mathsf{g}}(A^*)) \in \Gamma^* \quad \mathsf{g}(\overline{\Gamma}) \Rightarrow \mathsf{g}(x) \quad \dfrac{z : A^* \vdash_{\mathsf{g}(z)} \mathsf{F}_{\mathsf{g}}(C^*)}{y : \mathsf{F}_{\mathsf{g}}(A)^* \vdash_{\mathsf{y}} \mathsf{F}_{\mathsf{g}}(C^*)}\ \mathsf{FL}}{\Gamma^* \vdash_{\mathsf{g}(\overline{\Gamma})} \mathsf{F}_{\mathsf{g}}(C^*)}$$

(and an identity rule $\Gamma, P\ \mathsf{true} \vdash P\ \mathsf{true}$ would be translated as usual).

Conversely, suppose we have a derivation of $\Gamma^* \vdash_{\overline{\Gamma}} C^*$ or $\Gamma^* \vdash_{\mathsf{g}(\overline{\Gamma})} \mathsf{F}_{\mathsf{g}}(C^*)$. Lemma 7.1 can be used on such derivations: the equational axioms preserve variables and w removes but does not add, so we use Lemma 7.2; and $\mathsf{F}_{\mathsf{g}}(A)$ and $\mathsf{U}_{\mathsf{g}}(A)$ are both relevant, so by induction the encoding of any sequent is.

Suppose we have $\Gamma^* \vdash_{\overline{\Gamma}} C^*$. The hypothesis rule is translated back to itself as usual. Since there are no types that encode to F, any other final rule must be UR or UL, and the type must be $\mathsf{U}_{\mathsf{g}}(\mathsf{F}_{\mathsf{g}}(A^*))$.

- If we have
$$\frac{\Gamma^* \vdash_{\mathsf{g}(\overline{\Gamma})} \mathsf{F}_{\mathsf{g}}(A^*)}{\Gamma^* \vdash_{\overline{\Gamma}} \mathsf{U}_{\mathsf{g}}(\mathsf{F}_{\mathsf{g}}(A^*))}\ \mathsf{UR}$$
then in the inductive hypothesis gives $\Gamma \vdash A\ \mathsf{poss}$, so we have $\Gamma \vdash \Diamond A\ \mathsf{true}$ by rule.

- Suppose we have
$$\frac{x : \mathsf{U}_{\mathsf{g}}(\mathsf{F}_{\mathsf{g}}(A^*)) \in \Gamma \quad \overline{\Gamma} \Rightarrow \beta'[\mathsf{g}(x)/z] \quad \Gamma^*, z : \mathsf{F}_{\mathsf{g}}(A^*) \vdash_{\beta'} C^*}{\Gamma^* \vdash_{\overline{\Gamma}} C^*}\ \mathsf{UL}$$
For this mode theory, the constants do not allow embedding a p-mode term in a t-mode term. Therefore, the subterm $\mathsf{g}(x)$ cannot occur in a "reduct" of $\overline{\Gamma}$, which has mode $\mathtt{t}$. Thus, z does not occur in β', and $\overline{\Gamma} \Rightarrow \beta'$. Applying Lemma 4.1 to the premise gives a no-bigger derivation of $\Gamma^*, z : \mathsf{F}_{\mathsf{g}}(A^*) \vdash_{\overline{\Gamma}} C^*$, and applying Lemma 7.1 to strengthen away z gives a no-bigger derivation of $\Gamma^* \vdash_{\overline{\Gamma}} C^*$. Then the inductive hypothesis gives $\Gamma \vdash C\ \mathsf{true}$ as desired.

Suppose we have $\Gamma^* \vdash_{\mathsf{g}(\overline{\Gamma})} \mathsf{F}_{\mathsf{g}}(C^*)$ and want $\Gamma \vdash C\ \mathsf{poss}$. Since there are no F's in the context, the only possibilities are UL and FR:

- Suppose we have
$$\frac{\mathsf{g}(\overline{\Gamma}) \Rightarrow \mathsf{g}(\alpha) \quad \Gamma^* \vdash_{\alpha} C^*}{\Gamma^* \vdash_{\mathsf{g}(\overline{\Gamma})} \mathsf{F}_{\mathsf{g}}(C^*)}\ \mathsf{FR}$$
For this mode theory, there are no there are no equalities or transformations between terms of the form $\mathsf{g}(\alpha)$ and any other p-mode term besides congruence on $\alpha \Rightarrow \alpha'$, so we can extract a transformation $\overline{\Gamma} \Rightarrow \alpha$ (such that the given one is equal to congruence with g on it). So we get $\overline{\Gamma} \Rightarrow \alpha$ and can use Lemma 4.1 to get a no-bigger derivation of $\Gamma^* \vdash_{\overline{\Gamma}} C^*$. By the inductive hypothesis on this premise we get $\Gamma \vdash A\ \mathsf{true}$, which gives $\Gamma \vdash A\ \mathsf{poss}$ as desired.

- Suppose we have
$$\frac{\begin{array}{l} x : \mathsf{U}_{\mathsf{g}}(\mathsf{F}_{\mathsf{g}}(A^*)) \in \Gamma \\ \mathsf{g}(\overline{\Gamma}) \Rightarrow \beta'[\mathsf{g}(x)/z] \\ \Gamma^*, z : \mathsf{F}_{\mathsf{g}}(A^*) \vdash_{\beta'} \mathsf{F}_{\mathsf{g}}(C^*) \end{array}}{\Gamma^* \vdash_{\mathsf{g}(\overline{\Gamma})} \mathsf{F}_{\mathsf{g}}(C^*)}\ \mathsf{UL}$$
We have $x_i : \mathtt{t}, z : \mathtt{p} \vdash \beta' : \mathtt{p}$, so by inversion the only possibilities are z or $\mathsf{g}(-)$, and in the latter case z does not occur, as argued above.

  If β' is z, then we have $\Gamma^*, z : \mathsf{F}_{\mathsf{g}}(A^*) \vdash_z \mathsf{F}_{\mathsf{g}}(C^*)$. By Lemma 7.1 (strengthening away Γ^*) we have a no-bigger derivation of $z : \mathsf{F}_{\mathsf{g}}(A^*) \vdash_z \mathsf{F}_{\mathsf{g}}(C^*)$. By Lemma 4.5, we can left-invert to get a no-bigger derivation of

$z' : A^* \vdash_{\mathsf{g}(z')} \mathsf{F_g}(C^*)$. By the inductive hypothesis, this translates to a derivation of A $\mathsf{true} \vdash C$ poss, so applying $\Diamond$-left gives the result.

If z does not occur in β' we have $\mathsf{g}(\overline{\Gamma}) \Rightarrow \beta'$ so pushing this into the premise gives by Lemma 4.1 gives a no-bigger derivation of $\Gamma^*, z : \mathsf{F_g}(A^*) \vdash_{\mathsf{g}(\overline{\Gamma})} \mathsf{F_g}(C^*)$. Then strengthening z by Lemma 7.1 gives a no-bigger derivation of $\Gamma^* \vdash_{\mathsf{g}(\overline{\Gamma})} \mathsf{F_g}(C^*)$. so the inductive hypothesis gives the result. That is, we did an elimination to produce a 0-use variable, which we can strengthen away.

□

7.7 Strong Monad ($\bigcirc A$)

For the linear logic (commutative monoid) mode theory, we compare the rules

$$\frac{\Gamma \vdash A\ \mathsf{true}}{\Gamma \vdash A\ \mathsf{poss}} \qquad \frac{\Gamma \vdash A\ \mathsf{poss}}{\Gamma \vdash \bigcirc A\ \mathsf{true}} \qquad \frac{\Gamma, \Gamma', A\ \mathsf{true} \vdash C\ \mathsf{poss}}{\Gamma, \bigcirc A\ \mathsf{true}, \Gamma' \vdash C\ \mathsf{poss}}$$

with the mode theory consisting of a commutative monoid $(\otimes, 1)$ on p, a functor g from t to p, and

$$x : \mathsf{t}, y : \mathsf{p} \vdash x \otimes_{\mathsf{tp}} y : \mathsf{p} \qquad \mathsf{g}(x \otimes y) \equiv x \otimes_{\mathsf{tp}} \mathsf{g}(y)$$

(We elide the equations $(x \otimes y) \otimes_{\mathsf{tp}} z \equiv x \otimes_{\mathsf{tp}} (y \otimes_{\mathsf{tp}} z)$ and $1 \otimes_{\mathsf{tp}} z \equiv z$ discussed above because they provable when z is $\mathsf{g}(x)$, which are the only terms that come up in this encoding.)

Translating $\bigcirc A$ by $\mathsf{U_g}(\mathsf{F_g}(A^*))$, we have the same adequacy statement as above:

THEOREM 7.14: LOGICAL ADEQUACY FOR A STRONG MONAD.

$$\begin{gathered} A_1\ \mathsf{true}, \ldots, A_1\ \mathsf{true} \vdash C\ \mathsf{true} \\ \textit{iff} \\ x_1 : A_1^*, \ldots, x_1 : A_n^* \vdash_{x_1 \otimes \ldots \otimes x_n} C^* \end{gathered}$$

and

$$\begin{gathered} A_1\ \mathsf{true}, \ldots, A_n\ \mathsf{true} \vdash C\ \mathsf{poss} \\ \textit{iff} \\ x_1 : A_1^*, \ldots, x_1 : A_n^* \vdash_{\mathsf{g}(x_1 \otimes \ldots \otimes x_n)} \mathsf{F_g}(C^*). \end{gathered}$$

Proof. When Γ is $x_1 : A_1\ \mathsf{true}, \ldots, x_n : A_n\ \mathsf{true}$, we write $\overline{\Gamma}$ for $x_1 \otimes \ldots \otimes x_n$.

The three rules are represented by

$$\frac{\mathsf{g}(\overline{\Gamma}) \Rightarrow \mathsf{g}(\overline{\Gamma}) \quad \Gamma^* \vdash_{\overline{\Gamma}} A^*}{\Gamma^* \vdash_{\mathsf{g}(\overline{\Gamma})} \mathsf{F_g}(C^*)}\ \mathsf{FR}$$

$$\frac{\Gamma^* \vdash_{\mathsf{g}(\overline{\Gamma})} \mathsf{F_g}(C^*)}{\Gamma^* \vdash_{\overline{\Gamma}} \mathsf{U_g}(\mathsf{F_g}(C^*))}\ \mathsf{UR}$$

$$\frac{x : \mathsf{U_g}(\mathsf{F_g}(A^*)) \in \Gamma^* \quad \mathsf{g}(\overline{\Gamma}) \Rightarrow (\overline{\Gamma_1} \otimes \overline{\Gamma_2}) \otimes_{\mathsf{tp}} \mathsf{g}(x) \quad \dfrac{\dfrac{\Gamma_1^*, \Gamma_2^*, y : A^* \vdash_{\mathsf{g}(\overline{\Gamma_1} \otimes \overline{\Gamma_2} \otimes y)} \mathsf{F_g}(C^*)}{\Gamma^*, y : A^* \vdash_{(\overline{\Gamma_1} \otimes \overline{\Gamma_2} \otimes \mathsf{g}(y)) \equiv \mathsf{g}(\overline{\Gamma_1} \otimes y \otimes \overline{\Gamma_2})} \mathsf{F_g}(C^*)}\ \textit{Lemma}\ 4.2}{\Gamma^*, z : \mathsf{F_g}(A^*) \vdash_{\overline{\Gamma_1} \otimes \overline{\Gamma_2} \otimes z} \mathsf{F_g}(C^*)}\ \mathsf{FL}}{(\Gamma^* = \Gamma_1^*, x : \mathsf{U_g}(\mathsf{F_g}(A^*)), \Gamma_2^*) \vdash_{\mathsf{g}(\overline{\Gamma})} \mathsf{F_g}(C^*)}$$

The transformation $\mathsf{g}(\overline{\Gamma}) \Rightarrow (\overline{\Gamma_1} \otimes \overline{\Gamma_2}) \otimes_{\mathsf{tp}} \mathsf{g}(x)$ is an equality given by associating and commuting $\overline{\Gamma} = (\overline{\Gamma_1} \otimes x \otimes \overline{\Gamma_2})$ to $(\overline{\Gamma_1} \otimes \overline{\Gamma_2}) \otimes x$ and then using

$$\mathsf{g}((\overline{\Gamma_1} \otimes \overline{\Gamma_2}) \otimes x) \equiv (\overline{\Gamma_1} \otimes \overline{\Gamma_2}) \otimes_{\mathsf{tp}} \mathsf{g}(x)$$

An identity rule P true $\vdash P$ true is translated as usual.

Conversely, suppose we have a derivation of $\Gamma^* \vdash_{\overline{\Gamma}} C^*$ or $\Gamma^* \vdash_{\mathsf{g}(\overline{\Gamma})} \mathsf{F}_{\mathsf{g}}(C^*)$. Lemma 7.1 can be used on such derivations: the equational axioms preserve variables and there are no transformations, so we use Lemma 7.2; and $\mathsf{F}_{\mathsf{g}}(A)$ and $\mathsf{U}_{\mathsf{g}}(A)$ are both relevant, so by induction the encoding of any sequent is. We use the following properties of the mode theory:

- If $\alpha \Rightarrow \beta$ then $\alpha \equiv \beta$, because there are no structural transformation axioms.
- If $x_1 : \mathsf{t}, \ldots, x_n : \mathsf{t}, z : \mathsf{p} \vdash \beta : \mathsf{p}$ then β is of the form $\alpha_1 \otimes_{\mathsf{tp}} \alpha_2 \otimes_{\mathsf{tp}} \ldots z$, or $\alpha_1 \otimes_{\mathsf{tp}} \alpha_2 \otimes_{\mathsf{tp}} \ldots \mathsf{g}(\alpha_n)$ (in which case z does not occur). Moreover, if $x_1 : \mathsf{t}, \ldots, x_n : \mathsf{t} \vdash \beta : \mathsf{p}$ then β is of the form $\alpha_1 \otimes_{\mathsf{tp}} \alpha_2 \otimes_{\mathsf{tp}} \ldots \mathsf{g}(\alpha_n)$. This is because the only constants of mode p are $\otimes_{\mathsf{tp}}$, which has only 1 p-mode subposition, and g, which has 0, so any term of mode p must be an iterated application of $\otimes_{\mathsf{tp}}$ ending in either a variable (if there is one in the context), or g.
- If $\vec{x_i : \mathsf{t}} \vdash \mathsf{g}(\alpha) \equiv \mathsf{g}(\beta)$ then $\alpha \equiv \beta$. We generalize and define a meta-operation $\psi \vdash t(\alpha) : \mathsf{t}$ when $\psi \vdash \alpha : \mathsf{p}$ that picks out the t parts of α:
$$t(z) = 1 t(\alpha \otimes_{\mathsf{tp}} \beta) = \alpha \otimes t(\beta)$$
$$t(\mathsf{g}(\alpha)) = \alpha$$
  It now suffices to show that $\alpha \equiv \beta$ implies $t(\alpha) \equiv t(\beta)$. The cases for reflexivity, symmetry, and transitivity all follow from the inductive hypotheses and the corresponding rules. For the axiom $\mathsf{g}(x \otimes y) \equiv x \otimes_{\mathsf{tp}} \otimes \mathsf{g}(y)$, we get $x \otimes y$ on both sides. For congruence, suppose $\psi, x : q \vdash \alpha' \equiv \beta' : \mathsf{p}$ and $\psi \vdash \alpha \equiv \beta : q$. By the inductive hypothesis, $t(\alpha) \equiv t(\beta)$. We distinguish cases on whether q is p or t.

  If the congruence variable x has mode t, then the result follows from the fact that $x : \mathsf{t} \vdash \alpha : \mathsf{p}$ implies $t(\alpha[\beta/x]) \equiv t(\alpha)[\beta/x]$, because we get $t(\alpha)[\alpha'/x] \equiv t(\beta)[\beta'/x]$ by the congruence rule. To prove this, the case for z is immediate, because both sides are 1. In the case for $\mathsf{g}(\alpha)$, both sides are $\alpha[\beta/x]$. In the case for $\alpha_1 \otimes_{\mathsf{tp}} \alpha_2$, by the inductive hypothesis we get $t(\alpha_2[\beta/x]) \equiv t(\alpha_2)[\beta/x]$ by the inductive hypothesis, and we need to show that $\alpha_1[\beta/x] \otimes t(\alpha_2[\beta/x]) \equiv (\alpha_1 \otimes t(\alpha_2))[\beta/x]$, which is true by definition of substitution.

  If the congruence variable x has mode p, then because all equational axioms have the same variables on the left and right, x occurs either on both sides or on neither. If it occurs on neither, then the inductive hypothesis $t(\alpha) \equiv t(\beta)$ is enough, because $t(\alpha[\beta/x]) = t(\alpha) \equiv t(\beta) = t(\alpha'[\beta'/x])$. If it occurs on both, then the result follows from the fact that $t(\alpha[\beta/x]) \equiv t(\alpha) \otimes t(\beta)$, because $t(\alpha) \equiv t(\beta)$ and $t(\alpha') \equiv t(\beta')$ by the inductive hypotheses on the two subderivations, both of which have mode p. We prove the lemma that $x : \mathsf{p} \vdash \alpha : \mathsf{p}$ (where x occurs in α) and $\beta : \mathsf{p}$ imply $t(\alpha[\beta/x]) \equiv t(\alpha) \otimes t(\beta)$ by induction on α. If it is x, then we have $t(\beta) \equiv 1 \otimes t(\beta)$. It cannot be some other y or $\mathsf{g}(\alpha)$ because then x has no place to occur. If it is $\alpha_1 \otimes_{\mathsf{tp}} \alpha_2$, then by the inductive hypothesis, $t(\alpha_2[\beta/x]) \equiv t(\alpha_2) \otimes t(\beta)$, and we have to show that $\alpha_1[\beta/x] \otimes t(\alpha_2[\beta/x]) \equiv \alpha_1 \otimes t(\alpha_2) \otimes t(\beta)$. This follows because x, which has mode p, cannot occur in α_1, which has mode t.

Suppose we have $\Gamma^* \vdash_{\overline{\Gamma}} C^*$. The hypothesis rule is translated back to itself as usual. Since there are no types that encode to F, any other final rule must be UR or UL, and the type must be $\mathsf{U}_{\mathsf{g}}(\mathsf{F}_{\mathsf{g}}(A^*))$.

- If we have
$$\frac{\Gamma^* \vdash_{\mathsf{g}(\overline{\Gamma})} \mathsf{F}_{\mathsf{g}}(A^*)}{\Gamma^* \vdash_{\overline{\Gamma}} \mathsf{U}_{\mathsf{g}}(\mathsf{F}_{\mathsf{g}}(A^*))}\ \mathsf{UR}$$
  then the inductive hypothesis gives $\Gamma \vdash A$ poss, so we have $\Gamma \vdash \bigcirc A$ true by rule.
- Suppose we have
$$\frac{x : \mathsf{U}_{\mathsf{g}}(\mathsf{F}_{\mathsf{g}}(A^*)) \in \Gamma \quad \overline{\Gamma} \Rightarrow \beta'[\mathsf{g}(x)/z] \quad \Gamma^*, z : \mathsf{F}_{\mathsf{g}}(A^*) \vdash_{\beta'} C^*}{\Gamma^* \vdash_{\overline{\Gamma}} C^*}\ \mathsf{UL}$$
  For this mode theory, there are no transformations besides identities, so $\overline{\Gamma} \equiv \beta'[\mathsf{g}(x)/z]$. The constants do not allow embedding a p-mode term in a t-mode term. Therefore, the subterm $\mathsf{g}(x)$ cannot occur in a anything equal to $\overline{\Gamma}$, which has mode t. Thus, z does not occur in β', and $\overline{\Gamma} \equiv \beta'$. Applying Lemma 7.1 to strengthen away z gives a no-bigger derivation of $\Gamma^* \vdash_{\overline{\Gamma}} C^*$. Then the inductive hypothesis gives $\Gamma \vdash C$ true as desired.

Suppose we have $\Gamma^* \vdash_{\mathsf{g}(\overline{\Gamma})} \mathsf{F}_{\mathsf{g}}(C^*)$ and want $\Gamma \vdash C$ poss. Since there are no F's in the context, the only possibilities are UL and FR:

- Suppose we have
$$\frac{\mathsf{g}(\overline{\Gamma}) \Rightarrow \mathsf{g}(\alpha) \quad \Gamma^* \vdash_\alpha C^*}{\Gamma^* \vdash_{\mathsf{g}(\overline{\Gamma})} \mathsf{F}_{\mathsf{g}}(C^*)}\ \mathsf{FR}$$
We have $\mathsf{g}(\overline{\Gamma}) \equiv \mathsf{g}(\alpha)$, which implies $\overline{\Gamma} \equiv \alpha$. By the inductive hypothesis on the premise we get $\Gamma \vdash A$ true, which gives $\Gamma \vdash A$ poss as desired.

- Suppose we have
$$\frac{\begin{array}{l} x : \mathsf{U}_{\mathsf{g}}(\mathsf{F}_{\mathsf{g}}(A^*)) \in \Gamma^* \\ \mathsf{g}(\overline{\Gamma}) \equiv \beta'[\mathsf{g}(x)/z] \\ \Gamma^*, z : \mathsf{F}_{\mathsf{g}}(A^*) \vdash_{\beta'} \mathsf{F}_{\mathsf{g}}(C^*) \end{array}}{\Gamma^* \vdash_{\mathsf{g}(\overline{\Gamma})} \mathsf{F}_{\mathsf{g}}(C^*)}\ \mathsf{UL}$$
We have $x_1 : \mathsf{t}, \ldots, x_n : \mathsf{t}, z : \mathsf{p} \vdash \beta' : \mathsf{p}$, so β' is either $\alpha_1 \otimes_{\mathsf{tp}} \ldots \otimes_{\mathsf{tp}} z$ or $\alpha_1 \otimes_{\mathsf{tp}} \ldots \otimes_{\mathsf{tp}} \mathsf{g}(\alpha)$, and in this case z does not occur.

  If β' is $\alpha_1 \otimes_{\mathsf{tp}} \ldots \otimes_{\mathsf{tp}} z$, then we have $\Gamma^*, z : \mathsf{F}_{\mathsf{g}}(A^*) \vdash_{\alpha_1 \otimes_{\mathsf{tp}} \ldots \otimes_{\mathsf{tp}} \alpha_n \otimes_{\mathsf{tp}} z} \mathsf{F}_{\mathsf{g}}(C^*)$ and $\mathsf{g}(\overline{\Gamma}) \equiv \alpha_1 \otimes_{\mathsf{tp}} \ldots \otimes_{\mathsf{tp}} \mathsf{g}(x)$. This entails $\overline{\Gamma} \equiv \alpha_1 \otimes \ldots \otimes \alpha_n \otimes x$, and because the only equations are the commutative monoid laws, $\overline{\Gamma - x} \equiv \alpha_1 \otimes \ldots \otimes \alpha_n$. By Lemma 4.5, we have a no-bigger derivation of $\Gamma^*, y : A^* \vdash_{\alpha_1 \otimes_{\mathsf{tp}} \ldots \otimes_{\mathsf{tp}} \mathsf{g}(y)} \mathsf{F}_{\mathsf{g}}(C^*)$. The subscript $\alpha_1 \otimes_{\mathsf{tp}} \ldots \otimes_{\mathsf{tp}} \mathsf{g}(y)$ is equal to $\mathsf{g}(\alpha_1 \otimes \ldots \otimes \alpha_n \otimes y)$ and therefore $\mathsf{g}(\overline{\Gamma - x} \otimes y)$, so we have $\Gamma^*, y : A^* \vdash_{\overline{\Gamma - x, y : A}} \mathsf{F}_{\mathsf{g}}(C^*)$. By Lemma 7.1, we can strengthen away x, giving a no-bigger derivation of $(\Gamma^* - x), y : A^* \vdash_{\overline{\Gamma - x, y : A}} \mathsf{F}_{\mathsf{g}}(C^*)$. By the inductive hypothesis, this translates to a derivation of $\Gamma - x, A$ true $\vdash C$ poss, so applying $\bigcirc$-left gives the result.

  If z does not occur in β' we have $\mathsf{g}(\overline{\Gamma}) \equiv \beta'$, so the premise is $\Gamma^*, z : \mathsf{F}_{\mathsf{g}}(A^*) \vdash_{\mathsf{g}(\overline{\Gamma})} \mathsf{F}_{\mathsf{g}}(C^*)$. Then strengthening z by Lemma 7.1 gives a no-bigger derivation of $\Gamma^* \vdash_{\mathsf{g}(\overline{\Gamma})} \mathsf{F}_{\mathsf{g}}(C^*)$. So the inductive hypothesis gives the result. That is, we did an elimination to produce a 0-use variable, which we can strengthen away.

□

8 Permutative Equality

While the axiomatization of equality from Section 5 is quite concise, for adequacy proofs we will need an alternative characterization that is easier to reason from. For example, in the above equational theory, it seems possible that an equation between two cut-free derivations can be proved using an intermediate term that introduces a cut at a completely unrelated formula. It turns out that this is not the case, as we can show by relating the above equational theory to the following one.

We say that a derivation is *normal* if it uses only the rules in Figure 2—i.e. it does not use the cut rule, and only uses the hypothesis rule and respect for transformations in the form $(s_*(x))$ where x has a base type. In Figure 5, we write out the proofs of respect for transformation (Lemma 4.1), identity (Theorem 4.4), left-inversion (Lemma 4.5), and cut (Theorem 4.6) as operations on normal derivations. We also include a corresponding right inversion lemma for U, which takes a derivation of $\Gamma \vdash_\beta \mathsf{U}_{x.\alpha}(\Delta \mid A)$ to $\Gamma \vdash_{\alpha[\beta/x]} A$ when Δ is isomorphic to a subcontext of Γ.

Derivations where all cuts, identities, and transformations have been expanded are not unique representatives of $\equiv$-equivalence classes: what remains is to move transformations around the derivation and permute the order of rules. We define permutative equality as the least congruence on *normal derivations* containing the rules in Figure 6.

The first two rules are the uniqueness principles for F and U, which allow moving FL and UR to the bottom of any derivation.

The next rule allows permuting UL. We write c for a *context*, which is an arbitrary normal derivation, except it is only allowed to use the variable x_0 in subterms of the form $\mathsf{id}\{x_0\}$ at the leaf of a derivation, not in any other left/identity rule:
$$c ::= \mathsf{id}\{x_0\} \mid s_*(x) \mid \mathsf{FR}(s, \overrightarrow{c_i/x_i}) \mid \mathsf{FL}^x(\Delta.c) \mid \mathsf{UR}(\Delta.c) \mid \mathsf{UL}^x(s, \overrightarrow{c_i/x_i}, z.c)$$

$$
\begin{array}{rcl}
\mathsf{FL}^x(\Delta.d)\!\downarrow & := & \mathsf{FL}^x(\Delta.d\!\downarrow) \\
\mathsf{UR}(\Delta.d)\!\downarrow & := & \mathsf{UR}(\Delta.d\!\downarrow) \\
\mathsf{FR}(s,\overrightarrow{d/x})\!\downarrow & := & \mathsf{FR}(s,\overrightarrow{d\!\downarrow/x}) \\
\mathsf{UL}^x(s,\overrightarrow{d/x},z.d)\!\downarrow & := & \mathsf{UL}^x(s,\overrightarrow{d\!\downarrow/x},z.d\!\downarrow) \\
s_*(d)\!\downarrow & := & s_*\{d\!\downarrow\} \\
e[d/x]\!\downarrow & := & e\!\downarrow\!\{d\!\downarrow/x\} \\
x\!\downarrow & := & \mathsf{id}\{x\}
\end{array}
$$

$$
\begin{array}{rcl}
s_*\{\mathsf{FR}(s',\overrightarrow{d/x})\} & := & \mathsf{FR}(s;s',\overrightarrow{d/x}) \\
s_*\{\mathsf{FL}^x(\Delta.d)\} & := & \mathsf{FL}^x(\Delta.(s[1_\alpha/x])_*\{d\}) \\
s_*\{\mathsf{UL}^x(s',\overrightarrow{d/y},z.d')\} & := & \mathsf{UL}^x(s;s',\overrightarrow{d/y},z.d') \\
s_*\{\mathsf{UR}(\Delta.d)\} & := & \mathsf{UR}(\Delta.(1_\alpha[s/x])_*\{d\})
\end{array}
$$

$$
\begin{array}{rcl}
\mathsf{id}\{x\} :: (\Gamma,x:\mathsf{F}_\alpha(\Delta),\Gamma' \vdash_x \mathsf{F}_\alpha(\Delta)) & := & \mathsf{FL}^x(\Delta.\mathsf{FR}_{\overrightarrow{\mathsf{id}\{y\}/y}}(1,\overrightarrow{\mathsf{id}\{y\}/y})) \\
\mathsf{id}\{x\} :: (\Gamma,x:\mathsf{U}_\alpha(\Delta \mid A),\Gamma' \vdash_x \mathsf{U}_\alpha(\Delta \mid A)) & := & \mathsf{UR}(\Delta.\mathsf{UL}^x(\overrightarrow{\mathsf{id}\{y\}/y},1,z.\mathsf{id}\{z\}))
\end{array}
$$

$$
\begin{array}{rcll}
s_*(x_0)\{d/x_0\} & := & s_*\{d\} & (x_0 : Q) \\
s_*(y)\{d/x_0\} & := & s_*(y) & (x_0 \neq y) \\
\mathsf{FL}^{x_0}(\Delta.e)\{\mathsf{FR}(s,\overrightarrow{d_i/x_i})/x_0\} & := & (1_\beta[s/x])_*(e\{\overrightarrow{d_i/x_i}\}) & \\
\mathsf{UL}^{x_0}(s,\vec{e}_i/x_i,z.e')\{\mathsf{UR}(\Delta.d)/x_0\} & := & (s[1_{\alpha_0}])_*(e'\{(d\{\overrightarrow{e_i/x_i}\})/z\}) & \\
\mathsf{FR}(s,e)\{d/x_0\} & := & \mathsf{FR}(s[1_{\alpha_0}/x_0],e\{d/x_0\}) & \\
\mathsf{UR}(\Delta.e)\{d/x_0\} & := & \mathsf{UR}(\Delta.e\{d/x_0\}) & \\
\mathsf{FL}^x(\Delta.e)\{d/x_0\} & := & \mathsf{FL}^x(\Delta.e\{\mathsf{linv}(d,\Delta,x)/x_0\}) & (x \neq x_0) \\
\mathsf{UL}^x(s,\vec{e}_i/x_i,z.e')\{d/x_0\} & := & \mathsf{UL}^x(s[1_{\alpha_0}/x_0],(e_i\{\overrightarrow{d/x_0}\})/x_i,z.e'\{d/x_0\}) & (x \neq x_0) \\
e\{\mathsf{FL}^x(\Delta.d)/x_0\} & := & \mathsf{FL}^x(\Delta.\mathsf{linv}(e,\Delta,x)\{d/x_0\}) & (e = \mathsf{FL}^{x_0},\mathsf{UL}^{x_0}) \\
e\{\mathsf{UL}^x(s,\overrightarrow{d/y},z.d_2)/x_0\} & := & \mathsf{UL}^x(1_\beta[s/x_0],\overrightarrow{d/y},z.e\{d_2/x_0\}) & (e = \mathsf{FL}^{x_0},\mathsf{UL}^{x_0})
\end{array}
$$

$$
\begin{array}{rcll}
\mathsf{linv}(s_*(x),\vec{x},x_0) & := & s_*(x) & \\
\mathsf{linv}(\mathsf{FL}^{x_0}(\Delta.d),\vec{x},x_0) & := & d[\Delta \leftrightarrow \vec{x}] & \\
\mathsf{linv}(\mathsf{FL}^x(\Delta.d),\vec{x},x_0) & := & \mathsf{FL}^x(\Delta.\mathsf{linv}(d,\vec{x},x_0)) & (x \neq x_0) \\
\mathsf{linv}(\mathsf{FR}(s,\overrightarrow{d_i/x_i}),\vec{x},x_0) & := & \mathsf{FR}(s[1_\alpha/x_0],\mathsf{linv}(d_i,\vec{x},y)/x_i) & \\
\mathsf{linv}(\mathsf{UR}(\Delta.d),\vec{x},x_0) & := & \mathsf{UR}(\Delta.\mathsf{linv}(d,\vec{x},x_0)) & \\
\mathsf{linv}(\mathsf{UL}^y(s,\overrightarrow{d_i/y_i},z.d),\vec{x},x_0) & := & \mathsf{UL}^y(s[1_{\alpha_0}/x_0],\mathsf{linv}(d_i,\vec{x},x_0)/y_i,z.\mathsf{linv}(d,\vec{x},x_0)) &
\end{array}
$$

$$
\begin{array}{rcl}
\mathsf{rinv}(\mathsf{UR}(\Delta.d),\vec{x}) & := & d[\Delta \leftrightarrow \vec{x}] \\
\mathsf{rinv}(\mathsf{FL}^x(\Delta.d),\vec{x}) & := & \mathsf{FL}^x(\Delta.\mathsf{rinv}(d,\vec{x})) \\
\mathsf{rinv}(\mathsf{UL}^x(s,\overrightarrow{d_i/y_i},z.d),\vec{x}) & := & \mathsf{UL}^x(1_{\alpha_0}[s/x],\overrightarrow{d_i/y_i},z.\mathsf{rinv}(d,\vec{x}))
\end{array}
$$

Figure 5: Definitions of Admissible Rules

$$
\begin{array}{rcl}
d :: \Gamma,x:\mathsf{F}_\alpha(\Delta),\Gamma' \vdash_\beta C & \equiv_{\mathsf{p}} & \mathsf{FL}^x(\Delta.\mathsf{linv}(d,\Delta,x)) \\
d :: \Gamma \vdash_\beta \mathsf{U}_{x.\alpha}(\Delta \mid A) & \equiv_{\mathsf{p}} & \mathsf{UR}(\Delta.\mathsf{rinv}(d,\Delta)) \\
c\{\mathsf{UL}^x(s,\overrightarrow{d/y},z.d_2)/x_0\} & \equiv_{\mathsf{p}} & \mathsf{UL}^x(1_\beta[s/x_0],\overrightarrow{d/y},z.c\{d_2/x_0\})
\end{array}
$$

$$
\begin{array}{rcl}
\mathsf{FR}(s,d_1/x_1,\ldots,s_{i*}(d_i)/x_i,\ldots) & \equiv_{\mathsf{p}} & \mathsf{FR}(s;(1_{\alpha[\alpha_1/x_1,\ldots,\alpha_{i-1}/x_{i-1},\ldots]}[s_i/x_i]),\overrightarrow{d_j/x_j}) \\
\mathsf{UL}^x(s,(d_1/x_1,\ldots,s_{i*}(d_i)/x_i,\ldots),z.d) & \equiv_{\mathsf{p}} & \mathsf{UL}^x(s;(1_{\beta'[\alpha[\alpha_1/x_1,\ldots,\alpha_{i-1}/x_{i-1},\ldots]/z]}[s_i/x_i]),\overrightarrow{d_j/x_j},z.d) \\
\mathsf{UL}^x(s,\overrightarrow{d/x},z.s'_*(d)) & \equiv_{\mathsf{p}} & \mathsf{UL}^x((s;s'[1_{\alpha[\gamma]}/z]),\overrightarrow{d/x},z.d)
\end{array}
$$

Figure 6: Permutative Equality

The intention is a cut $c\{d/x_0\}$ is a simple substitution, which right-commutes into c, replacing all derivation leaves of x_0 with d.

The next three rules correspond to instances of functoriality of the respect-for-transformations in UL and FR.

LEMMA 8.1.

$$\mathsf{FR}(s, \vec{d/x}) \equiv s_*(\mathsf{FR}^*[\vec{d/x}])$$

and

$$\mathsf{UL}^x(s, \vec{d/y}, z.d') \equiv s_*(d'[\mathsf{UL}^*_x[\vec{d/y}]/z])$$

Proof. For FR:

$$\begin{aligned}
\mathsf{FR}(s, \vec{d/x}) &\equiv y[\mathsf{FR}(s, \vec{d/x})/y] \\
&\equiv \mathsf{FL}^y(\Delta.y[\mathsf{FR}^*/y])[\mathsf{FR}(s, \vec{d/x})/y] \\
&\equiv \mathsf{FL}^y(\Delta.\mathsf{FR}^*)[\mathsf{FR}(s, \vec{d/x})/y] \\
&\equiv (1_y[s/y])_*(\mathsf{FR}^*[\vec{d/x}]) \\
&\equiv s_*(\mathsf{FR}^*[\vec{d/x}])
\end{aligned}$$

using the η rule, the β rule, then the unit law for horizontal composition.

For UL:

$$\begin{aligned}
\mathsf{UL}^x(s, \vec{d/y}, z.d') &\equiv \mathsf{UL}^x(s, \vec{d/y}, z.d')[x/x] \\
&\equiv \mathsf{UL}^x(s, \vec{d/y}, z.d')[\mathsf{UR}(\Delta.\mathsf{UL}^*_x[x/x])/x] \\
&\equiv \mathsf{UL}^x(s, \vec{d/y}, z.d')[\mathsf{UR}(\Delta.\mathsf{UL}^*_x)/x] \\
&\equiv (s[x/x])_*(d'[\mathsf{UL}^*_x[\vec{d/y}]/z]) \\
&\equiv s_*(d'[\mathsf{UL}^*_x[\vec{d/y}]/z])
\end{aligned}$$

by η and β for U. □

LEMMA 8.2.

$$\begin{aligned}
\mathsf{FL}^x(\Delta.d)[d'/x_0] &\equiv \mathsf{FL}^x(\Delta.d[d'/x_0]) && \textit{if } x \neq x_0 \\
\mathsf{FR}(s, \vec{d_i/x_i})[d'/x_0] &\equiv \mathsf{FR}(s[1_\alpha/x_0], (d_i[d'/x_0])/x_i) \\
\mathsf{UL}^x(s, \vec{d_i/y_i}, z.d)[d'/x_0] &\equiv \mathsf{UL}^x(s[1_{\alpha_0}/x_0], (d_i[d'/x_0])/y_i, z.d[d'/x_0]) && \textit{if } x \neq x_0 \\
\mathsf{UR}(\Delta.d)[d'/x_0] &\equiv \mathsf{UR}(\Delta.d[d'/x_0])
\end{aligned}$$

Proof.

$$\begin{aligned}
\mathsf{FL}^x(\Delta.d[d'/x_0]) &\equiv \mathsf{FL}^x(\Delta.1_*(d)[d'/x_0]) \\
&\equiv \mathsf{FL}^x(\Delta.\mathsf{FL}^x(\Delta.d)[\mathsf{FR}^*/x][d'/x_0]) \\
&\equiv \mathsf{FL}^x(\Delta.\mathsf{FL}^x(\Delta.d)[d'/x_0][\mathsf{FR}^*/x]) \\
&\equiv \mathsf{FL}^x(\Delta.d)[d'/x_0]
\end{aligned}$$

where we use β, then associativity of cut (exploiting $x \neq x_0$), then η.

$$\begin{aligned}
\mathsf{FR}(s[1_\alpha/x_0], (d_i[d'/x_0])/x_i) &\equiv (s[1_\alpha/x_0])_*(\mathsf{FR}^*[(d_i[d'/x_0])/x_i]) \\
&\equiv (s[1_\alpha/x_0])_*(\mathsf{FR}^*[d_i/x_i][d'/x_0]) \\
&\equiv s_*(\mathsf{FR}^*[d_i/x_i])[(1_\alpha)_*(d')/x_0] \\
&\equiv s_*(\mathsf{FR}^*[d_i/x_i])[d'/x_0] \\
&\equiv \mathsf{FR}(s, \vec{d_i/x_i})[d'/x_0]
\end{aligned}$$

by the previous Lemma, associativity of cut, the interchange law and the previous Lemma again.

$$\begin{aligned}\mathsf{UL}^x(s[1_{\alpha_0}/x_0],(d_i[d'/x_0])/y_i,z.d[d'/x_0]) &\equiv (s[1_{\alpha_0}/x_0])_*((d[d'/x_0])[\mathsf{UL}^*_x[d_i[d'/x_0]/y_i]/z])\\ &\equiv (s[1_{\alpha_0}/x_0])_*((d[d'/x_0])[\mathsf{UL}^*_x[d_i/y_i][d'/x_0]/z])\\ &\equiv (s[1_{\alpha_0}/x_0])_*(d[\mathsf{UL}^*_x[d_i/y_i]/z][d'/x_0])\\ &\equiv s_*(d[\mathsf{UL}^*_x[d_i/y_i]/z])[1_*(d')/x_0]\\ &\equiv s_*(d[\mathsf{UL}^*_x[d_i/y_i]/z])[d'/x_0]\\ &\equiv \mathsf{UL}^x(s,\overrightarrow{d_i/}y_i,z.d)[d'/x_0]\end{aligned}$$

by the Lemma, associativity of cut twice, the interchange law, and the Lemma.

$$\begin{aligned}\mathsf{UR}(\Delta.d[d'/x_0]) &\equiv \mathsf{UR}(\Delta.1_*(d)[d'/x_0])\\ &\equiv \mathsf{UR}(\Delta.\mathsf{UL}^*_x[\mathsf{UR}(\Delta.d)/x][d'/x_0])\\ &\equiv \mathsf{UR}(\Delta.\mathsf{UL}^*_x[(\mathsf{UR}(\Delta.d)[d'/x_0])/x])\\ &\equiv \mathsf{UR}(\Delta.d)[d'/x_0]\end{aligned}$$

by the β rule, associativity of cut and the η rule. □

LEMMA 8.3.

$$d'[\mathsf{UL}^x(s,\overrightarrow{d_i/}y_i,z.d)/x_0] \equiv \mathsf{UL}^x(1_{\alpha_0}[s/x],\overrightarrow{d_i/}y_i,z.d'[d/x_0])$$

Proof.

$$\begin{aligned}\mathsf{UL}^x(1_{\alpha_0}[s/x],\overrightarrow{d_i/}y_i,z.d'[d/x_0]) &\equiv (1_{\alpha_0}[s/x])_*(d'[d/x_0][\mathsf{UL}^*_x[\overrightarrow{d_i/}y_i]/z])\\ &\equiv (1_{\alpha_0}[s/x])_*(d'[d[\mathsf{UL}^*_x[\overrightarrow{d_i/}y_i]/z]/x_0])\\ &\equiv (1_\alpha)_*(d')[s_*(d[\mathsf{UL}^*_x[\overrightarrow{d_i/}y_i]/z])/x_0]\\ &\equiv d'[s_*(d[\mathsf{UL}^*_x[\overrightarrow{d_i/}y_i]/z])/x_0]\\ &\equiv d'[\mathsf{UL}^x(s,\overrightarrow{d_i/}y_i,z.d)/x_0]\end{aligned}$$

by the first Lemma, associativity of cut, the interchange law, and the first Lemma again. □

THEOREM 8.4. *Soundness of Permutative Equality*

1. $s_*\{d\} \equiv s_*(d)$ *(for normal d)*
2. $x \equiv \mathsf{id}\{x\}$
3. $\mathsf{linv}(d,\vec{y},x_0) \equiv d[\mathsf{FR}^*/x_0]$
4. $\mathsf{rinv}(d,\vec{y}) \equiv \mathsf{UL}^*_{x_0}[d/x_0]$
5. $e\{d/x\} \equiv e[d/x]$ *(for normal d,e)*
6. $d \equiv d{\downarrow}$
7. *If* $d \equiv_{\mathsf{p}} d'$ *then* $d \equiv d'$
8. *If* $d{\downarrow} \equiv_{\mathsf{p}} d'{\downarrow}$ *then* $d \equiv d'$.

Proof. **Respect for transformations**:

$$\begin{aligned}
s_*\{\mathsf{FR}(s',\vec{d/x})\} &= \mathsf{FR}(s;s',\vec{d/x})\\
&\equiv (s;s')_*(\mathsf{FR}^*[\vec{d/x}])\\
&\equiv s_*(s'_*(\mathsf{FR}^*[\vec{d/x}]))\\
&\equiv s_*(\mathsf{FR}(s',\vec{d/x}))
\end{aligned}$$

using the first Lemma and functoriality of $(-)_*$.

$$\begin{aligned}
s_*\{\mathsf{FL}^x(\Delta.d)\} &= \mathsf{FL}^x(\Delta.(s[1_\alpha/x])_*\{d\})\\
&\equiv \mathsf{FL}^x(\Delta.(s[1_\alpha/x])_*(d))\\
&\equiv \mathsf{FL}^x(\Delta.(s[1_\alpha/x])_*(\mathsf{FL}^x(\Delta.d)[\mathsf{FR}^*/x]))\\
&\equiv \mathsf{FL}^x(\Delta.s_*(\mathsf{FL}^x(\Delta.d))[(1_\alpha)_*(\mathsf{FR}^*)/x])\\
&\equiv \mathsf{FL}^x(\Delta.s_*(\mathsf{FL}^x(\Delta.d))[\mathsf{FR}^*/x])\\
&\equiv s_*(\mathsf{FL}^x(\Delta.d))
\end{aligned}$$

using the β law, the interchange law, and the η law.

$$\begin{aligned}
s_*\{\mathsf{UL}^x(s',\vec{d/y},z.d')\} &= \mathsf{UL}^x(s;s',\vec{d/y},z.d')\\
&\equiv (s;s')_*(d'[\mathsf{UL}^*_x[\vec{d/y}]/z])\\
&\equiv s_*(s'_*(d'[\mathsf{UL}^*_x[\vec{d/y}]/z]))\\
&\equiv s_*(\mathsf{UL}^x(s',\vec{d/y},z.d'))
\end{aligned}$$

again by the first Lemma.

$$\begin{aligned}
s_*\{\mathsf{UR}(\Delta.d)\} &= \mathsf{UR}(\Delta.(1_\alpha[s/x])_*\{d\})\\
&\equiv \mathsf{UR}(\Delta.(1_\alpha[s/x])_*(d))\\
&\equiv \mathsf{UR}(\Delta.(1_\alpha[s/x])_*(\mathsf{UL}^*_x[\mathsf{UR}(\Delta.d)/x]))\\
&\equiv \mathsf{UR}(\Delta.(1_\alpha)_*(\mathsf{UL}^*_x)[s_*(\mathsf{UR}(\Delta.d))/x])\\
&\equiv \mathsf{UR}(\Delta.\mathsf{UL}^*_x[s_*(\mathsf{UR}(\Delta.d))/x])\\
&\equiv s_*(\mathsf{UR}(\Delta.d))
\end{aligned}$$

again by β followed by η.

Identity: This is straightforward: in both cases apply the induction hypothesis and then the η law.
If $x : \mathsf{F}_\alpha(\Delta)$, then

$$\begin{aligned}
\mathsf{id}\{x\} &= \mathsf{FL}^x(\Delta.\mathsf{FR}_{\mathsf{id}\{\vec{y}\}/y}(1,\mathsf{id}\{\vec{y}\}/y))\\
&\equiv \mathsf{FL}^x(\Delta.\mathsf{FR}_{\vec{y/y}}(1,\vec{y/y}))\\
&\equiv \mathsf{FL}^x(\Delta.\mathsf{FR}^*)\\
&\equiv x
\end{aligned}$$

If $x : \mathsf{U}_\alpha(\Delta \mid A)$, then

$$\begin{aligned}\mathsf{id}\{x\} &= \mathsf{UR}(\Delta.\mathsf{UL}^x(\overrightarrow{\mathsf{id}\{y\}/y}, 1, z.\mathsf{id}\{z\})) \\ &\equiv \mathsf{UR}(\Delta.\mathsf{UL}^x(\overrightarrow{y/y}, 1, z.z)) \\ &\equiv \mathsf{UR}(\Delta.\mathsf{UL}^*_x) \\ &\equiv x\end{aligned}$$

Left invertibility:

$$\begin{aligned}\mathsf{linv}(s_*(x), \vec{x}, x_0) &= s_*(x) \\ &\equiv s_*(x)[\mathsf{FR}^*/x_0]\end{aligned}$$

as x_0 does not appear in $s_*(x)$.

$$\begin{aligned}\mathsf{linv}(\mathsf{FL}^{x_0}(\Delta.d), \vec{x}, x_0) &= d[\Delta \leftrightarrow \vec{x}] \\ &\equiv \mathsf{FL}^{x_0}(\Delta.d)[\mathsf{FR}^*/x_0]\end{aligned}$$

by η.

If $x \neq x_0$:

$$\begin{aligned}\mathsf{linv}(\mathsf{FL}^{x}(\Delta.d), \vec{x}, x_0) &= \mathsf{FL}^x(\Delta.\mathsf{linv}(d, \vec{x}, x_0)) \\ &\equiv \mathsf{FL}^x(\Delta.d[\mathsf{FR}^*/x_0]) \\ &\equiv \mathsf{FL}^x(\Delta.d)[\mathsf{FR}^*/x_0]\end{aligned}$$

by the second Lemma.

$$\begin{aligned}\mathsf{linv}(\mathsf{FR}(s, \overrightarrow{d_i/x_i}), \vec{x}, x_0) &= \mathsf{FR}(s[1_\alpha/x_0], \mathsf{linv}(d_i, \vec{x}, y)/x_i) \\ &\equiv \mathsf{FR}(s[1_\alpha/x_0], (d_i[\mathsf{FR}^*/x_0])/x_i) \\ &\equiv \mathsf{FR}(s, \overrightarrow{d_i/x_i})[\mathsf{FR}^*/x_0]\end{aligned}$$

by the second Lemma.

$$\begin{aligned}\mathsf{linv}(\mathsf{UR}(\Delta.d), \vec{x}, x_0) &= \mathsf{UR}(\Delta.\mathsf{linv}(d, \vec{x}, x_0)) \\ &\equiv \mathsf{UR}(\Delta.d[\mathsf{FR}^*/x_0]) \\ &\equiv \mathsf{UR}(\Delta.d)[\mathsf{FR}^*/x_0]\end{aligned}$$

by the second Lemma.

$$\begin{aligned}\mathsf{linv}(\mathsf{UL}^y(s, \overrightarrow{d_i/y_i}, z.d), \vec{x}, x_0) &= \mathsf{UL}^y(s[1_{\alpha_0}/x_0], \mathsf{linv}(d_i, \vec{x}, x_0)/y_i, z.\mathsf{linv}(d, \vec{x}, x_0)) \\ &\equiv \mathsf{UL}^y(s[1_{\alpha_0}/x_0], (d_i[\mathsf{FR}^*/x_0])/y_i, z.d[\mathsf{FR}^*/x_0]) \\ &\equiv \mathsf{UL}^y(s, \overrightarrow{d_i/y_i}, z.d)[\mathsf{FR}^*/x_0]\end{aligned}$$

by the second Lemma.

Right invertibility:

$$\begin{aligned}\mathsf{rinv}(\mathsf{UR}(\Delta.d),\vec{x}) &= d[\Delta \leftrightarrow \vec{x}]\\ &\equiv \mathsf{UL}^*_{x_0}[\mathsf{UR}(\Delta.d)/x_0]\end{aligned}$$

by β.

$$\begin{aligned}\mathsf{rinv}(\mathsf{FL}^x(\Delta.d),\vec{x}) &= \mathsf{FL}^x(\Delta.\mathsf{rinv}(d,\vec{x}))\\ &\equiv \mathsf{FL}^x(\Delta.\mathsf{UL}^*_{x_0}[d/x_0])\\ &\equiv \mathsf{FL}^x(\Delta.\mathsf{UL}^*_{x_0}[(\mathsf{FL}^x(\Delta.d)[\mathsf{FR}^*/x])/x_0])\\ &\equiv \mathsf{FL}^x(\Delta.\mathsf{UL}^*_{x_0}[\mathsf{FL}^x(\Delta.d)/x_0][\mathsf{FR}^*/x])\\ &\equiv \mathsf{UL}^*_{x_0}[\mathsf{FL}^x(\Delta.d)/x_0]\end{aligned}$$

by β, associativity of cut and η.

$$\begin{aligned}\mathsf{rinv}(\mathsf{UL}^x(s,\overrightarrow{d_i/y_i},z.d),\vec{x}) &= \mathsf{UL}^x(1_{\alpha_0}[s/x],\overrightarrow{d_i/y_i},z.\mathsf{rinv}(d,\vec{x}))\\ &\equiv \mathsf{UL}^x(1_{\alpha_0}[s/x],\overrightarrow{d_i/y_i},z.\mathsf{UL}^*_{x_0}[d/x_0])\\ &\equiv \mathsf{UL}^*_{x_0}[\mathsf{UL}^x(s,\overrightarrow{d_i/y_i},z.d)/x_0]\end{aligned}$$

by the third Lemma.

Cut: The first few cases are immediate by applying the above. For the remainder:

$$\begin{aligned}\mathsf{FR}(s,e)\{d/x_0\} &= \mathsf{FR}(s[1_{\alpha_0}/x_0],e\{d/x_0\})\\ &\equiv \mathsf{FR}(s[1_{\alpha_0}/x_0],e[d/x_0])\\ &\equiv \mathsf{FR}(s,e)[d/x_0]\end{aligned}$$

by the second Lemma.

$$\begin{aligned}\mathsf{UR}(\Delta.e)\{d/x_0\} &= \mathsf{UR}(\Delta.e\{d/x_0\})\\ &\equiv \mathsf{UR}(\Delta.e[d/x_0])\\ &\equiv \mathsf{UR}(\Delta.e)[d/x_0]\end{aligned}$$

by the second Lemma.

If $x \neq x_0$

$$\begin{aligned}\mathsf{FL}^x(\Delta.e)\{d/x_0\} &= \mathsf{FL}^x(\Delta.e\{\mathsf{linv}(d,\Delta,x)/x_0\})\\ &\equiv \mathsf{FL}^x(\Delta.e[d[\mathsf{FR}^*/x]/x_0])\\ &\equiv \mathsf{FL}^x(\Delta.\mathsf{FL}^x(\Delta.e)[\mathsf{FR}^*/x][d[\mathsf{FR}^*/x]/x_0])\\ &\equiv \mathsf{FL}^x(\Delta.\mathsf{FL}^x(\Delta.e)[d/x_0][\mathsf{FR}^*/x])\\ &\equiv \mathsf{FL}^x(\Delta.e)[d/x_0]\end{aligned}$$

by β applied to e, associativity of cut, and η.

If $x \neq x_0$

$$\begin{aligned}\mathsf{UL}^x(s,\vec{e}_i/x_i,z.e')\{d/x_0\} &= \mathsf{UL}^x(s[1_{\alpha_0}/x_0],(e_i\{\overrightarrow{d/x_0}\})/x_i,z.e'\{d/x_0\})\\ &\equiv \mathsf{UL}^x(s[1_{\alpha_0}/x_0],(e_i[\overrightarrow{d/x_0}])/x_i,z.e'[d/x_0])\\ &\equiv \mathsf{UL}^x(s,\vec{e}_i/x_i,z.e')[d/x_0]\end{aligned}$$

by the second Lemma.

For either $e = \mathsf{FL}^{x_0}$ or UL^{x_0}:

$$\begin{aligned}
e\{\mathsf{FL}^x(\Delta.d)/x_0\} &= \mathsf{FL}^x(\Delta.\mathsf{linv}(e,\Delta,x)\{d/x_0\}) \\
&\equiv \mathsf{FL}^x(\Delta.e[\mathsf{FR}^*/x][d/x_0]) \\
&\equiv e[\mathsf{FL}^x(\Delta.d)/x_0]
\end{aligned}$$

by the second Lemma.

For either $e = \mathsf{FL}^{x_0}$ or UL^{x_0}:

$$\begin{aligned}
e\{\mathsf{UL}^x(s,\vec{d/y},z.d_2)/x_0\} &= \mathsf{UL}^x(1_\beta[s/x_0],\vec{d/y},z.e\{d_2/x_0\}) \\
&\equiv \mathsf{UL}^x(1_\beta[s/x_0],\vec{d/y},z.e[d_2/x_0]) \\
&\equiv e[\mathsf{UL}^x(s,\vec{d/y},z.d_2)/x_0]
\end{aligned}$$

by the third Lemma.

Cut elimination: This is immediate by induction and the previous equalities.

Permutative equality: For this we verify that the given equations for $\equiv_{\mathsf{p}}$ also hold for $\equiv$. The first two equations are exactly the η rules for F and U, after applying the above equations for linv and rinv. For the remainder:

$$\begin{aligned}
c\{\mathsf{UL}^x(s,\vec{d/y},z.d_2)/x_0\} &\equiv c[\mathsf{UL}^x(s,\vec{d/y},z.d_2)/x_0] \\
&\equiv \mathsf{UL}^x(1_\beta[s/x_0],\vec{d/y},z.c[d_2/x_0]) \\
&\equiv \mathsf{UL}^x(1_\beta[s/x_0],\vec{d/y},z.c\{d_2/x_0\})
\end{aligned}$$

by the third Lemma.

For the following, let $[\widehat{d_j/x_j}]$ denote the substitution $[d_1/x_1,\ldots,d_{i-1}/x_{i-1},d_{i+1}/x_{i+1},\ldots]$ that drops d_i/x_i. In each case we apply the Lemma, then interchange, then the Lemma again.

$$\begin{aligned}
\mathsf{FR}(s,d_1/x_1,\ldots,s_{i*}(d_i)/x_i,\ldots) &\equiv s_*(\mathsf{FR}^*[\widehat{d_j/x_j}][s_{i*}(d_i)/x_i]) \\
&\equiv s_*((1_{\alpha[\alpha_1/x_1,\ldots]})_*(\mathsf{FR}^*[\widehat{d_j/x_j}])[s_{i*}(d_i)/x_i]) \\
&\equiv s_*((1_{\alpha[\alpha_1/x_1,\ldots]}[s_i/x_i])_*(\mathsf{FR}^*[\widehat{d_j/x_j}][d_i/x_i])) \\
&\equiv s_*((1_{\alpha[\alpha_1/x_1,\ldots]}[s_i/x_i])_*(\mathsf{FR}^*[\vec{d_j/x_j}])) \\
&\equiv (s;(1_{\alpha[\alpha_1/x_1,\ldots]}[s_i/x_i]))_*(\mathsf{FR}^*[\vec{d_j/x_j}]) \\
&\equiv \mathsf{FR}(s;(1_{\alpha[\alpha_1/x_1,\ldots]}[s_i/x_i]),\vec{d_j/x_j})
\end{aligned}$$

For UL:

$$
\begin{aligned}
\mathsf{UL}^x(s,(d_1/x_1,\ldots,s_{i*}(d_i)/x_i,\ldots),z.d) &\equiv s_*(d[\mathsf{UL}^*_x[\widehat{d_j/x_j}][s_{i*}(d_i)/x_i]/z])\\
&\equiv s_*(d[\mathsf{UL}^*_x[\widehat{d_j/x_j}]/z][s_{i*}(d_i)/x_i])\\
&\equiv s_*((1_{\beta'[\alpha[\alpha_1/x_1,\ldots]/z]})_*(d[\mathsf{UL}^*_x[\widehat{d_j/x_j}]/z])[s_{i*}(d_i)/x_i])\\
&\equiv s_*((1_{\beta'[\alpha[\alpha_1/x_1,\ldots]/z]}[s_i/x_i])_*(d[\mathsf{UL}^*_x[\widehat{d_j/x_j}]/z][d_i/x_i]))\\
&\equiv s_*((1_{\beta'[\alpha[\alpha_1/x_1,\ldots]/z]}[s_i/x_i])_*(d[\mathsf{UL}^*_x[\vec{d_j/x_j}]/z]))\\
&\equiv (s;(1_{\beta'[\alpha[\alpha_1/x_1,\ldots]/z]}[s_i/x_i]))_*(d[\mathsf{UL}^*_x[\vec{d_j/x_j}]/z])\\
&\equiv \mathsf{UL}^x(s;(1_{\beta'[\alpha[\alpha_1/x_1,\ldots]/z]}[s_i/x_i]),\vec{d_j/x_j},z.d)
\end{aligned}
$$

And finally:

$$
\begin{aligned}
\mathsf{UL}^x(s,\vec{d/x},z.s'_*(d)) &\equiv s_*(s'_*(d)[\mathsf{UL}^*_x[\vec{d_i/y_i}]/z])\\
&\equiv s_*(s'_*(d)[(1_{\alpha[\gamma]})_*(\mathsf{UL}^*_x[\vec{d_i/y_i}])/z])\\
&\equiv s_*((s'[1_{\alpha[\gamma]}/z])_*(d[\mathsf{UL}^*_x[\vec{d_i/y_i}]/z]))\\
&\equiv (s;s'[1_{\alpha[\gamma]}/z])_*(d[\mathsf{UL}^*_x[\vec{d_i/y_i}]/z])\\
&\equiv \mathsf{UL}^x((s;s'[1_{\alpha[\gamma]}/z]),\vec{d/x},z.d)
\end{aligned}
$$

□

CONJECTURE 8.5. *Completeness of Permutative Equality.*
If $d \equiv d'$ *then* $d{\downarrow} \equiv_{\mathsf{p}} d'{\downarrow}$

9 Equational Adequacy

9.1 Template

In addition to the logical adequacy results above, we expect that the translation from an object logic into the framework extends to something like a full and faithful functor from the object logic to the framework. Unpacking this, the object part of the functor means we want a translation A^* from object language types to framework types—and an extension translating object-language sequents J to framework sequents J^*. The morphism part of the functor maps each object-logic derivation $d : J$ to a derivation $d^* : J^*$. Functoriality means that the translation takes identities to identities and cuts to cuts. Together, full and faithfullness say that for each sequent J, the object language derivations of J are bijective with framework derivations of J^*. In particular, fullness says that for any sequent J, the translation on derivations of that sequent is surjective: for every derivation e of J^*, there (merely) exists an object language derivation $d : J$ such that $d^* = e$. In terms of provability, this says that no more sequents can be proved in the framework, and in terms of proof identity, it says that every derivation could have been written in the object language. Faithfullness says that the translation on derivations is injective—$d_1^* = d_2^*$ implies $d_1 = d_2$—so no more equalities can be proved in the framework. The fact that a function is a bijection iff it is surjective and injective gives the overall result.

In the above discussion, we would like equality of derivations to correspond to the categorical universal properties for the connectives, which generally equate more morphisms than syntactic equality of cut-free proofs (unless one uses more sophisticated sequent calculi than we consider here, e.g. focusing/multifocusing). On the framework side, the equational theory of Section 5 already accounts for this. On the source side, will define a logic by the usual sequent calculus rules that make cut and identity admissible, along with primitive cut and identity rules, and an equality judgement analogous to Section 5, which is a concise description of $\beta\eta$ rules. The presentation of framework equality in Section 8 is helpful for translating framework equalities back to the source. Cut elimination for the source will be a corollary of the adequacy theorem (we could simplify the source syntax by removing the built-in cuts in the non-invertible rules, using the general cut rule in their place, but including them is convenient for stating the cut elimination corollary). Thus, we refine the discussion above by taking equality of derivations to be $\equiv$-classes.

We will generally focus on the following aspects of constructing such a full and faithful functor:

DEFINITION 9.1: THE INTERESTING PART OF AN ADEQUACY PROOF.

1. *The translation from types to types (A^*) and sequents to sequents (J^*).*
2. *For each source inference rule for each connective, a* derivation *d^* from the translated premises to the translated conclusion (not just an admissibility: each rule will be defined by a composition of framework inference rules).*
3. *A proof that equality axioms are preserved: for each connective-specific equality axiom (typically $\beta\eta$) $d_1 \equiv d_2$, $d_1^* \equiv d_2^*$.*
4. *A function $-^{\leftarrow}$ from normal derivations $e : J^*$ to source derivations of J. If the output does not use the cut rule, or identity at non-base-types, this gives cut and identity elimination for the source as a corollary.*
5. *A proof that $e^{\leftarrow *} \equiv e$.*
6. *A proof that the meta-operations are preserved by back-translation. For example, for identity $\mathsf{id}\{x\} : J^*$, we should have $\mathsf{id}\{x\}^{\leftarrow} \equiv x$. For normal e and e' in the image of "cutable" sequents J^* and J'^*, $(e\{e'/x\})^{\leftarrow} \equiv e^{\leftarrow}[e'^{\leftarrow}/x]$ (Note: this can be stated for $d^*\downarrow$ and $d'^*\downarrow$ if that is more convenient).*
7. *A proof that $(d^*\downarrow)^{\leftarrow} \equiv d$. The cases for identity and cut will use the previous bullet.*
8. *A proof that for normal $e, e' : J^*$, $e \equiv_{\mathsf{p}} e$ implies $e^{\leftarrow} \equiv e'^{\leftarrow}$. (This can be stated for $d^*\downarrow$ if that is convenient.)*

From this, the full construction is as follows:

REMARK 9.2: THE ROUTINE PART OF AN ADEQUACY PROOF.

1. *For the construction of the functor:*
   (a) *The translation of types and sequents was given in part 1 above.*
   (b) *The cases of the translation of derivations d^* given above are extended by sending identity to identity and cut to cut (possibly with some weakening-over-weakening and exchange-over-exchange), to determine a function from cutfull source derivations to cutfull framework derivations. So functoriality is true by definition.*
   (c) *We extend the above function d^* on derivations to $\equiv$-equivalence classes by proving $d_1 \equiv d_2$ implies $d_1^* \equiv d_2^*$. The type-specific cases are given by part 3 above. Reflexivity, symmetry, and transitivity are sent to reflexivity, symmetry, transitivity rules in the framework. The congruence rule for each source derivation constructor is sent to a composition of framework congruence rules, which works because because each inference rule is shown derivable (not just admissible) in part 2 above. The unit and associativity laws for cut will be modeled by the corresponding laws in the framework.*
2. *For fullness, every e is equal (by Theorem 8.4) to a cut/identity/transformation-free derivation $e\downarrow$, and the proof for cut-free derivations is given by $-^{\leftarrow}$ (parts 4 and 5 above). Even though we are constructing a bijection between derivations modulo $\equiv$, we do not need to show that this function respects the quotient: because of the "mere existence"/-1-truncation in the definition of surjective, the function on representatives automatically extends to the quotient.*

   If $-^{\leftarrow}$ does not use the cut rule in the source (or identity at non-base-type), then the composite $d^*\downarrow^{\leftarrow}$ witnesses cut/identity elimination for the source.
3. *For faithfulness, we need to show that $d_1^* \equiv d_2^*$ implies $d_1 \equiv d_2$. By part 8 above, it suffices to show $d_1^*\downarrow^{\leftarrow} \equiv d_2^*\downarrow^{\leftarrow}$. By completeness of permutative equality (Theorem 8.5), $d_1^* \equiv d_2^*$ implies $d_1^*\downarrow \equiv_{\mathsf{p}} d_2^*\downarrow$, so part 9 above gives the result.*

We do not abstract this "template" as a lemma because the class of "native sequent calculi" taken as input is not precisely defined.

LEMMA 9.3: EQUATIONAL 0-USE STRENGTHING. *Under the conditions of Lemma 7.1, write* $\mathsf{str}_{\vec{x}}(d)$ *for the derivation produced by the lemma. Then* $\mathsf{str}_{\vec{x}}(d)$ *(weakened with* $\vec{x}$*)* $\equiv d$.

Proof. Inspecting the proof, we have the following reductions:

$$\begin{array}{rcll}
\mathsf{str}_{\vec{x}}(\beta_*(y)) & := & \beta_*(y) & \\
\mathsf{str}_{\vec{x}}(\mathsf{FR}(s,\vec{d})) & := & \mathsf{FR}(s,\mathsf{str}_{\vec{x}}(\vec{d})) & \\
\mathsf{str}_{\vec{x}}(\mathsf{FL}^x(\Delta.d)) & := & \mathsf{str}_{\vec{x},x,\Delta}(d) & \text{if } x \in \vec{x} \\
\mathsf{str}_{\vec{x}}(\mathsf{FL}^x(\Delta.d)) & := & \mathsf{FL}^x(\Delta.\mathsf{str}_{\vec{x}}(d)) & \text{if } x \notin \vec{x} \\
\mathsf{str}_{\vec{x}}(\mathsf{UR}(\Delta.d)) & := & \mathsf{UR}(\Delta.\mathsf{str}_{\vec{x}}(d)) & \\
\mathsf{str}_{\vec{x}}(\mathsf{UL}^x_s(\vec{d},z.d',)) & := & s_*(\mathsf{str}_{\vec{x},z}(d')) & \text{if } z\#\beta' \\
\mathsf{str}_{\vec{x}}(\mathsf{UL}^x(s,\vec{d},z.d')) & := & \mathsf{UL}^x(s,\mathsf{str}_{\vec{x}}(d),z.\mathsf{str}_{\vec{x}}(d')) & \text{if } z \in \beta', x \notin \vec{x}
\end{array}$$

Most cases follow from the inductive hypothesis; the interesting ones are where a rule is deleted.

- Suppose we began with $\mathsf{FL}^x(\Delta.d) : \Gamma, x : \mathsf{F}_\alpha(\Delta) \vdash_\beta B$ with $x \in \vec{x}$ and produced $\mathsf{str}_{\vec{x},x,\Delta}(d) : \Gamma - \vec{x} \vdash_\beta B$ (because neither x nor Δ occur in β). Then weakening with x, Δ gives $\mathsf{str}_{\vec{x},x,\Delta}(d) : \Gamma, x : \mathsf{F}_\alpha(\Delta), \Delta \vdash_\beta B$, and the inductive hypothesis gives that this is equal to d.

  Thus, it suffices to show that for any $e : \Gamma, x : \mathsf{F}_\alpha(\Delta) \vdash_\beta B$ where x doesn't occur in β or e, then e is equal to $\mathsf{FL}^x(\Delta.e)$—i.e. we can introduce a "dead branch" on x. But by the η rule we have $e \equiv \mathsf{FL}^x(e[\mathsf{FR}/x])$, and the cut cancels because x does not occur.

- Suppose we began with $\mathsf{UL}^x(s,\vec{d},z.d') : \Gamma, x : \mathsf{U}_\alpha(\Delta \mid A) \vdash_\beta B$ and z does not occur in β', the resources of d', and we produce $s_*(\mathsf{str}_{\vec{x},z}(d'))$, and want to show this is equal to $\mathsf{UL}^x(s,\vec{d},z.d')$. By the IH, weakening $\mathsf{str}_{\vec{x},z}(d')$ is equal to d'.

  Thus, it suffices to show that $\mathsf{UL}^x(s,\vec{d},z.d') \equiv s_*(d')$ when z does not occur in the *derivation* d'. This is immediate from the semantic expansion of UL

  $$\mathsf{UL}^x(s,\vec{d/y},z.d') \equiv s_*((d'[\mathsf{UL}^*_x[\vec{d/y}]/z]))$$

  because in this the substitution for z, which does not occur, cancels.

□

LEMMA 9.4: PROPERTIES OF STRENGTHENING.

- $\mathsf{str}_\Gamma(\mathsf{str}_\Delta(e)) = \mathsf{str}_{\Gamma,\Delta}(e)$
- $\mathsf{str}_{\Delta,x}(d) = \mathsf{str}_{\Delta,x,\Delta'}(\mathsf{linv}(d,\Delta',x))$
- $\mathsf{str}_\Delta(\mathsf{linv}(d,\Delta',x)) = \mathsf{linv}(\mathsf{str}_\Delta(d),\Delta',x)$ *if* $x\#\Delta$
- $\mathsf{str}_{\vec{x}}(y) = y$ *if* $y \notin x$.
- $\mathsf{str}_{\vec{x}}(d) = d$ *if* $\vec{x}\#d$ *and for every* UL *in* d*,* z *occurs in* β'.

Proof. For the first part, deleting rules in two passes is the same one, as long as all of the same variables are deleted overall.

For the second part, deleting all uses of x is the same as first inverting x (which deletes all left rules on it) and then deleting all uses of the variables that are produced by the inversion (which are deleted when strengthening hits FL^x).

For the third, left inversion commutes with deleting variables other than the ones being deleted.

For the fourth, the η-expanded identity on y contains no variables beside y and those introduced by rules in it, so $\mathsf{str}_{\vec{x}}()$ deletes nothing.

For the fifth, strengthening only deletes (1) left rules on x and subsidiaries, and (2) unneeded occurrences of UL, and the premises say that there are none of these. □

LEMMA 9.5: CUT DOES NOT INTRODUCE LEFT RULES. *If* FL^x *or* UL^x *occurs in* $d\downarrow$*, then it occurs in* d

Proof. By induction on the execution trace of cut elimination: for each rule in Figure 5, FL^x or UL^x occurs on the right only if it occurs on the left. □

9.2 Ordered Logic (Product Only)

As a first example of an adequacy proof, we consider the following mode theory for ordered logic with only $A \odot B$:

$$\frac{}{A \vdash^{\circ} A} \qquad \frac{\Gamma, A, \Gamma' \vdash^{\circ} C \quad \Delta \vdash^{\circ} A}{\Gamma, \Delta, \Gamma' \vdash^{\circ} C} \qquad \frac{\Gamma, A, B, \Gamma' \vdash^{\circ} C}{\Gamma, A \odot B, \Gamma' \vdash^{\circ} C} \qquad \frac{\Gamma \vdash^{\circ} A \quad \Delta \vdash^{\circ} B}{\Gamma, \Delta \vdash^{\circ} A \odot B}$$

$$\odot\mathsf{L}^z(x, y.d)[\odot\mathsf{R}(d_1, d_2)/z] \equiv d[d_1/x][d_2/y]$$
$$d : \Gamma, z : A \odot B, \Gamma' \vdash^{\circ} C \equiv \odot\mathsf{L}^z(x, y.d[\odot\mathsf{R}(x, y)/z])$$

We use a mode theory with a monoid $(\odot, 1)$, so the only transformation axioms are equality axioms for associativity and unit.

The interesting parts of the adequacy proof are:

1. The type translation is given by $P^* := P$ and $(A \odot B)^* := \mathsf{F}_{x \odot y}(x : A^*, y : B^*)$. A context $(x_1 : A_1, \ldots, x_n : A_n)^* := x_1 : A_1^*, \ldots, x_n : A_n^*$. Writing $\overline{x_1 : A_1, \ldots, x_n : A_n} := x_1 \odot \ldots \odot x_n$, a sequent $\Gamma \vdash^{\circ} A$ is translated to $\Gamma^* \vdash_{\overline{\Gamma}} A^*$.

   We use the following properties of the mode theory:

   - If $\overline{\Gamma^*} \equiv x$ then Γ is $x : Q$ for some Q.
   - If $\overline{\Gamma} \equiv \alpha_1 \odot \alpha_2$, then there exist Γ_1, Γ_2 such that $\Gamma = \Gamma_1, \Gamma_2$ and $\overline{\Gamma_1} \equiv \alpha_1$ and $\overline{\Gamma_2} \equiv \alpha_2$.
   - A^* and Γ^* are relevant propositions, and the monoid axioms preserve variables, so by Lemma 7.1 we can strengthen away any variables that are not in the context descriptor.

2. As discussed in Section 7.1, the inference rules for $\odot$ are derived as follows:

$$\dfrac{\dfrac{\Gamma^*, x : A, y : B, \Gamma'^* \vdash_{\overline{\Gamma} \odot x \odot y \odot \overline{\Gamma'}} C}{\Gamma^*, \Gamma'^*, x : A, y : B \vdash_{\overline{\Gamma} \odot x \odot y \odot \overline{\Gamma'}} C}\ \textit{Lemma}\ 4.3}{\Gamma^*, z : \mathsf{F}_{x \odot y}(x : A^*, y : B^*), \Gamma'^* \vdash_{\overline{\Gamma} \odot z \odot \overline{\Gamma'}} C}\ \mathsf{FL}$$

$$\dfrac{\overline{\Gamma} \odot \overline{\Delta} \Rightarrow (x \odot y)[\overline{\Gamma}/x, \overline{\Delta}/y] \quad \dfrac{\Gamma^* \vdash^{\overline{\Gamma}} A}{\Gamma^*, \Delta^* \vdash^{\overline{\Gamma}} A}\ \textit{Lemma}\ 4.2 \quad \dfrac{\Delta^* \vdash^{\overline{\Delta}} B}{\Gamma^*, \Delta^* \vdash^{\overline{\Gamma}} A}\ \textit{Lemma}\ 4.2}{\Gamma^*, \Delta^* \vdash_{\overline{\Gamma} \odot \overline{\Delta}} \mathsf{F}_{x \odot y}(x : A, y : B)}$$

   Identity and cut are

$$\frac{}{x : A^* \vdash_x A}\ \textit{Thm}\ 4.4 \qquad \dfrac{\dfrac{\Gamma^*, x : A^*, \Gamma'^* \vdash_{\overline{\Gamma} \odot x \odot \overline{\Gamma'}} C}{\Gamma^*, \Delta^*, x : A^*, \Gamma'^* \vdash_{\overline{\Gamma} \odot x \odot \overline{\Gamma'}} C}\ \textit{Lem}\ 4.2 \quad \dfrac{\Delta^* \vdash_{\overline{\Delta}} A^*}{\Gamma^*, \Delta^*, x : A^*, \Gamma'^* \vdash_{\overline{\Delta}} A^*}\ \textit{Lem}\ 4.2}{\Gamma^*, \Delta^*, \Gamma'^* \vdash_{\overline{\Gamma} \odot \overline{\Delta} \odot \overline{\Gamma'}} C}\ \textit{Thm}\ 4.6$$

   Since we do not notate weakening and exchange, we can summarize these as:

$$\begin{array}{rcl} (\odot\mathsf{L}^z(x, y.d))^* & := & \mathsf{FL}^z(x, y.d^*) \\ (\odot\mathsf{R}(d_1, d_2))^* & := & \mathsf{FR}(1, (d_1^*/x, d_2^*/y)) \\ x^* & := & x \\ (e[d/x])^* & := & e^*[d^*/x] \end{array}$$

3. The $\beta\eta$ axioms for $\odot$ translate almost exactly to the corresponding axioms for $\mathsf{F}_{x\odot y}(x : A^*, y : B^*)$: for β, we also use the fact that $1_*(-)$ is the identity.

4. For the back-translation on normal derivations, suppose we have a normal derivation of $\Gamma^* \vdash_{\overline{\Gamma}} A^*$. Because there are no U-formulas in the context, the only possible rules are hypothesis and the F-rules.

   - For identity
   $$\frac{\overline{\Gamma^*} \Rightarrow x \quad x : P \in \Gamma^*}{\Gamma^* \vdash_{\overline{\Gamma}} P}$$
   Because the only structural transformation axioms are equalities for associativity and unit, we have $\overline{\Gamma^*} \equiv x$, which in turn implies that Γ is $x : Q$ for some Q (because if Γ is empty, does not contain x, or contains anything else, $\overline{\Gamma}$ will not equal x). By definition, this implies $Q = P$, so Γ is $x : P$. Therefore the identity rule applies.
   - For FR, because the only type that encodes to Fis $\odot$, we have
   $$\frac{\overline{\Gamma} \equiv \alpha_1 \odot \alpha_2 \quad \Gamma^* \vdash_{\alpha_1} A_1^* \quad \Gamma^* \vdash_{\alpha_2} A_2^*}{\Gamma^* \vdash_{\overline{\Gamma}} \mathsf{F}_{x\otimes y}(x : A_1^*, y : A_2^*)}$$
   By properties of the mode theory, $\Gamma = \Gamma_1, \Gamma_2$ with $\overline{\Gamma_i} \equiv \alpha_i$, so we have derivations of $\Gamma^* \vdash_{\overline{\Gamma_i}} A_i^*$. Because 0-use strengthening applies, we can strengthen these to $\Gamma_i^* \vdash_{\overline{\Gamma_i}} A_i^*$. Then the inductive hypothesis gives $\Gamma_i \vdash^{\mathsf{A_i}}$, so applying the $\odot$ right rule gives the result.
   - For FL, because the only type encoding to F is $A \odot B$, we have
   $$\frac{\Gamma^*, \Gamma'^*, x : A^*, y : B^* \vdash_{\overline{\Gamma} \otimes (x \otimes y) \otimes \overline{\Gamma'}} C^*}{\Gamma^*, z : \mathsf{F}_{x\odot y}(x : A^*, y : B^*), \Gamma'^* \vdash_{\overline{\Gamma} \otimes z \otimes \overline{\Gamma'}} C^*}$$
   By exchange (Lemma 4.3), we have a no-bigger derivation of $\Gamma^*, x : A^*, y : B^*, \Gamma'^* \vdash_{\overline{\Gamma} \otimes (x \otimes y) \otimes \overline{\Gamma'}} C^*$ so applying the IH gives $\Gamma, x : A, y : B, \Gamma' \vdash^{\circ} C$, and then $\odot$-left gives the result.

   That is,
   $$\begin{array}{rcl} 1_*(x)^{\leftarrow} & := & x \\ \mathsf{FR}(1, e_1/x, e_2/y)^{\leftarrow} & := & \odot\mathsf{R}(\mathsf{str}(e_1)^{\leftarrow}, \mathsf{str}(e_2)^{\leftarrow}) \\ \mathsf{FL}^z(x, y.e)^{\leftarrow} & := & \odot\mathsf{L}^z(x, y.e^{\leftarrow}) \end{array}$$
   where $\mathsf{str}_x(e_i)$ is the result of Lemma 7.1.

5. Next, we show that for normal $e : \Gamma^* \vdash_{\overline{\Gamma}} A^*$, $e^{\leftarrow *} \equiv e$.

   In the case for the hypothesis rule for atoms, we have
   $$1_*(x)^{\leftarrow *} = x^* = x \equiv 1_*(x)$$
   In the case for FL, we have
   $$(\mathsf{FL}^z(x, y.e)^{\leftarrow})^* = (\odot\mathsf{L}^z(x, y.e^{\leftarrow}))^* = \mathsf{FL}^z(x, y.e^{\leftarrow *})$$
   so the result follows from the inductive hypothesis.

   In the case for FR, we have
   $$(\mathsf{FR}(1, e_1/x, e_2/y)^{\leftarrow})^* = (\odot\mathsf{R}(\mathsf{str}(e_1)^{\leftarrow}, \mathsf{str}(e_2)^{\leftarrow}))^* = \mathsf{FR}(1, (\mathsf{str}(e_1)^{\leftarrow *}/x, \mathsf{str}(e_2)^{\leftarrow *}/y))$$
   By the inductive hypothesis, we have $\mathsf{str}(e_i)^{\leftarrow *} \equiv \mathsf{str}(e_i)$, but we have $\mathsf{str}(e_i) \equiv e_i$ by Lemma 7.1.

6. Meta-operations are preserved by back-translation:

- If $\mathsf{id}\{x\} : \Gamma^* \vdash_x A^*$, then $\mathsf{id}\{x\}^{\leftarrow} \equiv x$.
  - Case for $A = P$: We have $(1_*(x))^{\leftarrow} = x$ as required.
  - Case for $A = \mathsf{F}_{x_1 \otimes x_2}(x_1 : A_1^*, x_2 : A_2^*)$. By definition, $\mathsf{id}\{x\}$ is $\mathsf{FL}^x(x_1, x_2.\mathsf{FR}(1, \mathsf{id}\{x_1\}/x_1, \mathsf{id}\{x_2\}/x_2))$, so $\mathsf{id}\{x\}^{\leftarrow}$ is
  $$\odot\mathsf{L}^x(x_1, x_2.\odot\mathsf{R}((\mathsf{str}_{x_2}(\mathsf{id}\{x_1\}))^{\leftarrow}/x_1, (\mathsf{str}_{x_1}(\mathsf{id}\{x_2\}))^{\leftarrow}/x_2))$$
  Since x_2 doesn't occur in $\mathsf{id}\{x_1\}$, $\mathsf{str}_{x_2}(\mathsf{id}\{x_1\})$ is literally the same term as $\mathsf{id}\{x_1\}$ (interpreted in a bigger context), without rewriting by any definitional equalities. Therefore by the inductive hypothesis for A_1^* and A_2^* gives
  $$\odot\mathsf{L}^x(x_1, x_2.\odot\mathsf{R}(x_1/x_1, x_2/x_2))$$
  which is equal to x by the η law for $A \odot B$.
- Left-inversion: if $y, z \# d$, then $\mathsf{linv}(d, (y, z), x)^{\leftarrow} \equiv d^{\leftarrow}[\odot\mathsf{R}(y, z)/x]$.
  If d is $\mathsf{FL}^x(y, z.d')$ then we get $d'^{\leftarrow}$ on the left, and $\odot\mathsf{L}^x(y, z.d')[\odot\mathsf{R}(y, z)/x]$ on the right, so they are equal by β in the source. Otherwise the result follows from the inductive hypotheses.
- Cut: If $e : \Gamma'^*, \Gamma^*, \Delta^* \vdash_{\overline{\Gamma}} B^*$ and $d : \Gamma'^*, \Gamma^* - x_0, \Delta^* \vdash_{\overline{\Delta}} A^*$ (where $x_0 : A \in \Gamma$), then $\mathsf{str}_{\Gamma'}(e\{d/x_0\})^{\leftarrow} \equiv \mathsf{str}_{\Gamma', \Delta}(e)^{\leftarrow}[\mathsf{str}_{\Gamma - x, \Gamma'}(d)^{\leftarrow}/x_0]$.
  Here, Γ' is some extra variables that occur in neither side of the cut, which is necessary to get the induction to go through.
  Since the only transformations are the identity, $s_*\{d\} = d$ for any s and d.
  The proof is to go through each reduction of cut and check that it is valid in the source. We check a few cases of the definition of cut:
  - $x_0\{d/x_0\} = d$. In this case, Γ, Γ' are empty besides x_0, and the cut in the source reduces to $\mathsf{str}(d)^{\leftarrow}$.
  - $\mathsf{FL}^{x_0}(x_1, x_2.e)\{\mathsf{FR}(1, \overrightarrow{d_i/x_i})/x_0\} = e\{\overrightarrow{d_i/x_i}\}$
  We need to show
  $$\mathsf{str}_{\Gamma'}(e\{\overrightarrow{d_i/x_i}\})^{\leftarrow} = \mathsf{str}(\mathsf{FL}^{x_0}(x_1, x_2.e))^{\leftarrow}[\mathsf{str}(\mathsf{FR}(1, \overrightarrow{d_i/x_i}))^{\leftarrow}/x_0]$$
  The right-hand side is equal to $\odot\mathsf{L}^{x_0}(x_1, x_2.\mathsf{str}(e)^{\leftarrow})[\odot\mathsf{R}(\mathsf{str}(d_1)^{\leftarrow}, \mathsf{str}(d_2)^{\leftarrow})/x_0]$, so reducing the cut in the source gives $\mathsf{str}(e)^{\leftarrow}[\mathsf{str}(d_1)^{\leftarrow}/x_1, \mathsf{str}(d_2)^{\leftarrow}/x_2]$ and the inductive hypothesis gives the result.

7. Next, we show that $d^*\!\downarrow^{\leftarrow} \equiv d$.
   The proof is by induction on d.
   - In the case for $\odot\mathsf{L}()$, expanding definitions, we have
   $$(\odot\mathsf{L}^z(x, y.d))^*\!\downarrow^{\leftarrow} \equiv \mathsf{FL}^z(x, y.d^*)\!\downarrow^{\leftarrow} \equiv \mathsf{FL}^z(x, y.d^*\!\downarrow)^{\leftarrow} \equiv \odot\mathsf{L}^z(x, y.d^*\!\downarrow^{\leftarrow})$$
   so the inductive hypothesis gives the result.
   - In the case for $\odot\mathsf{R}(,)$, we have
   $$\begin{array}{ll} & (\odot\mathsf{R}(d_1, d_2))^*\!\downarrow^{\leftarrow} \\ \equiv & \mathsf{FR}(1, (d_1^*/x, d_2^*/y))\!\downarrow^{\leftarrow} \\ \equiv & \mathsf{FR}(1, (d_1^*\!\downarrow/x, d_2^*\!\downarrow/y))^{\leftarrow} \\ \equiv & \odot\mathsf{R}(\mathsf{str}(d_1^*\!\downarrow)^{\leftarrow}/x, \mathsf{str}(d_2^*\!\downarrow)^{\leftarrow}/y) \end{array}$$
   Note that the forward translation weakens d_i^* when it constructs $\mathsf{FR}(1, (d_1^*/x, d_2^*/y))$, and cut elimination does not introduce left rules on variables that are not case-analyzed somewhere in the proof by Lemma 9.5. So by Lemma 9.4, strengthening $\mathsf{str}(d_1\!\downarrow^*)$ will simply undo the weakening done in constructing the term, and $\mathsf{str}(d_1^*\!\downarrow)^{\leftarrow}$ equals $d_1^*\!\downarrow^{\leftarrow}$. Therefore the inductive hypotheses give the result.

- In the case for a variable, we have
$$x^*{\downarrow}^{\leftarrow} \equiv x{\downarrow}^{\leftarrow} \equiv \mathsf{id}\{x\}^{\leftarrow}$$
so the above part gives the result.

- In the case for cut, we have
$$(e[d/x])^*{\downarrow}^{\leftarrow} \equiv e^*[d^*/x]{\downarrow}^{\leftarrow} \equiv (e^*{\downarrow}\{d^*{\downarrow}/x\})^{\leftarrow}$$
By the previous part, this is $e^*{\downarrow}^{\leftarrow}[d^*{\downarrow}^{\leftarrow}/x]$ (the strengthening only undoes the weakening that the forward translation puts in), so the inductive hypothesis gives the result.

8. For normal derivations $e, e' : \Gamma^* \vdash_{\overline{\Gamma}} A^*$, if $e \equiv_{\mathsf{p}} e'$, then $e^{\leftarrow} \equiv e'^{\leftarrow}$.

   We generalize slightly and prove that for $e, e' : \Gamma^*, \Delta^* \vdash_{\overline{\Gamma}} A^*$, if $e \equiv_{\mathsf{p}} e'$, then $\mathsf{str}_\Delta(e)^{\leftarrow} \equiv \mathsf{str}_\Delta(e')^{\leftarrow}$.

   - The cases for reflexivity, symmetry, and transitivity follow by induction, using the the corresponding rules of source-language equality.
   - The cases for compatibility all have a similar structure. For example, suppose we have $\mathsf{FR}(1, e_1, e_2) \equiv_{\mathsf{p}} \mathsf{FR}(1, e_1', e_2)$ because $e_1 \equiv_{\mathsf{p}} e_1'$. Then by the inductive hypothesis we get $\mathsf{str}_{\Delta,\Gamma_2}(e_1)^{\leftarrow} \equiv \mathsf{str}_{\Delta,\Gamma_2}(e_1')^{\leftarrow}$. We need to show that $\mathsf{str}_\Delta(\mathsf{FR}(1, e_1, e_2))^{\leftarrow} \equiv \mathsf{str}_\Delta(\mathsf{FR}(1, e_1, e_2))^{\leftarrow}$. By expanding definitions on both sides, it suffices to show
   $$\mathsf{FR}(1, \mathsf{str}_\Delta(e_1), \mathsf{str}_\Delta(e_2))^{\leftarrow} \equiv \mathsf{FR}(1, \mathsf{str}_\Delta(e_1'), \mathsf{str}_\Delta(e_2))^{\leftarrow}$$
   and so
   $$\odot\mathsf{R}(\mathsf{str}_{\Gamma_2}(\mathsf{str}_\Delta(e_1))^{\leftarrow}, \mathsf{str}_{\Gamma_1}(\mathsf{str}_\Delta(e_2))^{\leftarrow}) \equiv \odot\mathsf{R}(\mathsf{str}_{\Gamma_2}(\mathsf{str}_\Delta(e_1'))^{\leftarrow}, \mathsf{str}_{\Gamma_1}(\mathsf{str}_\Delta(e_2))^{\leftarrow})$$
   By Lemma 9.4, $\mathsf{str}_{\Gamma_2}(\mathsf{str}_\Delta(e_1')) = \mathsf{str}_{\Gamma_2,\Delta}(e_1')$, so the inductive hypothesis gives the result.
   - For the axiom
   $$\mathsf{FR}(s, d_1/x_1, \ldots, s_{i*}(d_i)/x_i, \ldots) \equiv_{\mathsf{p}} \mathsf{FR}(s; (1_{\alpha[\alpha_1/x_1,\ldots,\alpha_{i-1}/x_{i-1},\ldots]}[s_i/x_i]), \overrightarrow{d_j/x_j})$$
   since there are only identity structural transformations in this mode theory, e and e' must be syntactically identical terms.
   - For
   $$d \equiv_{\mathsf{p}} \mathsf{FL}^x(y, z.\mathsf{linv}(d, (y,z), x))$$
   we need to show that
   $$\mathsf{str}_\Delta(d)^{\leftarrow} \equiv \mathsf{str}_\Delta(\mathsf{FL}^x(y, z.\mathsf{linv}(d, (y,z), x)))^{\leftarrow}$$
   We distinguish cases on whether $x \in \Delta$ (so the strengthening deletes the FL) or not.

     If it does, we need to show that
   $$\mathsf{str}_\Delta(d)^{\leftarrow} \equiv \mathsf{str}_{\Delta,x,y,z}(\mathsf{linv}(d, (y,z), x))^{\leftarrow}$$
   This follows from Lemma 9.4.

     If it is not, we need to show that
   $$\mathsf{str}_\Delta(d)^{\leftarrow} \equiv \odot\mathsf{L}^x(y, z.\mathsf{str}_\Delta(\mathsf{linv}(d, (y,z), x))^{\leftarrow})$$
   and by η-expanding the left-hand side gives $\odot\mathsf{L}^x(y, z.\mathsf{str}_\Delta(d)^{\leftarrow}[\odot\mathsf{R}(y,z)/x])$. But in this case $\mathsf{str}_\Delta(\mathsf{linv}(d, (y,z), x)) = \mathsf{linv}(\mathsf{str}_\Delta(d), (y,z), x)$, so the fact that back-translation preserves left inversion gives the result.

10 Conclusion

We have described a sequent calculus that can express a variety of substructural and modal logics through a suitable choice of mode theory. The framework itself enjoys identity and cut admissibility for all mode theories, and these properties are inherited by the logics that are represented in it. The logic corresponds semantically to a fibration between 2-dimensional cartesian multicategories, and so gives both a syntactic and semantic account of the idea that substructural and modal logics are constraints on structural proofs.

In future work, we plan to continue the preliminary investigation of equational adequacy that is discussed Section 9. Additionally, we plan to apply our framework to investigate more extensions of homotopy type theory like the spatial type theory considered here; in current work with Eric Finster, we are designing a variant of cohesion that is an internal language for spectra. We also plan to consider encodings of programming-focused type theories, such as specialized effect calculi. Finally, our adequacy proofs require reasoning about the 1- and 2-cells in the mode theory, which we have currently done entirely naïvely; we would like to investigate using techniques from higher-dimensional rewriting to simplify and possibly automate these proofs.

References

A. Abel. The next 700 modal type assignment systems. In *International Conference on Types for Proofs and Programs (TYPES)*, 2015.

N. Alechina, M. Mendler, V. de Paiva, and E. Ritter. Categorical and Kripke semantics for constructive S4 modal logic. In *Computer Science Logic*, 2001.

T. Altenkirch and A. Kaposi. Type theory in type theory using quotient inductive types. In *ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 2016.

R. Atkey. A λ-calculus for resource separation. In *Automata, Languages and Programming: 31st International Colloquium, ICALP 2004*, volume 3142 of *Lecture Notes in Computer Science*, pages 158–170. Springer, 2004.

N. D. Belnap Jr. Display logic. *Journal of Philosophical Logic*, 11:375–417, 1982.

N. Benton. A mixed linear and non-linear logic: Proofs, terms and models. In *Computer Science Logic*, volume 933 of *LNCS*. Springer-Verlag, 1995.

N. Benton and P. Wadler. Linear logic, monads and the lambda calculus. In *IEEE Symposium on Logic in Computer Science*. IEEE Computer Society Press, 1996.

I. Cervesato and F. Pfenning. A linear logical framework. *Information and Computation*, 179(1):19–75, 2002.

K. Crary. Higher-order representation of substructural logics. In *ACM SIGPLAN International Conference on Functional Programming*, pages 131–142. ACM, 2010.

V. Danos, J.-B. Joinet, , and H. Schellinx. The structure of exponentials: Uncovering the dynamics of linear logic proofs. In *Kurt Godel Colloquium*, LNCS, pages 159–171. Springer, 1993.

C. Hermida. Fibrations for abstract multicategories. Fields Institute Communications; available from `http://sqig.math.ist.utl.pt/pub/HermidaC/fib-mul.pdf`, 2002.

F. Hörmann. Fibered multiderivators and (co)homological descent. Available from `http://arxiv.org/abs/1505.00974`, 2015.

B. Jacobs. Semantics of weakening and contraction. *Annals of Pure and Applied Logic*, 69(1):73 – 106, 1994.

J. Lambek. The mathematics of sentence structure. *The American Mathematical Monthly*, 65:154–170, 1958.

D. R. Licata and M. Shulman. Adjoint logic with a 2-category of modes. In *Logical Foundations of Computer Science*, 2016.

C. McBride. *I Got Plenty o' Nuttin'*, pages 207–233. Springer International Publishing, 2016.

V. Nigam and D. Miller. Algorithmic specifications in linear logic with subexponentials. In *Principles and Practice of Declarative Programming*, 2009.

P. W. O'Hearn and D. J. Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 5(2):215–244, 1999.

F. Pfenning. A structural proof of cut elimination and its representation in a logical framework. Technical Report CMU-CS-94-218, Department of Computer Science, Carnegie Mellon University, 1994.

F. Pfenning and R. Davies. A judgmental reconstruction of modal logic. *Mathematical Structures in Computer Science*, 11: 511–540, 2001.

J. Reed. Names are (mostly) useless: Encoding nominal logic programming techniques with use-counting and dependent types. Talk at Working on Mechanizing Metatheory (WMM), 2008.

J. Reed. A judgemental deconstruction of modal logic. Available from `www.cs.cmu.edu/~jcreed/papers/jdml.pdf`, 2009a.

J. Reed. *A Hybrid Logical Framework*. PhD thesis, Carnegie Mellon University, 2009b.

U. Schreiber and M. Shulman. Quantum gauge field theory in cohesive homotopy type theory. In *Workshop on Quantum Physics and Logic*, 2012.

M. Shulman. Brouwer's fixed-point theorem in real-cohesive homotopy type theory. arXiv:1509.07584, 2015.

Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations Of Mathematics*. Available from homotopytypetheory.org/book, 2013.

V. Voevodsky. A very short note on homotopy λ-calculus. http://www.math.ias.edu/vladimir/files/2006_09_Hlambda.pdf, September 2006.

K. Watkins, I. Cervesato, F. Pfenning, and D. Walker. A concurrent logical framework I: Judgments and properties. Technical Report CMU-CS-02-101, Department of Computer Science, Carnegie Mellon University, 2002. Revised May 2003.