Categorical Glueing and Logical Predicates for Models of Linear Logic

Masahito Hasegawa
Research Institute for Mathematical Sciences, Kyoto University
email: `hassei@kurims.kyoto-u.ac.jp`

January 1999

Abstract

We give a series of glueing constructions for categorical models of fragments of linear logic. Specifically, we consider the glueing of (i) symmetric monoidal closed categories (models of Multiplicative Intuitionistic Linear Logic), (ii) symmetric monoidal adjunctions (for interpreting the modality !) and (iii) $*$-autonomous categories (models of Multiplicative Linear Logic); the glueing construction for $*$-autonomous categories is a mild generalization of the double glueing construction due to Hyland and Tan. Each of the glueing techniques can be used for creating interesting models of linear logic. In particular, we use them, together with the free symmetric monoidal cocompletion, for deriving Kripke-like parameterized logical predicates (logical relations) for the fragments of linear logic. As an application, we show full completeness results for translations between linear type theories.

Contents

1 Introduction

Logical predicates (*logical relations*, *reducibility methods*) have been a powerful tool for proving both syntactic and semantic results on intuitionistic type theories. In particular, since Plotkin's work [38], a substantial study of characterizing the definability on models of the simply typed lambda calculus (and related typed languages such as PCF) using logical predicates has been carried out, see for instance [26, 37, 3, 17].

From the category-theoretic point of view, it is known that a setting for logical predicates for the simply typed lambda calculus can be derived from the *categorical glueing construction* (also known as *sconing* and *Freyd covering*) on cartesian closed categories [29, 36]. In terms of categorical logic, a glueing is to construct a category of "predicates" from a (codomain or subobject) fibration by a change-of-base [24]. For cartesian closed categories, it suffices to assume that the change-of-base functor preserves finite products for making the glued category cartesian closed. In such cases, the category of predicates on a model of the lambda calculus again gives a model, and moreover the projection to the original model preserves the structure. This observation allows us to derive the *Basic Lemma* for logical predicates on categorical models of the simply typed lambda calculus [36].

This paper develops an analogous story for fragments of *linear logic* [20]. We first investigate the glueing techniques for symmetric monoidal structures which serve as category-theoretic models of linear logic. Specifically, we consider the glueing of

1. symmetric monoidal closed categories (models of Multiplicative Intuitionistic Linear Logic),
2. symmetric monoidal adjunctions (for interpreting the modality !), and
3. $*$-autonomous categories (models of Multiplicative Linear Logic).

The glueing construction for $*$-autonomous categories is a mild generalization of the double glueing construction due to Hyland and Tan [42]. Each of them can be used for creating interesting models of linear logic. For instance, though not central for our development, we demonstrate how phase semantics [21] and its variants can be derived systematically from the glueing techniques (Example 3.6, 3.9, 3.11, 3.18 and 3.23).

Then we are ready to introduce a notion of logical predicates for models of linear logic. The predicates we introduce are parameterized, in the same way as the Kripke-logical relations [3]; the role of parameterization is essential in dealing with connectives of linear logic, especially the multiplicatives and modalities, roughly by the following reason. Suppose that we have a predicate $P_b \subseteq \mathcal{A}_b$ for each base type b, where $\mathcal{A}_\sigma$ is a set in which the closed terms of type σ are interpreted. As the standard logical predicates, we hope to define a predicate $P_\sigma \subseteq \mathcal{A}_\sigma$ for every type σ in an inductive way. However, we soon face a difficulty in constructing $P_{\sigma\otimes\tau}$ from P_σ and P_τ. The naive construction $P_{\sigma\otimes\tau} = \{a \otimes b \mid a \in P_\sigma, b \in P_\tau\}$ makes sense but can miss some interesting "undecomposable" elements of $\mathcal{A}_{\sigma\otimes\tau}$; in particular assume a constant of type $\sigma \otimes \tau$, then its interpretation may not belong to $P_{\sigma\otimes\tau}$ for any P_σ and P_τ. The same trouble appears when we construct $P_{!\sigma}$ from P_σ.

We solve this problem by parameterizing the predicates on a symmetric monoidal category which specifies a property closed under the linear structural constructions, so that the parameter indicates the linearly used resource (or the linear context). Such parameterized predicates give rise to a model of the fragment of linear logic, and serve as a basis for constructing logical predicates. The problem observed above disappears if each interesting element satisfies the property.

These parameterized logical predicates are derived from the glueing constructions together with the *free symmetric monoidal cocompletion* [25] on symmetric monoidal categories. As mentioned above, it is known that a setting for standard logical predicates can be obtained by glueing a cartesian closed category to **Set**; ours is derived by glueing a symmetric monoidal closed category to the presheaf category $\mathbf{Set}^{\mathbb{C}_0^{\mathrm{op}}}$ (free symmetric monoidal cocompletion) of a small symmetric monoidal category $\mathbb{C}_0$ which plays the role of "worlds" in the Kripke semantics [35].

As a consequence of these observations, we show full completeness results for translations between linear type theories – equivalently the fullness of the embeddings into relatively free symmetric monoidal structures. This is carried out by constructing a logical predicate which specifies the elements of (the term model of) the target type theory definable in the source theory, and then by appealing to the Basic Lemma.

From a more application-oriented point of view, it might be fruitful to adapt our method to reasoning about the properties of programming languages. For example, the complexity-parameterized logical relation used in [19] for showing the safety of a type-directed compilation with respect to the time complexity seems to have some common idea with our parameterized logical predicates. Another interesting direction is to combine our approach to other techniques of specifying properties of semantic categories, for instance that of specification structures [1]. We also note that an application of parameterized logical predicates for models of full propositional classical linear logic is found in Streicher's work [41] (see Example 3.9).

Some of the results in this paper have appeared in a preliminary version [23], where only the cases of intuitionistic linear type theories are discussed.

Construction of This Paper

In Section 2 we review some basic concepts of symmetric monoidal categories and related structures which will be used throughout this paper, and also recall the definition of the categorical glueing. Section 3 describes a series of glueing constructions for symmetric monoidal structures, with several examples. In Section 4 we apply the glueing constructions for deriving a notion of parameterized logical predicates for three fragments of linear logic. As a direct application, in Section 5 we show the full completeness of translations between linear type theories. Appendices give some syntactic details of the linear type theories as well as the proofs of two results in Section 3.

Acknowledgements

I thank Gordon Plotkin for discussions at the initial stage of this work. Thanks are also due to Martin Hyland and Audrey Tan, for introducing me to their work on double glueing.

2 Preliminaries

2.1 Symmetric Monoidal Structures

While we will heavily make use of concepts related to (symmetric) monoidal categories, some of them are given several names in the literature; for avoiding possible confusion, here we summarise the notions and terminology to be used in this paper. Many of them are found in the classical references (e.g. [16]), in several articles on models of linear logic (e.g. [6, 11]), and also in the second edition of Mac Lane's book [33].

A *monoidal category* $\mathbb{C} = (\mathbb{C}, \otimes, I, a, l, r)$ consists of a category $\mathbb{C}$, a functor $\otimes : \mathbb{C} \times \mathbb{C} \to \mathbb{C}$ (called the *monoidal* or *tensor product*), an object $I \in \mathbb{C}$ (the *unit object*) and natural isomorphisms $a_{A,B,C} : (A \otimes B) \otimes C \xrightarrow{\sim} A \otimes (B \otimes C)$, $l_A : I \otimes A \xrightarrow{\sim} A$ and $r_A : A \otimes I \xrightarrow{\sim} A$ such that, for objects $A, B, C, D \in \mathbb{C}$, the following two diagrams commute:

$$\begin{tikzcd}
((A\otimes B)\otimes C)\otimes D \arrow[r, "a"] \arrow[d, "a\otimes D"'] & (A\otimes B)\otimes(C\otimes D) \arrow[dd, "a"] \\
(A\otimes(B\otimes C))\otimes D \arrow[d, "a"'] & \\
A\otimes((B\otimes C)\otimes D) \arrow[r, "A\otimes a"'] & A\otimes(B\otimes(C\otimes D))
\end{tikzcd}
\qquad
\begin{tikzcd}
(A\otimes I)\otimes B \arrow[rr, "a"] \arrow[dr, "r\otimes B"'] & & A\otimes(I\otimes B) \arrow[dl, "A\otimes l"] \\
& A\otimes B &
\end{tikzcd}$$

A *symmetry* for a monoidal category is a natural transformation $c_{A,B} : A\otimes B \to B\otimes A$ subject to the following two commutative diagrams:

$$\begin{tikzcd}
(A\otimes B)\otimes C \arrow[r, "a"] \arrow[d, "c\otimes C"'] & A\otimes(B\otimes C) \arrow[r, "c"] & (B\otimes C)\otimes A \arrow[d, "a"] \\
(B\otimes A)\otimes C \arrow[r, "a"'] & B\otimes(A\otimes C) \arrow[r, "B\otimes c"'] & B\otimes(C\otimes A)
\end{tikzcd}
\quad
\begin{tikzcd}
A\otimes B \arrow[d, "c"'] \arrow[dr, "A\otimes B"] & \\
B\otimes A \arrow[r, "c"'] & A\otimes B
\end{tikzcd}$$

A monoidal category equipped with a symmetry is called a *symmetric monoidal category*. A *symmetric monoidal closed category* is a symmetric monoidal category $\mathbb{C}$ such that the functor $-\otimes A : \mathbb{C} \to \mathbb{C}$ has a right adjoint for each object A; we often write $A \multimap -$ for a specified right adjoint functor, and call it the *exponent*. Also we write $\mathsf{ev}_{A,B} : (A \multimap B)\otimes A \to B$ for the counit of the adjunction, and $\Lambda : \mathbb{C}(B\otimes A, C) \overset{\cong}{\to} \mathbb{C}(B, A \multimap C)$ for the bijection.

For monoidal categories $\mathbb{C} = (\mathbb{C}, \otimes, I, a, l, r)$ and $\mathbb{C}' = (\mathbb{C}', \otimes', I', a', l', r')$, a *monoidal functor* from $\mathbb{C}$ to $\mathbb{C}'$ is a tuple (F, m, m_I) where F is a functor from $\mathbb{C}$ to $\mathbb{C}'$, m is a natural transformation from $F(-)\otimes' F(=)$ to $F(-\otimes =)$ and $m_I : I' \to FI$ is an arrow in $\mathbb{C}'$, satisfying the coherence conditions below.

$$\begin{tikzcd}
(FA\otimes' FB)\otimes' FC \arrow[r, "m\otimes' FC"] \arrow[d, "a'"'] & F(A\otimes B)\otimes' FC \arrow[r, "m"] & F((A\otimes B)\otimes C) \arrow[d, "Fa"] \\
FA\otimes'(FB\otimes' FC) \arrow[r, "FA\otimes' m"'] & FA\otimes' F(B\otimes C) \arrow[r, "m"'] & F(A\otimes(B\otimes C))
\end{tikzcd}$$

$$\begin{tikzcd}
I'\otimes' FA \arrow[r, "l'"] \arrow[d, "m_I\otimes FA"'] & FA \\
FI\otimes' FA \arrow[r, "m"'] & F(I\otimes A) \arrow[u, "Fl"']
\end{tikzcd}
\qquad
\begin{tikzcd}
FA\otimes' I' \arrow[r, "r'"] \arrow[d, "FA\otimes m_I"'] & FA \\
FA\otimes' FI \arrow[r, "m"'] & F(A\otimes I) \arrow[u, "Fr"']
\end{tikzcd}$$

Let $\mathbb{C} = (\mathbb{C}, \otimes, I, a, l, r, c)$ and $\mathbb{C}' = (\mathbb{C}', \otimes', I', a', l', r', c')$ be symmetric monoidal categories. A *symmetric monoidal functor* from $\mathbb{C}$ to $\mathbb{C}'$ is a monoidal functor (F, m, m_I) which additionally satisfies the following condition:

$$\begin{tikzcd}
FA\otimes' FB \arrow[r, "c'"] \arrow[d, "m"'] & FB\otimes' FA \arrow[d, "m"] \\
F(A\otimes B) \arrow[r, "Fc"'] & F(B\otimes A)
\end{tikzcd}$$

Composition of (symmetric) monoidal functors is guaranteed by the following observation [16].

Lemma 2.1 *Given (symmetric) monoidal functors $(F, m, m_I) : \mathbb{C} \to \mathbb{C}'$ and $(G, n, n_{I'}) : \mathbb{C}' \to \mathbb{C}''$, $(G, n, n_{I'}) \circ (F, m, m_I) \equiv (G \circ F, G(m) \circ n_{F,F}, G(m_I) \circ n_{I'})$ is also a (symmetric) monoidal functor from $\mathbb{C}$ to $\mathbb{C}''$. This composition is associative, and satisfies the identity law for the identity (symmetric) monoidal functor.* □

A monoidal functor (F, m, m_I) is
– *strong*, if m is a natural isomorphism and m_I is an isomorphism;
– *strict*, if all components of m and m_I are identities.
A strict symmetric monoidal functor between symmetric monoidal closed categories (with specified exponents) is *closed* if it preserves the exponents as well as the unit and counit.

Given monoidal functors (F, m, m_I), (G, n, n_I) with the same source and target monoidal categories, a *monoidal natural transformation* from (F, m, m_I) to (G, n, n_I) is a natural transformation $\varphi : F \to G$ such that the following diagrams commute:

$$\begin{tikzcd}
FA \otimes' FB \arrow[r, "m"] \arrow[d, "\varphi \otimes' \varphi"'] & F(A \otimes B) \arrow[d, "\varphi"] \\
GA \otimes' GB \arrow[r, "n"'] & G(A \otimes B)
\end{tikzcd}
\qquad
\begin{tikzcd}
 & I' \arrow[dl, "m_I"'] \arrow[dr, "n_I"] & \\
FI \arrow[rr, "\varphi"'] & & GI
\end{tikzcd}$$

A *(symmetric) monoidal adjunction* between (symmetric) monoidal categories is an adjunction in which both of the functors are (symmetric) monoidal and the unit and counit are monoidal natural transformations. The following result is standard (Kelly [27]):

Proposition 2.2 *The left adjoint part of a monoidal adjunction is strong. Conversely, if a strong (symmetric) monoidal functor has a right adjoint, then the adjunction is (symmetric) monoidal.* □

A $*$-*autonomous category* [7, 8] is a symmetric monoidal category $\mathbb{C}$ equipped with a fully faithful functor $(-)^{\perp} : \mathbb{C}^{\mathrm{op}} \to \mathbb{C}$ such that there exists a natural isomorphism $\mathbb{C}(A \otimes B, C^{\perp}) \simeq \mathbb{C}(A, (B \otimes C)^{\perp})$. A $*$-autonomous category is closed because we have $\mathbb{C}(A \otimes B, C) \simeq \mathbb{C}(A, (B \otimes C^{\perp})^{\perp})$. Also it is easy to verify $A \simeq A^{\perp\perp}$ and $\mathbb{C}(A, B) \simeq \mathbb{C}(B^{\perp}, A^{\perp})$. If $\mathbb{C}$ is $*$-autonomous so is $\mathbb{C}^{\mathrm{op}}$, for unit ("false") $\perp \equiv I^{\perp}$ and tensor ("par") $A \wp B \equiv (A^{\perp} \otimes B^{\perp})^{\perp}$.

Alternatively, we can specify a $*$-autonomous category as a symmetric monoidal closed category with a "dualising object", i.e. an object $\perp$ such that the canonical morphism $A \to (A \multimap \perp) \multimap \perp$ is an isomorphism for any $A \in \mathbb{C}$.

2.2 Categorical Glueing

Here we recall the notion of the categorical glueing constructions which will be used throughout this paper. See Section 7.7 of Taylor's forthcoming book [43] for a comprehensive survey on properties, usages and historical remarks on the glueing constructions.

Given categories $\mathbb{C}$, $\mathbb{C}'$ and $\mathbb{D}$ with functors $F : \mathbb{C} \to \mathbb{D}$ and $G : \mathbb{C}' \to \mathbb{D}$, we write $(F \downarrow G)$ for the *comma category* [33] whose object is a triple $(A \in \mathbb{C}, B \in \mathbb{C}', f : FA \to GB)$ and an arrow from

(A,B,f) to (A',B',f') is a pair $(a : A \to A', b : B \to B')$ such that the following diagram commutes.

$$\begin{tikzcd} FA \arrow[r, "f"] \arrow[d, "Fa"'] & GB \arrow[d, "Gb"] \\ FA' \arrow[r, "f'"'] & GB' \end{tikzcd}$$

In the sequel, we will be interested in the comma categories of the form $(\mathbb{D} \downarrow G)$ for a functor $G : \mathbb{C} \to \mathbb{D}$. An object of $(\mathbb{D} \downarrow G)$ may be written as $(D \in \mathbb{D}, C \in \mathbb{C}, f : D \to GC)$. An arrow from (D,C,f) to (D',C',f') is then a pair $(d : D \to D', c : C \to C')$ satisfying $Gc \circ f = f' \circ d$. We note that there is an obvious "projection" functor $p : (\mathbb{D} \downarrow G) \to \mathbb{C}$ given by $p(D,C,f) = C$ and $p(d,c) = c$. We may call $(\mathbb{D} \downarrow G)$ the *glueing* of $\mathbb{C}$ to $\mathbb{D}$ along G.

We also consider the full subcategory $(\mathbb{D} \downarrow G)_s$ of $(\mathbb{D} \downarrow G)$ whose objects are subobjects in $\mathbb{D}$. We may write (C,X) for an object of $(\mathbb{D} \downarrow G)_s$, where X is a subobject of GC. An arrow $f : (C,X) \to (C',X')$ is then an arrow $f : C \to C'$ in $\mathbb{C}$ so that (in set-theoretic notation) $x \in X$ implies $Gf \circ x \in X'$. So we regard X as a predicate (or a specification) on GC, and a map from (C,X) to (C',X') is a map from C to C' which respects the predicates. The projection $p : (\mathbb{D} \downarrow G)_s \to \mathbb{C}$ sends $f : (C,X) \to (C',X')$ to $f : C \to C'$. The category $(\mathbb{D} \downarrow G)_s$ will be called the *subglueing* of $\mathbb{C}$ to $\mathbb{D}$ along G.

Also it is useful to notice that $(\mathbb{D} \downarrow G)$ and $(\mathbb{D} \downarrow G)_s$ together with the projections are characterized by the following pullbacks:

$$\begin{tikzcd} (\mathbb{D} \downarrow G) \arrow[r] \arrow[d, "p"'] & \mathbb{D}^{\to} \arrow[d, "\mathrm{cod}"] \\ \mathbb{C} \arrow[r, "G"'] & \mathbb{D} \end{tikzcd} \qquad \begin{tikzcd} (\mathbb{D} \downarrow G)_s \arrow[r] \arrow[d, "p"'] & \mathbf{Sub}(\mathbb{D}) \arrow[d, "\iota"] \\ \mathbb{C} \arrow[r, "G"'] & \mathbb{D} \end{tikzcd}$$

where cod is a forgetful functor which takes the codomain, $\mathbf{Sub}(\mathbb{D})$ is the full subcategory of $\mathbb{D}^{\to}$ whose objects are subobjects in $\mathbb{D}$, and ι is the restriction of cod to $\mathbf{Sub}(\mathbb{D})$. It is often the case that $\mathbb{D}$ has pullbacks, thus cod and ι are fibrations (the codomain fibration and the subobject fibration). In such settings, p is a fibration obtained by change-of-base along G. In particular, the subglueing is the place where we talk about $\mathbb{C}$ via the internal logic of $\mathbb{D}$. This point of view based on fibrations and categorical logic is exploited in Hermida's thesis [24], in the context of models of typed lambda calculi. It turns out that being a (bi)fibration has similar (but slightly indirect) impact in giving the symmetric monoidal structures on the glued categories; later we will briefly address this issue (Proposition 3.2, 3.14 and 3.20).

3 Glueing Symmetric Monoidal Structures

In this section we give a series of the glueing constructions for symmetric monoidal structures. We start with a simple (perhaps folklore) result for glueing symmetric monoidal closed categories. Based on this, we then consider two more involved settings. The first is the glueing of symmetric monoidal adjunctions, that is, to construct a symmetric monoidal adjunction between the glued categories from those between the component categories. The second is the "double" glueing of $*$-autonomous categories introduced by Hyland and Tan. To realize the duality in $*$-autonomous categories, we combine the glueing with its dual, thus use the glueing construction twice. We provide these results together with several examples.

3.1 Glueing Symmetric Monoidal Closed Categories

Lemma 3.1 *Suppose that $\mathbb{C}$ and $\mathbb{D}$ are symmetric monoidal closed categories and that $\Gamma : \mathbb{C} \to \mathbb{D}$ is a symmetric monoidal functor. Moreover suppose that $\mathbb{D}$ has pullbacks. Then the category $(\mathbb{D} \downarrow \Gamma)$ can be given a symmetric monoidal closed structure, so that the projection $p : (\mathbb{D} \downarrow \Gamma) \to \mathbb{C}$ is strict symmetric monoidal closed.*

Proof: We define the symmetric monoidal structure on $(\mathbb{D} \downarrow \Gamma)$ by

$$\begin{aligned} I &\equiv (I_{\mathbb{D}}, I_{\mathbb{C}}, m_I) \\ (D,C,f) \otimes (D',C',f') &\equiv (D \otimes D', C \otimes C', m_{C,C'} \circ (f \otimes f')) \\ (d,c) \otimes (d',c') &\equiv (d \otimes d', c \otimes c') \end{aligned}$$

where m_I and $m_{C,C'}$ are the coherent morphisms of the symmetric monoidal functor Γ. Exponents are defined as

$$(D,C,f) \multimap (D',C',f') \equiv (X, C \multimap C', \pi_2)$$

which is given by the pullback in $\mathbb{D}$

$$\begin{tikzcd}
X \arrow[r, "\pi_2"] \arrow[dd, "\pi_1"'] & \Gamma(C \multimap C') \arrow[d, "\theta_{C,C'}"] \\
 & \Gamma C \multimap \Gamma C' \arrow[d, "f \multimap \Gamma C'"] \\
D \multimap D' \arrow[r, "D \multimap f'"'] & D \multimap \Gamma C'
\end{tikzcd}$$

where we write $\mathsf{ev}_{C,C'} : (C \multimap C') \otimes C \to C'$ for the counit of the adjunction, and $\theta_{C,C'} : \Gamma(C \multimap C') \to \Gamma C \multimap \Gamma C'$ is the adjunct of $\mathsf{ev}'_{C,C'} \equiv \Gamma \mathsf{ev}_{C,C'} \circ m_{C \multimap C',C} : \Gamma(C \multimap C') \otimes \Gamma C \to \Gamma C'$. It is routine to see the bijective correspondences between

$$\begin{tikzcd}
D \otimes D' \arrow[r, "m \circ (f \otimes f')"] \arrow[d, dotted, "-"'] & \Gamma(C \otimes C') \arrow[d, dotted, "\Gamma-"] \\
D'' \arrow[r, "f''"'] & \Gamma C''
\end{tikzcd}$$

and

$$\begin{tikzcd}
D \arrow[r, "f"] \arrow[dd, dotted, "-"'] & \Gamma C \arrow[d, dotted, "\Gamma-"] \\
 & \Gamma(C' \multimap C'') \arrow[d, "(f' \multimap \Gamma C'') \circ \theta"] \\
D' \multimap D'' \arrow[r, "D' \multimap f''"'] & D' \multimap \Gamma C''
\end{tikzcd}$$

and

$$\begin{tikzcd}
D \arrow[r, "f"] \arrow[d, dotted, "-"'] & \Gamma C \arrow[d, dotted, "\Gamma-"] \\
X \arrow[r, "\pi_2"'] & \Gamma(C' \multimap C'')
\end{tikzcd}$$

□

This seems to be a folklore – Lawvere has stated this result in his lectures in 1990, c.f. [31]. Casley et al. [13] describe this too. Also see Ambler's thesis [4] for a related observation.

In fact, a more abstract point of view is available, in terms of fibrations. Hermida [24] has shown that, if we have a fibred ccc $p : \mathbb{E} \to \mathbb{B}$, $\mathbb{B}$ with finite products and p with $\mathrm{Cons}_{\mathbb{B}}$-products, then so is the fibration obtained by a change-of-base of p along a functor preserving finite products. The following observation is in spirit a parallel result for symmetric monoidal closed categories:

Proposition 3.2 *Suppose that $\mathbb{C}$, $\mathbb{E}$ and $\mathbb{B}$ are symmetric monoidal closed categories, and that $\Gamma : \mathbb{C} \to \mathbb{B}$ is a symmetric monoidal functor, and moreover that $p : \mathbb{E} \to \mathbb{B}$ is a strict symmetric monoidal closed functor which is also a cloven bifibration(i.e. both p and p^{op} are cloven fibrations).Consider the following pullback:*

$$\begin{tikzcd}
\mathbb{G} \arrow[r] \arrow[d, "q"'] & \mathbb{E} \arrow[d, "p"] \\
\mathbb{C} \arrow[r, "\Gamma"'] & \mathbb{B}
\end{tikzcd}$$

Then $\mathbb{G}$ can be given a symmetric monoidal closed structure, so that the bifibration $q : \mathbb{G} \to \mathbb{C}$ is strict symmetric monoidal closed.

<u>Proof:</u> See Appendix D. The assumption that p is a cofibration is used for making $\mathbb{G}$ symmetric monoidal, while the exponents are given by using the fact that p is also a fibration (see Proposition 3.14 for a general result for glueing such adjunctions). □

Lemma 3.1 can be regarded as an instance of Proposition 3.2, where $p : \mathbb{E} \to \mathbb{B}$ is the codomain fibration $\mathrm{cod} : \mathbb{D}^{\to} \to \mathbb{D}$. We can then derive Lemma 3.1 from Proposition 3.2 just by checking that $\mathbb{D}^{\to}$ is symmetric monoidal closed and cod preserves the structure strictly. Lemma 3.5 below gives another example, where the subobject fibration $\iota : \mathbf{Sub}(\mathbb{D}) \to \mathbb{D}$ takes the place.

Remark 3.3 One important result in Hermida's thesis [24] states that, if $p : \mathbb{E} \to \mathbb{B}$ is a fibred-ccc with $\mathrm{Cons}_{\mathbb{B}}$-products and $\mathbb{B}$ is a cartesian closed category, then so is $\mathbb{E}$ and p strictly preserves the cartesian closed structure. However, we do not know any anologous result for symmetric monoidal closed categories (we have no adequate notion of "fibred smcc"). □

As a variation of Lemma 3.1, we have the standard result on glueing cartesian closed categories (see for instance [29, 15, 36]):

Corollary 3.4 *Suppose that $\mathbb{C}$ and $\mathbb{D}$ are cartesian closed categories and that $\Gamma : \mathbb{C} \to \mathbb{D}$ is a functor which preserves finite products. Moreover suppose that $\mathbb{D}$ has pullbacks. Then the category $(\mathbb{D} \downarrow \Gamma)$ is cartesian closed; and the projection $p : (\mathbb{D} \downarrow \Gamma) \to \mathbb{C}$ is a cartesian closed functor.* □

Similarly to Lemma 3.1, we can give a symmetric monoidal closed structure on the subglueing $(\mathbb{D} \downarrow \Gamma)_s$, provided that the base category $\mathbb{D}$ admits epi-mono factorization:

Lemma 3.5 *In addition to the assumptions in Lemma 3.1, suppose that $\mathbb{D}$ admits epi-mono factorization. Then $(\mathbb{D} \downarrow \Gamma)_s$ can be given a symmetric monoidal closed structure, so that the projection $p : (\mathbb{D} \downarrow \Gamma)_s \to \mathbb{C}$ is strict symmetric monoidal closed.*

<u>Proof Sketch:</u> The description of the symmetric monoidal closed structure of $(\mathbb{D} \downarrow \Gamma)_s$ is easier than that of $(\mathbb{D} \downarrow \Gamma)$, using set-theoretic notation:

$$\begin{array}{rcl}
I & \equiv & (I_{\mathbb{C}}, \{m_I \circ x \mid x \in I_{\mathbb{D}}\}) \\
(C,X) \otimes (C',X') & \equiv & (C \otimes C', \{m_{C,C'} \circ (x \otimes x') \mid x \in X, x' \in X'\}) \\
(C,X) \multimap (C',X') & \equiv & (C \multimap C', \{f \in \Gamma(C \multimap C') \mid x \in X \text{ implies } \mathsf{ev}'_{C,C'} \circ (f \otimes x) \in X'\})
\end{array}$$

Alternatively we can apply Proposition 3.2; since the base category $\mathbb{D}$ admits epi-mono factorization, the subobject fibration $\iota : \mathbf{Sub}(\mathbb{D}) \to \mathbb{D}$ is a bifibration. So we only need to check that $\mathbf{Sub}(\mathbb{D})$ is symmetric monoidal closed so that ι is strict symmetric monoidal closed. □

Example 3.6 (Phase semantics)
A symmetric monoidal functor from the one-object one-arrow category **1** (with the trivial symmetric monoidal closed structure) to **Set** (with cartesian closed structure as the symmetric monoidal closed structure) is no other than a commutative monoid $M = (|M|, e, \cdot)$ – the underlying set $|M|$ is the image of the unique object of **1**, while the unit e and the multiplication $\cdot$ correspond to m_I and m respectively. Its subglueing is the poset $\mathcal{P}(|M|)$ of subsets of $|M|$ with the inclusion ordering. By Lemma 3.5, we can give a symmetric monoidal closed structure on $\mathcal{P}(|M|)$ as follows.

$$\begin{array}{rcl} I & = & \{e\} \\ X \otimes Y & = & \{x \cdot y \mid x \in X,\ y \in Y\} \\ X \multimap Y & = & \{u \mid x \in X \text{ implies } u \cdot x \in Y\} \end{array}$$

In fact $\mathcal{P}(|M|)$ is a free (commutative) quantale on the monoid M. By the way, this symmetric monoidal structure determines the multiplicative structure of the *phase semantics* [21]; it follows that, for a fixed $X \subseteq |M|$, the subposet $\mathcal{P}(|M|)_X$ of $\mathcal{P}(|M|)$ whose objects take the form $A \multimap X$ has a $*$-autonomous structure given by $I' = X \multimap X$, $A \otimes' B = ((A \otimes B) \multimap X) \multimap X$ and $A^\perp = A \multimap X$. (In fact, for any symmetric monoidal closed preordered set $\mathbb{C}$ and $X \in \mathbb{C}$, the Kleisli category of the monad $((-) \multimap X) \multimap X$ becomes a $*$-autonomous preordered set in this way, c.f. Example 3.9.) □

Example 3.7 (Subsconing)
Let $\mathbb{C}$ be a locally small symmetric monoidal closed category. The functor $\mathbb{C}(I, -) : \mathbb{C} \to \mathbf{Set}$ is symmetric monoidal, with $m_I =$ ‘id_I’ $: 1 \to \mathbb{C}(I, I)$ and $m_{A,B} : \mathbb{C}(I, A) \times \mathbb{C}(I, B) \to \mathbb{C}(I, A \otimes B)$ which sends $(x : I \to A, y : I \to B)$ to $I \overset{\simeq}{\to} I \otimes I \overset{x \otimes y}{\to} A \otimes B$. The subglueing $(\mathbf{Set} \downarrow \mathbb{C}(I, -))_s$ has the following symmetric monoidal closed structure:

$$\begin{array}{rcl} I & = & (I, \{id_I\}) \\ (A, X) \otimes (B, Y) & = & (A \otimes B, \{(x \otimes y) \circ \simeq \ \mid x \in X, y \in Y\}) \\ (A, X) \multimap (B, Y) & = & (A \multimap B, \{f \in \mathbb{C}_1(I, A \multimap B) \mid x \in X \text{ implies } \mathsf{ev} \circ (f \otimes x) \circ \simeq \in Y\}) \end{array}$$

where $X \subseteq \mathbb{C}(I, A)$, $Y \subseteq \mathbb{C}(I, B)$, and $\simeq$ indicates the canonical isomorphism $I \overset{\simeq}{\to} I \otimes I$. We call this category the *subsconing* of $\mathbb{C}$ and may write $\widetilde{\mathbb{C}}$ for it. □

Example 3.8 (Parameterized predicates)
A more sophisticated (and useful) example is obtained by combining the (sub)glueing construction with the *free symmetric monoidal cocompletion*. Let $\mathbb{I} : \mathbb{C}_0 \to \mathbb{C}_1$ be a strict symmetric monoidal functor from a small symmetric monoidal category $\mathbb{C}_0$ to a locally small symmetric monoidal closed category $\mathbb{C}_1$. We first note that the presheaf category $\mathbf{Set}^{\mathbb{C}_0^{\mathrm{op}}}$ has a symmetric monoidal closed structure given by

$$\begin{array}{rcl} I(-) & = & \mathbb{C}_0(-, I) \\ (F \otimes G)(-) & = & \displaystyle\int^{X,Y} FX \times GY \times \mathbb{C}_0(-, X \otimes Y) \\ (F \multimap G)(-) & = & \mathbf{Set}^{\mathbb{C}_0^{\mathrm{op}}}(F(=), G(- \otimes =)) \end{array}$$

which in fact is a free symmetric monoidal cocompletion of $\mathbb{C}_0$ [25]. Now we have a symmetric monoidal functor $\Gamma : \mathbb{C}_1 \to \mathbf{Set}^{\mathbb{C}_0^{\mathrm{op}}}$ given by $\Gamma(X) = \mathbb{C}_1(\mathbb{I}(-), X)$, equipped with $m_I : \mathbb{C}_0(-, I) \to \mathbb{C}_1(\mathbb{I}(-), I)$ and $m_{A,B} : \mathbb{C}_1(\mathbb{I}(-), A) \otimes \mathbb{C}_1(\mathbb{I}(-), B) \to \mathbb{C}_1(\mathbb{I}(-), A \otimes B)$ where

$$\begin{array}{rll} (m_I)_X & \text{sends} & f \in \mathbb{C}_0(X, I) \text{ to } \mathbb{I}(f) \in \mathbb{C}_1(\mathbb{I}X, I) \text{ and} \\ (m_{A,B})_X & \text{sends} & \text{the equivalence class of } (f \in \mathbb{C}_1(\mathbb{I}U, A), g \in \mathbb{C}_1(\mathbb{I}V, B), h \in \mathbb{C}_0(X, U \otimes V)) \\ & & \text{to } (f \otimes g) \circ \mathbb{I}(h) \in \mathbb{C}_1(\mathbb{I}X, A \otimes B). \end{array}$$

The subglued category $(\mathbf{Set}^{\mathbb{C}_0^{op}} \downarrow \Gamma)_s$ has objects of the form (A,P), where $A \in \mathbb{C}_1$ and P is a subfunctor of $\mathbb{C}_1(\mathbb{I}(-),A)$. An arrow from (A,P) to (B,Q) is an arrow $f : A \to B$ in $\mathbb{C}_1$ such that, for any $X \in \mathbb{C}_0$, $x \in P(X)$ implies $f \circ x \in Q(X)$. Its symmetric monoidal closed structure is described as $I = (I,\bar{I})$, $(A,P) \otimes (B,Q) = (A \otimes B, P \otimes Q)$ and $(A,P) \multimap (B,Q) = (A \multimap B, P \multimap Q)$ where

$$\begin{array}{rcl} \bar{I}(X) & = & \{\mathbb{I}h \mid h \in \mathbb{C}_0(X,I)\} \\ (P \otimes Q)(X) & = & \{(a \otimes b) \circ \mathbb{I}h \mid \exists Y,Z \in \mathbb{C}_0\ h \in \mathbb{C}_0(X, Y \otimes Z), a \in P(Y), b \in Q(Z)\} \\ (P \multimap Q)(X) & = & \{f \in \mathbb{C}_1(\mathbb{I}X, A \multimap B) \mid \forall Y \in \mathbb{C}_0\ \forall a \in P(Y)\ \ \mathsf{ev} \circ (f \otimes a) \in Q(X \otimes Y)\} \end{array}$$

Note that the subscone in Example 3.7 is the special case of this construction in which $\mathbb{C}_0$ is equivalent to **1**. The subglueing $(\mathbf{Set}^{\mathbb{C}_0^{op}} \downarrow \Gamma)_s$ will appear in Section 4.1 as the category $\mathbb{C}_0\mathsf{PRED}$ of the parameterized predicates. □

Example 3.9 (Proof-relavant phase semantics)
The following construction appears in Streicher's work on "denotationally complete" models of classical linear logic [41]. In example 3.8, suppose that the category $\mathbb{C}_1$ is $*$-autonomous. The resulting subglueing need not be $*$-autonomous; however, it is easily seen that, for any subfunctor P of $\mathbb{C}_1(\mathbb{I}(-),\perp)$, the canonical natural transformation from $(-) \multimap (\perp,P)$ to $(((-) \multimap (\perp,P)) \multimap (\perp,P)) \multimap (\perp,P)$ is anisomorphism. It then follows that the Kleisli category of the monad $((-) \multimap (\perp,P)) \multimap (\perp,P)$ is a $*$-autonomous category, where the unit, tensor and duality are given as the phase semantics (Example 3.6). In [41] Streicher has shown that, with suitable conditions on $\mathbb{C}_0$, one can cover not only the $*$-autonomous structure (multiplicatives) but also additives and exponentials. □

Remark 3.10 We can drop all "symmetric" from the results above and talk about monoidal (bi)closed categories, which are models of non-commutative linear logic and also the syntactic calculus of Lambek [30]. The example below shows that a non-commutative version of the phase semantics can be given as an instance of the subglueing construction. We also note that Shirasu [40] studied the glueing of monoidal (bi)closed lattices (called "FL-algebras") along a monoidal meet-semilattice morphism ("fringe morphism") for showing disjunction and existence properties of substructural logics; his construction can be derived from the non-symmetric version of Lemma 3.1. □

Example 3.11 (Non-commutative phase semantics)
A monoidal functor from the one-object one-arrow category **1** to **Set** is no other than a monoid $M = (|M|, e, \cdot)$. Its subglueing is the poset $\mathcal{P}(|M|)$, as described in Example 3.6. The non-symmetric variant of Lemma 3.5 implies that $\mathcal{P}(|M|)$ has the following monoidal biclosed structure.

$$\begin{array}{rcl} I & = & \{e\} \\ X \otimes Y & = & \{x \cdot y \mid x \in X,\ y \in Y\} \\ Y/X & = & \{u \mid x \in X \text{ implies } u \cdot x \in Y\} \\ X\backslash Y & = & \{u \mid x \in X \text{ implies } x \cdot u \in Y\} \end{array}$$

It is easy to verify the adjunctions $- \otimes X \dashv -/X$ and $X \otimes - \dashv X\backslash -$. □

3.2 Glueing Symmetric Monoidal Adjunctions

Lemma 3.12 *Suppose that* $\mathbb{C}_1 \underset{U}{\overset{F}{\underset{\leftarrow}{\overset{\rightarrow}{\perp}}}} \mathbb{C}_2$ *and* $\mathbb{D}_1 \underset{U'}{\overset{F'}{\underset{\leftarrow}{\overset{\rightarrow}{\perp}}}} \mathbb{D}_2$ *are (symmetric monoidal) adjunctions, with (symmetric monoidal) functors* $\Gamma_1 : \mathbb{C}_1 \to \mathbb{D}_1$ *and* $\Gamma_2 : \mathbb{C}_2 \to \mathbb{D}_2$ *together with a (monoidal)* *natural isomorphism* $\tau : \Gamma_1 U \overset{\cong}{\to} U'\Gamma_2$. *For* $\mathcal{G}_1 \equiv (\mathbb{D}_1 \downarrow \Gamma_1)$ *and* $\mathcal{G}_2 \equiv (\mathbb{D}_2 \downarrow \Gamma_2)$*, there are (symmetric monoidal) functors* $\mathcal{F} : \mathcal{G}_1 \to \mathcal{G}_2$ *and* $\mathcal{U} : \mathcal{G}_2 \to \mathcal{G}_1$ *given by*

$$\begin{array}{rcl@{\qquad}rcl} \mathcal{F}(D,C,f) & = & (F'D,\ FC,\ \sigma_C \circ F'f), & \mathcal{F}(d,c) & = & (F'd, Fc), \\ \mathcal{U}(D',C',g) & = & (U'D',\ UC',\ \tau^{-1}_{C'} \circ U'g), & \mathcal{U}(d',c') & = & (U'd', Uc') \end{array}$$

where $\sigma_C = \varepsilon'_{\Gamma_2 FC} \circ F'\tau_{FC} \circ F'\Gamma_1\eta_C : F'\Gamma_1 C \to \Gamma_2 FC$ *(*η *is the unit of* $F \dashv U$ *and* ε' *is the counit of* $F' \dashv U'$*).* $\mathcal{F}$ *is (strong and) left adjoint to* $\mathcal{U}$*. Moreover the projections* $p_1 : \mathcal{G}_1 \to \mathbb{C}_1$ *and* $p_2 : \mathcal{G}_2 \to \mathbb{C}_2$ *give a map of adjunction [33] from* $\mathcal{G}_1 \underset{\mathcal{U}}{\overset{\mathcal{F}}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} \mathcal{G}_2$ *to* $\mathbb{C}_1 \underset{U}{\overset{F}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} \mathbb{C}_2$.

Proof Sketch: See the bijective correspondence between

$$\begin{tikzcd} F'D \arrow[r, "F'f"] \arrow[d, dotted, "-"'] & F'\Gamma_1 C \arrow[r, "\sigma"] & \Gamma_2 FC \arrow[d, dotted, "\Gamma_2-"] \\ D' \arrow[rr, "g"'] & & \Gamma_2 C' \end{tikzcd}$$

and

$$\begin{tikzcd} D \arrow[rr, "f"] \arrow[d, dotted, "-"'] & & \Gamma_1 C \arrow[d, dotted, "\Gamma_1-"] \\ U'D' \arrow[r, "U'g"'] & U'\Gamma_2 C' \arrow[r, "\tau^{-1}_{C'}"'] & \Gamma_1 UC' \end{tikzcd}$$

□

While for most of our development Lemma 3.12 is sufficiently general, we can drop the assumption that τ is an isomorphism, provided $\mathbb{D}_1$ has pullbacks:

Lemma 3.13 *Consider* $\mathbb{C}_1 \underset{U}{\overset{F}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} \mathbb{C}_2$, $\mathbb{D}_1 \underset{U'}{\overset{F'}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} \mathbb{D}_2$, $\Gamma_1 : \mathbb{C}_1 \to \mathbb{D}_1$ *and* $\Gamma_2 : \mathbb{C}_2 \to \mathbb{D}_2$ *as in Lemma 3.12, with a (monoidal) natural transformation* $\tau : \Gamma_1 U \to U'\Gamma_2$*. Moreover suppose that* $\mathbb{D}_1$ *has pullbacks. For* $\mathcal{G}_1 \equiv (\mathbb{D}_1 \downarrow \Gamma_1)$ *and* $\mathcal{G}_2 \equiv (\mathbb{D}_2 \downarrow \Gamma_2)$*, there are (symmetric monoidal) functors* $\mathcal{F} : \mathcal{G}_1 \to \mathcal{G}_2$ *and* $\mathcal{U} : \mathcal{G}_2 \to \mathcal{G}_1$ *given by* $\mathcal{F}(D,C,f) = (F'D, FC, \sigma_C \circ F'f)$ *and* $\mathcal{U}(D',C',g) = (X, UC', \pi_2)$*, where* σ *is given as in Lemma 3.12, and* $\pi_2 : X \to \Gamma_1 UC'$ *is given by the following pullback in* $\mathbb{D}_1$.

$$\begin{tikzcd} X \arrow[r, "\pi_2"] \arrow[d, "\pi_1"'] & \Gamma_1 UC' \arrow[d, "\tau_{C'}"] \\ U'D' \arrow[r, "U'g"'] & U'\Gamma_2 C' \end{tikzcd}$$

$\mathcal{F}$ *is (strong and) left adjoint to* $\mathcal{U}$*, and the projections* $p_1 : \mathcal{G}_1 \to \mathbb{C}_1$ *and* $p_2 : \mathcal{G}_2 \to \mathbb{C}_2$ *give a map of adjunction.*

Proof Sketch: It is easy to see the natural bijections between

$$\begin{tikzcd}
F'D \arrow[r, "\sigma\circ F'f"] \arrow[d, dotted, "-"'] & \Gamma_2 FC \arrow[d, dotted, "\Gamma_2-"] \\
D' \arrow[r, "g"'] & \Gamma_2 C'
\end{tikzcd}
\quad \text{and} \quad
\begin{tikzcd}
D \arrow[r, "f"] \arrow[dd, dotted, "-"'] & \Gamma_1 C \arrow[d, dotted, "\Gamma_1-"] \\
 & \Gamma_1 UC' \arrow[d, "\tau"] \\
U'D' \arrow[r, "U'g"'] & U'\Gamma_2 C'
\end{tikzcd}
\quad \text{and} \quad
\begin{tikzcd}
D \arrow[r, "f"] \arrow[d, dotted, "-"'] & \Gamma_1 C \arrow[d, dotted, "\Gamma_1-"] \\
X \arrow[r, "\pi_2"'] & \Gamma_1 UC'
\end{tikzcd}$$

□

Almost all examples below will satisfy the assumption of Lemma 3.12; we use Lemma 3.13 only in Example 3.18.

In terms of fibrations Lemma 3.13 can be generalized as follows.

Proposition 3.14 *Let $p_1 : \mathbb{E}_1 \to \mathbb{B}_1$ be a cloven fibration and $p_2 : \mathbb{E}_2 \to \mathbb{B}_2$ be a cloven cofibration so that they give a map of adjunction from $\mathbb{E}_1 \underset{U_{\mathbb{E}}}{\overset{F_{\mathbb{E}}}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} \mathbb{E}_2$ to $\mathbb{B}_1 \underset{U_{\mathbb{B}}}{\overset{F_{\mathbb{B}}}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} \mathbb{B}_2$. Also suppose that there are an adjunction $\mathbb{C}_1 \underset{U_{\mathbb{C}}}{\overset{F_{\mathbb{C}}}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} \mathbb{C}_2$, functors $\Gamma_1 : \mathbb{C}_1 \to \mathbb{B}_1$, $\Gamma_2 : \mathbb{C}_2 \to \mathbb{B}_2$, and a natural transformation $\tau : \Gamma_1 U_{\mathbb{C}} \to U_{\mathbb{B}}\Gamma_2$. Consider the following pullbacks:*

$$\begin{tikzcd}
\mathbb{G}_1 \arrow[r] \arrow[d, "q_1"'] & \mathbb{E}_1 \arrow[d, "p_1"] \\
\mathbb{C}_1 \arrow[r, "\Gamma_1"'] & \mathbb{B}_1
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbb{G}_2 \arrow[r] \arrow[d, "q_2"'] & \mathbb{E}_2 \arrow[d, "p_2"] \\
\mathbb{C}_2 \arrow[r, "\Gamma_2"'] & \mathbb{B}_2
\end{tikzcd}$$

Then there is an adjunction $\mathbb{G}_1 \underset{U_{\mathbb{G}}}{\overset{F_{\mathbb{G}}}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} \mathbb{G}_2$ with functors given by $F_{\mathbb{G}}(C,E) = (F_{\mathbb{C}}C, \sigma_{C!}(F_{\mathbb{E}}E))$ and $U_{\mathbb{G}}(C,E) = (U_{\mathbb{C}}C, \tau_C^*(U_{\mathbb{E}}E))$, where $\sigma : F_{\mathbb{B}}\Gamma_1 \to \Gamma_2 F_{\mathbb{C}}$ is given as in Lemma 3.12 (the notations for cartesian and cocartesian liftings are those in Appendix D). Moreover, q_1 and q_2 give a map of adjunction from $\mathbb{G}_1 \underset{U_{\mathbb{G}}}{\overset{F_{\mathbb{G}}}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} \mathbb{G}_2$ to $\mathbb{C}_1 \underset{U_{\mathbb{C}}}{\overset{F_{\mathbb{C}}}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} \mathbb{C}_2$. □

This not only generalises Lemma 3.13 but also covers the construction of exponents in Proposition 3.2 where we have $\tau = \theta : \Gamma(C \multimap -) \to \Gamma C \multimap \Gamma(-)$ and $\sigma = m : \Gamma(-) \otimes \Gamma C \to \Gamma(- \otimes C)$.

The subglueing versions of Lemma 3.12 and 3.13 are also available, provided the categories $\mathbb{D}_1$ and $\mathbb{D}_2$ admit epi-mono factorization; for Lemma 3.12, we have

$$\mathcal{F}(C,X) = (FC, \{\sigma_C \circ x \mid x \in F'X\}), \quad \mathcal{U}(C',Y) = (UC', \{\tau_{C'}^{-1} \circ y \mid y \in U'Y\})$$

and for Lemma 3.13

$$\mathcal{F}(C,X) = (FC, \{\sigma_C \circ x \mid x \in F'X\}), \quad \mathcal{U}(C',Y) = (UC', \{z \in \Gamma_1 UC' \mid \tau_{C'} \circ z \in U'Y\}).$$

Note that, if τ is an isomorphism, then these two results agree.

Remark 3.15 Again we can drop all “symmetric” from the results above. □

Example 3.16 (Adjunction between subsconings)
Let $\mathbb{C}_1 \substack{\xrightarrow{F} \\ \perp \\ \xleftarrow[U]{}} \mathbb{C}_2$ be a symmetric monoidal adjunction between small symmetric monoidal categories $\mathbb{C}$ and $\mathbb{D}$. By subglueing $\mathbb{C}_1 \substack{\xrightarrow{F} \\ \perp \\ \xleftarrow[U]{}} \mathbb{C}_2$ to the trivial adjunction $\mathbf{Set} \substack{\xrightarrow{Id} \\ \perp \\ \xleftarrow[Id]{}} \mathbf{Set}$ along the glueing functors $\mathbb{C}_1(I,-) : \mathbb{C}_1 \to \mathbf{Set}$ and $\mathbb{C}_2(I,-) : \mathbb{C}_2 \to \mathbf{Set}$ together with a monoidal natural isomorphism $\tau : \mathbb{C}_1(I,U-) \overset{\simeq}{\dot{\to}} \mathbb{C}_2(FI,-) \overset{\simeq}{\dot{\to}} \mathbb{C}_2(I,-)$, we obtain a symmetric monoidal adjunction between subsconings (Example 3.7) $\widetilde{\mathbb{C}_1} \substack{\xrightarrow{\mathcal{F}} \\ \perp \\ \xleftarrow[\mathcal{U}]{}} \widetilde{\mathbb{C}_2}$ where

$$\mathcal{F}(A,X) \;=\; (FA, \{Fx \circ m_I \mid x \in X\}), \quad \mathcal{U}(B,Y) \;=\; (UB, \{U(y \circ m_I^{-1}) \circ \eta_I \mid y \in Y\}).$$

□

Example 3.17 (Adjunction between categories of parameterized predicates)
In Example 3.8 we have constructed a category of parameterized predicates as a subglueing. Consider a commutative diagram of functors

$$\begin{tikzcd}
\mathbb{C}_0 \arrow[r, "F_0"] \arrow[d, "\mathbb{I}"'] & \mathbb{D}_0 \arrow[d, "\mathbb{J}"] \\
\mathbb{C}_1 \arrow[r, "F_1"'] & \mathbb{D}_1
\end{tikzcd}$$

in which $\mathbb{C}_0$, $\mathbb{C}_1$, $\mathbb{D}_0$ and $\mathbb{D}_1$ are symmetric monoidal categories, F_0, F_1 are strong symmetric monoidal, while $\mathbb{I}$, $\mathbb{J}$ are strict symmetric monoidal. Moreover assume that F_1 has a right adjoint $U_1 : \mathbb{D}_1 \to \mathbb{C}_1$. For this setting, we shall give a symmetric monoidal adjunction between the categories of parameterized predicates. We first note that there is a symmetric monoidal adjunction between the presheaf categories $\mathbf{Set}^{\mathbb{C}_0^{\mathrm{op}}} \substack{\xrightarrow{\mathbf{Lan}_{F_0^{\mathrm{op}}}(-)} \\ \perp \\ \xleftarrow[(-)\circ F_0]{}} \mathbf{Set}^{\mathbb{D}_0^{\mathrm{op}}}$, where $\mathbf{Lan}_{F_0^{\mathrm{op}}} G = \int^X \mathbb{D}_0(-,F_0X) \times GX : \mathbb{D}_0^{\mathrm{op}} \to \mathbf{Set}$ is a left Kan extension [33] of $G : \mathbb{C}_0^{\mathrm{op}} \to \mathbf{Set}$ along F_0^{op}. By subglueing $\mathbb{C}_1 \substack{\xrightarrow{F_1} \\ \perp \\ \xleftarrow[U_1]{}} \mathbb{D}_1$ to $\mathbf{Set}^{\mathbb{C}_0^{\mathrm{op}}} \substack{\xrightarrow{\mathbf{Lan}_{F_0^{\mathrm{op}}}(-)} \\ \perp \\ \xleftarrow[(-)\circ F_0]{}} \mathbf{Set}^{\mathbb{D}_0^{\mathrm{op}}}$ via the glueing functors $\Gamma_1 : X \mapsto \mathbb{C}_1(\mathbb{I}(-),X) : \mathbb{C}_1 \to \mathbf{Set}^{\mathbb{C}_0^{\mathrm{op}}}$ and $\Gamma_2 : X \mapsto \mathbb{D}_1(\mathbb{J}(-),X) : \mathbb{D}_1 \to \mathbf{Set}^{\mathbb{D}_0^{\mathrm{op}}}$ together with a monoidal natural isomorphism

$$\tau_X : \Gamma_1 U_1 X = \mathbb{C}_1(\mathbb{I}(-),U_1X) \overset{\simeq}{\dot{\to}} \mathbb{D}_1(F_1(\mathbb{I}(-)),X) = \mathbb{D}_1(\mathbb{J}(F_0(-)),X) = ((-)\circ F_0)(\Gamma_2 X),$$

we obtain a symmetric monoidal adjunction $(\mathbf{Set}^{\mathbb{C}_0^{\mathrm{op}}} \downarrow \Gamma_1)_s \substack{\xrightarrow{\mathcal{F}} \\ \perp \\ \xleftarrow[\mathcal{U}]{}} (\mathbf{Set}^{\mathbb{D}_0^{\mathrm{op}}} \downarrow \Gamma_2)_s$. Explicitly, $\mathcal{F}$ and $\mathcal{U}$ are given by $\mathcal{F}(A,P) = (F_1A, \mathcal{F}(P))$ and $\mathcal{U}(B,Q) = (U_1B, \mathcal{U}(Q))$ where

$$\begin{array}{rcl}
\mathcal{F}(P)(Y) & = & \{F_1(f) \circ \mathbb{J}(h) \mid \exists X \in \mathbb{C}_0 \; f \in P(X), h \in \mathbb{D}_0(Y,F_0X)\}, \\
\mathcal{U}(Q)(X) & = & \{U_1(g) \circ \eta_{\mathbb{I}X} \mid g \in Q(F_0X)\}.
\end{array}$$

This example will be used in Section 4.4 for modelling the modality ! of intuitionistic linear logic in the category of parameterized predicates. □

Example 3.18 (Adjunction between phase semantics and the interpretation of !)
As noted in Example 3.6, to give a symmetric monoidal functor from **1** to **Set** is to give a commutative monoid. It is easily seen that to give a monoidal natural transformation between such symmetric monoidal functors is to give a monoid homomorphism between the corresponding monoids. Now let us consider commutative monoids M, N and a monoid homomorphism $\tau : M \to N$. We then obtain a symmetric monoidal adjunction $\mathcal{P}(|M|) \underset{\mathcal{U}}{\overset{\mathcal{F}}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} \mathcal{P}(|N|)$ by subglueing $\mathbf{1} \underset{Id}{\overset{Id}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} \mathbf{1}$ to $\mathbf{Set} \underset{Id}{\overset{Id}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} \mathbf{Set}$ via the glueing functors and natural transformation corresponding to M, N and τ, using the subglueing version of Lemma 3.13. Explicitly, for $X \subseteq |M|$ and $Y \subseteq |N|$,

$$\mathcal{F}(X) = \tau(X) = \{\tau(x) \mid x \in X\}, \quad \mathcal{U}(Y) = \tau^{-1}(Y) = \{x \mid \tau(x) \in Y\}.$$

The induced comonad on $\mathcal{P}(|N|)$ sends Y to $Y \cap \tau(M)$. By composing this comonad with the Kleisli adjunction of the monad $((-) \multimap X) \multimap X$ described in Example 3.6, we obtain a comonad T on $\mathcal{P}(|N|)_X$ which sends Y to $((Y \cap \tau(M)) \multimap X) \multimap X$. The (co)Kleisli adjunction of T is symmetric monoidal (with respect to the finite products of the (co)Kleisli category and the $*$-autonomous structure of $\mathcal{P}(|N|)_X$) if and only if $\tau(M) \subseteq I'$ and also $(Y \cap Z \cap \tau(M)) \multimap X = ((Y \cap \tau(M)) \otimes (Z \cap \tau(M))) \multimap X$ for $Y, Z \in \mathcal{P}(|N|)_X$; in such cases T gives a sound interpretation of the modality !. In [21], M is chosen to be the submonoid $\{u \in I' \mid u \cdot u = u\}$ of N, with $\tau : M \to N$ the inclusion. □

3.3 Glueing $*$-Autonomous Categories

We give a mild generalization of the *double glueing construction* of Hyland and Tan [42] (see Example 3.24 below). The essential idea is that we *double* the objects of the glued category so that the duality of the underlying $*$-autonomous category scales up to the glued category. While an object in the glued category considered so far is essentially a predicate on an object of the underlying category, in the double-glued category an object is a pair of predicates, one on an object of the underlying category and the other on its dual. Though such a doubled category obviously has a self-duality, it is perhaps surprising to see that we can also give a symmetric monoidal structure on it so that together with the duality it forms a $*$-autonomous category.

Proposition 3.19 *Suppose that $\mathbb{C} = (\mathbb{C}, \otimes, I, (-)^{\perp})$ is a $*$-autonomous category, $\mathbb{D}$ is a symmetric monoidal closed category and that $\Gamma : \mathbb{C} \to \mathbb{D}$ is a symmetric monoidal functor. Moreover suppose that $\mathbb{D}$ has pullbacks. Then the category $\mathbf{DG}(\Gamma)$ determined by the following pullback can be given a $*$-autonomous structure, so that the projection $p : \mathbf{DG}(\Gamma) \to \mathbb{C}$ preserves the $*$-autonomous structure strictly.*

$$\begin{tikzcd}
\mathbf{DG}(\Gamma) \arrow[r] \arrow[d] \arrow[dr, dotted, "p"] & (\mathbb{D} \downarrow \Gamma(-)^{\perp})^{\mathrm{op}} \arrow[d, "p_2^{\mathrm{op}}"] \\
(\mathbb{D} \downarrow \Gamma) \arrow[r, "p_1"'] & \mathbb{C}
\end{tikzcd}$$

($p_1 : (\mathbb{D} \downarrow \Gamma) \to \mathbb{C}$ and $p_2 : (\mathbb{D} \downarrow \Gamma(-)^{\perp}) \to \mathbb{C}^{\mathrm{op}}$ are the projections from the glued categories.)

Proof Sketch: Explicitly, objects of $\mathbf{DG}(\Gamma)$ are tuples (D_s, D_t, C, f_s, f_t) where $D_s, D_t \in \mathbb{D}$, $C \in \mathbb{C}$, $f_s : D_s \to \Gamma C$ and $f_t : D_t \to \Gamma C^\perp$ in $\mathbb{D}$. An arrow from (D_s, D_t, C, f_s, f_t) to $(D'_s, D'_t, C', f'_s, f'_t)$ is a tuple $(d_s : D_s \to D'_s, d_t : D'_t \to D_t, c : C \to C')$ such that the following diagrams commute.

$$\begin{tikzcd}
D_s \arrow[r, "f_s"] \arrow[d, "d_s"'] & \Gamma C \arrow[d, "\Gamma c"] & D_t \arrow[r, "f_t"] & \Gamma C^\perp \\
D'_s \arrow[r, "f'_s"'] & \Gamma C' & D'_t \arrow[r, "f'_t"'] \arrow[u, "d_t"] & \Gamma C'^\perp \arrow[u, "\Gamma c^\perp"']
\end{tikzcd}$$

The $*$-autonomous structure on $\mathbf{DG}(\Gamma)$ is given as follows.

$$\begin{aligned}
I &\equiv (I_{\mathbb{D}}, \Gamma I_{\mathbb{C}}^\perp, I_{\mathbb{C}}, m_I, id_{\Gamma I_{\mathbb{C}}^\perp}) \\
(D_s, D_t, C, f_s, f_t) \otimes (D'_s, D'_t, C', f'_s, f'_t) &\equiv (D_s \otimes D'_s, X, C \otimes C', m_{C,C'} \circ (f_s \otimes f'_s), x) \\
(D_s, D_t, C, f_s, f_t)^\perp &\equiv (D_t, D_s, C^\perp, f_t, D_s \xrightarrow{f_s} \Gamma C \xrightarrow{\simeq} \Gamma C^{\perp\perp})
\end{aligned}$$

X and $x : X \to \Gamma(C \otimes C')^\perp$ are given by the following three pullbacks in $\mathbb{D}$.

$$\begin{tikzcd}
X \arrow[r] \arrow[d] \arrow[dr, dotted, "x"] & {} \arrow[rrr] \arrow[d] & & & D'_s \multimap D_t \arrow[d, "D'_s \multimap f_t"] \\
{} \arrow[r] \arrow[dddd] & \Gamma(C \otimes C')^\perp \arrow[r, "\Gamma\simeq"] \arrow[d, "\Gamma\simeq"] & \Gamma(C' \multimap C^\perp) \arrow[r, "\theta_{C',C^\perp}"] & \Gamma C' \multimap \Gamma C^\perp \arrow[r, "f'_s \multimap \Gamma C^\perp"] & D'_s \multimap \Gamma C^\perp \\
& \Gamma(C \multimap C'^\perp) \arrow[d, "\theta_{C,C'^\perp}"] & & & \\
& \Gamma C \multimap \Gamma C'^\perp \arrow[d, "f_s \multimap \Gamma C'^\perp"] & & & \\
& & & & \\
D_s \multimap D'_t \arrow[r, "D_s \multimap f'_t"'] & D_s \multimap \Gamma C'^\perp & & &
\end{tikzcd}$$

$\theta_{C,C'} : \Gamma(C \multimap C') \to \Gamma C \multimap \Gamma C'$ is given as in the proof of Lemma 3.1. It is not hard to see that the unit and tensor product given above determine a symmetric monoidal structure on $\mathbf{DG}(\Gamma)$; also it is easy to see that $(-)^\perp$ determines a contravariant fully faithful functor on $\mathbf{DG}(\Gamma)$. The most nontrivial part is to check the natural bijection $\mathbf{DG}(\Gamma)({}_{-}\otimes{}_{=}, ({}_{\equiv})^\perp) \simeq \mathbf{DG}(\Gamma)({}_{-}, ({}_{=}\otimes{}_{\equiv})^\perp)$ – it requires some calculation, as found in Appendix E. $p : \mathbf{DG}(\Gamma) \to \mathbb{C}$ sends (D_s, D_t, C, f_s, f_t) to C and (d_s, d_t, c) to c, and obviously preserves the $*$-autonomous structure strictly. □

We call $\mathbf{DG}(\Gamma)$ the *double glueing* of $\mathbb{C}$ to $\mathbb{D}$ along Γ.

In fact it is possible to give a more general statement in terms of fibrations, in the similar way to Proposition 3.2, as follows. It subsumes Proposition 3.19 and also 3.22 below. The proof is essentially the reworking of that of Proposition 3.19 (Appendix E) using the idiom of fibrations.

Proposition 3.20 *Suppose that* $\mathbb{C}$ *is a* $*$*-autonomous category,* $\mathbb{B}$*,* $\mathbb{E}$ *are symmetric monoidal closed categories, and that* $\Gamma : \mathbb{C} \to \mathbb{E}$ *is a symmetric monoidal functor while* $p : \mathbb{E} \to \mathbb{B}$ *is a strict symmetric monoidal closed functor which is also a cloven bifibration with fibred finite products. Consider the category* $\mathbf{DG}(\Gamma, p)$ *determined by the following pullbacks:*

$$\begin{tikzcd}
\mathbf{G}_s \arrow[r] \arrow[d, "p_s"'] & \mathbb{E} \arrow[d, "p"] \\
\mathbb{C} \arrow[r, "\Gamma"'] & \mathbb{B}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{G}_t \arrow[r] \arrow[d, "p_t"'] & \mathbb{E} \arrow[d, "p"] \\
\mathbb{C}^{\mathrm{op}} \arrow[r, "\Gamma(-)^{\perp}"'] & \mathbb{B}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{DG}(\Gamma, p) \arrow[r] \arrow[d] & \mathbf{G}_t^{\mathrm{op}} \arrow[d, "p_t^{\mathrm{op}}"] \\
\mathbf{G}_s \arrow[r, "p_s"'] & \mathbb{C}
\end{tikzcd}$$

Then $\mathbf{DG}(\Gamma, p)$ *can be given a* $*$*-autonomous structure, so that the projection to* $\mathbb{C}$ *strictly preserves the* $*$*-autonomous structure.*

Proof Sketch: An object of $\mathbf{DG}(\Gamma, p)$ is a triple $A = (|A| \in \mathbb{C}, A_s \in \mathbb{E}, A_t \in \mathbb{E})$ such that $\Gamma|A| = p(A_s)$ and $\Gamma|A|^{\perp} = p(A_t)$ hold. An arrow from $A = (|A|, A_s, A_t)$ to $B = (|B|, B_s, B_t)$ is a triple $f = (|f| : |A| \to |B|, f_s : A_s \to B_s, f_t : B_t \to A_t)$ which satisfies $\Gamma|f| = p(f_s)$ and $\Gamma|f|^{\perp} = p(f_t)$. Let us define (using the notations in Appendix D)

$$\begin{array}{rcl}
I & \equiv & (I_{\mathbb{C}},\ m_{I!}(I_{\mathbb{E}}),\ 1_{\Gamma I_{\mathbb{C}}^{\perp}}) \\
A \otimes B & \equiv & (|A| \otimes |B|,\ m_{|A|,|B|!}(A_s \otimes B_s),\ \theta_1^*(A_s \multimap B_t) \times_{\Gamma(|A| \otimes |B|)^{\perp}} \theta_2^*(B_s \multimap A_t)) \\
A^{\perp} & \equiv & (|A|^{\perp},\ A_t,\ \sigma^*(A_s))
\end{array}$$

where

$$\begin{array}{rcl}
\theta_1 & : & \Gamma(|A| \otimes |B|)^{\perp} \overset{\simeq}{\to} \Gamma(|A| \multimap |B|^{\perp}) \overset{\theta}{\to} \Gamma|A| \multimap \Gamma|B|^{\perp} \\
\theta_2 & : & \Gamma(|A| \otimes |B|)^{\perp} \overset{\simeq}{\to} \Gamma(|B| \multimap |A|^{\perp}) \overset{\theta}{\to} \Gamma|B| \multimap \Gamma|A|^{\perp} \\
\sigma & : & \Gamma|A|^{\perp\perp} \overset{\simeq}{\to} \Gamma|A|
\end{array}$$

and we write 1_X and $E \times_X E'$ for the terminal object and binary product in the fibre over $X \in \mathbb{B}$. Then one can check that these data give a $*$-autonomous structure on $\mathbf{DG}(\Gamma, p)$, similarly to the proof of Proposition 3.19. □

Remark 3.21 The functor $\Gamma(-)^{\perp}$ is symmetric monoidal with respect to the symmetric monoidal structure $(\perp, \wp)$ on $\mathbb{C}^{\mathrm{op}}$. Double glueing makes use of this duality between $(\mathbb{C}, I, \otimes)$ and $(\mathbb{C}^{\mathrm{op}}, \perp, \wp)$. □

Proposition 3.22 *In addition to the assumptions in Proposition 3.19, suppose that* $\mathbb{D}$ *admits epi-mono factorization. Then the category* $\mathbf{DG}(\Gamma)_s$ *determined by the following pullback can be given a* $*$*-autonomous structure, so that the projection* $p : \mathbf{DG}(\Gamma)_s \to \mathbb{C}$ *preserves the* $*$*-autonomous structure strictly.*

$$\begin{tikzcd}
\mathbf{DG}(\Gamma)_s \arrow[r] \arrow[d] \arrow[dr, dotted, "p"] & (\mathbb{D} \downarrow \Gamma(-)^{\perp})_s^{\mathrm{op}} \arrow[d, "p_2^{\mathrm{op}}"] \\
(\mathbb{D} \downarrow \Gamma)_s \arrow[r, "p_1"'] & \mathbb{C}
\end{tikzcd}$$

Proof Sketch: Let us write $A = (|A|, A_s, A_t)$ for an object of $\mathbf{DG}(\Gamma)_s$, where $|A| \in \mathbb{C}$ and A_s, A_t are subobjects of $\Gamma|A|$ and $\Gamma|A|^{\perp}$ respectively. Then a map f from A to B is an arrow $f : |A| \to |B|$ in $\mathbb{C}$ such that, $x \in A_s$ implies $\Gamma f \circ x \in B_s$ and $y \in B_t$ implies $\Gamma f^{\perp} \circ y \in A_t$. The $*$-autonomous structure is given as

$$\begin{array}{rcl}
I & \equiv & (I_{\mathbb{C}}, \{m_I \circ x \mid x \in I_{\mathbb{D}}\}, \Gamma I_{\mathbb{C}}^{\perp}) \\
A \otimes B & \equiv & (|A| \otimes |B|, \\
 & & \{m_{|A|,|B|} \circ (x \otimes y) \mid x \in A_s, y \in B_s\}, \\
 & & \{u \in \Gamma(|A| \otimes |B|)^{\perp} \simeq \Gamma(|A| \multimap |B|^{\perp}) \simeq \Gamma(|B| \multimap |A|^{\perp}) \mid \\
 & & \quad x \in A_s \text{ implies } \mathsf{ev}' \circ (u \otimes x) \in B_t,\ y \in B_s \text{ implies } \mathsf{ev}' \circ (u \otimes y) \in A_t\}) \\
A^{\perp} & \equiv & (|A|^{\perp}, A_t, \{x^{\perp^{\perp}} \mid x \in A_s\})
\end{array}$$

□

We call $\mathbf{DG}(\Gamma)_s$ the *double subglueing* of $\mathbb{C}$ to $\mathbb{D}$ along Γ.

Example 3.23 (“Double” phase semantics)
As in Example 3.6, let us consider the case that $\mathbb{C}$ is $\mathbf{1}$ and $\mathbb{D}$ is $\mathbf{Set}$, thus a symmetric monoidal functor $\Gamma : \mathbb{C} \to \mathbb{D}$ is determined by a commutative monoid $M = (|M|, e, \cdot)$. By applying the double subglueing construction we have a poset $\mathcal{P}(|M|) \times \mathcal{P}(|M|)^{\mathrm{op}}$ (with the ordering $(A_s, A_t) \le (B_s, B_t)$ iff $A_s \subseteq B_s$ and also $B_t \subseteq A_t$) with the following $*$-autonomous structure.

$$\begin{array}{rcl}
I & = & (I, |M|) \\
(A_s, A_t) \otimes (B_s, B_t) & = & (A_s \otimes B_s, (A_s \multimap B_t) \cap (B_s \multimap A_t)) \\
(A_s, A_t)^{\perp} & = & (A_t, A_s)
\end{array}$$

where I, $\otimes$ and $\multimap$ in the right hand sides are those of Example 3.6. Other connectives are derived from them. Explicitly:

$$\begin{array}{rcl}
\perp & = & (|M|, I) \\
(A_s, A_t) \wp (B_s, B_t) & = & ((A_t \multimap B_s) \cap (B_t \multimap A_s), A_t \otimes B_t) \\
(A_s, A_t) \multimap (B_s, B_t) & = & ((A_s \multimap B_s) \cap (B_t \multimap A_t), A_s \otimes B_t)
\end{array}$$

□

Example 3.24 (Double glueing of Hyland and Tan)
If $\mathbb{C}$ is a locally small compact closed category [28], $\mathbb{D} \equiv \mathbf{Set}$ and $\Gamma \equiv \mathbb{C}(I, -)$, then $\mathbf{DG}(\Gamma)_s$ is exactly the double glueing $\mathbf{G}\mathbb{C}$ of Hyland and Tan. Explicitly, $\mathbf{G}\mathbb{C}$'s object is a triple $A = (|A| \in \mathbb{C}, A_s \subseteq \mathbb{C}(I, |A|), A_t \subseteq \mathbb{C}(I, |A|^*))$ and an arrow $f : A \to B$ in $\mathbf{G}\mathbb{C}$ is an arrow $f : |A| \to |B|$ in $\mathbb{C}$ satisfying $f \circ x \in B_s$ for $x \in A_s$ and also $f^* \circ y \in A_t$ for $y \in B_t$. As a leading example, from the compact closed category **Rel** of sets and binary relations, we obtain its double glueing **GRel** which is the category of “linear logical predicates” of Loader [32]. In her thesis [42], Tan has shown that the full completeness of $\mathbf{G}\mathbb{C}$ (as a $*$-autonomous category) is reduced to that of $\mathbb{C}$ (as a compact closed category), and derived full completeness results of various models of MLL. □

Example 3.25 (Parameterized predicates)
We can consider the “double” version of the parameterized predicates in Example 3.8. Let $\mathbb{I} : \mathbb{C}_0 \to \mathbb{C}_1$ be a strict symmetric monoidal functor from a small symmetric monoidal category $\mathbb{C}_0$ to a locally small $*$-autonomous category $\mathbb{C}_1$. By applying the double subglueing construction to the symmetric monoidal functor $\Gamma : X \mapsto \mathbb{C}_1(\mathbb{I}(-), X) : \mathbb{C}_1 \to \mathbf{Set}^{\mathbb{C}_0^{\mathrm{op}}}$, we obtain the $*$-autonomous category $\mathbf{DG}(\Gamma)_s$. We will concretely describe it in Section 4.5 as the category $\mathsf{D}\mathbb{C}_0\mathsf{PRED}$ of the “double parameterized predicates”. □

4 Logical Predicates for Linear Logic

We consider the notion of parameterized logical predicates for three fragments of linear logic: MILL (Multiplicative Intuitionistic Linear Logic), DILL (Dual Intuitionistic Linear Logic = MILL + modality !), and MLL (Multiplicative Linear Logic). In fact, the category-theoretic setting for these logical predicates have already appeared as examples of the glueing constructions in last section (Example 3.8, 3.17 and 3.25); in this section we turn them in a more syntax-oriented form.

4.1 Parameterized Predicates

Let $\mathbb{C}_0$ be a (small) symmetric monoidal category, $\mathbb{C}_1$ a (locally small) symmetric monoidal closed category and $\mathbb{I}$ be a strict symmetric monoidal functor from $\mathbb{C}_0$ to $\mathbb{C}_1$.

Definition 4.1 An $\mathrm{Obj}(\mathbb{C}_0)$-indexed set $P = \{P(X)\}_{X \in \mathbb{C}_0}$ is a $\mathbb{C}_0$*-predicate on* $A \in \mathbb{C}_1$ when

- $P(X) \subseteq \mathbb{C}_1(\mathbb{I}X, A)$ for $X \in \mathbb{C}_0$, and
- for $f \in \mathbb{C}_0(X,Y)$, $g \in P(Y)$ implies $g \circ \mathbb{I} f \in P(X)$. □

We may intuitively think that $\mathbb{C}_1(\mathbb{I}X, A)$ represents (a denotation of) the set of proofs of a sequent $X \vdash A$, and $\mathbb{C}_0$ (imported into $\mathbb{C}_1$ via $\mathbb{I}$) determines a property on proofs which is closed under tensor, composition and structural constructions. Unlike the traditional non-linear calculi and logical predicates over them, we explicitly state the “resource” X, which will play some significant role in our work. Then, for a $\mathbb{C}_0$-predicate P on A, $P(X)$ is a predicate on the proofs of $X \vdash A$. The second condition tells us that P is stable under the change of resource along some proof of $X \vdash Y$, provided that it satisfies the property $\mathbb{C}_0$.

Definition 4.2 Define the category of $\mathbb{C}_0$-predicates $\mathbb{C}_0\mathsf{PRED}$ as follows:

- an object of $\mathbb{C}_0\mathsf{PRED}$ is a pair (A, P) where P is a $\mathbb{C}_0$-predicate on $A \in \mathbb{C}_1$;
- an arrow from (A,P) to (B,Q) is an arrow $h \in \mathbb{C}_1(A,B)$ such that $g \in P(X)$ implies $h \circ g \in Q(X)$. □

Definition 4.3 For $\mathbb{C}_0$-predicates P on A and Q on B, define $\mathbb{C}_0$-predicates $P \otimes Q$ on $A \otimes B$ and $P \multimap Q$ on $A \multimap B$ as follows.

$$\begin{array}{rcl}(P \otimes Q)(X) & = & \{(g \otimes h) \circ \mathbb{I} f \mid \exists Y, Z \in \mathbb{C}_0 \ f \in \mathbb{C}_0(X, Y \otimes Z), g \in P(Y), h \in Q(Z)\} \\ (P \multimap Q)(X) & = & \{f \in \mathbb{C}_1(\mathbb{I}X, A \multimap B) \mid \forall Y \in \mathbb{C}_0 \ \forall g \in P(Y) \ \ \mathsf{ev} \circ (f \otimes g) \in Q(X \otimes Y)\}\end{array}$$

□

The reader should notice that $\mathbb{C}_0\mathsf{PRED}$ is no other than the subglueing of $\mathbb{C}_1$ to $\mathbf{Set}^{\mathbb{C}_0^{\mathrm{op}}}$ along the glueing functor $X \mapsto \mathbb{C}_1(\mathbb{I}-, X)$. As explained in Example 3.8, the definition of $P \otimes Q$ above is derived from the free symmetric monoidal cocompletion together with Lemma 3.5. However, here we also give a proof-theoretic explanation: a sequent $X \vdash A \otimes B$ can be derived as

$$\dfrac{\begin{array}{c}\Pi_f \\ \vdots \\ X \vdash Y \otimes Z\end{array} \qquad \dfrac{\begin{array}{c}\Pi_g \\ \vdots \\ Y \vdash A\end{array} \quad \begin{array}{c}\Pi_h \\ \vdots \\ Z \vdash B\end{array}}{Y, Z \vdash A \otimes B}\,(\otimes\textbf{-Intro})}{X \vdash A \otimes B}\,(\otimes\textbf{-Elim})$$

where $X \vdash Y \otimes Z$ splits a resource X to Y and Z which are used to prove A and B respectively. In general, such a splitting of resource is not unique, so we consider all possible cases such that (i) the proof Π_f of the splitting satisfies the "tensor-closed property" $\mathbb{C}_0$ and (ii) the proofs Π_g of $Y \vdash A$ and Π_h of $Z \vdash B$ satisfy the predicates $P(Y)$ and $Q(Z)$ respectively – in such cases we say that the derivation satisfies the property $(P \otimes Q)(X)$.

The definition of $P \multimap Q$ is in spirit the same as the usual definition of logical predicates; $M : A \Rightarrow B$ satisfies $P \Rightarrow Q$ if and only if $MN : B$ belongs to Q for any $N : A$ satisfying P. However, since our type theory is linear, we have to deal with the resources of terms linearly, and we explicitly state them in the definition: intuitively, $\Delta \vdash M : A \multimap B$ satisfies $P \multimap Q$ if and only if $\Delta, \Delta' \vdash MN : B$ satisfies Q for any $\Delta' \vdash N : A$ satisfying P.

Lemma 4.4 *For each $X, A \in \mathbb{C}_0$ define $\mathbb{P}_A(X) = \{\mathbb{I} f \mid f \in \mathbb{C}_0(X,A)\}$. Then:*

- *$\mathbb{P}_A$ is a $\mathbb{C}_0$-predicate on $\mathbb{I}A$.*
- *$f : (\mathbb{I}A, \mathbb{P}_A) \to (\mathbb{I}B, \mathbb{P}_B)$ in $\mathbb{C}_0\mathsf{PRED}$ if and only if $f = \mathbb{I}g$ for some $g \in \mathbb{C}_0(A,B)$.*
- *$\mathbb{P}_A \otimes \mathbb{P}_B = \mathbb{P}_{A \otimes B}$.* □

Proposition 4.5 *$\mathbb{C}_0\mathsf{PRED}$ becomes a symmetric monoidal closed category by the following data: the unit object is $(I, \mathbb{P}_I)$, tensor is given by $(A,P) \otimes (B,Q) = (A \otimes B, P \otimes Q)$, and we have exponents $(A,P) \multimap (B,Q) = (A \multimap B, P \multimap Q)$. Moreover $\mathbb{P}$ extends to a strict symmetric monoidal functor from $\mathbb{C}_0$ to $\mathbb{C}_0\mathsf{PRED}$ which is full.* □

Remark 4.6 If $\mathbb{C}_0$ is closed and $\mathbb{I}$ preserves exponents strictly, then so is $\mathbb{P}$ – in particular we have $\mathbb{P}_{A \multimap B} = \mathbb{P}_A \multimap \mathbb{P}_B$. □

4.2 Logical $\mathbb{C}_0$-Predicates

Suppose that we have $\mathbb{C}_0$, $\mathbb{C}_1$ and $\mathbb{I} : \mathbb{C}_0 \to \mathbb{C}_1$ as before. Also we fix an interpretation $[\![-]\!]_1$ of MILL in $\mathbb{C}_1$ (see Appendix A).

Definition 4.7 A type-indexed family $\{P_\sigma\}$ of $\mathbb{C}_0$-predicates is a *logical $\mathbb{C}_0$-predicate* if

- P_σ is a $\mathbb{C}_0$-predicate on $[\![\sigma]\!]_1$,
- $P_I = \mathbb{P}_I$,
- $P_{\sigma \otimes \tau} = P_\sigma \otimes P_\tau$,
- $P_{\sigma \multimap \tau} = P_\sigma \multimap P_\tau$, and
- $[\![c]\!]_1 : ([\![\sigma]\!]_1, P_\sigma) \to ([\![\tau]\!]_1, P_\tau)$ for each constant $c : \sigma \to \tau$. □

Note that a logical $\mathbb{C}_0$-predicate is completely determined by its instances at base types.

Given a logical $\mathbb{C}_0$-predicate $\{P_\sigma\}$, we can interpret MILL in $\mathbb{C}_0\mathsf{PRED}$ by $[\![b]\!] = ([\![b]\!]_1, P_b)$ for each base type b and $[\![c]\!] = [\![c]\!]_1 : ([\![\sigma]\!]_1, P_\sigma) \to ([\![\tau]\!]_1, P_\tau)$ for each constant $c : \sigma \to \tau$. Thus we have

Lemma 4.8 *(Basic Lemma for MILL) Let $\{P_\sigma\}$ be a logical $\mathbb{C}_0$-predicate. Then, for any term $\Delta \vdash M : \tau$, $[\![\Delta \vdash M : \tau]\!]_1 : ([\![|\Delta|]\!]_1, P_{|\Delta|}) \to ([\![\tau]\!]_1, P_\tau)$ holds.* □

Corollary 4.9 *If no $\multimap$ occurs in Δ, $[\![\Delta \vdash M : \tau]\!]_1 \in P_\tau([\![|\Delta|]\!]_0)$ holds for any $\Delta \vdash M : \tau$.* □

$\mathbb{C}_0$ itself determines a logical $\mathbb{C}_0$-predicate in a canonical way, provided that

- for each base type b there is an object $[\![b]\!]_0 \in \mathbb{C}_0$, and
- for each constant $c : \sigma \to \tau$ there is an arrow $[\![c]\!]_0 \in \mathbb{C}_0([\![\sigma]\!]_0, [\![\tau]\!]_0)$

where $[\![\sigma]\!]_0$ is defined inductively by $[\![I]\!]_0 = I$ and $[\![\sigma \otimes \tau]\!]_0 = [\![\sigma]\!]_0 \otimes [\![\tau]\!]_0$. Then we automatically have an interpretation $[\![-]\!]_1$ in $\mathbb{C}_1$ determined by $[\![b]\!]_1 = \mathbb{I}([\![b]\!]_0)$ and $[\![c]\!]_1 = \mathbb{I}([\![c]\!]_0)$. Now define the *canonical logical* $\mathbb{C}_0$*-predicate* $\{\mathbb{P}^*_\sigma\}$ by $\mathbb{P}^*_b = \mathbb{P}_{[\![b]\!]_0}$ (as noted above, a logical $\mathbb{C}_0$-predicate is determined by its instances at base types). Basic Lemma for the canonical logical $\mathbb{C}_0$-predicate implies that, at $\multimap$-free types (at any types if $\mathbb{C}_0$ and $\mathbb{I}$ are closed) a definable element must be in the image of $\mathbb{I}$.

Example 4.10

- If $\mathbb{C}_0$ is equivalent to the one object one arrow category (thus we have no base type nor constant), then the canonical logical $\mathbb{C}_0$-predicate specifies the canonical isomorphisms between objects generated from I.
- If $\mathbb{C}_0 = \mathbb{C}_1$ and $\mathbb{I}$ is the identity functor, then every morphism between MILL-definable objects satisfies the canonical logical $\mathbb{C}_0$-predicate. $\square$

4.3 Binary Logical $\mathbb{C}_0$-Relations

It is straightforward to generalize (or specialize) our logical predicates to multiple arguments, i.e. *logical relations*, in the same way as demonstrated in [36]. Here we spell out the case of binary logical relations. Suppose that $\mathbb{C}_0$ is a (small) symmetric monoidal category, $\mathbb{C}_1$ and $\mathbb{C}_2$ are (locally small) symmetric monoidal closed categories and that $\mathbb{I}_1 : \mathbb{C}_0 \to \mathbb{C}_1$ and $\mathbb{I}_2 : \mathbb{C}_0 \to \mathbb{C}_2$ are strict symmetric monoidal functors. A binary $\mathbb{C}_0$-relation is no other than a $\mathbb{C}_0$-predicate obtained by replacing $\mathbb{C}_1$ by $\mathbb{C}_1 \times \mathbb{C}_2$ and $\mathbb{I}$ by $\langle \mathbb{I}_1, \mathbb{I}_2 \rangle : \mathbb{C}_0 \to \mathbb{C}_1 \times \mathbb{C}_2$. Explicitly:

Definition 4.11 An Obj($\mathbb{C}_0$)-indexed set $R = \{R(X)\}_{X \in \mathbb{C}_0}$ is a $\mathbb{C}_0$*-relation on* $(A,B) \in \mathbb{C}_1 \times \mathbb{C}_2$ when

- $R(X) \subseteq \mathbb{C}_1(\mathbb{I}_1 X, A) \times \mathbb{C}_2(\mathbb{I}_2 X, B)$ for $X \in \mathbb{C}_0$, and
- for $f \in \mathbb{C}_0(X,Y)$, $(g,h) \in P(Y)$ implies $(g \circ \mathbb{I}_1 f, h \circ \mathbb{I}_2 f) \in P(X)$. $\square$

Definition 4.12 Define the category of $\mathbb{C}_0$-relations $\mathbb{C}_0\mathsf{REL}$ as follows:

- an object of $\mathbb{C}_0\mathsf{REL}$ is a triple (A,B,R) where R is a $\mathbb{C}_0$-relation on (A,B);
- an arrow from (A,B,R) to (A',B',R') is a pair of arrows $(h \in \mathbb{C}_1(A,A'), k \in \mathbb{C}_2(B,B'))$ such that $(f,g) \in R(X)$ implies $(h \circ f, k \circ g) \in R'(X)$. $\square$

From Proposition 4.5 we know that $\mathbb{C}_0\mathsf{REL}$ is a symmetric monoidal closed category. Again explicitly:

Definition 4.13 For $\mathbb{C}_0$-relations R on (A,B) and R' on (A',B'), define $\mathbb{C}_0$-relations $R \otimes R'$ on $(A \otimes A', B \otimes B')$ and $R \multimap R'$ on $(A \multimap A', B \multimap B')$ as follows.

$$(R\otimes R')(X) \;=\; \left\{ ((g\otimes g')\circ \mathbb{I}_1 f, (h\otimes h')\circ \mathbb{I}_2 f) \;\middle|\; \begin{array}{l} \exists Y,Z\in \mathbb{C}_0 \;\; f\in \mathbb{C}_0(X, Y\otimes Z), \\ (g,h)\in R(Y),\; (g',h')\in R'(Z) \end{array} \right\}$$

$$(R\multimap R')(X) \;=\; \left\{ (f,g) \;\middle|\; \begin{array}{l} \forall Y\in \mathbb{C}_0 \;\; (a,b)\in R(Y) \text{ implies} \\ (\mathsf{ev}\circ(f\otimes a), \mathsf{ev}\circ(g\otimes b))\in R'(X\otimes Y) \end{array} \right\}$$

□

Note that, in the relational notation, $R\multimap R'$ can be given as

$$f\ (R\multimap R')(X)\ g \quad \text{iff} \quad a\ R(Y)\ b \ \text{ implies } \ \mathsf{ev}\circ(f\otimes a)\ R'(X\otimes Y)\ \mathsf{ev}\circ(g\otimes b)$$

Therefore the results of the applications of related functions to related elements are again related – this is the key property of the logical relations. The only novelty in our definition is that we take care about the linearly used resources X and Y.

Now fix interpretations $[\![-]\!]_1$ and $[\![-]\!]_2$ of MILL in $\mathbb{C}_1$ and $\mathbb{C}_2$ respectively.

Definition 4.14 A type-indexed family $\{R_\sigma\}$ of $\mathbb{C}_0$-relations is a *logical* $\mathbb{C}_0$*-relation* if

- R_σ is a $\mathbb{C}_0$-relation on $([\![\sigma]\!]_1, [\![\sigma]\!]_2)$,
- $R_I(X) = \{(\mathbb{I}_1 f, \mathbb{I}_2 f) \mid f\in \mathbb{C}_0(X,I)\}$, $R_{\sigma\otimes\tau} = R_\sigma \otimes R_\tau$, $R_{\sigma\multimap\tau} = R_\sigma \multimap R_\tau$, and
- $([\![c]\!]_1, [\![c]\!]_2) : ([\![\sigma]\!]_1, [\![\sigma]\!]_2, R_\sigma) \to ([\![\tau]\!]_1, [\![\tau]\!]_2, R_\tau)$ for each constant $c:\sigma\to\tau$. □

Lemma 4.15 *(Basic Lemma, binary version) Let $\{R_\sigma\}$ be a logical $\mathbb{C}_0$-relation. Then, for any term $\Delta\vdash M:\tau$, $([\![\Delta\vdash M:\tau]\!]_1, [\![\Delta\vdash M:\tau]\!]_2) : ([\![|\Delta|]\!]_1, [\![|\Delta|]\!]_2, R_{|\Delta|}) \to ([\![\tau]\!]_1, [\![\tau]\!]_2, R_\tau)$ holds.* □

4.4 Dual Intuitionistic Linear Logic

Now we enrich our logic and calculus with the modality !. There are many possible choices for this, see for instance [10]. Here we choose the formulation due to Barber and Plotkin, called Dual Intuitionistic Linear Logic (DILL) [6] for its simple syntax and equational theory, as well as for the well-established category-theoretic models of DILL in terms of symmetric monoidal adjunctions[1]. Alternatively we could use Benton's Linear Non-Linear Logic (LNL Logic) [9] which has essentially the same class of category-theoretic models as DILL. The syntax and semantics of DILL are recalled in Appendix B.

Consider a commutative diagram of functors

$$\begin{tikzcd}
\mathbb{C}_0 \arrow[r, "F_0"] \arrow[d, "\mathbb{I}"'] & \mathbb{D}_0 \arrow[d, "\mathbb{J}"] \\
\mathbb{C}_1 \arrow[r, "F_1"'] & \mathbb{D}_1
\end{tikzcd}$$

---

[1] The term "dual" refers to the double context of the type system, and has nothing to do with the duality in classical linear logic

in which $\mathbb{C}_0$ and $\mathbb{C}_1$ are cartesian categories, $\mathbb{D}_0$ symmetric monoidal and $\mathbb{D}_1$ symmetric monoidal closed; and F_0, F_1 are strong symmetric monoidal while $\mathbb{I}$, $\mathbb{J}$ are strict symmetric monoidal. Moreover assume that F_1 has a right adjoint $U_1 : \mathbb{D}_1 \to \mathbb{C}_1$.

As before, we define the categories of $\mathbb{C}_0$- and $\mathbb{D}_0$-predicates – let us call them $\mathbb{C}_0\mathsf{PRED}$ and $\mathbb{D}_0\mathsf{PRED}$ respectively. Note that $\mathbb{C}_0\mathsf{PRED}$ is a cartesian category (actually cartesian closed if $\mathbb{C}_1$ is closed) with products given by $(A,P)\times(B,Q) = (A\times B, P\times Q)$ where

$$(P\times Q)(X) \;=\; \{\langle f,g\rangle \mid f\in P(X), g\in Q(X)\}$$

for $\mathbb{C}_0$-predicates P and Q (which coincides with $P\otimes Q$ in Definition 4.3).

Now we give a symmetric monoidal adjunction between $\mathbb{C}_0\mathsf{PRED}$ and $\mathbb{D}_0\mathsf{PRED}$, by applying Example 3.17. For a $\mathbb{C}_0$-predicate P on $A\in\mathbb{C}_1$, define a $\mathbb{D}_0$-predicate $L(P)$ on $F_1A\in\mathbb{D}_1$ by

$$L(P)(Y) \;=\; \{F_1 g\circ \mathbb{J}f \mid \exists X\in\mathbb{C}_0 \;\; f\in\mathbb{D}_0(Y,F_0X), g\in P(X)\}$$

and, for a $\mathbb{D}_0$-predicate Q on $B\in\mathbb{D}_1$, define a $\mathbb{C}_0$-predicate $\widehat{F_0}(Q)$ on $U_1B\in\mathbb{C}_1$ by

$$(\widehat{F_0}(Q))(X) \;=\; \{f^*\in\mathbb{C}_1(\mathbb{I}X,U_1B) \mid f\in Q(F_0X)\subseteq\mathbb{D}_1(\mathbb{J}F_0X,B)=\mathbb{D}_1(F_1\mathbb{I}X,B)\}$$

where $f^* : \mathbb{I}X \to U_1B$ is the adjunct of $f : F_1\mathbb{I}X \to B$.

Proposition 4.16 *L and $\widehat{F_0}$ extend to functors between $\mathbb{C}_0\mathsf{PRED}$ and $\mathbb{D}_0\mathsf{PRED}$. Moreover L is strong symmetric monoidal, and left adjoint to $\widehat{F_0}$.* □

Therefore we have a symmetric monoidal adjunction $\mathbb{C}_0\mathsf{PRED} \overset{L}{\underset{\widehat{F_0}}{\overset{\longrightarrow}{\underset{\longleftarrow}{\perp}}}} \mathbb{D}_0\mathsf{PRED}$ between a cartesian category $\mathbb{C}_0\mathsf{PRED}$ and a symmetric monoidal closed category $\mathbb{D}_0\mathsf{PRED}$. Let ! be the induced comonad on $\mathbb{D}_0\mathsf{PRED}$, that is, we define a $\mathbb{D}_0$-predicate $!P$ on F_1U_1A by

$$(!P)(Y) \;=\; \{F_1 g^*\circ\mathbb{J}f \mid \exists X\in\mathbb{C}_0 \;\; f\in\mathbb{D}_0(Y,F_0X), g\in P(F_0X)\}$$

for a $\mathbb{D}_0$-predicate P on A. As explained in Example 3.17, $!P$ is derived from a left Kan extension together with the subglueing version of Lemma 3.12, but it can be explained more or less intuitively (proof-theoretically) as follows. A sequent $\emptyset\,;\,Y\vdash !A$ can be proved as

$$\dfrac{\begin{array}{c}\Pi_f\\ \vdots\\ \emptyset\,;\,Y\vdash !X\end{array}\qquad \dfrac{\begin{array}{c}\Pi_g\\ \vdots\\ X\,;\,\emptyset\vdash A\end{array}}{X\,;\,\emptyset\vdash !A}\,(!\text{-}\mathbf{Intro})}{\emptyset\,;\,Y\vdash !A}\,(!\text{-}\mathbf{Elim})$$

where $\emptyset\,;\,Y\vdash !X$ converts a linear resource Y to $!X$ which is used non-linearly in $X\,;\,\emptyset\vdash !A$ to produce $!A$. Taking all such possible cases into account, we say that the proof satisfies $(!P)(Y)$ when Π_f belongs to $\mathbb{D}_0$ and Π_g satisfies $P(X)$.

Now let us fix an interpretation $[\![-]\!]_1$ of DILL in $\mathbb{C}_1 \overset{F_1}{\underset{U_1}{\overset{\longrightarrow}{\underset{\longleftarrow}{\perp}}}} \mathbb{D}_1$ (see Appendix B).

Definition 4.17 A type-indexed family $\{P_\sigma\}$ of $\mathbb{D}_0$-predicates is a *logical* $(\mathbb{C}_0 \xrightarrow{F_0} \mathbb{D}_0)$*-predicate* if

- P_σ is a $\mathbb{D}_0$-predicate on $[\![\sigma]\!]_1$,

- $P_I = \mathbb{P}_I$, $P_{\sigma\otimes\tau} = P_\sigma \otimes P_\tau$, $P_{\sigma\multimap\tau} = P_\sigma \multimap P_\tau$ and $P_{!\sigma} = !P_\sigma$ hold, and
- $[\![c]\!]_1 : ([\![\sigma]\!]_1, P_\sigma) \to ([\![\tau]\!]_1, P_\tau)$ for each constant $c : \sigma \to \tau$. □

Lemma 4.18 *(Basic Lemma for DILL) Let $\{P_\sigma\}$ be a logical $(\mathbb{C}_0 \xrightarrow{F_0} \mathbb{D}_0)$-predicate. Then, for any term $\Gamma\,;\,\Delta \vdash M : \tau$, $[\![\Gamma\,;\,\Delta \vdash M : \tau]\!]_1 : ([\![|\Gamma\,;\,\Delta|]\!]_1, P_{|\Gamma\,;\,\Delta|}) \to ([\![\tau]\!]_1, P_\tau)$ holds.* □

$(\mathbb{C}_0 \xrightarrow{F_0} \mathbb{D}_0)$ itself determines the *canonical logical* $(\mathbb{C}_0 \xrightarrow{F_0} \mathbb{D}_0)$*-predicate* when

- for each base type b there is an object $[\![b]\!]_0 \in \mathbb{D}_0$, and
- for each constant $c : \sigma \to \tau$ there is an arrow $[\![c]\!]_0 \in \mathbb{D}_0([\![\sigma]\!]_0, [\![\tau]\!]_0)$

where $[\![\sigma]\!]_0$ is defined inductively by $[\![I]\!]_0 = I$ and $[\![\sigma \otimes \tau]\!]_0 = [\![\sigma]\!]_0 \otimes [\![\tau]\!]_0$. In such cases we automatically have an interpretation $[\![-]\!]_1$ in $\mathbb{D}_1$ determined by $[\![b]\!]_1 = \mathbb{J}([\![b]\!]_0)$ and $[\![c]\!]_1 = \mathbb{J}([\![c]\!]_0)$, and the canonical logical $(\mathbb{C}_0 \xrightarrow{F_0} \mathbb{D}_0)$-predicate $\{\mathbb{P}^*_\sigma\}$ is determined by $\mathbb{P}^*_b = \mathbb{P}_{[\![b]\!]_0}$.

4.5 Multiplicative Linear Logic

Let $\mathbb{C}_0$ be a (small) symmetric monoidal category, $\mathbb{C}_1$ a (locally small) $*$-autonomous category and $\mathbb{I}$ be a strict symmetric monoidal functor from $\mathbb{C}_0$ to $\mathbb{C}_1$.

Definition 4.19 A *double* $\mathbb{C}_0$*-predicate on* $A \in \mathbb{C}_1$ is a pair $P = (P_s, P_t)$ such that P_s is a $\mathbb{C}_0$-predicate on A and P_t is a $\mathbb{C}_0$-predicate on $A^\perp$. □

Definition 4.20 Define the category of double $\mathbb{C}_0$-predicates $\mathsf{D}\mathbb{C}_0\mathsf{PRED}$ as follows:

- an object of $\mathsf{D}\mathbb{C}_0\mathsf{PRED}$ is a pair (A, P) where $P = (P_s, P_t)$ is a double $\mathbb{C}_0$-predicate on $A \in \mathbb{C}_1$;
- an arrow from (A, P) to (B, Q) is an arrow $h \in \mathbb{C}_1(A, B)$ such that $g \in P_s(X)$ implies $h \circ g \in Q_s(X)$ and $g \in Q_t(X)$ implies $h^\perp \circ g \in P_t(X)$. □

Definition 4.21 For double $\mathbb{C}_0$-predicates $P = (P_s, P_t)$ on A and $Q = (Q_s, Q_t)$ on B, define a double $\mathbb{C}_0$-predicate $P \otimes Q = ((P \otimes Q)_s, (P \otimes Q)_t)$ on $A \otimes B$ by

$$\begin{array}{rcl}
(P\otimes Q)_s(X) & = & \{(a \otimes b) \circ \mathbb{I}(h) \mid h \in \mathbb{C}_0(X, Y \otimes Z), a \in P_s(Y), b \in Q_s(Z)\} \\
(P\otimes Q)_t(X) & = & \left\{ \begin{array}{rcl|l} f & \in & \mathbb{C}_1(\mathbb{I}(X), (A \otimes B)^\perp) & a \in P_s(Y) \text{ implies} \\ & \simeq & \mathbb{C}_1(\mathbb{I}(X) \otimes A, B^\perp) & f \circ (id \otimes a) \in Q_t(X \otimes Y), \\ & \simeq & \mathbb{C}_1(\mathbb{I}(X) \otimes B, A^\perp) & b \in Q_s(Y) \text{ implies} \\ & & & f \circ (id \otimes b) \in P_t(X \otimes Y) \end{array} \right\}
\end{array}$$

Also, for a double $\mathbb{C}_0$-predicate $P = (P_s, P_t)$ on A, define a double $\mathbb{C}_0$-predicate $P^\perp = (P^\perp_s, P^\perp_t)$ on $A^\perp$ by $P^\perp_s = P_t$ and $P^\perp_t(X) = \{\mathbb{I}X \xrightarrow{f} A \xrightarrow{\simeq} (A^\perp)^\perp \mid f \in P_s(X)\}$. □

Lemma 4.22 *For each $X, A \in \mathbb{C}_0$, define $\mathbb{P}_{As}(X) = \{\mathbb{I}f \mid f \in \mathbb{C}_0(X, A)\}$ and $\mathbb{P}_{At}(X) = \mathbb{C}_1(\mathbb{I}X, (\mathbb{I}A)^\perp)$. Then*

- *$\mathbb{P}_A = (\mathbb{P}_{As}, \mathbb{P}_{At})$ is a double $\mathbb{C}_0$-predicate on $\mathbb{I}A$.*
- *$f : (\mathbb{I}A, \mathbb{P}_A) \to (\mathbb{I}B, \mathbb{P}_B)$ in $\mathsf{D}\mathbb{C}_0\mathsf{PRED}$ if and only if $f = \mathbb{I}g$ for some $g \in \mathbb{C}(A, B)$.*

- $\mathbb{P}_A \otimes \mathbb{P}_B = \mathbb{P}_{A\otimes B}$. ☐

Proposition 4.23 *$\mathsf{D}\mathbb{C}_0\mathsf{PRED}$ becomes a $*$-autonomous category by the following data: the unit object is $(I, \mathbb{P}_I)$, tensor is given by $(A,P)\otimes(B,Q) = (A\otimes B, P\otimes Q)$ while the duality is $(A,P)^\perp = (A^\perp, P^\perp)$. Moreover $\mathbb{P}$ extends to a strict symmetric monoidal functor from $\mathbb{C}_0$ to $\mathsf{D}\mathbb{C}_0\mathsf{PRED}$ which is full.* ☐

Let us fix an interpretation $[\![-]\!]_1$ of MLL in $\mathbb{C}_1$.

Definition 4.24 A type-indexed family $\{P_\sigma\}$ of double $\mathbb{C}_0$-predicates is a *double logical $\mathbb{C}_0$-predicate* if

- P_σ is a double $\mathbb{C}_0$-predicate on $[\![\sigma]\!]_1$,
- $P_I = \mathbb{P}_I$, $P_{\sigma\otimes\tau} = P_\sigma \otimes P_\tau$ and $P_{\sigma^\perp} = P_\sigma^\perp$,
- $[\![c]\!]_1 : ([\![\sigma]\!]_1, P_\sigma) \to ([\![\tau]\!]_1, P_\tau)$ for each constant $c : \sigma \to \tau$. ☐

Lemma 4.25 *(Basic Lemma for MLL) Let $\{P_\sigma\}$ be a double logical $\mathbb{C}_0$-predicate. Then, for any term $\vdash M : \sigma$ of MLL, $[\![\vdash M : \sigma]\!]_1 \in (P_\sigma)_s(I)$ holds.* ☐

5 Fully Complete Translations

As an application of our logical predicates (hence of our glueing constructions), we can show that several translations between fragments of linear logic are *fully complete*, i.e. it is not just conservative but also full.

5.1 A Case Study: From MILL to MLL

Let us spell out the case of the embedding from MILL to MLL, under the assumption that they have the same base types and constants. The translation $(-)^\circ$ at the type level is given by $b^\circ = b$, $I^\circ = I$, $(\sigma\otimes\tau)^\circ = \sigma^\circ \otimes \tau^\circ$ and $(\sigma \multimap \tau)^\circ = (\sigma \otimes \tau^\perp)^\perp$. We omit the translation at the proof level, since it should be obvious for those familar with linear logic, and also it requires us to present the proof theory (proof nets) of MLL which needs some space; see e.g. [12] for a complete description. While MLL is a richer theory than MILL (MLL contains lots of types which are not definable in MILL), it is not very obvious how proofs of these two theories can be related; we show that they are in fact in a strongest relation.

Let $\mathbb{C}_0$ be a term model of MILL, and $\mathbb{C}_1$ be that of MLL. Thus $\mathbb{C}_0$ is a small symmetric monoidal closed category which is freely generated from a fixed set of base objects and constant arrows, and $\mathbb{C}_1$ is a small $*$-autonomous category freely generated from the same set of base objects and constant arrows.

The strict symmetric monoidal closed embedding $\mathbb{I} : \mathbb{C}_0 \to \mathbb{C}_1$ then corresponds to the syntactic translation from MILL to MLL. The translation is sound, because $\mathbb{I}$ preserves the symmetric monoidal closed structure strictly. It is also conservative, because $\mathbb{I}$ is faithful; this can be shown by the following observation: given a small symmetric monoidal closed category $\mathbb{C}$, we can always construct a $*$-autonomous category $\mathbb{C}'$ to which $\mathbb{C}$ (fully and) faithfully embeds – such $\mathbb{C}'$ is obtained, for example, by applying the Chu construction [8] to $\mathbf{Set}^{\mathbb{C}^{op}}$. In general this kind of model construction technique is useful for showing the conservativity of syntactic translations, see for example [18, 5].

A much harder property to show is that the translation is *full*, i.e.

> *if an expression of the target theory has types which are definable in the source theory, it is already definable in the source theory.*

Equivalently it amounts to the fullness of $\mathbb{I}$. We say that a translation between type theories is *fully complete* if it is both conservative and full. (The term "full completeness" was coined by Abramsky and Jagadeesan in [2]. In their original use, full completeness means that any arrow of the semantic category is the denotation of a syntactic element; we adapt this for the case that the semantic category itself is syntactically defined as the term model of a type theory.)

We show that fullness is an immediate consequence of our logical predicates method described in Section 4.5. First, we define a double logical $\mathbb{C}_0$-predicate $\{\mathbb{P}^*_\sigma\}$ by $\mathbb{P}^*_b = \mathbb{P}_b$ – note that $\{\mathbb{P}^*_\sigma\}$ is completely determined by its instances at base types. Since we suppose that MILL and MLL share the same constants, the interpretation of a constant trivially respects this double logical $\mathbb{C}_0$-predicate. Also we note that $\mathbb{P}_{\sigma\multimap\tau} = \mathbb{P}_\sigma \multimap \mathbb{P}_\tau$ holds (in fact we can show that $\mathbb{P}$ extends to a strict symmetric monoidal closed functor from $\mathbb{C}_0$ to $\mathsf{D}\mathbb{C}_0\mathsf{PRED}$).

Now we see that, by definition, at a MILL-definable type σ, $x \in (\mathbb{P}^*_\sigma)_s(I)$ means that x is in the image of $\mathbb{I}$; therefore, by the Basic Lemma for MLL, an element of MLL with MILL-definable type is in fact definable in MILL.

Proposition 5.1 *For any term* $\vdash M : \sigma^\circ$ *of MLL with a MILL-definable type* σ*, there exists a term* $\vdash N : \sigma$ *of MILL such that* $\vdash M = N^\circ : \sigma^\circ$ *holds in MLL.* □

Theorem 5.2 *The translation from MILL into MLL is fully complete.* □

5.2 Full Completeness, Semantically

In fact the argument above shows a stronger result, which can be stated without mentioning the specific syntactic theories. Given a small symmetric monoidal closed category $\mathbb{C}$, we can construct its *relatively free* $*$*-autonomous category* $\mathcal{F}\mathbb{C}$ together with a strict symmetric monoidal closed functor $i : \mathbb{C} \to \mathcal{F}\mathbb{C}$, which satisfy the following universal property:

> *For any strict symmetric monoidal closed functor* $F : \mathbb{C} \to \mathbb{D}$ *to a* $*$*-autonomous category* $\mathbb{D}$*, there is a unique* $*$*-autonomous functor* $F^* : \mathcal{F}\mathbb{C} \to \mathbb{D}$ *making the following diagram commute.*

$$\begin{tikzcd}
\mathbb{C} \arrow[r, "i"] \arrow[dr, "F"'] & \mathcal{F}\mathbb{C} \arrow[d, dotted, "F^*"] \\
 & \mathbb{D}
\end{tikzcd}$$

(Equivalently, we can define i as the unit of the adjunction of the free construction.) Then we have

Theorem 5.3 *The embedding* $i : \mathbb{C} \to \mathcal{F}\mathbb{C}$ *is full and faithful.*

Proof: Faithfulness is shown by the model construction argument. Fullness follows from the commutative diagrams

$$\begin{tikzcd}
 & \mathcal{F}\mathbb{C} \arrow[d, "\mathbb{P}^*"] \arrow[ddr, "Id"] & \\
\mathbb{C} \arrow[ur, "i"] \arrow[r, "\mathbb{P}"] \arrow[drr, "i"'] & \mathsf{D}\mathbb{C}\mathsf{PRED} \arrow[dr, "p"] & \\
 & & \mathcal{F}\mathbb{C}
\end{tikzcd}$$

where the right triangle commutes because of the universal property of i. Since both $\mathbb{P}$ and p are full, so is $i = p \circ \mathbb{P}$. □

Note that the assumptions we have used here are

- $i : \mathbb{C} \to \mathcal{F}\mathbb{C}$ is relatively free, i.e., satisfies some suitable universal property
- $\mathbb{P} : \mathbb{C} \to \mathsf{D}\mathbb{C}\mathsf{PRED}$ preserves the structure that $\mathbb{C}$ has; and
- $\mathbb{P}$ is full .

This argument is applicable for many similar situations as well. The first condition follows by definition, whereas the third is always true for our glued categories (Proposition 4.5 and 4.23). Therefore, for each case we have only to check the second point, i.e., if the embedding $\mathbb{P}$ into the glued category preserves the required structure. If this is done, then the fullness follows immediately. To turn the fullness of i to the fullness of a translation between syntactically defined theories, we need to show that the term models of the theories and the induced functor between them satisfy the relative freeness, which often follows from just a routine verification (this is the case for MILL and MLL).

In the rest of this section we give a series of full completeness results obtained by this argument.

5.3 Examples

Theorem 5.4 *The embedding from a small symmetric monoidal category into its relatively free symmetric monoidal closed category is fully faithful.*

Proof: We follow the same argument as the last section. Let $\mathbb{C}$ be a small symmetric monoidal category and $\mathcal{F}\mathbb{C}$ be its relatively free symmetric monoidal closed category, with $\mathbb{I} : \mathbb{C} \to \mathcal{F}\mathbb{C}$ the embedding. From Proposition 4.5 we know that $\mathbb{P} : \mathbb{C} \to \mathbb{C}\mathsf{PRED}$ strictly preserves the symmetric monoidal structure and is full. Therefore the same argument as the case of MILL to MILL shows that $\mathbb{I}$ is full. Faithfulness follows by constructing a symmetric monoidal closed category to which $\mathbb{C}$ faithfully embeds (use the free symmetric monoidal cocompletion). □

Corollary 5.5 *The embedding from the $I, \otimes$-fragment of MILL into MILL is fully complete.* □

For proving syntax-oriented (hence restricted) results directly, it is convenient to use the Basic Lemma.

Theorem 5.6 *The translation from MILL to DILL (which sends $\Delta \vdash M : \sigma$ to $\emptyset\ ;\ \Delta \vdash M : \sigma$) is fully complete.*

Proof: Let $\mathbb{C}_0$ be equivalent to $\mathbf{1}$, $\mathbb{D}_0$ be the term model of MILL and $\mathbb{C}_1 \underset{\xleftarrow[U_1]{}}{\overset{\xrightarrow{F_1}}{\perp}} \mathbb{D}_1$ be the term model of DILL, while $\mathbb{I} : \mathbb{C}_0 \rightarrow \mathbb{C}_1$ and $F : \mathbb{C}_0 \rightarrow \mathbb{D}_0$ is the obvious functor and $\mathbb{J} : \mathbb{D}_0 \rightarrow \mathbb{D}_1$ is the strict symmetric monoidal closed functor corresponding to the translation from MILL to DILL. By applying the Basic Lemma for DILL (Lemma 4.18) to the canonical logical predicate, we know that a DILL-definable element at MILL-definable type satisfies the canonical logical predicate, thus is definable in MILL. $\Box$

Theorem 5.7 *The translation from a sharing theory [22] (action calculus [34]) to DILL is fully complete.* $\Box$

This is spelled out in Appendix C. The proof makes use of the Basic Lemma for DILL (Lemma 4.18).

Remark 5.8 Yet there are examples of fully faithful relatively free constructions for which our method cannot be applied (at least directly). One such case is the embedding of a small symmetric linearly distributive category to its relatively free $*$-autonomous category; its full faithfullness is shown in [12] using a normal form result on proof nets. Our argument does not work because, for a small symmetric linearly distributive category $\mathbb{C}$, the functor $\mathbb{P} : \mathbb{C} \rightarrow \mathsf{D}\mathbb{C}\mathsf{PRED}$ may not preserve $\perp$ and $\wp$. $\Box$

References

[1] Abramsky, S., Gay, S.J., and Nagarajan, R. (1996), Specification structures and propositions-as-types for concurrency, *in* “Proceedings, 8th Banff Higher Order Workshop”, Lecture Notes in Computer Science, Vol. 1043, pp. 5–40, Springer-Verlag.

[2] Abramsky, S., and Jagadeesan, R. (1994), Games and full completeness for multiplicative linear logic, *J. Symbolic Logic* **59**, 543–574.

[3] Alimohamed, M. (1995), A characterization of lambda definability in categorical models of implicit polymorphism, *Theoret. Comp. Sci.* **146**, 5–23.

[4] Ambler, S.J. (1992), “First Order Linear Logic in Symmetric Monoidal Closed Categories”, Ph.D. thesis, ECS-LFCS-92-194, University of Edinburgh.

[5] Barber, A., Gardner, P., Hasegawa, M., and Plotkin, G. (1998), From action calculi to linear logic, *in* “Computer Science Logic (CSL’97), Selected Papers”, Lecture Notes in Computer Science, Vol. 1414, pp. 78–97, Springer-Verlag.

[6] Barber, A., and Plotkin, G. (1998), Dual intuitionistic linear logic, submitted.

[7] Barr, M. (1979), “$*$-Autonomous Categories”, Lecture Notes in Mathematics, Vol. 752, Springer-Verlag.

[8] Barr, M. (1991), $*$-autonomous categories and linear logic, *Math. Struct. Comp. Sci.* **1**, 159–178.

[9] Benton, N. (1995), A mixed linear non-linear logic: proofs, terms and models, *in* “Computer Science Logic (CSL’94), Selected Papers”, Lecture Notes in Computer Science, Vol. 933, pp. 121–135, Springer-Verlag.

[10] Benton, N., Bierman, G.M., de Paiva, V., and Hyland, J.M.E. (1993), Linear lambda-calculus and categorical models revisited, *in* "Computer Science Logic (CSL'92), Selected Papers", Lecture Notes in Computer Science, Vol. 702, pp. 61–84, Springer-Verlag.

[11] Bierman, G.M. (1995), What is a categorical model of intuitionistic linear logic? *in* "Proceedings, Typed Lambda Calculi and Applications (TLCA'95)", Lecture Notes in Computer Science, Vol. 902, pp. 78–93, Springer-Verlag.

[12] Blute, R.F., Cockett, J.R.B., Seely, R.A.G., and Trimble, T.H. (1996), Natural deduction and coherence for weakly distributive categories, *J. Pure Appl. Algebra* **113**, 229–296.

[13] Casley, R.T., Crew, R.F., Meseguer, J., and Pratt, V.R. (1991), Temporal structures, *Math. Struct. Comp. Sci.* **1**, 179–213.

[14] Corradini, A., and Gadducci, F. (1997), A 2-categorical presentation of term graphs, *in* "Proceedings, Category Theory and Computer Science (CTCS'97)", Lecture Notes in Computer Science, Vol. 1290, pp. 87–105, Springer-Verlag.

[15] Crole, R. (1993), "Categories for Types", Cambridge University Press.

[16] Eilenberg, S., and Kelly, G.M. (1966), Closed categories, *in* "Proceedings, Categorical Algebra (La Jolla 1965)", pp. 421-562, Springer-Verlag.

[17] Fiore, M., and Simpson, A. (1998), Lambda definability with sums via Grothendieck logical relations, *to appear in* "Proceedings, Typed Lambda Calculi and Applications (TLCA'99)", Lecture Notes in Computer Science, Springer-Verlag.

[18] Gardner, P., and Hasegawa, M. (1997), Types and models for higher-order action calculi, *in* "Proceedings, Theoretical Aspects of Computer Software (TACS'97)", Lecture Notes in Computer Science , Vol. 1281, pp. 583–603, Springer-Verlag.

[19] Garrigue, J., and Minamide, Y. (1998), On the runtime complexity of type-directed unboxing, *in* "Proceedings, International Conference on Functional Programming (ICFP'98)", pp. 1–12, ACM Press.

[20] Girard, J.-Y. (1987), Linear logic, *Theoret. Comp. Sci.* **50**, 1–102.

[21] Girard, J.-Y. (1995), Linear logic: its syntax and semantics, *in* "Advances in Linear Logic", pp. 1–42, Cambridge University Press.

[22] Hasegawa, M. (1997), "Models of Sharing Graphs (A Categorical Semantics of Let and Letrec)", Ph.D. thesis, ECS-LFCS-97-360, University of Edinburgh; to be published from Springer-Verlag as a volume of Distinguished Dissertation Series.

[23] Hasegawa, M. (1998), Logical predicates for intuitionistic linear type theories, *to appear in* "Proceedings, Typed Lambda Calculi and Applications (TLCA'99)", Lecture Notes in Computer Science, Springer-Verlag.

[24] Hermida, C. (1993), "Fibrations, Logical Predicates and Indeterminates", Ph.D. thesis, ECS-LFCS-93-277, University of Edinburgh.

[25] Im, G.B., and Kelly, G.M. (1986), A universal property of the convolution monoidal structure, *J. Pure Appl. Algebra* **43**, 75–88.

[26] Jung, A., and Tiuryn, J. (1993), A new characterisation of lambda definability, *in* "Proceedings, Typed Lambda Calculi and Applications (TLCA'93)", Lecture Notes in Computer Science, Vol. 664, pp. 230–244, Springer-Verlag.

[27] Kelly, G.M. (1974), Doctorinal adjunction, *in* "Proceedings, Sydney Category Theory Seminar", Lecture Notes in Mathematics, Vol. 420, pp 257–280, Springer-Verlag.

[28] Kelly, G.M. and Laplaza M.L. (1980), Coherence for compact closed categories, *J. Pure Appl. Algebra* **19**, 193–213.

[29] Lafont, Y. (1988), "Logiques, Catégories et Machines", Thèse de Doctorat, Université Paris VII.

[30] Lambek, J. (1995), Bilinear logic in algebra and linguistics, *in* "Advances in Linear Logic", pp. 43–59, Cambridge University Press.

[31] Lawvere, F.W. (1992), Categories of space and of quantity, *in* "The Space of Mathematics: Philosophical, Epistemological and Historical Explorations", pp. 14–30, DeGruyter, Berlin.

[32] Loader, R. (1994), "Models of Lambda Calculi and Linear Logic: Structural, Equational and Proof-Theoretic Characterisations", Ph.D. thesis, St. Hugh's College, Oxford.

[33] Mac Lane, S. (1998), "Categories for the Working Mathematician (Second Edition)", Graduate Texts in Mathematics 5, Springer-Verlag.

[34] Milner, R. (1996), Calculi for interaction, *Acta Inform.* **33**(8), 707–737.

[35] Mitchell, J.C., and Moggi, E. (1991), Kripke-style models for typed lambda calculus, *Ann. Pure Appl. Logic* **51**, 99–124.

[36] Mitchell, J.C., and Scedrov, A. (1992), Notes on sconing and relators, *in* "Computer Science Logic (CSL'92), Selected Papers", Lecture Notes in Computer Science, Vol. 702, pp. 352–378, Springer-Verlag.

[37] O'Hearn, P.W., and Riecke, J.G. (1995), Kripke logical relations and PCF, *Inform. and Comput.* **120**(1), 107–116.

[38] Plotkin, G.D. (1980), Lambda-definability in the full type hierarchy, *in* "To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism", Academic Press.

[39] Power, A.J. (1996), Elementary control structures, *in* "Proceedings, Concurrency Theory (CONCUR'96)", Lecture Notes in Computer Science, Vol. 1119, pp. 115–130, Springer-Verlag.

[40] Shirasu, H. (1995), Glueing of algebras for substructural logics, *in* "Proceedings, Non-Classical Logics and Their Kripke Semantics", RIMS Kokyuroku 927, pp. 127–139, Kyoto University.

[41] Streicher, T. (1998), Denotational completeness revisited, draft.

[42] Tan, A.M. (1997), "Full Completeness for Models of Linear Logic", Ph.D. thesis, University of Cambridge.

[43] Taylor, P. (1998), "Practical Foundations of Mathematics", to be published from Cambridge University Press.

A Syntax and Semantics of MILL

A.1 Syntax of MILL

We briefly recall the syntax and equational theory of MILL. The detail is discussed e.g. in [10]; our presentation is chosen so that it will be compatible with DILL (Appendix B). A set of base types (write b for one) and also a set of constants are fixed throughout this paper.

Types and Terms

$$\begin{array}{rcl} \sigma & ::= & b \mid I \mid \sigma\otimes\sigma \mid \sigma\multimap\sigma \\ M & ::= & c(M) \mid x \mid * \mid \mathsf{let}\ *\ \mathsf{be}\ M\ \mathsf{in}\ M \mid M\otimes M \mid \mathsf{let}\ x\otimes x\ \mathsf{be}\ M\ \mathsf{in}\ M \mid \lambda x.M \mid MM \end{array}$$

We assume that each constant c has a fixed arity $\sigma\to\tau$, where σ and τ are types which do not involve $\multimap$. (This restriction on arity is for ease of presentation and not essential.)

Typing

$$\frac{c:\sigma\to\tau \quad \Delta\vdash M:\sigma}{\Delta\vdash c(M):\tau}\ (\textbf{Constant}) \qquad \frac{}{x:\sigma\vdash x:\sigma}\ (\textbf{Variable})$$

$$\frac{}{\vdash *:I}\ (I\textbf{-Intro}) \qquad \frac{\Delta_1\vdash M:I \quad \Delta_2\vdash N:\sigma}{\Delta_1\sharp\Delta_2\vdash \mathsf{let}\ *\ \mathsf{be}\ M\ \mathsf{in}\ N:\sigma}\ (I\textbf{-Elim})$$

$$\frac{\Delta_1\vdash M:\sigma \quad \Delta_2\vdash N:\tau}{\Delta_1\sharp\Delta_2\vdash M\otimes N:\sigma\otimes\tau}\ (\otimes\textbf{-Intro}) \qquad \frac{\begin{array}{c}\Delta_1\vdash M:\sigma\otimes\tau\\ \Delta_2,x:\sigma,y:\tau\vdash N:\theta\end{array}}{\Delta_1\sharp\Delta_2\vdash \mathsf{let}\ x\otimes y\ \mathsf{be}\ M\ \mathsf{in}\ N:\theta}\ (\otimes\textbf{-Elim})$$

$$\frac{\Delta,x:\sigma\vdash M:\tau}{\Delta\vdash\lambda x.M:\sigma\multimap\tau}\ (\multimap\textbf{-Intro}) \qquad \frac{\Delta_1\vdash M:\sigma\multimap\tau \quad \Delta_2\vdash N:\sigma}{\Delta_1\sharp\Delta_2\vdash MN:\tau}\ (\multimap\textbf{-Elim})$$

where $\Delta_1\sharp\Delta_2$ is a merge of Δ_1 and Δ_2 (this notation is taken from [6]). We note that any typing judgement has a unique derivation.

Axioms

$$\begin{array}{rcl@{\qquad}rcl} \mathsf{let}\ *\ \mathsf{be}\ *\ \mathsf{in}\ M & = & M & \mathsf{let}\ *\ \mathsf{be}\ M\ \mathsf{in}\ * & = & M \\ \mathsf{let}\ x\otimes y\ \mathsf{be}\ M\otimes N\ \mathsf{in}\ L & = & L[M/x,N/y] & \mathsf{let}\ x\otimes y\ \mathsf{be}\ M\ \mathsf{in}\ x\otimes y & = & M \\ (\lambda x.M)N & = & M[N/x] & \lambda x.Mx & = & M \end{array}$$

$$\begin{array}{rcl} C[\mathsf{let}\ *\ \mathsf{be}\ M\ \mathsf{in}\ N] & = & \mathsf{let}\ *\ \mathsf{be}\ M\ \mathsf{in}\ C[N] \\ C[\mathsf{let}\ x\otimes y\ \mathsf{be}\ M\ \mathsf{in}\ N] & = & \mathsf{let}\ x\otimes y\ \mathsf{be}\ M\ \mathsf{in}\ C[N] \end{array}$$

In the above $C[-]$ indicates a (well-typed) context – we assume suitable conditions on variables for avoiding undesirable captures. The equational theory of MILL is defined as the congruence relation on the terms with typing judgement generated from these axioms.

A.2 Semantics of MILL

Let $\mathbb{C}$ be a symmetric monoidal closed category with tensor product $\otimes$, unit object I and exponent $\multimap$. Assume that there is an object $[\![b]\!]$ for each base type b and an arrow $[\![c]\!]:[\![\sigma]\!]\to[\![\tau]\!]$ for each constant $c : \sigma \to \tau$, where $\llbracket \sigma \rrbracket$ is defined by $\llbracket I \rrbracket = I$, $\llbracket \sigma \otimes \tau \rrbracket = \llbracket \sigma \rrbracket \otimes \llbracket \tau \rrbracket$ and $\llbracket \sigma \multimap \tau \rrbracket = \llbracket \sigma \rrbracket \multimap \llbracket \tau \rrbracket$. For each typing judgement $\Delta \vdash M : \tau$, we define its interpretation $\llbracket \Delta \vdash M : \tau \rrbracket : \llbracket |\Delta| \rrbracket \to \llbracket \tau \rrbracket$ in $\mathbb{C}$ as follows, where $|\Delta| = (\ldots(\llbracket \sigma_1 \rrbracket \otimes \llbracket \sigma_2 \rrbracket)\ldots) \otimes \llbracket \sigma_n \rrbracket$ for $\Delta \equiv x_1 : \sigma_1, x_2 : \sigma_2, \ldots, x_n : \sigma_n$.

$$
\begin{array}{rcl}
\llbracket \Delta \vdash c(M) : \tau \rrbracket & = & \llbracket |\Delta| \rrbracket \xrightarrow{\llbracket \Delta \vdash M:\sigma \rrbracket} \llbracket \sigma \rrbracket \xrightarrow{\llbracket c \rrbracket} \llbracket \tau \rrbracket \\
\llbracket x : \sigma \vdash x : \sigma \rrbracket & = & \llbracket \sigma \rrbracket \xrightarrow{id_{\llbracket \sigma \rrbracket}} \llbracket \sigma \rrbracket \\
\llbracket \vdash * : I \rrbracket & = & I \xrightarrow{id_I} I \\
\llbracket \Delta_1 \sharp \Delta_2 \vdash \mathsf{let}\ *\ \mathsf{be}\ M\ \mathsf{in}\ N : \sigma \rrbracket & = & \llbracket |\Delta_1 \sharp \Delta_2| \rrbracket \overset{\simeq}{\to} \llbracket |\Delta_1| \rrbracket \otimes \llbracket |\Delta_2| \rrbracket \xrightarrow{\llbracket \Delta_1 \vdash M:I \rrbracket \otimes \llbracket \Delta_2 \vdash N:\sigma \rrbracket} I \otimes \llbracket \sigma \rrbracket \overset{\simeq}{\to} \llbracket \sigma \rrbracket \\
\llbracket \Delta_1 \sharp \Delta_2 \vdash M \otimes N : \sigma \otimes \tau \rrbracket & = & \llbracket |\Delta_1 \sharp \Delta_2| \rrbracket \overset{\simeq}{\to} \llbracket |\Delta_1| \rrbracket \otimes \llbracket |\Delta_2| \rrbracket \xrightarrow{\llbracket \Delta_1 \vdash M:\sigma \rrbracket \otimes \llbracket \Delta_2 \vdash N:\tau \rrbracket} \llbracket \sigma \rrbracket \otimes \llbracket \tau \rrbracket
\end{array}
$$

$$
\begin{array}{l}
\llbracket \Delta_1 \sharp \Delta_2 \vdash \mathsf{let}\ x \otimes y\ \mathsf{be}\ M\ \mathsf{in}\ N : \theta \rrbracket \ = \\
\qquad \llbracket |\Delta_1 \sharp \Delta_2| \rrbracket \overset{\simeq}{\to} \llbracket |\Delta_1| \rrbracket \otimes \llbracket |\Delta_2| \rrbracket \xrightarrow{\llbracket \Delta_1 \vdash M:\sigma \otimes \tau \rrbracket \otimes id_{\llbracket |\Delta_2| \rrbracket}} \\
\qquad\qquad (\llbracket \sigma \rrbracket \otimes \llbracket \tau \rrbracket) \otimes \llbracket |\Delta_2| \rrbracket \overset{\simeq}{\to} (\llbracket |\Delta_2| \rrbracket \otimes \llbracket \sigma \rrbracket) \otimes \llbracket \tau \rrbracket \xrightarrow{\llbracket \Delta_2, x:\sigma, y:\tau \vdash N:\theta \rrbracket} \llbracket \theta \rrbracket
\end{array}
$$

$$
\llbracket \Delta \vdash \lambda x.M : \sigma \multimap \tau \rrbracket \ = \ \llbracket |\Delta| \rrbracket \xrightarrow{\Lambda(\llbracket \Delta, x:\sigma \vdash M:\tau \rrbracket)} \llbracket \sigma \rrbracket \multimap \llbracket \tau \rrbracket
$$

$$
\llbracket \Delta_1 \sharp \Delta_2 \vdash MN : \tau \rrbracket \ = \ \llbracket |\Delta_1 \sharp \Delta_2| \rrbracket \overset{\simeq}{\to} \llbracket |\Delta_1| \rrbracket \otimes \llbracket |\Delta_2| \rrbracket \xrightarrow{\llbracket \Delta_1 \vdash M:\sigma \multimap \tau \rrbracket \otimes \llbracket \Delta_2 \vdash N:\sigma \rrbracket} (\llbracket \sigma \rrbracket \multimap \llbracket \tau \rrbracket) \otimes \llbracket \sigma \rrbracket \overset{\mathsf{ev}}{\to} \llbracket \tau \rrbracket
$$

where "$\simeq$" denotes a (uniquely determined) canonical isomorphism.

Proposition A.1 *This semantics is sound and complete.* □

B Syntax and Semantics of DILL

In DILL a typing judgement takes the form $\Gamma\,;\,\Delta \vdash M : \sigma$ in which Γ represents an intuitionistic (or additive) context whereas Δ is a linear (multiplicative) context.

B.1 Syntax of DILL

Types and Terms

$$
\begin{array}{rcl}
\sigma & ::= & b \mid I \mid \sigma \otimes \sigma \mid \sigma \multimap \sigma \mid !\sigma \\
M & ::= & c(M) \mid x \mid * \mid \mathsf{let}\ *\ \mathsf{be}\ M\ \mathsf{in}\ M \mid M \otimes M \mid \mathsf{let}\ x \otimes x\ \mathsf{be}\ M\ \mathsf{in}\ M \mid \\
 & & \lambda x.M \mid MM \mid !M \mid \mathsf{let}\ !x\ \mathsf{be}\ M\ \mathsf{in}\ M
\end{array}
$$

Typing

$$
\frac{c : \sigma \to \tau \quad \Gamma\,;\,\Delta \vdash M : \sigma}{\Gamma\,;\,\Delta \vdash c(M) : \tau}\ (\textbf{Constant})
\qquad
\frac{}{\Gamma\,;\,x : \sigma \vdash x : \sigma}\ (\textbf{Variable}_{\textbf{lin}})
$$

$$
\frac{}{\Gamma\,;\,\emptyset \vdash * : I}\ (I\textbf{-Intro})
\qquad
\frac{\Gamma\,;\,\Delta_1 \vdash M : I \quad \Gamma\,;\,\Delta_2 \vdash N : \sigma}{\Gamma\,;\,\Delta_1 \sharp \Delta_2 \vdash \mathsf{let}\ *\ \mathsf{be}\ M\ \mathsf{in}\ N : \sigma}\ (I\textbf{-Elim})
$$

$$
\frac{\Gamma\,;\,\Delta_1 \vdash M : \sigma \quad \Gamma\,;\,\Delta_2 \vdash N : \tau}{\Gamma\,;\,\Delta_1 \sharp \Delta_2 \vdash M \otimes N : \sigma \otimes \tau}\ (\otimes\textbf{-Intro})
\qquad
\frac{\begin{array}{c}\Gamma\,;\,\Delta_1 \vdash M : \sigma \otimes \tau \\ \Gamma\,;\,\Delta_2, x : \sigma, y : \tau \vdash N : \theta\end{array}}{\Gamma\,;\,\Delta_1 \sharp \Delta_2 \vdash \mathsf{let}\ x \otimes y\ \mathsf{be}\ M\ \mathsf{in}\ N : \theta}\ (\otimes\textbf{-Elim})
$$

$$\frac{\Gamma\,;\,\Delta, x:\sigma \vdash M:\tau}{\Gamma\,;\,\Delta \vdash \lambda x.M : \sigma \multimap \tau}\ (\multimap\textbf{-Intro}) \qquad \frac{\Gamma\,;\,\Delta_1 \vdash M:\sigma \multimap \tau \quad \Gamma\,;\,\Delta_2 \vdash N:\sigma}{\Gamma\,;\,\Delta_1 \sharp \Delta_2 \vdash MN:\tau}\ (\multimap\textbf{-Elim})$$

$$\frac{}{\Gamma_1, x:\sigma, \Gamma_2\,;\,\emptyset \vdash x:\sigma}\ (\textbf{Variable}_{\textbf{int}})$$

$$\frac{\Gamma\,;\,\emptyset \vdash M:\sigma}{\Gamma\,;\,\emptyset \vdash !M:!\sigma}\ (\textbf{!-Intro}) \qquad \frac{\Gamma\,;\,\Delta_1 \vdash M:!\sigma \quad \Gamma, x:\sigma\,;\,\Delta_2 \vdash N:\tau}{\Gamma\,;\,\Delta_1 \sharp \Delta_2 \vdash \mathsf{let}\ !x\ \mathsf{be}\ M\ \mathsf{in}\ N:\tau}\ (\textbf{!-Elim})$$

Axioms

$$\begin{array}{rcl@{\qquad}rcl}
\mathsf{let}\ *\ \mathsf{be}\ *\ \mathsf{in}\ M & = & M & \mathsf{let}\ *\ \mathsf{be}\ M\ \mathsf{in}\ * & = & M \\
\mathsf{let}\ x \otimes y\ \mathsf{be}\ M \otimes N\ \mathsf{in}\ L & = & L[M/x, N/y] & \mathsf{let}\ x \otimes y\ \mathsf{be}\ M\ \mathsf{in}\ x \otimes y & = & M \\
(\lambda x.M)N & = & M[N/x] & \lambda x.Mx & = & M \\
\mathsf{let}\ !x\ \mathsf{be}\ !M\ \mathsf{in}\ N & = & N[M/x] & \mathsf{let}\ !x\ \mathsf{be}\ M\ \mathsf{in}\ !x & = & M
\end{array}$$

$$\begin{array}{rcl}
C[\mathsf{let}\ *\ \mathsf{be}\ M\ \mathsf{in}\ N] & = & \mathsf{let}\ *\ \mathsf{be}\ M\ \mathsf{in}\ C[N] \\
C[\mathsf{let}\ x \otimes y\ \mathsf{be}\ M\ \mathsf{in}\ N] & = & \mathsf{let}\ x \otimes y\ \mathsf{be}\ M\ \mathsf{in}\ C[N] \\
C[\mathsf{let}\ !x\ \mathsf{be}\ M\ \mathsf{in}\ N] & = & \mathsf{let}\ !x\ \mathsf{be}\ M\ \mathsf{in}\ C[N]
\end{array}$$

where $C[-]$ is a linear context (no ! binds $[-]$).

B.2 Semantics of DILL

Let $\mathbb{C}$ be a cartesian category (category with finite products), $\mathbb{D}$ a symmetric monoidal closed category and $\mathbb{C} \underset{U}{\overset{F}{\underset{\perp}{\rightleftarrows}}} \mathbb{D}$ a symmetric monoidal adjunction; we understand that the symmetric monoidal structure on $\mathbb{C}$ is given by (a choice of) the terminal object and binary product. Assume that there is an object $\llbracket b \rrbracket \in \mathbb{D}$ for each base type b and an arrow $\llbracket c \rrbracket \in \mathbb{D}(\llbracket \sigma \rrbracket, \llbracket \tau \rrbracket)$ for each constant $c : \sigma \to \tau$, where $\llbracket \sigma \rrbracket \in \mathbb{D}$ is inductively defined by $\llbracket I \rrbracket = I$, $\llbracket \sigma \otimes \tau \rrbracket = \llbracket \sigma \rrbracket \otimes \llbracket \tau \rrbracket$, $\llbracket \sigma \multimap \tau \rrbracket = \llbracket \sigma \rrbracket \multimap \llbracket \tau \rrbracket$ and $\llbracket !\sigma \rrbracket = FU\llbracket \sigma \rrbracket$. For each typing judgement $\Gamma\,;\,\Delta \vdash M:\sigma$, we define $\llbracket \Gamma\,;\,\Delta \vdash M:\sigma \rrbracket : \llbracket |\Gamma\,;\,\Delta| \rrbracket \to \llbracket \tau \rrbracket$ in $\mathbb{D}$ as follows, where $|\Gamma\,;\,\Delta| = |!\Gamma, \Delta|$ in which $!\Gamma = x_1 :!\sigma_1, \ldots, x_n :!\sigma_n$ for $\Gamma \equiv x_1 : \sigma_1, \ldots, x_n : \sigma_n$. First eight cases are dealt with as in MILL, with care for discarding or duplicating the intuitionistic context, using

$$\begin{array}{rcl}
\mathsf{discard}_{\Gamma,\Delta} & : & \llbracket |\Gamma\,;\,\Delta| \rrbracket \to \llbracket |\Delta| \rrbracket \\
\mathsf{split}_{\Gamma,\Delta_1,\Delta_2} & : & \llbracket |\Gamma\,;\,\Delta_1 \sharp \Delta_2| \rrbracket \to \llbracket |\Gamma\,;\,\Delta_1| \rrbracket \otimes \llbracket |\Gamma\,;\,\Delta_2| \rrbracket
\end{array}$$

which are defined in terms of projections and diagonal maps in $\mathbb{C}$ and imported into $\mathbb{D}$ via F. For last three cases we have

$$\llbracket \Gamma_1, x:\sigma, \Gamma_2\,;\,\emptyset \vdash x:\sigma \rrbracket \;=\; \llbracket |\Gamma_1, x:\sigma, \Gamma_2| \rrbracket \xrightarrow{\simeq} F(\ldots \times U\llbracket \sigma \rrbracket \times \ldots) \xrightarrow{F\mathsf{proj}} FU\llbracket \sigma \rrbracket \xrightarrow{\varepsilon} \llbracket \sigma \rrbracket$$

$$\begin{aligned}
\llbracket \Gamma\,;\,\emptyset \vdash !M:!\sigma \rrbracket = \llbracket |\Gamma\,;\,\emptyset| \rrbracket \xrightarrow{\simeq} \textstyle\bigotimes_i FU\llbracket \sigma_i \rrbracket \xrightarrow{\bigotimes_i \delta} \textstyle\bigotimes_i FUFU\llbracket \sigma_i \rrbracket \xrightarrow{m} \\
FU(\textstyle\bigotimes_i FU\llbracket \sigma_i \rrbracket) \xrightarrow{\simeq} FU\llbracket |\Gamma\,;\,\emptyset| \rrbracket \xrightarrow{FU\llbracket \Gamma\,;\,\emptyset \vdash M:\sigma \rrbracket} FU\llbracket \sigma \rrbracket
\end{aligned}$$

$$\begin{aligned}
&\llbracket \Gamma\,;\,\Delta_1 \sharp \Delta_2 \vdash \mathsf{let}\ !x\ \mathsf{be}\ M\ \mathsf{in}\ N:\tau \rrbracket = \\
&\qquad \llbracket |\Gamma\,;\,\Delta_1 \sharp \Delta_2| \rrbracket \xrightarrow{\mathsf{split}} \llbracket |\Gamma\,;\,\Delta_1| \rrbracket \otimes \llbracket |\Gamma\,;\,\Delta_2| \rrbracket \xrightarrow{\llbracket \Gamma\,;\,\Delta_1 \vdash M:!\sigma \rrbracket \otimes id} \\
&\qquad\qquad \llbracket !\sigma \rrbracket \otimes \llbracket |\Gamma\,;\,\Delta_2| \rrbracket \xrightarrow{\simeq} \llbracket |\Gamma, x:\sigma\,;\,\Delta_2| \rrbracket \xrightarrow{\llbracket \Gamma, x:\sigma\,;\,\Delta_2 \vdash N:\tau \rrbracket} \llbracket \tau \rrbracket
\end{aligned}$$

where proj is a suitable projection in $\mathbb{C}$, ε and δ are the counit and comultiplication of the comonad FU while m is an induced coherent morphism.

Proposition B.1 *(Barber and Plotkin [6]) This semantics is sound and complete.* □

C Sharing Theories (Action Calculi)

We reproduce the *(acyclic) sharing theories* (ST) which was introduced in [22] as type theories for sharing graphs (term graphs) of graph rewriting systems (also see [14] for a related structure). They can also be regarded as type theories for *action calculi* (AC) [34]; see [18] for an exposition in this direction. It has been shown that DILL (with suitable constants) is a conservative extension of an action calculus [5]. We can show that the translation is in fact fully complete, using a logical predicate.

Remark C.1 In the previous work [5], Benton's LNL Logic [9] is used instead of DILL, and also the type theory for action calculi is not the one presented here. However these cosmetic changes do not affect our results, as they share the same class of categorical models as those used in this paper. □

C.1 Syntax

Types A type is a finite list of base types; let σ, τ, ... range over types. We write ε for the empty sequence and $\sigma \otimes \tau$ for the concatenation of σ and τ.

Terms $M \ ::= \ c(M) \mid x \mid \mathsf{let}\ (\vec{x})\ \mathsf{be}\ M\ \mathsf{in}\ M \mid 0 \mid M \otimes M$

Typing

$$\frac{c : \sigma \to \tau \quad \Gamma \vdash M : \sigma}{\Gamma \vdash c(M) : \tau}\ (\textbf{Constant}) \qquad \frac{}{\Gamma, x : b, \Gamma' \vdash x : (b)}\ (\textbf{Variable})$$

$$\frac{\Gamma \vdash M : (\vec{b}) \quad \Gamma, \vec{x} : \vec{b} \vdash N : \tau}{\Gamma \vdash \mathsf{let}\ (\vec{x})\ \mathsf{be}\ M\ \mathsf{in}\ N : \tau}\ (\textbf{Let}) \qquad \frac{}{\Gamma \vdash 0 : \varepsilon}\ (\textbf{Unit}) \qquad \frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash N : \tau}{\Gamma \vdash M \otimes N : \sigma \otimes \tau}\ (\textbf{Tensor})$$

Axioms

$$\begin{array}{rcl}
0 \otimes M \ = \ M & = & M \otimes 0 \\
(L \otimes M) \otimes N & = & L \otimes (M \otimes N) \\
\mathsf{let}\ (x)\ \mathsf{be}\ y\ \mathsf{in}\ M & = & M[y/x] \\
\mathsf{let}\ (x_1, \ldots, x_n)\ \mathsf{be}\ M\ \mathsf{in}\ x_1 \otimes \ldots \otimes x_n & = & M \\
\mathsf{let}\ (\vec{x}, \vec{y})\ \mathsf{be}\ L \otimes M\ \mathsf{in}\ N & = & \mathsf{let}\ (\vec{x})\ \mathsf{be}\ L\ \mathsf{in}\ \mathsf{let}\ (\vec{y})\ \mathsf{be}\ M\ \mathsf{in}\ N \\
C[\mathsf{let}\ (x)\ \mathsf{be}\ M\ \mathsf{in}\ N] & = & \mathsf{let}\ (x)\ \mathsf{be}\ M\ \mathsf{in}\ C[N]
\end{array}$$

Intuitively, $\mathsf{let}\ (\vec{x})\ \mathsf{be}\ M\ \mathsf{in}\ N$ represents a notion of sharing; $\vec{x}$ are the pointers to a shared resource M which can be referred many times in N. The equation $\mathsf{let}\ (x)\ \mathsf{be}\ M\ \mathsf{in}\ N = N[M/x]$ is not always true, as M can be duplicated or discarded in $N[M/x]$.

C.2 Semantics

The following semantics is that presented in [22], though we slightly relax the requirements on the semantic categorical structure. Let $F : \mathbb{C} \to \mathbb{D}$ be a strong symmetric monoidal functor from a cartesian category $\mathbb{C}$ to a symmetric monoidal category $\mathbb{D}$. Assume that there is an object $\llbracket b \rrbracket_{\mathbb{C}} \in \mathbb{C}$ for each base type b. Define $\llbracket \sigma \rrbracket_{\mathbb{C}} \in \mathbb{C}$ and $\llbracket \sigma \rrbracket \in \mathbb{D}$ by $\llbracket \varepsilon \rrbracket_{\mathbb{C}} = 1$, $\llbracket (\vec{b}, b) \rrbracket_{\mathbb{C}} = \llbracket (\vec{b}) \rrbracket_{\mathbb{C}} \times \llbracket b \rrbracket_{\mathbb{C}}$, and $\llbracket \varepsilon \rrbracket = I$ and $\llbracket (\vec{b}, b) \rrbracket = \llbracket (\vec{b}) \rrbracket \otimes F(\llbracket b \rrbracket_{\mathbb{C}})$. Since F is strong, it follows that $F(\llbracket \sigma \rrbracket_{\mathbb{C}}) \simeq \llbracket \sigma \rrbracket$. Moreover suppose that there is an arrow $\llbracket c \rrbracket : \llbracket \sigma \rrbracket \to \llbracket \tau \rrbracket$ in $\mathbb{D}$ for each constant $c : \sigma \to \tau$. We use the following notations:

$$\begin{aligned}
\mathsf{discard} &= \llbracket \sigma \rrbracket \xrightarrow{\simeq} F(\llbracket \sigma \rrbracket_{\mathbb{C}}) \xrightarrow{F!} F1 \xrightarrow{\simeq} I \\
\mathsf{diag} &= \llbracket \sigma \rrbracket \xrightarrow{\simeq} F(\llbracket \sigma \rrbracket_{\mathbb{C}}) \xrightarrow{F\Delta} F(\llbracket \sigma \rrbracket_{\mathbb{C}} \times \llbracket \sigma \rrbracket_{\mathbb{C}}) \xrightarrow{\simeq} F(\llbracket \sigma \rrbracket_{\mathbb{C}}) \otimes F(\llbracket \sigma \rrbracket_{\mathbb{C}}) \xrightarrow{\simeq} \llbracket \sigma \rrbracket \otimes \llbracket \sigma \rrbracket
\end{aligned}$$

We define $\llbracket \Gamma \vdash M : \sigma \rrbracket : \llbracket |\Gamma| \rrbracket \to \llbracket \sigma \rrbracket$ in $\mathbb{D}$ as follows.

$$\begin{aligned}
\llbracket \Gamma \vdash c(M) : \tau \rrbracket &= \llbracket |\Gamma| \rrbracket \xrightarrow{\llbracket \Gamma \vdash M : \sigma \rrbracket} \llbracket \sigma \rrbracket \xrightarrow{\llbracket c \rrbracket} \llbracket \tau \rrbracket \\
\llbracket \Gamma, x : \sigma, \Gamma' \vdash x : \sigma \rrbracket &= \llbracket |\Gamma, x : \sigma, \Gamma'| \rrbracket \xrightarrow{\simeq} F(\llbracket |\Gamma| \rrbracket_{\mathbb{C}} \times \llbracket \sigma \rrbracket_{\mathbb{C}} \times \llbracket |\Gamma'| \rrbracket_{\mathbb{C}}) \xrightarrow{F\mathsf{proj}} F(\llbracket \sigma \rrbracket_{\mathbb{C}}) \xrightarrow{\simeq} \llbracket \sigma \rrbracket
\end{aligned}$$

$$\begin{aligned}
&\llbracket \Gamma \vdash \mathsf{let}\ (\vec{x})\ \mathsf{be}\ M\ \mathsf{in}\ N : \tau \rrbracket = \\
&\quad \llbracket |\Gamma| \rrbracket \xrightarrow{\mathsf{diag}} \llbracket |\Gamma| \rrbracket \otimes \llbracket |\Gamma| \rrbracket \xrightarrow{\llbracket |\Gamma| \rrbracket \otimes \llbracket \Gamma \vdash M : (\vec{b}) \rrbracket} \llbracket |\Gamma| \rrbracket \otimes \llbracket (\vec{b}) \rrbracket \xrightarrow{\simeq} \llbracket |\Gamma, \vec{x} : (\vec{b})| \rrbracket \xrightarrow{\llbracket \Gamma, \vec{x} : \vec{b} \vdash N : \tau \rrbracket} \llbracket \tau \rrbracket
\end{aligned}$$

$$\begin{aligned}
\llbracket \Gamma \vdash 0 : \varepsilon \rrbracket &= \llbracket |\Gamma| \rrbracket \xrightarrow{\mathsf{discard}} I \\
\llbracket \Gamma \vdash M \otimes N : \sigma \otimes \tau \rrbracket &= \llbracket |\Gamma| \rrbracket \xrightarrow{\mathsf{diag}} \llbracket |\Gamma| \rrbracket \otimes \llbracket |\Gamma| \rrbracket \xrightarrow{\llbracket \Gamma \vdash M : \sigma \rrbracket \otimes \llbracket \Gamma \vdash N : \tau \rrbracket} \llbracket \sigma \rrbracket \otimes \llbracket \tau \rrbracket \xrightarrow{\simeq} \llbracket \sigma \otimes \tau \rrbracket
\end{aligned}$$

Proposition C.2 *(c.f. [39, 22, 18]) This semantics is sound and complete.* □

C.3 Translation into DILL

The translation of types and terms of ST into DILL is defined as follows.

$$\varepsilon^\circ = I \quad (b)^\circ = !b \quad (\vec{b}, b)^\circ = (\vec{b})^\circ \otimes !b \quad \text{(where } \vec{b} \text{ is not empty)}$$

$$\begin{aligned}
(c(M))^\circ &= c^\circ(M^\circ) \quad (c^\circ : \sigma^\circ \to \tau^\circ \text{ for } c : \sigma \to \tau) \\
x^\circ &= !x \\
(\mathsf{let}\ ()\ \mathsf{be}\ M\ \mathsf{in}\ N)^\circ &= \mathsf{let}\ *\ \mathsf{be}\ M^\circ\ \mathsf{in}\ N^\circ \\
(\mathsf{let}\ (x)\ \mathsf{be}\ M\ \mathsf{in}\ N)^\circ &= \mathsf{let}\ !x\ \mathsf{be}\ M^\circ\ \mathsf{in}\ N^\circ \\
(\mathsf{let}\ (\vec{x}, x)\ \mathsf{be}\ M\ \mathsf{in}\ N)^\circ &= \mathsf{let}\ y \otimes z\ \mathsf{be}\ M^\circ\ \mathsf{in}\ \mathsf{let}\ !x\ \mathsf{be}\ z\ \mathsf{in}\ (\mathsf{let}\ (\vec{x})\ \mathsf{be}\ \underline{y}\ \mathsf{in}\ N)^\circ \\
& \ \text{(where } \vec{x} \text{ is not empty, and } \underline{y} \text{ is a “dummy” constant s.t. } (\underline{y})^\circ = y) \\
0^\circ &= * \\
(M \otimes N)^\circ &= M^\circ \otimes' N^\circ \\
M \otimes' N &= \begin{cases} \mathsf{let}\ *\ \mathsf{be}\ N\ \mathsf{in}\ M & (\text{if } N : I) \\ \mathsf{let}\ x \otimes y\ \mathsf{be}\ N\ \mathsf{in}\ (M \otimes' x) \otimes y & (\text{if } N : \sigma \otimes \tau) \\ M \otimes N & (\text{otherwise}) \end{cases}
\end{aligned}$$

The soundness results below tell us that the translation corresponds to a structure-preserving functor.

Lemma C.3 *(Type soundness) If $\Gamma \vdash M : \sigma$ in ST, then $\Gamma\ ;\ \emptyset \vdash M^\circ : \sigma^\circ$ in DILL.* □

Lemma C.4 *(Soundness) If* $\Gamma \vdash M = N : \sigma$ *in ST, then* $\Gamma\,;\,\emptyset \vdash M^\circ = N^\circ : \sigma^\circ$ *in DILL.* ☐

By constructing a model of DILL to which the term model of ST faithfully embeds (c.f. [5]), we can show that

Proposition C.5 *(Conservativity) If* $\Gamma\,;\,\emptyset \vdash M^\circ = N^\circ : \sigma^\circ$ *in DILL, then* $\Gamma \vdash M = N : \sigma$ *in ST.* ☐

C.4 Control Operators and Proof of Fullness

In an action calculus, we can additionally have a sort of parameterized constants called *control operators*. For ease of presentation, we shall consider the case with just one parameter; the generalization to the multiple arguments is straightforward. A control operator c with arity $(\sigma_0 \to \tau_0) \to (\sigma \to \tau)$ obeys the following term construction and typing

$$\frac{\Gamma, \vec{x} : \sigma_0 \vdash M : \tau_0 \quad \Gamma \vdash N : \sigma}{\Gamma \vdash c((\vec{x})M)(N) : \tau}\ (\textbf{Control})$$

and also the α-conversion of the bound variables $\vec{x}$. Intuitively, a control operator amounts to a rank 2 functional in a type theory with higher-order types. Power [39] has shown that, in the semantic setting described above, such a control operator is modelled by a family of functions

$$[\![c]\!]_X : \mathbb{D}(F\,X \otimes [\![\sigma_0]\!], [\![\tau_0]\!]) \to \mathbb{D}(F\,X \otimes [\![\sigma]\!], [\![\tau]\!])$$

which is natural in $X \in \mathbb{C}$. To make the Basic Lemma effective under the presence of such constructs, we need to show the following lemma for the setting of Section 4.4.

Lemma C.6 *Assume that there are natural families of functions*

$$\begin{array}{rcll} \alpha_X & : & \mathbb{D}_0(F_0X \otimes A, B) \to \mathbb{D}_0(F_0X \otimes C, D) & (\textit{natural in } X \in \mathbb{C}_0) \\ \beta_Y & : & \mathbb{D}_1(F_1Y \otimes \mathbb{J}A, \mathbb{J}B) \to \mathbb{D}_1(F_1Y \otimes \mathbb{J}C, \mathbb{J}D) & (\textit{natural in } Y \in \mathbb{C}_1) \end{array}$$

such that $\mathbb{J}(\alpha_X(h)) = \beta_{\mathbb{I}X}(\mathbb{J}h)$ *holds for* $h \in \mathbb{D}_0(F_0X \otimes A, B)$*. Then, for* $f \in \mathbb{D}_1(F_1Y \otimes \mathbb{J}A, \mathbb{J}B)$*,*

$$f \in \mathbb{D}_0\mathsf{PRED}(L(Y,P) \otimes (\mathbb{J}F_0A, \mathbb{P}_{F_0A}), (\mathbb{J}F_0B, \mathbb{P}_{F_0B}))$$

implies

$$\beta_Y(f) \in \mathbb{D}_0\mathsf{PRED}(L(Y,P) \otimes (\mathbb{J}F_0C, \mathbb{P}_{F_0C}), (\mathbb{J}F_0D, \mathbb{P}_{F_0B})).$$

Proof: We note that $f \in \mathbb{D}_0\mathsf{PRED}(L(Y,P) \otimes (\mathbb{J}F_0A, \mathbb{P}_{F_0A}), (\mathbb{J}F_0B, \mathbb{P}_{F_0B}))$ if and only if

> for any $g \in P(X) \subseteq \mathbb{C}_1(\mathbb{I}X, Y)$, $f \circ (F_1g \otimes \mathbb{J}A) = \mathbb{J}h \in \mathbb{D}_1(\mathbb{J}(F_0X \otimes A), \mathbb{J}B)$ for some $h \in \mathbb{D}_0(F_0X \otimes A, B)$.

Similarly, $\beta_Y(f) \in \mathbb{D}_0\mathsf{PRED}(L(Y,P) \otimes (\mathbb{J}F_0C, \mathbb{P}_{F_0C}), (\mathbb{J}F_0D), \mathbb{P}_{F_0B})$. if and only if

> for any $g \in P(X) \subseteq \mathbb{C}_1(\mathbb{I}X, Y)$, $\beta_Y(f) \circ (F_1g \otimes \mathbb{J}C) = \mathbb{J}h \in \mathbb{D}_1(\mathbb{J}(F_0X \otimes C), \mathbb{J}D)$ for some $h \in \mathbb{D}_0(F_0X \otimes C, D)$.

We show that the former condition implies the latter. For $g \in P(X) \subseteq \mathbb{C}_1(\mathbb{I}X, Y)$, we have $\beta_Y(f) \circ (F_1g \otimes \mathbb{J}C) = \beta_{\mathbb{I}X}(f \circ (F_1g \otimes \mathbb{J}A))$ by the naturality of β. By the condition on f, there exists $h \in \mathbb{D}_0(F_0X \otimes A, B)$ such that $f \circ (F_1g \otimes \mathbb{J}A) = \mathbb{J}h$ holds. Therefore

$$\begin{array}{rcll} \beta_Y(f) \circ (F_1g \otimes \mathbb{J}C) & = & \beta_{\mathbb{I}X}(f \circ (F_1g \otimes \mathbb{J}A)) & \text{naturality of } \beta \\ & = & \beta_{\mathbb{I}X}(\mathbb{J}h) & \text{condition on } f \\ & = & \mathbb{J}(\alpha_X(h)) & \text{assumption on } \alpha \text{ and } \beta \end{array}$$

□

On the other hand, as noted in [5], giving a natural family β above amounts to giving an arrow of $\mathbb{D}_1(F_1U_1(\mathbb{J}A \multimap \mathbb{J}B), \mathbb{J}C \multimap \mathbb{J}D)$. Therefore, we assume that, for each control operator c with arity $(\sigma_0, \tau_0) \to (\sigma \to \tau)$, the corresponding DILL has a constant c° of type $!(\sigma_0^\circ \multimap \tau_0^\circ) \multimap (\sigma^\circ \multimap \tau^\circ)$. Under this assumption we have the translation of control operators $(c((\vec{x})M)(N))^\circ = c^\circ(!(\lambda y.(\mathsf{let}\ (\vec{x})\ \mathsf{be}\ \underline{y}\ \mathsf{in}\ M)^\circ))N^\circ$.

Theorem C.7 *The translation from an action calculus to the corresponding DILL is full; that is, if $\Gamma\,;\,\emptyset \vdash M : \sigma^\circ$ in DILL and Γ consists of variables with base types, then there exists a term $\Gamma \vdash N : \sigma$ in AC such that $\Gamma\,;\,\emptyset \vdash M = N^\circ : \sigma^\circ$ holds in DILL.*

Proof: Let $\mathbb{C}_0 \overset{F_0}{\to} \mathbb{D}_0$ be the term model of the action calculus and $\mathbb{C}_1 \underset{U_1}{\overset{F_1}{\underset{\leftarrow}{\overset{\rightarrow}{\perp}}}} \mathbb{D}_1$ be that of DILL, while $\mathbb{I}$ and $\mathbb{J}$ are determined by the translation. Lemma C.6 ensures the Basic Lemma for the the canonical logical $(\mathbb{C}_0 \overset{F_0}{\to} \mathbb{D}_0)$-predicate under the presence of control operators. Thus any DILL-definable element satisfies the canonical logical predicate, hence is AC-definable if it has AC-definable types.

□

D Proof of Proposition 3.2

Let us use the following notations for the chosen cartesian and cocartesian liftings:

$$\frac{B \xrightarrow{\ u\ } pE \ \text{ in } \mathbb{B}}{u^*(E) \xrightarrow{\ \overline{u}(E)\ } E \ \text{ in } \mathbb{E}} \qquad \frac{pE \xrightarrow{\ u\ } B \ \text{ in } \mathbb{B}}{E \xrightarrow{\ \underline{u}(E)\ } u_!(E) \ \text{ in } \mathbb{E}}$$

For $u : B \to pE$ and $e : E' \to E$ such that $pe = u \circ w$ for $w : pE' \to B$, let us write $\varphi^u_e : E' \to u^*(E)$ for the unique map satisfying $p(\varphi^u_e) = w$ and $e = \overline{u}(E) \circ \varphi^u_e$.

$$\begin{tikzcd}
E' \arrow[d, dotted, "\varphi^u_e"'] \arrow[dr, "e"] & & & pE' \arrow[d, "w"'] \arrow[dr, "pe"] & \\
u^*(E) \arrow[r, "\overline{u}(E)"'] & E & \overset{p}{\mapsto} & B \arrow[r, "u"'] & pE
\end{tikzcd}$$

Dually, for $u : pE \to B$ and $e : E \to E'$ with $pe = w \circ u$ for $w : B \to pE'$, we write $\psi^u_e : u_!(E) \to E'$ for the unique map such that $p(\psi^u_e) = w$ and $e = \psi^u_e \circ \underline{u}(E)$ hold.

$$\begin{tikzcd}
E \arrow[r, "\underline{u}(E)"] \arrow[dr, "e"'] & u_!(E) \arrow[d, dotted, "\psi^u_e"] & \overset{p}{\mapsto} & pE \arrow[r, "u"] \arrow[dr, "pe"'] & B \arrow[d, "w"] \\
& E' & & & pE'
\end{tikzcd}$$

An object of $\mathbb{G}$ is a pair $(C \in \mathbb{C}, E \in \mathbb{E})$ such that $\Gamma C = pE$ holds, and an arrow from (C,E) to (D,F) is a pair $(c : C \to D, e : E \to F)$ satisfying $\Gamma c = pe$. We give a symmetric monoidal structure on $\mathbb{G}$ using the fact that p is a cofibration:

$$\begin{array}{rcl} I_{\mathbb{G}} & \equiv & (I_{\mathbb{C}}, m_{I!}(I_{\mathbb{E}})) \\ (C,E) \otimes (C',E') & \equiv & (C \otimes C', m_{C,C'!}(E \otimes E')) \end{array}$$

For $(c,e):(C,E)\to(D,F)$ and $(c',e'):(C',E')\to(D',F')$, we define

$$(c,e)\otimes(c',e')\equiv(c\otimes c',\psi^{m_{C,C'}}_{\underline{m_{D,D'}}(F\otimes F')\circ(e\otimes e')}):(C,E)\otimes(C',E')\to(D,F)\otimes(D',F')$$

where $\psi^{m_{C,C'}}_{\underline{m_{D,D'}}(F\otimes F')\circ(e\otimes e')}:m_{C,C'!}(E\otimes E')\to m_{D,D'!}(F\otimes F')$ is well-defined and satisfies $\Gamma(c\otimes c')=p(\psi^{m_{C,C'}}_{\underline{m_{D,D'}}(F\otimes F')\circ(e\otimes e')})$ because

$$\begin{array}{rcl}
p(\underline{m_{D,D'}}(F\otimes F')\circ(e\otimes e')) & = & m_{D,D'}\circ p(e\otimes e') \\
 & = & m_{D,D'}\circ(pe\otimes pe') \\
 & = & m_{D,D'}\circ(\Gamma c\otimes\Gamma c') \\
 & = & \Gamma(c\otimes c')\circ m_{C,C'}.
\end{array}$$

Exponents are given by appealing to the fact that p is a fibration:

$$(C,E)\multimap(C',E') \quad\equiv\quad (C\multimap C',\theta^{*}_{C,C'}(E\multimap E'))$$

where $\theta_{C,C'}:\Gamma(C\multimap C')\to\Gamma C\multimap\Gamma C'$ is given as in the proof of Lemma 3.1. We have a natural bijection

$$\cfrac{\cfrac{(c,e):(C,E)\otimes(C',E')\to(C'',E'')}{(\Lambda(c),\varphi^{\theta_{C',C''}}_{\Lambda(e\circ\underline{m_{C,C'}}(E\otimes E'))}):(C,E)\to(C',E')\multimap(C'',E'')}}{}$$

$$\cfrac{(d,f):(C,E)\to(C',E')\multimap(C'',E'')}{(\Lambda^{-1}(d),\psi^{m_{C,C'}}_{\Lambda^{-1}(\overline{\theta_{C',C''}}(E'\multimap E'')\circ f)}):(C,E)\otimes(C',E')\to(C'',E'')}$$

$\varphi^{\theta_{C',C''}}_{\Lambda(e\circ\underline{m_{C,C'}}(E\otimes E'))}:E\to\theta^{*}_{C',C''}(E'\multimap E'')$ is well-defined and satisfies $p(\varphi^{\theta_{C',C''}}_{\Lambda(e\circ\underline{m_{C,C'}}(E\otimes E'))})=\Gamma(\Lambda(c))$ because

$$\begin{array}{rcl}
p(\Lambda(e\circ\underline{m_{C,C'}}(E\otimes E'))) & = & \Lambda(p(e\circ\underline{m_{C,C'}}(E\otimes E'))) \\
 & = & \Lambda(pe\circ m_{C,C'}) \\
 & = & \Lambda(\Gamma c\circ m_{C,C'}) \\
 & = & \theta_{C',C''}\circ\Gamma(\Lambda(c))
\end{array}$$

$\psi^{m_{C,C'}}_{\Lambda^{-1}(\overline{\theta_{C',C''}}(E'\multimap E'')\circ f)}:m_{C,C'!}(E\otimes E')\to E''$ is well-defined and satisfies $p(\psi^{m_{C,C'}}_{\Lambda^{-1}(\overline{\theta_{C',C''}}(E'\multimap E'')\circ f)})=\Gamma(\Lambda^{-1}(d))$ because

$$\begin{array}{rcl}
p(\Lambda^{-1}(\overline{\theta_{C',C''}}(E'\multimap E'')\circ f)) & = & \Lambda^{-1}(p(\overline{\theta_{C',C''}}(E'\multimap E'')\circ f)) \\
 & = & \Lambda^{-1}(\theta_{C',C''}\circ pf) \\
 & = & \Lambda^{-1}(\theta_{C',C''}\circ\Gamma d) \\
 & = & \Gamma(\Lambda^{-1}(d))\circ m_{C,C'}
\end{array}$$

Bijectivity is verified as

$$\begin{array}{rcl}
\psi^{m_{C,C'}}_{\Lambda^{-1}(\overline{\theta_{C',C''}}(E'\multimap E'')\circ\varphi^{\cdots}_{\cdots})} & = & \psi^{m_{C,C'}}_{\Lambda^{-1}(\Lambda(e\circ\underline{m_{C,C'}}(E\otimes E')))} \\
 & = & \psi^{m_{C,C'}}_{e\circ\underline{m_{C,C'}}(E\otimes E')} \\
 & = & e
\end{array}$$

$$\begin{array}{rcl}
\varphi^{\theta_{C',C''}}_{\Lambda(\psi^{\cdots}_{\cdots}\circ\underline{m_{C,C'}}(E\otimes E'))} & = & \varphi^{\theta_{C',C''}}_{\Lambda(\Lambda^{-1}(\overline{\theta_{C',C''}}(E'\multimap E'')\circ f))} \\
 & = & \varphi^{\theta_{C',C''}}_{\overline{\theta_{C',C''}}(E'\multimap E'')\circ f} \\
 & = & f
\end{array}$$

(Note that $\psi^u_{e\circ\underline{u}(E)} = e$ and $\varphi^u_{\underline{u}(E)\circ e} = e$ hold.)
The bifibration $q : \mathbb{G} \to \mathbb{C}$ sends $(c,e) : (C,E) \to (D,F)$ to $c : C \to D$, and obviously preserves the symmetric monoidal closed structure strictly.

E Proof of Proposition 3.19

We are going to check the following bijective correspondence:

$$\frac{(D_s,D_t,C,f_s,f_t)\otimes(D'_s,D'_t,C',f'_s,f'_t) \xrightarrow{(d_s,d_t,c)} (D''_s,D''_t,C'',f''_s,f''_t)^\perp}{(D_s,D_t,C,f_s,f_t) \xrightarrow{(v_s,v_t,u)} ((D'_s,D'_t,C',f'_s,f'_t)\otimes(D''_s,D''_t,C'',f''_s,f''_t))^\perp}$$

The former means that the following diagrams commute:

$$\begin{tikzcd}
D_s\otimes D'_s \arrow[r, "f_s\otimes f'_s"] \arrow[d, dotted, "d_s"'] & \Gamma C\otimes\Gamma C' \arrow[r, "m"] & \Gamma(C\otimes C') \arrow[d, dotted, "\Gamma c"] \\
D''_t \arrow[rr, "f''_t"'] & & \Gamma C''^\perp
\end{tikzcd}
\qquad
\begin{tikzcd}
X \arrow[rr, "x"] & & \Gamma(C\otimes C')^\perp \\
D''_s \arrow[u, dotted, "d_t"] \arrow[r, "f''_s"'] & \Gamma C'' \arrow[r, "\Gamma\simeq"'] & \Gamma C''^{\perp\perp} \arrow[u, dotted, "\Gamma c^\perp"']
\end{tikzcd}$$

where X and $x : X \to \Gamma(C\otimes C')^\perp$ are given as in the proof sketch (p.16). On the other hand, the latter amounts to the following commutative diagrams:

$$\begin{tikzcd}
D_s \arrow[r, "f_s"] \arrow[d, dotted, "v_s"'] & \Gamma C \arrow[d, dotted, "\Gamma u"] \\
Y \arrow[r, "y"'] & \Gamma(C'\otimes C'')^\perp
\end{tikzcd}$$

$$\begin{tikzcd}
D_t \arrow[rrr, "f_t"] & & & \Gamma C^\perp \\
D'_s\otimes D''_s \arrow[u, dotted, "v_t"] \arrow[r, "f'_s\otimes f''_s"'] & \Gamma C'\otimes\Gamma C'' \arrow[r, "m"'] & \Gamma(C'\otimes C'') \arrow[r, "\Gamma\simeq"'] & \Gamma(C'\otimes C'')^{\perp\perp} \arrow[u, dotted, "\Gamma u^\perp"']
\end{tikzcd}$$

where Y and $y : Y \to \Gamma(C'\otimes C'')^\perp$ are given by the following pullbacks.

$$\begin{tikzcd}
Y \arrow[r] \arrow[d] \arrow[dr, dotted, "y"] & {} \arrow[r] \arrow[d] & D''_s \multimap D'_t \arrow[d, "D''_s\multimap f'_t"] \\
{} \arrow[r] \arrow[d] & \Gamma(C'\otimes C'')^\perp \arrow[r, "(f''_s\multimap\Gamma C'^\perp)\circ\theta\circ\Gamma\simeq"'] \arrow[d, "(f'_s\multimap\Gamma C''^\perp)\circ\theta\circ\Gamma\simeq"] & D''_s \multimap \Gamma C'^\perp \\
D'_s\multimap D''_t \arrow[r, "D'_s\multimap f''_t"'] & D'_s\multimap\Gamma C''^\perp &
\end{tikzcd}$$

Let us write $\pi_Y : Y \to D'_s \multimap D''_t$ and $\pi'_Y : Y \to D''_s \multimap D'_t$ for the left vertical arrow and the top horizontal arrow respectively. Similarly, we shall define $\pi_X : X \to D_s \multimap D'_t$ and $\pi'_X : X \to D'_s \multimap D_t$. From (d_s, d_t, c) we derive (v_s, v_t, u) as follows. u is derived from c by the bijection $\mathbb{C}(C \otimes C', C''^{\perp}) \simeq \mathbb{C}(C, (C' \otimes C'')^{\perp})$. v_s and v_t are derived as

$$\dfrac{\dfrac{D_s \otimes D'_s \xrightarrow{d_s} D''_t}{D_s \xrightarrow{\alpha} D'_s \multimap D''_t} \qquad \dfrac{\dfrac{D''_s \xrightarrow{d_t} X \xrightarrow{\pi_X} D_s \multimap D'_t}{D''_s \longrightarrow D_s \multimap D'_t}}{D_s \xrightarrow{\beta} D''_s \multimap D'_t}}{D_s \xrightarrow{v_s} Y}\,\star \qquad \dfrac{\dfrac{D''_s \xrightarrow{d_t} X \xrightarrow{\pi'_X} D'_s \multimap D_t}{D''_s \longrightarrow D'_s \multimap D_t}}{D'_s \otimes D''_s \xrightarrow{v_t} D_t}$$

where the step $\star$ follows from the commutative diagrams below and the definition of Y and y.

$$\begin{tikzcd}
D_s \arrow[rr, "\beta"] \arrow[dd, "\alpha"'] \arrow[dr, "\Gamma u \circ f_s"] & & D''_s \multimap D'_t \arrow[d, "D''_s \multimap f'_t"] \\
& \Gamma(C' \otimes C'')^{\perp} \arrow[r, "(f''_s \multimap \Gamma C'^{\perp}) \circ \theta \circ \Gamma\simeq"'] \arrow[d, "(f'_s \multimap \Gamma C''^{\perp}) \circ \theta \circ \Gamma\simeq"] & D''_s \multimap \Gamma C'^{\perp} \\
D'_s \multimap D''_t \arrow[r, "D'_s \multimap f''_t"'] & D'_s \multimap \Gamma C''^{\perp} &
\end{tikzcd}$$

It is routine to see that (v_s, v_t, u) makes the required diagrams commute. The converse direction is similar: from (v_s, v_t, u) we can derive (d_s, d_t, c) as

$$\dfrac{\dfrac{D_s \xrightarrow{v_s} Y \xrightarrow{\pi_Y} D'_s \multimap D''_t}{D_s \longrightarrow D'_s \multimap D''_t}}{D_s \otimes D'_s \xrightarrow{d_s} D''_t} \qquad \dfrac{\dfrac{\dfrac{D_s \xrightarrow{v_s} Y \xrightarrow{\pi'_Y} D''_s \multimap D'_t}{D_s \longrightarrow D''_s \multimap D'_t}}{D''_s \xrightarrow{\alpha} D_s \multimap D'_t} \qquad \dfrac{D'_s \otimes D''_s \xrightarrow{v_t} D_t}{D''_s \xrightarrow{\beta} D'_s \multimap D_t}}{D''_s \xrightarrow{d_t} X}\,\star$$

(the derivation of c from u is just the inverse of that of u from c) where again the step $\star$ follows from the commutativity of diagrams below and the definition of X and x.

$$\begin{tikzcd}
D''_s \arrow[rr, "\beta"] \arrow[dd, "\alpha"'] \arrow[dr, "\Gamma c^{\perp} \circ \Gamma\simeq \circ f''_t"] & & D'_s \multimap D_t \arrow[d, "D'_s \multimap f_t"] \\
& \Gamma(C \otimes C')^{\perp} \arrow[r, "(f'_s \multimap \Gamma C^{\perp}) \circ \theta \circ \Gamma\simeq"'] \arrow[d, "(f_s \multimap \Gamma C'^{\perp}) \circ \theta \circ \Gamma\simeq"] & D'_s \multimap \Gamma C^{\perp} \\
D_s \multimap D'_t \arrow[r, "D_s \multimap f'_t"'] & D_s \multimap \Gamma C'^{\perp} &
\end{tikzcd}$$

Again it is routine to see that (v_s, v_t, u) makes the required diagrams commute. Also it is now easy to see that these mappings are inverses of each other. The check of naturality is straightforward.