

RIMS-1932

**Generalized Bounded Linear Logic and
its Categorical Semantics**

By

Yōji FUKIHARA and Shin-ya KATSUMATA

November 2020



京都大学 数理解析研究所

RESEARCH INSTITUTE FOR MATHEMATICAL SCIENCES

KYOTO UNIVERSITY, Kyoto, Japan

Generalized Bounded Linear Logic and its Categorical Semantics

Yōji Fukihara¹ and Shin-ya Katsumata²

¹ Kyoto University

² National Institute of Informatics

Abstract. We introduce GBLL and GBAL, generalizations of Girard et al.'s BLL. The calculus extracts an underlying fundamental structure of BLL, while separates complexity-related issues in BLL. We analyze the complexity of cut-elimination in GBLL, and give a translation of QBLL, a fragment of QBAL (a variant of BLL by Dal Lago and Hofmann), into GBAL. We then introduce *indexed linear exponential comonads* (ILEC for short) as a categorical structure for interpreting the !-modality of GBLL. This is obtained by extending the grading semiring of graded linear exponential comonads to the 2-category **Idx**, which may be seen as a multi-object pseudo-semiring. We give an elementary example of ILEC using folding product, and its modification via symmetric monoidal comonads. We then instantiate this elementary example with the category of assemblies of a BCI-algebra, and discuss (dis)similarity with the realizability category studied by [18,9].

Keywords: Linear Logic · Categorical Semantics · Linear Exponential Comonad · Graded Comonad

1 Introduction

Girard's *linear logic* is a refinement of the ordinary (non-linear) logic by restricting the usage of weakening and contraction in proofs [14]. The linear logic also has the *of-course modality* !, which partially restores these structural rules to the formulas of the form !A.

Later, Girard et al. introduced an extension of the !-modality called *bounded !-modality*, and applied it to a logical characterization of P-time computations [15]. In their paper, the bounded !-modality is introduced in two steps.

First, a simple !-modality $!_r A$ is introduced [15, Section 2.4]. The index r ranges over a semiring, and in modern terminology r is called a *grade* [10,12]. This simple !-modality and its variants has been considered in various logics and programming languages [7,29,13,25,27]. The categorical structure corresponding to $!_r A$ has been identified as *graded linear exponential comonad* in [6,12], and its double-category theoretic characterization was given in [21]

Second, the bounded !-modality $!_{x < p} A$ involving *variable binding* is introduced [15, Section 3]. This is the actual modality employed in the *bounded linear*

logic (BLL for short) [15]; p is called a *resource polynomial*. The logic thus allows two kinds of dependency: *grade-dependent grades*, which occurs as p in the modal formula $!_{x < p} A$, and *grade-dependent formulas*, such as $!_{x < p} A$ itself. These two kinds of dependency significantly increases the expressiveness and flexibility of the bounded linear logic, leading to the characterization of P-time complexity. This characterization result is later proved through a *realizability semantics* of BLL [15,18,9]. Inside this semantics, however, mechanisms for controlling the complexity of proof reductions are hard-coded, and it is not very clear which categorical structure is behind the semantics.

In this paper we propose a generalized bounded linear logic GBLL and its categorical semantics. GBLL extracts the essence of the bounded linear logic, while abstracts away complexity-related issues in BLL, such as resource polynomials and computability constraints. The main feature of GBLL is the new format of !-modal formula. It is of the form $\Delta \vdash !_f A$, where f is a function of type $\Delta \rightarrow (\Delta')^*$, $\Delta' \vdash A$ is a formula and Δ, Δ' are *indexing sets* of formulas. Intuitively, the !-modal formula $!_f A$ evaluated at an index $i \in \Delta$ invokes A with each index in the list $f(i) = j_1 \cdots j_n$, then form the meaning of $!_f A$ from $A(j_1) \cdots A(j_n)$, which is typically $A(j_1) \otimes \cdots \otimes A(j_n)$. This mechanism is enough to encode the bounded quantification $!_{x < p} A$ of BLL. We regard the inequality $x < p$ as the function sending an environment ρ to the list of extended environments $\rho\{x \mapsto 0\}, \dots, \rho\{x \mapsto \llbracket p \rrbracket \rho - 1\}$. They are then passed to A in this order, resulting the tensor product $\bigotimes_{0 \leq i < \llbracket p \rrbracket \rho} A\rho\{x \mapsto i\}$.

We equip this new format of !-modality with the structure of linear exponential comonad, referred as weakening, contraction, dereliction, digging and monoidality. For the case of graded linear exponential comonads, these structures interact with the semiring structure over grades. We follow the same strategy in this paper: we find a semiring-like structure on the collection of functions of the form $\Delta \rightarrow (\Delta')^*$. For the multiplication $g \bullet f$, we adopt the *Kleisli composition* of the free monoid monad $(-)^*$, while for the addition $f + g$, the pointwise concatenation $(f + g)(x) = f(x)g(x)$. However, these operations fail to satisfy one of the semiring axioms: $(f + g) \bullet h = f \bullet h + g \bullet h$. We make it hold up-to-isomorphism by introducing *2-cells* between functions of type $\Delta \rightarrow (\Delta')^*$ given by (pointwise) list permutations. The resulting 2-category **Idx** may be naturally seen as a *multi-object pseudo-semiring*. This 2-category is the common structure for both the syntax and semantics of GBLL.

Upon this indexing mechanism, we first study syntactic properties of GBLL. We introduce the cut elimination rules and study its complexity property. It turns out that the proof technique used in BLL naturally extends to GBLL - as done in [15], we classify cuts into *reducible* and *irreducible*, introduce *weights* of proofs, and show that the reduction steps of reducible cuts will terminate in cubic time of the weight of the proof. We also examine the expressive power of GBLL by giving a translation of QBLL, which is a monomorphic linear variant of Dal Lago and Hofmann's QBAL, to GBLL. The key idea of this translation is the aforementioned encoding mentioned.

We next give a categorical semantics of GBL. We introduce the concept of *indexed linear exponential comonad (ILEC)*; it is a \mathbf{Idx} -graded linear exponential comonad satisfying a commutativity condition with respect to an underlying indexed SMCCs. We then present a construction of ILEC from a symmetric monoidal closed category \mathbb{C} with a symmetric monoidal comonad on it. We demonstrate this construction with the category of assemblies over a BCI algebra [2,19] and relate the constructed model of GBL with the realizability category studied in [18,9].

Acknowledgment The authors are grateful to Masahito Hasegawa, Naohiko Hoshino, Clovis Eberherth and J  r  my Dubut for fruitful discussions. This research was supported by JST ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603).

Preliminaries For a set Δ , by Δ^* we mean the set of finite sequences of Δ . The empty sequence is denoted by $()$. Juxtaposition of Δ^* -elements denotes the concatenation of sequences. For $x \in \Delta^*$, by $|x|$ we mean the length of x . We identify a natural number n and the set $\{0, \dots, n-1\}$; note that $0 = \emptyset$. We also identify a sequence $x \in \Delta^*$ and a function of type $|x| \rightarrow \Delta$ where $x(i)$ is the i -th element of x .

2 Generalized Bounded Linear Logic

2.1 Indexing Category

As mentioned in introduction, we consider indexing sets Δ, Δ' and functions $f, g : \Delta \rightarrow (\Delta')^*$ listing up instantiation parameters of a formula A indexed by Δ' . For two functions f, g such that $f(i)$ is a permutation of $g(i)$ for each $i \in \Delta$, we would like to have isomorphisms between formulas $!_f A \cong !_g A$. To validate this argument, 1) we introduce a notion of morphisms between functions of type $\Delta \rightarrow (\Delta')^*$, and 2) we assume that the modality $!_f$ is functorial on f . Especially, the notion of morphism introduced in 1) is designed so that f, g are isomorphic when $f(i)$ is a permutation of $g(i)$ for each i .

To formulate this idea, we introduce the following 2-category \mathbf{Idx} ³. 0-cells are sets, and the hom-category $\mathbf{Idx}(\Delta, \Delta')$ is defined by:

- An object (1-cell) is a function $f : \Delta \rightarrow (\Delta')^*$.
- A morphism (2-cell) from f to g in $\mathbf{Idx}(\Delta, \Delta')$ is a Δ -indexed family of bijection $\{\sigma_x : |g(x)| \rightarrow |f(x)|\}_{x \in \Delta}$ such that $f(x)(\sigma_x(i)) = g(x)(i)$.

The identity 1-cell and the composition of 1-cells in \mathbf{Idx} are denoted by i_Δ and (\bullet) , respectively. The composition functor is defined by $(g \bullet f)(x) \stackrel{\text{def}}{=} g(y_1) \cdots g(y_n)$ where $y_1 \cdots y_n = f(x)$. The hom-category $\mathbf{Idx}(\Delta, \Delta')$ has a symmetric strict monoidal structure:

³ This is a full sub-2-category of the Kleisli 2-category \mathbf{CAT}_S , where S is the 2-monad of symmetric strict monoidal category [20].

- the monoidal unit is the constant empty-sequence function $0(x) = ()$,
- the tensor product of f, g , denoted by $f + g$, is defined by the index-wise concatenation $(f + g)(x) \stackrel{\text{def}}{=} f(x)g(x)$.

We write $J : \mathbf{Set} \rightarrow \mathbf{Idx}$ for the inclusion, namely $J\Delta = \Delta$ and $(Jf)(x) = f(x)$ (the singleton sequence).

Proposition 2.1. *The composition \bullet is symmetric strong monoidal in each argument. Especially, we have*

$$f \bullet 0 = 0 \quad 0 \bullet f = 0 \quad f \bullet (g + h) = f \bullet g + f \bullet h \quad (f + g) \bullet h \cong f \bullet h + g \bullet h.$$

We also define \mathbf{Idx}_a by replacing “bijection” in the definition of 2-cell of \mathbf{Idx} with “injection”. The hom-category $\mathbf{Idx}_a(\Delta, \Delta')$ has the 1-cell 0 as the terminal object, hence is a symmetric *affine* monoidal category.

2.2 Formulas and Proofs

Definition of GBLL Formulas We first fix an indexed set $\{\mathcal{A}(\Delta)\}_{\Delta \in \mathbf{Set}}$ of atomic propositions. We define the set of formulas by the following BNF:

$$A ::= a \star r \mid A \otimes B \mid A \multimap B \mid !_f A$$

where $a \in \mathcal{A}(\Delta)$ for some set Δ , r is a function and f is a 1-cell in \mathbf{Idx} . Formula formation rules are introduced to derive the pair $\Delta \vdash A$ of an index set Δ and a formula A , and are defined as follows:

$$\frac{a \in \mathcal{A}(\Delta') \quad r \in \Delta \rightarrow \Delta'}{\Delta \vdash a \star r} \quad \frac{\Delta \vdash A \quad \Delta \vdash B}{\Delta \vdash A \otimes B} \quad \frac{\Delta \vdash A \quad \Delta \vdash B}{\Delta \vdash A \multimap B}$$

$$\frac{\Delta' \vdash A \quad f \in \mathbf{Idx}(\Delta, \Delta')}{\Delta \vdash !_f A}$$

The formula $a \star r$ represents the atomic formula a precomposed with a reindexing function r . We next introduce reindexing of formulas.

Definition 2.1. *Let $r \in \mathbf{Set}(\Delta, \Delta')$. We define the reindexing operation $(-)|_r$ as follows:*

$$a \star r|_{r'} \stackrel{\text{def}}{=} a \star (r \circ r'), \quad (A \otimes B)|_r \stackrel{\text{def}}{=} A|_r \otimes B|_r,$$

$$(A \multimap B)|_r \stackrel{\text{def}}{=} A|_r \multimap B|_r, \quad (!_f A)|_r \stackrel{\text{def}}{=} !_f \bullet J_r A.$$

It holds that for any $r \in \mathbf{Set}(\Delta, \Delta')$, $\Delta' \vdash A$ implies $\Delta \vdash A|_r$. We identify well-formed formulas by the congruent equivalence relation on well-formed formulas generated from (below $r \in \mathbf{Set}(\Delta', \Delta'')$, $f \in \mathbf{Idx}(\Delta, \Delta')$, $\Delta'' \vdash A$)

$$!_{J_r \bullet f} A = !_f (A|_r). \tag{2.1}$$

We see some formations of formulas in GBLL.

Example 2.1. Let us illustrate how a formula $!_{y < x^2} !_{z < x+y} A$ in BLL is represented in GBL; here we assume that x, y, z are the only resource variables used in this formula. We first introduce a notation. Let E be a mathematical expression using variables $\mathbf{x}_1 \cdots \mathbf{x}_n$. Then by $[E]_n : \mathbb{N}^n \rightarrow (\mathbb{N}^{n+1})^*$ we mean the function

$$[E]_n(x_1, \dots, x_n) = (x_1, \dots, x_n, 0) \cdots (x_1, \dots, x_n, E[x_1/\mathbf{x}_1, \dots, x_n/\mathbf{x}_n] - 1).$$

For instance, $[\mathbf{x}_1^2]_1(x) = (x, 0), \dots, (x, x^2 - 1)$. Then from a well-formed formula $\mathbb{N}^3 \vdash A$, we obtain $\mathbb{N} \vdash !_{[\mathbf{x}_1^2]_1} !_{[\mathbf{x}_1 + \mathbf{x}_2]_2} A$. Generalizing this, a BLL formula $!_{x < E} A$ under a resource variable $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ corresponds to the GBL formula $!_{[E]_n} A$.

Example 2.2. We look at how we express the substitution of a resource polynomial $A[x := p(x_1, \dots, x_n)]$. We define a function $\langle p \rangle_n : \mathbb{N}^n \rightarrow \mathbb{N}^{n+1}$ by

$$\langle p \rangle_n(x_1, \dots, x_n) \stackrel{\text{def}}{=} (x_1, \dots, x_n, p(x_1, \dots, x_n))$$

Then the reindexed formula $\mathbb{N}^n \vdash A|_{\langle p \rangle_n}$ corresponds to $A[x := p(x_1, \dots, x_n)]$.

Example 2.3. We illustrate the equality between well-formed formulas. Consider a formula $\mathbb{N} \vdash A$ and a function $r : \mathbb{N}^3 \rightarrow \mathbb{N}$. Then we equate formulas $\mathbb{N}^2 \vdash !_{[\mathbf{x}_1 + \mathbf{x}_2]_2} (A|_r)$ and $\mathbb{N}^2 \vdash !_h A$, where $h : \mathbb{N}^2 \rightarrow \mathbb{N}^*$ is given by

$$h \stackrel{\text{def}}{=} Jr \bullet [\mathbf{x}_1 + \mathbf{x}_2]_2(x, y) = r(x, y, 0), \dots, r(x, y, x + y - 1).$$

Definition of GBL Proofs A *judgment* of the GBL is the form $\Delta \mid \Gamma \vdash A$, where Δ is a set, Γ is a sequence of formulas well-formed under Δ , and A is a well-formed formula under Δ , respectively. The inference rules of GBL is presented in Table 1. Similarly, we define GBAL to be the system obtained by replacing **Idx** in Table 1 with **Idx_a**.

Example 2.4. We mimic a special case of the contraction rule in BLL

$$\frac{\Gamma, !_{x < x_i} A, !_{y < x_j} A\{x_i + y/x\} \vdash B}{\Gamma, !_{x < x_i + x_j} A \vdash B}$$

see also (!C)-rule of QBL in Section 3.2. We use the *shift function* $s_{n,i} : \mathbb{N}^{n+1} \rightarrow \mathbb{N}^{n+1}$ defined by $s_{n,i}(x_1, \dots, x_n, y) \stackrel{\text{def}}{=} (x_1, \dots, x_n, x_i + y)$. Then we easily see $[\mathbf{x}_i]_n + Js_{n,i} \bullet [\mathbf{x}_j]_n = [\mathbf{x}_i + \mathbf{x}_j]_n$. Therefore by Contraction rule of GBL, we obtain the following derivation for well-formed formulas $\mathbb{N}^{n+1} \vdash A$ and $\mathbb{N}^n \vdash B$, mimicking the contraction of BLL:

$$\frac{!_{[\mathbf{x}_i]_n} A, !_{[\mathbf{x}_j]_n} (A|_{s_{n,i}}) \vdash B}{!_{[\mathbf{x}_i + \mathbf{x}_j]_n} A = !_{[\mathbf{x}_i]_n + Js_{n,i} \bullet [\mathbf{x}_j]_n} A \vdash B}$$

Here, we use the formula equality $!_{Js_{n,i} \bullet [\mathbf{x}_j]_n} A = !_{[\mathbf{x}_j]_n} (A|_{s_{n,i}})$.

$$\begin{array}{c}
\frac{\Delta \vdash A}{\Delta \mid A \vdash A} \text{ (Ax) Axiom} \quad \frac{\Delta \mid \Gamma, X, Y, \Gamma' \vdash A}{\Delta \mid \Gamma, Y, X, \Gamma' \vdash A} \text{ (Exch) Exchange} \\
\\
\frac{\Delta \mid \Gamma_1 \vdash A \quad \Delta \mid \Gamma_2, A \vdash B}{\Delta \mid \Gamma_1, \Gamma_2 \vdash B} \text{ (Cut)} \\
\\
\frac{\Delta \mid \Gamma, X, Y \vdash A}{\Delta \mid \Gamma, X \otimes Y \vdash A} \text{ (\otimes L)} \quad \frac{\Delta \mid \Gamma_1 \vdash X \quad \Delta \mid \Gamma_2 \vdash Y}{\Delta \mid \Gamma_1, \Gamma_2 \vdash X \otimes Y} \text{ (\otimes R)} \\
\\
\frac{\Delta \mid \Gamma_1 \vdash X \quad \Delta \mid \Gamma_2, Y \vdash B}{\Delta \mid \Gamma_1, \Gamma_2, X \multimap Y \vdash B} \text{ (\multimap L)} \quad \frac{\Delta \mid \Gamma, X \vdash Y}{\Delta \mid \Gamma \vdash X \multimap Y} \text{ (\multimap R)} \\
\\
\frac{\Delta \mid \Gamma \vdash B}{\Delta \mid \Gamma, !_0 A \vdash B} \text{ (!W) Weakening} \quad \frac{\Delta \mid \Gamma, A \vdash B}{\Delta \mid \Gamma, !_d A \vdash B} \text{ (!D) Dereliction} \\
\\
\frac{\Delta \mid \Gamma, !_g A \vdash B \quad \sigma \in \mathbf{Idx}(\Delta, \Delta')(f, g)}{\Delta \mid \Gamma, !_f A \vdash B} \text{ (!F) !-Functor} \\
\\
\frac{\Delta \mid \Gamma, !_f A, !_f A \vdash B}{\Delta \mid \Gamma, !_f A \vdash B} \text{ (!C) Contraction} \\
\\
\frac{\Delta' \mid !_g A_1, \dots, !_g A_k \vdash B \quad f \in \mathbf{Idx}(\Delta, \Delta')}{\Delta \mid !_g \bullet f A_1, \dots, !_g \bullet f A_k \vdash !_f B} \text{ (P!) Composition}
\end{array}$$

Table 1. GBLL Proof Rules

Example 2.5. We consider reindexing of proofs. Let $r \in \mathbf{Set}(\Delta, \Delta')$ and $\Gamma|_r$ denotes a sequence $C_1|_r, \dots, C_n|_r$ for $\Gamma = C_1, \dots, C_n$. For axiom rule $\Delta' \mid A \vdash A$, we give reindexing by r as $\Delta \mid A|_r \vdash A|_r$. Other cases except (P!) are easily defined. Each judgment $\Delta' \mid \Gamma \vdash A$ in each rule is replaced $\Delta \mid \Gamma|_r \vdash A|_r$ by reindexing. For (P!) rule, reindexing by r is given as follows:

$$\frac{\Delta'' \mid !_g A_1, \dots, !_g A_k \vdash B \quad f \bullet Jr \in \mathbf{Idx}(\Delta, \Delta'')}{\Delta \mid (!_{g_1 \bullet f A_1})|_r, \dots, (!_{g_k \bullet f A_k})|_r \vdash (!_f B)|_r}$$

2.3 Complexity of Cut Elimination in GBLL

By a similar discussion to BLL [15], cut inferences are divided in two classes: *reducible* cuts and *irreducible* cuts. We define a *weight* of proof $|\pi|$ (for each proof π) and *reduction steps* of proofs such that every reducible cut will be eliminated by reduction steps and every reduction steps will terminate in polynomial of $|\pi|$ (specifically, in $|\pi|^3$ steps).

Definition 2.2. *In GBLL (resp. GBAL) proofs, an instance of the Cut inference is irreducible if there are at least one Composition rule below it or if its left premise is obtained by a Composition rule with nonempty context and the*

other premise is obtained by a Weakening, !-Functor, Dereliction, Contraction or Composition inference. A reducible Cut is Cut inferences that is not irreducible.

Definition 2.3. A GBL or GBAL proof is irreducible if it contains only irreducible Cut inferences.

Definition 2.4. For a given proof $\pi \triangleright \Delta \mid \Gamma \vdash A$ of GBL or GBAL, the weight of π is a function $|\pi| : \Delta \rightarrow \mathbb{N}$ defined as follows:

1. For an Axiom rule $\pi \triangleright \Delta \mid A \vdash A$, $|\pi|(\delta) \stackrel{\text{def}}{=} 1$.
2. If π is obtained from π' by an unary rule except Contraction and Composition, $|\pi| \stackrel{\text{def}}{=} |\pi'| + 1$.
3. If π is obtained from π_1 and π_2 by a binary rule except Cut, $|\pi| \stackrel{\text{def}}{=} |\pi_1| + |\pi_2| + 1$.
4. If π is obtained from π_1 and π_2 by a Cut rule, $|\pi| \stackrel{\text{def}}{=} |\pi_1| + |\pi_2|$.
5. If π is obtained from π' by a Contraction rule, $|\pi| \stackrel{\text{def}}{=} |\pi'| + 2$
6. If π is obtained from π' by a Composition rule, such as

$$\pi \triangleright \frac{\begin{array}{c} \vdots \pi' \\ \Delta' \mid !_{\alpha_1} A_1, \dots, !_{\alpha_k} A_k \vdash B \end{array}}{\Delta \mid !_{\alpha_1 \bullet f} A_1, \dots, !_{\alpha_k \bullet f} A_k \vdash !_f B}$$

then $|\pi|(\delta) \stackrel{\text{def}}{=} \sum_{\gamma \in f(\delta)} (|\pi'|(\gamma) + 2k + 1) + k + 1$. Note that the summation $\sum_{\gamma \in f(\delta)}$ scans all elements in the list $f(\delta)$, hence the weight depends on the length of $f(\delta)$.

Definitions of reduction steps for reducible cuts are shown in Section A. In each reduction step, the weight of proof is not increasing (some cases keep the weight). We call such reductions (keeping the weight of proof) *commutative reductions*, specifically $L\otimes$, $R\otimes$, $L\multimap$, $R\multimap$, $R\multimap$ -L, $R\multimap$ -R, W, F, D and C. And we call *symmetric reductions* as reductions of $S\otimes$, $S\multimap$, !W, !F, !D, !C and !P. The *axiom reductions* Ax are the different case.

Definition 2.5. For a proof π of GBL or GBAL, the cut size of the proof $\|\pi\|$ is defined as the same rule to the weight of the proof except Cut rule. When π is obtained by Cut rule from π_1 and π_2 , then the cut size is defined as $\|\pi\| \stackrel{\text{def}}{=} \|\pi_1\| + \|\pi_2\| + |\pi_1| + |\pi_2|$.

Proposition 2.2. In each commutative reduction step, the cut size decreases.

Proof. We use π', κ , etc. as the same subproof named in each case of definition of reduction steps.

- ($L\otimes$), ($R\multimap$ -R), (W), (F), (D) The cut size decreases from $\|\pi_1\| + \|\pi_2\| + |\pi_1| + |\pi_2| + 2$ to $\|\pi_1\| + \|\pi_2\| + |\pi_1| + |\pi_2| + 1$.

- (R \otimes), (R \otimes -L) The cut size decreases from $\|\pi'\| + \|\kappa_1\| + \|\kappa_2\| + |\pi'| + |\kappa_1| + |\kappa_2| + 2$ to $\|\pi'\| + \|\kappa_i\| + |\pi'| + |\kappa_1| + |\kappa_2| + 1$ ($i = 0, 1$).
- (L \multimap) The cut size decreases from $\|\pi_1\| + \|\pi_2\| + \|\kappa\| + |\pi_1| + |\pi_2| + |\kappa| + 2$ to $\|\pi_2\| + \|\kappa\| + |\pi_1| + |\pi_2| + |\kappa| + 1$.
- (C) The cut size decreases from $\|\pi_1\| + \|\pi_2\| + |\pi_1| + |\pi_2| + 4$ to $\|\pi_1\| + \|\pi_2\| + |\pi_1| + |\pi_2| + 2$.

Lemma 2.1. *For every proof π , it holds $\|\pi\| \leq |\pi|^2$.*

Proof. By induction on the structure of π . Clearly $|\pi'|^2 + 1 \leq (|\pi'| + 1)^2$, $|\pi_1|^2 + |\pi_2|^2 + 1 \leq (|\pi_1| + |\pi_2| + 1)^2$ and so on.

In the case that π is obtained from π_1 and π_2 by Cut rule, it holds $|\pi_1|^2 + |\pi_2|^2 + |\pi_1| + |\pi_2| \leq (|\pi_1| + |\pi_2|)^2$ because $|\pi_1|, |\pi_2| \geq 1$.

Theorem 2.1. *For every proof π , reduction steps will terminate in $|\pi|^3$ steps.*

Proof. Since non-commutative reduction steps decrease the weight of proof $|\pi|$, these steps are at most $|\pi|$. In each non-commutative reduction step, commutative reduction steps are at most $\|\pi\| \leq |\pi|^2$. Therefore reduction steps are at most $|\pi|^3$.

3 Translation from Constrained BLL

To examine the expressiveness of our general framework, we give a translation from Dal Lago and Hofmann’s QBAL [9], a modern extension of BLL, to a minor extension of GBLL. Actually, the source of the translation is a monomorphic, linear fragment of QBAL, which we call QBLL⁴. Following QBAL, a judgment of QBLL is of the form $\Gamma \vdash_{\mathcal{C}} A$, and is inferred under a set \mathcal{C} of *constraints* on free resource variables. Annotating judgments with constraints is one of the main differences between the BLL and QBLL / QBAL. Especially each judgment $\Gamma \vdash A$ in BLL corresponds to the QBLL judgment $\Gamma \vdash_{\emptyset} A$.

The target of the translation is a minor extension of GBLL; the plain GBLL is insufficient for the translation. The extension involves two points. First, we add the weakening $\Delta \mid !_{f+g} A \vdash !_f A$ to GBLL; thus we actually work with GBAL. This is to soundly interpret the weakening $!_{p+q} A \vdash_{\mathcal{C}} !_p A$ in QBAL. Second, we assume the positivity of each atomic formulas, which is assumed in QBAL. For every n -ary atomic formula $a \in \mathcal{A}$ in QBAL, we introduce an atomic formula $[a] \in \mathcal{A}(\mathbb{N}^n)$ in GBLL together with the *positivity axiom*:

$$V_{\mathcal{C}}(F) \mid \emptyset \vdash [a] \star \langle p_1, \dots, p_n \rangle \multimap [a] \star \langle q_1, \dots, q_n \rangle \quad (\forall i. p_i \sqsubseteq_{\mathcal{C}} q_i).$$

Here the definition of each notation is given in section 3.1 and 3.3. Positivity axiom induces proofs $V_{\mathcal{C}}(F) \mid A' \vdash A$ for every formulas A, A' such that $A' \sqsubseteq_{\mathcal{C}} A$ (the relation $\sqsubseteq_{\mathcal{C}}$ for formulas defined in section 3.2).

⁴ QBAL admits unrestricted weakening, which is not available in GBLL/GBAL.

3.1 Resource Polynomials and Constraints

We introduce basic concepts around QBLL, referring to its original calculus QBAL [9]. We put a reference in the beginning of each paragraph when the contents comes from QBAL.

[9, Definition 2.1] Given a countably infinite set \mathcal{RV} of *resource variables*, a *resource monomial* over \mathcal{RV} is a finite product of binomial coefficients $\prod_{i=1}^m \binom{x_i}{n_i}$, where the resource variables x_1, \dots, x_m are distinct and $n_1, \dots, n_m \in \mathbb{N}$ are natural numbers. A *resource polynomial* over \mathcal{RV} is a finite sum of resource monomials. We write 1 as $\binom{x}{0}$ and x as $\binom{x}{1}$ for short. Each positive natural number n denotes a resource polynomial $1 + 1 + \dots + 1$. Resource polynomials are closed under sum, product, bounded sum and composition [9, Lemma 2.2].

[9, Definition 2.3] A *constraint* is an inequality $p \leq q$, where p and q are resource polynomials. We abbreviate $p + 1 \leq q$ as $p < q$. A constraint $p \leq q$ *holds* (written $\vDash p \leq q$) if it is true in the standard model. A *constraint set* (denoted with \mathcal{C}, \mathcal{D}) is a finite set of constraints. A constraint $p \leq q$ is a *consequence* of a constraint set \mathcal{C} (written $\mathcal{C} \vDash p \leq q$) if $p \leq q$ is logical consequence of \mathcal{C} . For every constraint sets \mathcal{C} and \mathcal{D} , we write $\mathcal{C} \vDash \mathcal{D}$ iff $\mathcal{C} \vDash p \leq q$ for every constraints $p \leq q$ in \mathcal{D} . For each constraint set \mathcal{C} , we define an order $\sqsubseteq_{\mathcal{C}}$ on resource polynomials by $p \sqsubseteq_{\mathcal{C}} q$ iff $\mathcal{C} \vDash p \leq q$.

[9, Definition 2.3] We define the polarity of occurrences of free resource variables. For a constraint $p \leq q$, we say that an occurrence of a resource variable x in p is called *negative*, while the one in q is called *positive*.

3.2 Formulas and Inference Rules of QBLL

Let \mathcal{A} be a set of atomic formulas and assume that each atomic formula $a \in \mathcal{A}$ is associated with an arity $\text{ar}(a)$. Formulas of QBLL are defined by:

$$A, B ::= a(p_1, \dots, p_{\text{ar}(a)}) \mid A \otimes B \mid A \multimap B \mid !_{x < p} A$$

where a formula $!_{x < p} A$ satisfies $x \notin \text{FV}(p)$.

[9, Definition 2.6] Each occurrence of a free resource variable in a formula is classified into *positive* or *negative*. Below we inductively define a *positive occurrence* of a resource variable. An occurrence of x in:

- $a(p_1, \dots, p_{\text{ar}(a)})$ is always positive.
- $A \otimes B$ is positive iff it is in A and positive, or so in B .
- $A \multimap B$ is positive iff it is in A and negative, or it is in B and positive.
- $!_{x' < p} A$ is positive iff it is in A and positive. We remark that an occurrence of a free resource variable in p is counted as negative in $!_{x' < p} A$.

[9, Definition 2.8] We extend the order $\sqsubseteq_{\mathcal{C}}$ on resource polynomials to the one on QBLL formulas.

$$\begin{aligned} a(p_1, \dots, p_{\text{ar}(a)}) \sqsubseteq_{\mathcal{C}} a(q_1, \dots, q_{\text{ar}(a)}) & \text{ iff } \forall i. p_i \sqsubseteq_{\mathcal{C}} q_i \\ A \otimes B \sqsubseteq_{\mathcal{C}} C \otimes D & \text{ iff } (A \sqsubseteq_{\mathcal{C}} C) \wedge (B \sqsubseteq_{\mathcal{C}} D) \\ A \multimap B \sqsubseteq_{\mathcal{C}} C \multimap D & \text{ iff } (C \sqsubseteq_{\mathcal{C}} A) \wedge (B \sqsubseteq_{\mathcal{C}} D) \\ !_{x < p} A \sqsubseteq_{\mathcal{C}} !_{x < q} B & \text{ iff } (q \sqsubseteq_{\mathcal{C}} p) \wedge (x \notin \text{FV}(\mathcal{C})) \wedge (A \sqsubseteq_{\mathcal{C} \cup \{x < q\}} B) \end{aligned}$$

$$\begin{array}{c}
\frac{A \sqsubseteq_{\mathcal{C}} B}{A \vdash_{\mathcal{C}} B} \text{ (Ax)} \qquad \frac{\Gamma_1 \vdash_{\mathcal{C}} A \quad \Gamma_2, A \vdash_{\mathcal{C}} B}{\Gamma_1, \Gamma_2 \vdash_{\mathcal{C}} B} \text{ (Cut)} \\
\\
\frac{\Gamma, A, B \vdash_{\mathcal{C}} C}{\Gamma, A \otimes B \vdash_{\mathcal{C}} C} \text{ (\otimes L)} \qquad \frac{\Gamma_1 \vdash_{\mathcal{C}} A \quad \Gamma_2 \vdash_{\mathcal{C}} B}{\Gamma_1, \Gamma_2 \vdash_{\mathcal{C}} A \otimes B} \text{ (\otimes R)} \\
\\
\frac{\Gamma_1 \vdash_{\mathcal{C}} A \quad \Gamma_2, B \vdash_{\mathcal{C}} C}{\Gamma_1, \Gamma_2, A \multimap B \vdash_{\mathcal{C}} C} \text{ (\multimap L)} \qquad \frac{\Gamma, A \vdash_{\mathcal{C}} B}{\Gamma \vdash_{\mathcal{C}} A \multimap B} \text{ (\multimap R)} \\
\\
\frac{\Gamma \vdash_{\mathcal{C}} A \quad \mathcal{D} \vDash \mathcal{C}}{\Gamma \vdash_{\mathcal{D}} A} \text{ (Str)} \qquad \frac{\Gamma \vdash_{\mathcal{C}} B}{\Gamma, !_{x < 0} A \vdash_{\mathcal{C}} B} \text{ (!W)} \\
\\
\frac{A\{0/x\}, \Gamma \vdash_{\mathcal{C}} B}{!_{x < 1} A, \Gamma \vdash_{\mathcal{C}} B} \text{ (!D)} \qquad \frac{\Gamma, !_{x < p} A, !_{y < q} A\{p+y/x\} \vdash_{\mathcal{C}} B}{\Gamma, !_{x < p+q} A \vdash_{\mathcal{C}} B} \text{ (!C)} \\
\\
\frac{A_1, \dots, A_n \vdash_{\mathcal{C} \cup \{x < p\}} B \quad x \notin \text{FV}(\mathcal{C})}{!_{x < p} A_1, \dots, !_{x < p} A_n \vdash_{\mathcal{C}} !_{x < p} B} \text{ (!P)} \\
\\
\frac{!_{y < p} !_{z < q} A\{y/w\} A\{(z + \sum_{w < y} q(w))/x\}, \Gamma \vdash_{\mathcal{C}} B}{!_{x < \sum_{w < p} q(w)} A, \Gamma \vdash_{\mathcal{C}} B} \text{ (!N)}
\end{array}$$

Table 2. Inference rules for QBLL

[9, Section 2.3] A QBLL *judgment* is an expression $\Gamma \vdash_{\mathcal{C}} A$, where \mathcal{C} is a constraint set, Γ is a multiset of formulas and A is a formula. A judgment $\Gamma \vdash_{\mathcal{C}} A$ means that A is a consequence of Γ under the constraints \mathcal{C} .

Inference rules are shown in Table 2. Basically this is a fragment of QBAL. We restricted the Weakening rule (!W) and separated two operations from each rule: (1) $\Gamma \vdash_{\mathcal{C}} A$ and $\mathcal{D} \vDash \mathcal{C}$ implies $\Gamma \vdash_{\mathcal{D}} A$ and (2) $\Gamma, !_{x < p} X \vdash_{\mathcal{C}} A$ implies $\Gamma, !_{x < p+q} X \vdash_{\mathcal{C}} A$.

3.3 Translation into GBAL

As mentioned at the beginning of section 3, the translation we will give is from QBLL into GBAL. For a QBLL proof $\Gamma \vdash_{\mathcal{C}} A$, its translation is dependent on the constraint set \mathcal{C} and a set of free resource variables F (satisfying $\text{FV}(\Gamma) \cup \text{FV}(A) \cup \text{FV}(\mathcal{C}) \subseteq F$). That is to say, the translation of $\Gamma \vdash_{\mathcal{C}} A$ has the form $V_{\mathcal{C}}(F) \mid [\Gamma]^{(F; \mathcal{C})} \vdash [A]^{(F; \mathcal{C})}$ in GBAL.

For Constraints We first define an *environment* over a finite set F of resource variables to be a function from F to \mathbb{N} ; by $V(F)$ we mean the set of environments over F . Given an environment $\rho \in V(F)$ and a resource variable $x \notin F$ and $n \in \mathbb{N}$, by $\rho\{x \mapsto n\}$ we mean the environment over $F \cup \{x\}$ that extends ρ with a mapping $x \mapsto n$. Given a resource polynomial p such that $\text{FV}(p) \subseteq F$, by $\llbracket p \rrbracket : V(F) \rightarrow \mathbb{N}$ we mean the function that evaluates the resource polynomial p under a given environment. For resource polynomials p_1, \dots, p_n such that

$\text{FV}(p_i) \subseteq F$, we give a map $\langle p_1, \dots, p_n \rangle : (V(F)) \rightarrow \mathbb{N}^n$ by $\langle p_1, \dots, p_n \rangle \rho = (\llbracket p_1 \rrbracket \rho, \dots, \llbracket p_n \rrbracket \rho)$.

For a constraint $p \leq q$ with a set F of free resource variables (such that $\text{FV}(p) \cup \text{FV}(q) \subseteq F$) and for an environment $\rho \in V(F)$, let $\rho \models p \leq q$ denote $\llbracket p \rrbracket \rho \leq \llbracket q \rrbracket \rho$. For a subset $S \subseteq V(F)$ and for a constraint set \mathcal{C} , $S \models \mathcal{C}$ is also defined similarly: for every $\rho \in S$ and for every $p \leq q \in \mathcal{C}$, $\rho \models p \leq q$. Given a constraint set \mathcal{C} and a set F of resource variables such that $\text{FV}(\mathcal{C}) \subseteq F$, let $V_{\mathcal{C}}(F)$ be a set given by:

$$V_{\mathcal{C}}(F) \stackrel{\text{def}}{=} \{\rho \in V(F) \mid \rho \models \mathcal{C}\}$$

For a resource polynomial p , a free resource variable x such that $x \notin \text{FV}(p)$, a constraint set \mathcal{C} and a set F of resource variables such that $\text{FV}(p) \cup \text{FV}(\mathcal{C}) \subseteq F$, we introduce a map $[x < p]_{(F, \mathcal{C})} : V_{\mathcal{C}}(F) \rightarrow V_{\mathcal{C} \cup \{x < p\}}(F \cup \{x\})^*$ by

$$[x < p]_{(F, \mathcal{C})} \rho \stackrel{\text{def}}{=} \rho\{x \mapsto 0\}, \rho\{x \mapsto 1\}, \dots, \rho\{x \mapsto (\llbracket p \rrbracket \rho - 1)\}$$

For Formulas For a QBLL formula A , a constraint set \mathcal{C} and a set of resource variables F such that $\text{FV}(A) \cup \text{FV}(\mathcal{C}) \subseteq F$, the translation $[A]^{(F; \mathcal{C})}$ of A under F and \mathcal{C} is defined inductively as follows:

$$\begin{aligned} [a(p_1, \dots, p_n)]^{(F; \mathcal{C})} &\stackrel{\text{def}}{=} [a] \star \langle p_1, \dots, p_n \rangle |_{V_{\mathcal{C}}(F)} & [A \otimes B]^{(F; \mathcal{C})} &\stackrel{\text{def}}{=} [A]^{(F; \mathcal{C})} \otimes [B]^{(F; \mathcal{C})} \\ [A \multimap B]^{(F; \mathcal{C})} &\stackrel{\text{def}}{=} [A]^{(F; \mathcal{C})} \multimap [B]^{(F; \mathcal{C})} & [!_{x < p} A]^{(F; \mathcal{C})} &\stackrel{\text{def}}{=} !_{[x < p]_{(F, \mathcal{C})}} [A]^{(F \cup \{x\}; \mathcal{C} \cup \{x < p\})} \end{aligned}$$

Every translation $[A]^{(F; \mathcal{C})}$ of QBLL formula is well-formed under $V_{\mathcal{C}}(F)$.

For Proofs To give a translation of proofs, we define another notation. For a resource polynomial p, q , a set F of resource variables and a constraint set \mathcal{C} such that $\text{FV}(p) \cup \text{FV}(\mathcal{C}) \subseteq F$, a set $[p, q]^{(F; \mathcal{C})}$ of environments is defined by

$$[p, q]^{(F; \mathcal{C})} = \{\rho \in V(F \cup \{t\}) \mid \rho \models \mathcal{C}, \llbracket p \rrbracket(\rho) \leq \rho(t) < \llbracket p + q \rrbracket \rho\}$$

here t be a ‘‘fresh’’ resource variable such that $t \notin F$.

For a proof $\pi \triangleright \Gamma \vdash_{\mathcal{C}} A$, a translation $[\pi]^{(F; \mathcal{C})} \triangleright V_{\mathcal{C}}(F) \mid [\Gamma]^{(F; \mathcal{C})} \vdash [A]^{(F; \mathcal{C})}$ is defined inductively on the structure of the proof:

- For Axiom rule, we can prove $V_{\mathcal{C}}(F) \mid [A]^{(F; \mathcal{C})} \vdash [B]^{(F; \mathcal{C})}$ for formulas A, B such that $A \sqsubseteq_{\mathcal{C}} B$.
- For rules (Cut), (\otimes L), (\otimes R), (\multimap L), (\multimap R) and (!W), the translation is simple replacement of each formula A with $[A]^{(F; \mathcal{C})}$.
- For (Str) rule, we have a map $r : V_{\mathcal{C}}(F) \rightarrow V_{\mathcal{C}}(F)$ then the translation is given as reindexed proof $[\pi']^{(F; \mathcal{C})}|_r$ of the translation $[\pi']^{(F; \mathcal{C})}$ of the premise.
- For (!D) rule, the premise is translated $V_{\mathcal{C}}(F) \mid A', [\Gamma]^{(F; \mathcal{C})} \vdash [B]^{(F; \mathcal{C})}$, where $A' = [A]^{(F \cup \{x\}; \mathcal{C} \cup \{x < 1\})}|_r$ and r is a map such that $Jr = [x < 1]_{(F, \mathcal{C})}$.

- For (!C) rule, let a morphism $s_{p,q}^{(F;\mathcal{C})}$ and maps $r_{p,q}^{(F;\mathcal{C})}$, $i_1^{(p,q;F;\mathcal{C})}$, $i_2^{(p,q;F;\mathcal{C})}$ (s, r, i_1 and i_2 for short) be defined by

$$\begin{aligned}
s_{p,q}^{(F;\mathcal{C})} &: V_{\mathcal{C}}(F) \rightarrow [p, q]^{(F;\mathcal{C})} \\
&\quad \rho \mapsto \rho\{t \mapsto \llbracket p \rrbracket \rho\}, \dots, \rho\{t \mapsto (\llbracket p + q \rrbracket \rho - 1)\} \\
r_{p,q}^{(F;\mathcal{C})} &: [p, q]^{(F;\mathcal{C})} \xrightarrow{\sim} V_{\mathcal{C} \cup \{y < q\}}(F \cup \{y\}) \\
&\quad \rho\{t \mapsto (\llbracket p \rrbracket \rho + k)\} \mapsto \rho\{y \mapsto k\} \\
i_1^{(p,q;F;\mathcal{C})} &: V_{\mathcal{C} \cup \{x < p\}}(F \cup \{x\}) \rightarrow V_{\mathcal{C} \cup \{x < p+q\}}(F \cup \{x\}) \\
&\quad \rho\{x \mapsto k\} \mapsto \rho\{x \mapsto k\} \\
i_2^{(p,q;F;\mathcal{C})} &: [p, q]^{(F;\mathcal{C})} \rightarrow V_{\mathcal{C} \cup \{x < p+q\}}(F \cup \{x\}) \\
&\quad \rho\{t \mapsto (\llbracket p \rrbracket \rho + k)\} \mapsto \rho\{x \mapsto \llbracket p \rrbracket \rho + k\}
\end{aligned}$$

In this time, $!_{[x < p]}[A]^{(F \cup \{x\}; \mathcal{C} \cup \{x < p\})} = !_{[x < p]}([A]^{(F \cup \{x\}; \mathcal{C} \cup \{x < p+q\})}|_{i_1})$ and $!_{[y < q]}[A\{p+y/x\}]^{(F \cup \{y\}; \mathcal{C} \cup \{y < q\})} = !_{Jr \bullet s}([A]^{(F \cup \{x\}; \mathcal{C} \cup \{x < p+q\})}|_{i_2 \circ r^{-1}})$ hold. We obtain the conclusion of (!C) as

$$V_{\mathcal{C}}(F) \mid [I]^{(F;\mathcal{C})}, !_{(Ji_1 \bullet [x < p]) + (Ji_2 \bullet s)}[A]^{(F \cup \{x\}; \mathcal{C} \cup \{x < p+q\})} \vdash [B]^{(F;\mathcal{C})}$$

- For (!P) rule, let $F' = F \cup \{x\}$ and $\mathcal{C}' = \mathcal{C} \cup \{x < p\}$. We can prove the translated conclusion from the translated premise by the following proof:

$$\frac{V_{\mathcal{C}'}(F') \mid [A_1]^{(F';\mathcal{C}')} , \dots , [A_n]^{(F';\mathcal{C}')} \vdash [B]^{(F';\mathcal{C}')}}{n \text{ times (!D)'s :}}
\frac{V_{\mathcal{C}'}(F') \mid !_{\text{id}}[A_1]^{(F';\mathcal{C}')} , \dots , !_{\text{id}}[A_n]^{(F';\mathcal{C}')} \vdash [B]^{(F';\mathcal{C}')}}{V_{\mathcal{C}}(F) \mid !_{[x < p]}[A_1]^{(F';\mathcal{C}')} \dots !_{[x < p]}[A_n]^{(F';\mathcal{C}')} \vdash !_{[x < p]}[B]^{(F';\mathcal{C}')}}$$

- For (!N) rule, let sets $\Delta_0, \Delta_1, \Delta_2$ and constraint sets $\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2$ be

$$\begin{aligned}
\mathcal{C}_0 &= \mathcal{C} \cup \{y < p\} & \Delta_0 &= V_{\mathcal{C}_0}(F \cup \{y\}) \\
\mathcal{C}_1 &= \mathcal{C} \cup \{y < p, z < q\{y/w\}\} & \Delta_1 &= V_{\mathcal{C}_1}(F \cup \{y, z\}) \\
\mathcal{C}_2 &= \mathcal{C} \cup \{x < \sum_{w < p} q(w)\} & \Delta_2 &= V_{\mathcal{C}_2}(F \cup \{x\})
\end{aligned}$$

In this time, there is an isomorphism $r : \Delta_1 \xrightarrow{\sim} \Delta_2$ and an equation of morphisms $[z < q\{y/w\}]_{(F \cup \{y\}, \mathcal{C}_0)} \bullet [y < p]_{(F, \mathcal{C})} = Jr^{-1} \bullet [x < \sum_{w < p} q(w)]_{(F, \mathcal{C})}$ holds. Therefore (!N) rule can be translated by using the following provable judgment:

$$V_{\mathcal{C}}(F) \mid !_{[x < \sum_{w < p} q]}[A]^{(F \cup \{x\}; \mathcal{C}_2)} \vdash !_{[y < p]}!_{[z < q\{y/w\}]}[A\{z + \sum_{w < y} q/w\}]^{(F \cup \{y, z\}; \mathcal{C}_1)}$$

Since every BLL proof $\Gamma \vdash A$ can be translated as a QBALL proof $\Gamma \vdash_{\emptyset} A$, it can be translated as a GBAL proof $V_{\emptyset}(F) \mid [\Gamma]^{(F;\emptyset)} \vdash [A]^{(F;\emptyset)}$.

4 Categorical Semantics for GBL

In this section, we give a categorical semantics of the GBL. First, notice that each index set Δ determines a multiplicative linear logic under Δ . We model this situation by a *set-indexed symmetric monoidal closed categories*, given by a functor $C : \mathbf{Set}^{op} \rightarrow \mathbf{SMCC}_{\text{strict}}$. That is, for each $\Delta \in \mathbf{Set}$, a symmetric monoidal closed category $C\Delta$ is given, and any function $f : \Delta \rightarrow \Delta'$ induces a strict symmetric monoidal closed functor $Cf : C\Delta' \rightarrow C\Delta$, performing renaming of indexes.

Upon this indexed symmetric monoidal closed categories, we introduce a categorical structure that models $!_f$ modality. We call it *indexed linear exponential comonad*. This is a generalization of the *semiring-graded linear exponential comonad* studied in [12,21]. Our generalization replaces the semiring with \mathbf{Idx} , which may be regarded as a many-object pseudo-semiring (Proposition 2.1).

We write $[\mathbb{C}, \mathbb{D}]_l$ for the category of symmetric lax monoidal functors from \mathbb{C} to \mathbb{D} and monoidal natural transformations between them. We equip it with the pointwise symmetric monoidal structure $(\dot{I}, \dot{\otimes})$ given by $\dot{I}X = I$ and $(F \dot{\otimes} G)X = FX \otimes GX$ for $X \in \mathbb{C}$.

Definition 4.1. *An indexed linear exponential comonad (ILEC for short) over a set-indexed SMCC C consists of:*

- A collection of symmetric colax monoidal functors

$$(D, w^{\Delta, \Delta'}, c^{\Delta, \Delta'}) : \mathbf{Idx}(\Delta, \Delta') \rightarrow [C\Delta', C\Delta]_l \quad (\Delta, \Delta' \in \mathbf{Set}).$$

The symmetric lax monoidal structure of Df is denoted by $m_f : I \rightarrow DfI$ and $m_{f, A, B} : DfA \otimes DfB \rightarrow Df(A \otimes B)$.

- Monoidal natural transformations $\epsilon^\Delta : D(\text{id}_\Delta) \rightarrow \text{Id}_{D\Delta}$ and $\delta_{g, f} : D(g \bullet f) \rightarrow Df \circ Dg$ satisfying axioms in Figure 1.
- $Cr' \circ Df \circ Cr = D(Jr \bullet f \bullet Jr')$ holds for any morphism f in \mathbf{Idx} and r, r' in \mathbf{Set} of appropriate type.

The last axiom has two purposes: the equality $Cr'(DfA) = D(f \bullet Jr')A$ is to allow reindexing functions to act from outside, and the other equality $Df(CrA) = D(Cr \bullet f)A$ is to make D invariant under internal reindexing of formulas. This will be tied up with the formula equivalence in Definition (2.1) and the definition of reindexing at $!_f A$ in Definition (2.1). We postpone a concrete example of ILEC to Section 4.2.

4.1 Semantics of GBL

We interpret well-formed formula $\Delta \vdash A$ as an object $\llbracket \Delta \vdash A \rrbracket \in C\Delta$. This is done by induction on the structure of formula. We assume that each atomic formula $a \in \mathcal{A}(\Delta)$ comes with its interpretation as an object $\llbracket a \rrbracket \in C\Delta$.

$$\begin{aligned} \llbracket \Delta \vdash a \star r \rrbracket &\stackrel{\text{def}}{=} Cr[\llbracket a \rrbracket] & \llbracket \Delta \vdash !_f A \rrbracket &\stackrel{\text{def}}{=} Df[\llbracket \Delta' \vdash A \rrbracket] \\ \llbracket \Delta \vdash A \otimes B \rrbracket &\stackrel{\text{def}}{=} \llbracket \Delta \vdash A \rrbracket \otimes \llbracket \Delta \vdash B \rrbracket & \llbracket \Delta \vdash A \multimap B \rrbracket &\stackrel{\text{def}}{=} \llbracket \Delta \vdash A \rrbracket \multimap \llbracket \Delta \vdash B \rrbracket \end{aligned}$$

$$\begin{array}{ccc}
D(f \bullet h + g \bullet h)A & \equiv & D((f + g) \bullet h)A & D0A & \equiv & D(0 \bullet h)A \\
\downarrow & & \downarrow & \downarrow & & \downarrow \\
D(f \bullet h)A \otimes D(g \bullet h)A & & Dh(D(f + g)A) & Dh(D0A) & & \\
\downarrow & & \downarrow & \downarrow & & \\
Dh(DfA) \otimes Dh(DgA) & \longrightarrow & Dh(DfA \otimes DgA) & I & \longrightarrow & DhI \\
\\
D(h \bullet f + h \bullet g)A & \xrightarrow{D \cong A} & D(h \bullet (f + g))A & D0A & \equiv & D(h \bullet 0)A \\
\downarrow & & \downarrow & \downarrow & & \downarrow \\
D(h \bullet f)A \otimes D(h \bullet g)A & & D(f + g)(DhA) & D0(DhA) & & \\
\downarrow & & \downarrow & \downarrow & & \\
Df(DhA) \otimes Dg(DhA) & \equiv & (Df \otimes Dg)(DhA) & I & \equiv & I \\
\\
DfA & \longrightarrow & D(\text{id}_\Delta)(DfA) & D(h \bullet g \bullet f)A & \longrightarrow & D(g \bullet f)(DhA) \\
\downarrow & \searrow & \downarrow & \downarrow & & \downarrow \\
Df(D(\text{id}_\Delta)A) & \longrightarrow & DfA & Df(D(h \bullet g)A) & \longrightarrow & Df(Dg(DhA))
\end{array}$$

Fig. 1. Axioms of Indexed Linear Exponential Comonad

Proposition 4.1. *For any $r : \Delta \rightarrow \Delta'$ and well-formed formula $\Delta' \vdash A$, we have $\llbracket \Delta \vdash A \rrbracket_r = Cr \llbracket \Delta' \vdash A \rrbracket$.*

Proposition 4.2. $\llbracket \Delta \vdash !_{J,r} f A \rrbracket = \llbracket \Delta' \vdash !_f(A|_r) \rrbracket$.

Each proof $\pi \triangleright \Delta \mid \Gamma \vdash A$ of GBL is interpreted as a morphism $\llbracket \Delta \mid \Gamma \vdash A \rrbracket : \llbracket \Delta \vdash \Gamma \rrbracket \rightarrow \llbracket \Delta \vdash A \rrbracket$ in $C\Delta$. Here, for a sequence $\Gamma = C_1, \dots, C_m$ of formulas, $\llbracket \Delta \vdash \Gamma \rrbracket$ denotes $\llbracket \Delta \vdash C_1 \rrbracket \otimes \dots \otimes \llbracket \Delta \vdash C_m \rrbracket$. We write this interpretation only the cases of modalities, because the other rules, Axiom, Exchange, Cut, \otimes (L,R) and \multimap (L,R) are interpreted similarly to the semantics of multiplicative intuitionistic linear logic. The rules related to the modality are interpreted as in Table 3.

Theorem 4.1. *For a proof $\pi \triangleright \Delta \mid \Gamma \vdash A$, if π has a reducible cut and reduces into π' by a reduction step, then $\llbracket \pi \rrbracket = \llbracket \pi' \rrbracket$ in $C\Delta$.*

4.2 Construction of an Indexed Linear Exponential Comonad

We present a construction of an indexed SMCCs $C : \mathbf{Set}^{op} \rightarrow \mathbf{SMCC}_{\text{strict}}$ and an ILEC $D : \mathbf{Idx}(\Delta, \Delta') \rightarrow [C\Delta', C\Delta]_l$ over C from a SMCC $\langle \mathbb{C}, \otimes, I, \multimap \rangle$, and a symmetric lax monoidal comonad $\langle V, m^V, m_{X,Y}^V, \epsilon, \delta \rangle$ on \mathbb{C} .

Construction of Indexed SMCCs First, for each set Δ , we define the category $\Delta \pitchfork \mathbb{C}$ to be the product of Δ -many copies of \mathbb{C} . We represent objects and

$$\begin{aligned}
 & \left[\frac{\begin{array}{c} \vdots \pi' \\ \Delta \mid \Gamma \quad \vdash B \end{array}}{\Delta \mid \Gamma, !_{0_{\Delta, \Delta'}} A \vdash B} \right] = \llbracket \Gamma, !_{0_{\Delta, \Delta'}} A \rrbracket \xrightarrow[\llbracket \pi' \rrbracket \circ \cong]{\text{id} \otimes w_{\llbracket A \rrbracket}^{\Delta, \Delta'}} \llbracket \Gamma \rrbracket \otimes I \\
 & \left[\frac{\begin{array}{c} \vdots \pi' \\ \Delta \mid \Gamma, \quad A \vdash B \end{array}}{\Delta \mid \Gamma, !_{\text{id}_{\Delta}} A \vdash B} \right] = \llbracket \Gamma, !_{\text{id}_{\Delta}} A \rrbracket \xrightarrow[\llbracket \pi' \rrbracket \rrbracket]{\text{id} \otimes \epsilon_{\llbracket A \rrbracket}^{\Delta}} \llbracket \Gamma \rrbracket \otimes \llbracket A \rrbracket \\
 & \left[\frac{\begin{array}{c} \vdots \pi' \\ \Delta \mid \Gamma, !_{g} A \vdash B \quad \sigma : f \Rightarrow g \end{array}}{\Delta \mid \Gamma, !_{f} A \vdash B} \right] = \llbracket \Gamma, !_{f} A \rrbracket \xrightarrow[\llbracket \pi' \rrbracket \rrbracket]{\text{id} \otimes (D\sigma)_{\llbracket A \rrbracket}} \llbracket \Gamma, !_{g} A \rrbracket \\
 & \left[\frac{\begin{array}{c} \vdots \pi' \\ \Delta \mid \Gamma, !_{f} A, !_{g} A \vdash B \end{array}}{\Delta \mid \Gamma, !_{f+g} A \vdash B} \right] = \llbracket \Gamma, !_{f+g} A \rrbracket \xrightarrow[\llbracket \pi' \rrbracket \circ \cong]{\text{id} \otimes c_{f, g, \llbracket A \rrbracket}^{\Delta, \Delta'}} \llbracket \Gamma \rrbracket \otimes (\llbracket !_{f} A \rrbracket \otimes \llbracket !_{g} A \rrbracket) \\
 & \left[\frac{\begin{array}{c} \vdots \pi' \\ \Delta, g \mid !_{g_1} A_1, \dots, !_{g_k} A_k \vdash B \end{array}}{\Delta \mid !_{g_1 \bullet f} A_1, \dots, !_{g_k \bullet f} A_k \vdash !_{f} B} \right] = \begin{array}{c} \otimes_i \llbracket !_{g_i \bullet f} A_i \rrbracket \xrightarrow{\otimes_i \delta_{g_i, f, \llbracket A_i \rrbracket}} \otimes_i D(f)(\llbracket !_{g_i} A_i \rrbracket) \\ \xrightarrow{m_{f, \dots, \llbracket !_{g_i} A_i \rrbracket} \dots} D(f) (\otimes_i \llbracket !_{g_i} A_i \rrbracket) \\ \xrightarrow{D(f)(\llbracket \pi' \rrbracket)} \llbracket !_{f} B \rrbracket \end{array}
 \end{aligned}$$

Table 3. Interpretations of Modal Rules ($\llbracket A \rrbracket$ denotes $\llbracket \Delta \vdash A \rrbracket$ for each well-formed formula $\Delta \vdash A$).

morphisms of this category by maps $X : \Delta \rightarrow \text{Obj}(\mathbb{C})$ and maps $f : \Delta \rightarrow \text{Mor}(\mathbb{C})$, respectively. Since SMCCs are closed under products, $\Delta \pitchfork \mathbb{C}$ is a SMCC by the component-wise tensor product and internal hom:

$$I(d) \stackrel{\text{def}}{=} I, \quad X \otimes Y(d) \stackrel{\text{def}}{=} X(d) \otimes Y(d), \quad X \multimap Y(d) \stackrel{\text{def}}{=} X(d) \multimap Y(d)$$

We then define the indexed SMCCs C by $C\Delta \stackrel{\text{def}}{=} \Delta \pitchfork C$.

Folding Product We next introduce the *folding product* functor T ; we later compose it with the symmetric lax monoidal comonad V so that we can derive various ILECs over C . Note that T itself is also an ILEC; set $V = \text{Id}$. The type of T is $\mathcal{S}\Delta \times \Delta \pitchfork C \rightarrow C$ and defined by

$$T(i_1 i_2 \dots i_n, A) \stackrel{\text{def}}{=} A(i_1) \otimes A(i_2) \otimes \dots \otimes A(i_n), \quad T((), A) \stackrel{\text{def}}{=} I$$

On morphisms, T maps a list permutation in the first argument to the symmetry morphism in \mathbb{C} . T is symmetric strong monoidal in each argument. Moreover, each strong monoidal structure interacts well with each other, concluding that it becomes a multi-symmetric strong monoidal functor in the sense of [20].

Proposition 4.3. *For $\Delta \xrightarrow{f} \Delta'$ in \mathbf{Idx} and $l = i_1 \cdots i_k \in \mathcal{S} \Delta$, let $f(l)$ denote $f(i_1) \cdots f(i_k)$. Then it holds $\mathbb{T}(f(l), \mathbf{A}) \simeq \mathbb{T}(l, \mathbb{T}(f(-), \mathbf{A}))$ and this isomorphism is natural for \mathbf{A} .*

Construction of ILEC We now compose the folding product functor with the symmetric lax monoidal comonad V , to derive an ILEC. Fixed two sets Δ, Δ' , we define a symmetric strong (hence colax) monoidal functor $D : \mathbf{Idx}(\Delta, \Delta') \rightarrow [C\Delta', C\Delta]_l$ by

$$Df\mathbf{A}(i) \stackrel{\text{def}}{=} \mathbb{T}(f(i), V \circ \mathbf{A}) \quad Df\mathbf{p}(i) \stackrel{\text{def}}{=} \mathbb{T}(f(i), V\mathbf{p}) \quad D\alpha\mathbf{A} \stackrel{\text{def}}{=} \mathbb{T}(\alpha, V \circ \mathbf{A}). \quad (4.1)$$

Here, $\mathbf{A} \in \Delta' \pitchfork \mathbb{C}$, and \mathbf{p} and α are morphisms in $\Delta' \pitchfork \mathbb{C}$ and $\mathbf{Idx}(\Delta, \Delta')$, respectively. We also define a helper morphism $\gamma_{\mathbf{A}}^l : \mathbb{T}(l, V \circ \mathbf{A}) \rightarrow V\mathbb{T}(l, \mathbf{A})$ for $(l_1 \cdots l_k) \in \mathcal{S} \Delta$ and $\mathbf{A} \in \Delta \pitchfork \mathbb{C}$. It is the multiple composite of $m_{A,B}$:

$$VA(l_1) \otimes \cdots \otimes VA(l_k) \rightarrow V(\mathbf{A}(l_1) \otimes \cdots \otimes \mathbf{A}(l_k)).$$

It is routine to verify that this morphism is monoidal natural on l and \mathbf{A} .

Two monoidal natural transformations $\epsilon : \text{Did}_{\Delta} \rightarrow \text{Id}_{\Delta \pitchfork \mathbb{C}}$ and $\delta_{g,f} : D(g \bullet f) \rightarrow Df \circ Dg$ is defined by:

$$\epsilon_{\mathbf{A},i} : \mathbb{T}(i, V \circ \mathbf{A}) = VA(i) \quad (4.2)$$

$$\begin{aligned} \delta_{g,f;\mathbf{A};i} : \mathbb{T}((g \bullet f)(i), V \circ \mathbf{A}) &\xrightarrow{\sim} \mathbb{T}(f, \mathbb{T}(g(-), V \circ \mathbf{A})) & (4.3) \\ &\xrightarrow{\mathbb{T}(f, \mathbb{T}(g(-), \delta_{\mathbf{A}}))} \mathbb{T}(f, \mathbb{T}(g(-), V \circ V \circ \mathbf{A})) \\ &\xrightarrow{\mathbb{T}(f, \gamma_{\mathbf{A}}^{g(-)})} Df(Dg\mathbf{A})(i) \end{aligned}$$

Theorem 4.2. *The symmetric colax monoidal functor D defined by (4.1) with monoidal natural transformations ϵ, δ defined by (4.2) and (4.3) determines an ILEC over C .*

4.3 GBL Semantics by Realizability Category

Hofmann et al., and also Dal Lago et al. employ a *realizability semantics* to show that the complexity of BLL proof reductions belongs to P-time [18,9]. In this section we compare their semantics and the simple semantics of GBL constructed in the previous section.

We instantiate \mathbb{C} in the previous section with the realizability category over a BCI algebra (A, \cdot) , which is a combinatory algebra based on B, C, I -combinators; see e.g. [2,19]. We then form the realizability category $\mathbf{Ass}(A)$ by the following data: an object is a function f into P^+A , where P^+ is the nonempty powerset construction, and a morphism from f to g is a function $h : \text{dom } f \rightarrow \text{dom } g$ with the following property: there exists an element $e \in A$ such that for any

$x \in \text{dom } f$ and $a \in f(x)$, we have $e \cdot a \in g(h(x))$. The category $\mathbf{Ass}(A)$ is symmetric monoidal closed; see e.g. [19, Proposition 4]. The tensor product of f and g is given by $(f \otimes g)(x, y) = \{u \boxtimes v \mid u \in f(x), v \in g(y)\}$, where $u \boxtimes v$ is the BCL-algebra element corresponding to $\lambda x.xuv$ [19, Section 2].

Next, let Δ be a set and consider the power category $\Delta \pitchfork \mathbf{Ass}(A)$. Under the axiom of choice, $\Delta \pitchfork \mathbf{Ass}(A)$ is equivalently described as follows: an object is a family of functions $\{f_i\}_{i \in \Delta}$ into P^+A , and a morphism from $\{f_i\}_{i \in \Delta}$ to $\{g_i\}_{i \in \Delta}$ is a family of functions $\{h_i : \text{dom } f_i \rightarrow \text{dom } g_i\}_{i \in \Delta}$ with the following property: there exists a function $e : \Delta \rightarrow A$ such that for any $i \in \Delta$, $x \in \text{dom } f_i$ and $a \in f_i(x)$, we have $e(i) \cdot a \in g_i(h_i(x))$.

This power category is quite close to the realizability category introduced in [18, Section 4] and [9, Section 4]. This becomes apparent when identifying a membership statement $a \in f_i(x)$ for an object $\{f_i\}_{i \in \Delta} \in \Delta \pitchfork \mathbf{Ass}(A)$ and a realizability statement $i, a \Vdash x$ in the realizability category (see [18]). The major difference between these categories is twofold: 1) In the realizability category, a computability constraint is imposed on $e : \Delta \rightarrow A$ to achieve the characterization of P-time complexity. 2) Objects in the realizability category are limited to $\Delta \pitchfork \mathbf{Ass}(A)$ -objects such that all f_i share the common domain. This is to synchronize with the a set-theoretic semantics *ignoring* resource polynomials [18, Section 3] [9, Section 3].

We compute the bounded !-modality using the folding product ILEC \mathbb{T} with respect to the indexed SMCC $(-) \pitchfork \mathbf{Ass}(A)$. Let F be a finite set of variables, $x \notin F$ be a resource variable, p be a resource polynomial and \mathcal{C} be a constraint set under F . For any object \mathbf{X} in $V_{\mathcal{C} \cup \{v \leq p\}}(F \cup \{v\}) \pitchfork \mathbf{Ass}(A)$, the folding product $\mathbb{T}([v \leq p]_{(F, \mathcal{C})}, \mathbf{X})$ is an object in $V_{\mathcal{C}}(F) \pitchfork \mathbf{Ass}(A)$ satisfying

$$\begin{aligned} & \mathbb{T}([v < p]_{(F, \mathcal{C})}, \mathbf{X})(i) \\ &= \lambda(x_0, \dots, x_{\llbracket p \rrbracket i-1}) . \{a_0 \otimes \dots \otimes a_{\llbracket p \rrbracket i-1} \mid a_j \in \mathbf{X}(i\{v \mapsto j\})(x_j)\} \end{aligned} \quad (4.4)$$

This is different from the modality over the realizability category introduced in [18, Definition 16] and [9, Definition 4.6]:

$$(!_{v < p} \mathbf{X})(i) = \lambda x . \{a_0 \otimes \dots \otimes a_{\llbracket p \rrbracket i-1} \mid a_j \in \mathbf{X}(i\{v \mapsto j\})(x)\};$$

it only takes a single argument. This is again because their realizability semantics is designed to synchronize with the set-theoretic semantics ignoring resource polynomials - especially it interprets $\llbracket !_{x \leq p} A \rrbracket = \llbracket A \rrbracket$. On the other hand, the bounded quantification computed in (4.4) does *not* ignore resource polynomials and indexing, as the domain of (4.4) is the index-dependent product $\prod_j \text{dom}(\mathbf{X}(i\{v \mapsto j\}))$. From this, we conjecture that the semantics of BLL using the ILEC \mathbb{T} over $(-) \pitchfork \mathbf{Ass}(A)$ realizes an *index-dependent* set-theoretic semantics of BLL - we leave this semantics as a future work.

5 Conclusion and Related Work

We introduced GBLL, a generalization of Girard et al.'s BLL. The calculus extracts an underlying fundamental structure of BLL while separates complexity-related issues in BLL. We analyzed the complexity of cut-elimination in GBLL,

and gave a translation of QBLL, a variant of BLL by Dal Lago and Hofmann, into GBLL. We then introduced ILEC as a categorical structure for interpreting the !-modality of GBLL. The ILEC is an extension of semiring-graded linear exponential comonad, replacing semirings with the 2-category \mathbf{Idx} , which may be seen as a multi-object pseudo-semiring. Additionally, it should interact well with a specified indexed SMCCs. We gave an elementary construction of ILEC using folding product, and its variant inserting symmetric monoidal comonads. We instantiated this elementary example with the category of assemblies of a BCI-algebra, and discussed (dis)similarity with the realizability category studied by [18,9].

Girard's BLL has a great influence on the subsequent development of indexed modalities and implicit complexity theory [15]. Hofmann and Scott introduced the realizability technique to BLL and semantically proved that BLL characterizes P-time complexity [18]. Their work was further enriched and studied by Dal Lago and Hofmann [9]. Gaboardi combined the !-modality involving variable binding with PCF and showed that the combined system is relatively complete [23].

Bucciarelli and Ehrhard's *indexed linear logic with exponential* [8] is one of the closest systems to GBLL. However, the type of the !-modality is different: their system derives $\Delta \vdash !_f A$ from $\Delta' \vdash A$ and an *almost injective function* $f : \Delta' \rightarrow \Delta$; it is a function where each $f^{-1}(i)$ is finite. To relate their system and GBLL, let us use the finite powerset construction P_{fin} and convert f into its inverse $f^{-1} : \Delta \rightarrow P_{\text{fin}}(\Delta')$. This exhibit the similarity with GBLL: GBLL relaxes P_{fin} to $(-)^*$, and takes the inverse as the parameter for the !-modality. The novelty of this work to [8] is that a categorical axiomatics for the $!_f$ modality is identified as an extension of the graded linear exponential comonads [6,21]. Another novelty is to show that GBLL is enough to encode BLL.

As described in Section 1, the simple form of !-modality $!_r A$ is also widely used in various type systems and programming languages. Examples include: INTML [29], coeffct calculus [27,7] and its combination with effect systems [12], Granule language [25], bounded linear type system [13,25], type systems for the analysis of higher-order model-checking [17,16], a generic BLL-like logic $B_{\mathcal{S}}\text{LL}$ over semirings [5], Fuzz type system for function sensitivity and differential privacy [28,11,3], and many more. A combination of $!_r A$ with dependent type theory called QTT is also introduced in [24] and [4]. Among these systems, each of [11,25,1] supports 1) full universal and existential, 2) full universal and 3) partial universal quantification over grades, respectively.

The categorical structure corresponding to the simple form of !-modality appears in [6,12,21], and is identified as *semiring-graded linear exponential comonad*. Breuvert constructed various examples of semiring-graded linear exponential comonads on relational models of linear logic [5] using his *slicing* technique. In this work we replaced semirings to \mathbf{Idx} , which may be seen as a multi-object pseudo-semiring. In the study of graded monad, Orchard et al. generalize the grading structure from ordered monoids to 2-categories [26]. The main difference from this work is that their generalized graded monad is defined over a single categories, while an ILEC is defined over an *indexed* SMCCs.

References

1. Abel, A., Bernardy, J.P.: A unified view of modalities in type systems. *Proc. ACM Program. Lang.* **4**(ICFP) (Aug 2020). <https://doi.org/10.1145/3408972>, <https://doi.org/10.1145/3408972>
2. Abramsky, S., Lenisa, M.: Linear realizability and full completeness for typed lambda-calculi. *Ann. Pure Appl. Log.* **134**(2-3), 122–168 (2005). <https://doi.org/10.1016/j.apal.2004.08.003>, <https://doi.org/10.1016/j.apal.2004.08.003>
3. de Amorim, A.A., Gaboardi, M., Hsu, J., Katsumata, S., Cherigui, I.: A semantic account of metric preservation. In: Castagna, G., Gordon, A.D. (eds.) *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages, POPL 2017, Paris, France, January 18-20, 2017*. pp. 545–556. ACM (2017). <https://doi.org/10.1145/3009837>, <http://dl.acm.org/citation.cfm?id=3009890>
4. Atkey, R.: Syntax and semantics of quantitative type theory. In: Dawar, A., Grädel, E. (eds.) *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*. pp. 56–65. ACM (2018). <https://doi.org/10.1145/3209108.3209189>, <https://doi.org/10.1145/3209108.3209189>
5. Breuvert, F., Pagani, M.: Modelling coefficients in the relational semantics of linear logic. In: Kreutzer [22], pp. 567–581. <https://doi.org/10.4230/LIPICs.CSL.2015.567>, <https://doi.org/10.4230/LIPICs.CSL.2015.567>
6. Brunel, A., Gaboardi, M., Mazza, D., Zdancewic, S.: A core quantitative coefficient calculus. In: Shao, Z. (ed.) *Programming Languages and Systems*. pp. 351–370. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
7. Brunel, A., Gaboardi, M., Mazza, D., Zdancewic, S.: A core quantitative coefficient calculus. In: Shao, Z. (ed.) *Programming Languages and Systems*. pp. 351–370. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
8. Bucciarelli, A., Ehrhard, T.: On phase semantics and denotational semantics: the exponentials. *Annals of Pure and Applied Logic* **109**(3), 205 – 241 (2001). [https://doi.org/https://doi.org/10.1016/S0168-0072\(00\)00056-7](https://doi.org/https://doi.org/10.1016/S0168-0072(00)00056-7), <http://www.sciencedirect.com/science/article/pii/S0168007200000567>
9. Dal Lago, U., Hofmann, M.: Bounded linear logic, revisited. In: Curien, P.L. (ed.) *Typed Lambda Calculi and Applications*. pp. 80–94. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
10. Fujii, S., Katsumata, S., Melliès, P.: Towards a formal theory of graded monads. In: Jacobs, B., Löding, C. (eds.) *Foundations of Software Science and Computation Structures - 19th International Conference, FOSSACS 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings*. *Lecture Notes in Computer Science*, vol. 9634, pp. 513–530. Springer (2016). https://doi.org/10.1007/978-3-662-49630-5_30, https://doi.org/10.1007/978-3-662-49630-5_30
11. Gaboardi, M., Haeberlen, A., Hsu, J., Narayan, A., Pierce, B.C.: Linear dependent types for differential privacy. In: Giacobazzi, R., Cousot, R. (eds.) *The 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '13, Rome, Italy - January 23 - 25, 2013*. pp. 357–370. ACM (2013). <https://doi.org/10.1145/2429069.2429113>, <https://doi.org/10.1145/2429069.2429113>

12. Gaboardi, M., Katsumata, S.y., Orchard, D., Breuvar, F., Uustalu, T.: Combining effects and coeffects via grading. *SIGPLAN Not.* **51**(9), 476–489 (Sep 2016). <https://doi.org/10.1145/3022670.2951939>, <https://doi.org/10.1145/3022670.2951939>
13. Ghica, D.R., Smith, A.I.: Bounded linear types in a resource semiring. In: Shao, Z. (ed.) *Programming Languages and Systems*. pp. 331–350. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
14. Girard, J.: Linear logic. *Theor. Comput. Sci.* **50**, 1–102 (1987). [https://doi.org/10.1016/0304-3975\(87\)90045-4](https://doi.org/10.1016/0304-3975(87)90045-4), [https://doi.org/10.1016/0304-3975\(87\)90045-4](https://doi.org/10.1016/0304-3975(87)90045-4)
15. Girard, J., Scedrov, A., Scott, P.J.: Bounded linear logic: A modular approach to polynomial-time computability. *Theor. Comput. Sci.* **97**(1), 1–66 (1992). [https://doi.org/10.1016/0304-3975\(92\)90386-T](https://doi.org/10.1016/0304-3975(92)90386-T), [https://doi.org/10.1016/0304-3975\(92\)90386-T](https://doi.org/10.1016/0304-3975(92)90386-T)
16. Grellois, C., Melliès, P.: An infinitary model of linear logic. In: Pitts, A.M. (ed.) *Foundations of Software Science and Computation Structures - 18th International Conference, FoSSaCS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11–18, 2015. Proceedings. Lecture Notes in Computer Science*, vol. 9034, pp. 41–55. Springer (2015). https://doi.org/10.1007/978-3-662-46678-0_3, https://doi.org/10.1007/978-3-662-46678-0_3
17. Grellois, C., Melliès, P.: Relational semantics of linear logic and higher-order model checking. In: Kreutzer [22], pp. 260–276. <https://doi.org/10.4230/LIPICs.CSL.2015.260>, <https://doi.org/10.4230/LIPICs.CSL.2015.260>
18. Hofmann, M., Scott, P.J.: Realizability models for bll-like languages. *Theor. Comput. Sci.* **318**(1-2), 121–137 (2004). <https://doi.org/10.1016/j.tcs.2003.10.019>, <https://doi.org/10.1016/j.tcs.2003.10.019>
19. Hoshino, N.: Linear realizability. In: Duparc, J., Henzinger, T.A. (eds.) *Computer Science Logic, 21st International Workshop, CSL 2007, 16th Annual Conference of the EACSL, Lausanne, Switzerland, September 11–15, 2007, Proceedings. Lecture Notes in Computer Science*, vol. 4646, pp. 420–434. Springer (2007). https://doi.org/10.1007/978-3-540-74915-8_32, https://doi.org/10.1007/978-3-540-74915-8_32
20. Hyland, M., Power, J.: Pseudo-commutative monads and pseudo-closed 2-categories. *Journal of Pure and Applied Algebra* **175**(1), 141 – 185 (2002). [https://doi.org/https://doi.org/10.1016/S0022-4049\(02\)00133-0](https://doi.org/https://doi.org/10.1016/S0022-4049(02)00133-0), <http://www.sciencedirect.com/science/article/pii/S0022404902001330>, special Volume celebrating the 70th birthday of Professor Max Kelly
21. Katsumata, S.y.: A double category theoretic analysis of graded linear exponential comonads. In: Baier, C., Dal Lago, U. (eds.) *Foundations of Software Science and Computation Structures*. pp. 110–127. Springer International Publishing, Cham (2018)
22. Kreutzer, S. (ed.): *24th EACSL Annual Conference on Computer Science Logic, CSL 2015, September 7–10, 2015, Berlin, Germany, LIPIcs*, vol. 41. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2015), <http://www.dagstuhl.de/dagpub/978-3-939897-90-3>
23. Lago, U.D., Gaboardi, M.: Linear dependent types and relative completeness. In: *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011, June 21–24, 2011, Toronto, Ontario, Canada*. pp. 133–142. IEEE Com-

- puter Society (2011). <https://doi.org/10.1109/LICS.2011.22>, <https://doi.org/10.1109/LICS.2011.22>
24. McBride, C.: I got plenty o' nuttin'. In: Lindley, S., McBride, C., Trinder, P.W., Sannella, D. (eds.) A List of Successes That Can Change the World - Essays Dedicated to Philip Wadler on the Occasion of His 60th Birthday. Lecture Notes in Computer Science, vol. 9600, pp. 207–233. Springer (2016). https://doi.org/10.1007/978-3-319-30936-1_12, https://doi.org/10.1007/978-3-319-30936-1_12
 25. Orchard, D., Liepelt, V.B., Eades III, H.: Quantitative program reasoning with graded modal types. *Proc. ACM Program. Lang.* **3**(ICFP) (Jul 2019). <https://doi.org/10.1145/3341714>, <https://doi.org/10.1145/3341714>
 26. Orchard, D., Wadler, P., Eades, H.: Unifying graded and parameterised monads. *Electronic Proceedings in Theoretical Computer Science* **317**, 18–38 (May 2020). <https://doi.org/10.4204/eptcs.317.2>, <http://dx.doi.org/10.4204/EPTCS.317.2>
 27. Petricek, T., Orchard, D., Mycroft, A.: Coeffects: Unified static analysis of context-dependence. In: Fomin, F.V., Freivalds, R., Kwiatkowska, M., Peleg, D. (eds.) *Automata, Languages, and Programming*. pp. 385–397. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
 28. Reed, J., Pierce, B.C.: Distance makes the types grow stronger: a calculus for differential privacy. In: Hudak, P., Weirich, S. (eds.) *Proceeding of the 15th ACM SIGPLAN international conference on Functional programming, ICFP 2010, Baltimore, Maryland, USA, September 27-29, 2010*. pp. 157–168. ACM (2010). <https://doi.org/10.1145/1863543.1863568>, <https://doi.org/10.1145/1863543.1863568>
 29. Schöpp, U.: Computation-by-interaction with effects. In: Yang, H. (ed.) *Programming Languages and Systems*. pp. 305–321. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)

A Proofs for Section 2 (Generalized Bounded Linear Logic)

In this section, we show reduction steps for reducible Cut eliminations of GBLL and GBAL and the proof size is non-increasing in each reduction step.

Case 1. (Axiom)

$$\frac{\frac{\Delta \vdash A}{\Delta \mid A \vdash A} (\text{Ax}) \quad \frac{\Delta \mid \Gamma, A \vdash B}{\Delta \mid \Gamma \vdash B} (\text{Cut})}{\Delta \mid \Gamma \vdash B} \quad \begin{array}{c} \vdots \pi' \\ \Delta \mid \Gamma, A \vdash B \end{array} \longrightarrow \frac{\Delta \mid \Gamma, A \vdash B}{\Delta \mid \Gamma, A \vdash B} (\text{Cut}) \quad \begin{array}{c} \vdots \pi' \\ \Delta \mid \Gamma, A \vdash B \end{array}$$

$$\frac{\frac{\Delta \mid \Gamma \vdash A}{\Delta \mid \Gamma \vdash A} (\text{Cut}) \quad \frac{\Delta \vdash A}{\Delta \mid A \vdash A} (\text{Ax})}{\Delta \mid \Gamma \vdash A} \quad \begin{array}{c} \vdots \pi' \\ \Delta \mid \Gamma, A \vdash B \end{array} \longrightarrow \frac{\Delta \mid \Gamma, A \vdash B}{\Delta \mid \Gamma, A \vdash B} (\text{Cut}) \quad \begin{array}{c} \vdots \pi' \\ \Delta \mid \Gamma, A \vdash B \end{array}$$

The weight of each proof decreases from $|\pi'| + 1$ to $|\pi'|$.

Case 2. (L \otimes , R \otimes -L)

$$\frac{\frac{\Delta \mid \Gamma_1, X, Y \vdash A}{\Delta \mid \Gamma_1, X \otimes Y \vdash A} (\otimes\text{L}) \quad \frac{\Delta \mid \Gamma_2, A \vdash B}{\Delta \mid \Gamma_1, \Gamma_2, X \otimes Y \vdash B} (\text{Cut})}{\Delta \mid \Gamma_1, \Gamma_2, X \otimes Y \vdash B} \quad \begin{array}{c} \vdots \pi_1 \\ \Delta \mid \Gamma_1, X, Y \vdash A \end{array} \quad \begin{array}{c} \vdots \pi_2 \\ \Delta \mid \Gamma_2, A \vdash B \end{array} \longrightarrow \frac{\frac{\Delta \mid \Gamma_1, X, Y \vdash A}{\Delta \mid \Gamma_1, \Gamma_2, X, Y \vdash B} (\otimes\text{L}) \quad \frac{\Delta \mid \Gamma_2, A \vdash B}{\Delta \mid \Gamma_1, \Gamma_2, X \otimes Y \vdash B} (\otimes\text{L})}{\Delta \mid \Gamma_1, \Gamma_2, X \otimes Y \vdash B} (\text{Cut}) \quad \begin{array}{c} \vdots \pi_1 \\ \Delta \mid \Gamma_1, X, Y \vdash A \end{array} \quad \begin{array}{c} \vdots \pi_2 \\ \Delta \mid \Gamma_2, A \vdash B \end{array}$$

$$\frac{\frac{\Delta \mid \Gamma_1 \vdash A}{\Delta \mid \Gamma_1, \Gamma_2, X \otimes Y \vdash B} (\text{Cut}) \quad \frac{\Delta \mid \Gamma_2, X, Y, A \vdash B}{\Delta \mid \Gamma_2, X \otimes Y, A \vdash B} (\otimes\text{L})}{\Delta \mid \Gamma_1, \Gamma_2, X \otimes Y \vdash B} \quad \begin{array}{c} \vdots \pi_2 \\ \Delta \mid \Gamma_2, X, Y, A \vdash B \end{array} \longrightarrow \frac{\frac{\Delta \mid \Gamma_1 \vdash A}{\Delta \mid \Gamma_1, \Gamma_2, X, Y \vdash B} (\otimes\text{L}) \quad \frac{\Delta \mid \Gamma_2, X, Y, A \vdash B}{\Delta \mid \Gamma_1, \Gamma_2, X \otimes Y \vdash B} (\otimes\text{L})}{\Delta \mid \Gamma_1, \Gamma_2, X \otimes Y \vdash B} (\text{Cut}) \quad \begin{array}{c} \vdots \pi_1 \\ \Delta \mid \Gamma_1 \vdash A \end{array} \quad \begin{array}{c} \vdots \pi_2 \\ \Delta \mid \Gamma_2, X, Y, A \vdash B \end{array}$$

The weight of the proof keeps $|\pi_1| + |\pi_2| + 1$ in this reduction step.

Case 3. (R \otimes -R)

$$\frac{\frac{\Delta \mid \Gamma \vdash A}{\Delta \mid \Gamma, \Gamma'_1, \Gamma'_2 \vdash X \otimes Y} (\text{Cut}) \quad \frac{\Delta \mid \Gamma'_1, A \vdash X}{\Delta \mid \Gamma'_1, \Gamma'_2, A \vdash X \otimes Y} (\otimes\text{R}) \quad \frac{\Delta \mid \Gamma'_2 \vdash Y}{\Delta \mid \Gamma'_1, \Gamma'_2, A \vdash X \otimes Y} (\otimes\text{R})}{\Delta \mid \Gamma, \Gamma'_1, \Gamma'_2 \vdash X \otimes Y} \quad \begin{array}{c} \vdots \kappa_1 \\ \Delta \mid \Gamma'_1, A \vdash X \end{array} \quad \begin{array}{c} \vdots \kappa_2 \\ \Delta \mid \Gamma'_2 \vdash Y \end{array} \longrightarrow \frac{\Delta \mid \Gamma, \Gamma'_1, \Gamma'_2 \vdash X \otimes Y}{\Delta \mid \Gamma, \Gamma'_1, \Gamma'_2 \vdash X \otimes Y} (\text{Cut}) \quad \begin{array}{c} \vdots \pi' \\ \Delta \mid \Gamma \vdash A \end{array} \quad \begin{array}{c} \vdots \kappa_1 \\ \Delta \mid \Gamma'_1, A \vdash X \end{array} \quad \begin{array}{c} \vdots \kappa_2 \\ \Delta \mid \Gamma'_2 \vdash Y \end{array}$$

$$\begin{aligned}
 &\rightarrow \frac{\frac{\frac{\vdash \pi'}{\Delta \mid \Gamma \vdash A} \quad \frac{\vdash \kappa_1}{\Delta \mid \Gamma'_1, A \vdash X}}{\Delta \mid \Gamma, \Gamma'_1 \vdash X} \text{ (Cut)} \quad \frac{\vdash \kappa_2}{\Delta \mid \Gamma'_2 \vdash Y}}{\Delta \mid \Gamma, \Gamma'_1, \Gamma'_2 \vdash X \otimes Y} \text{ } (\otimes R) \\
 &\frac{\frac{\vdash \pi'}{\Delta \mid \Gamma \vdash A} \quad \frac{\frac{\frac{\vdash \kappa_1}{\Delta \mid \Gamma'_1 \vdash X} \quad \frac{\vdash \kappa_2}{\Delta \mid \Gamma'_2, A \vdash Y}}{\Delta \mid \Gamma'_1, \Gamma'_2, A \vdash X \otimes Y} \text{ } (\otimes R)}{\Delta \mid \Gamma, \Gamma'_1, \Gamma'_2 \vdash X \otimes Y} \text{ (Cut)}}{\rightarrow \frac{\frac{\frac{\vdash \pi'}{\Delta \mid \Gamma \vdash A} \quad \frac{\vdash \kappa_2}{\Delta \mid \Gamma'_2, A \vdash X}}{\Delta \mid \Gamma, \Gamma'_2 \vdash X} \text{ (Cut)} \quad \frac{\vdash \kappa_1}{\Delta \mid \Gamma'_1 \vdash Y}}{\Delta \mid \Gamma, \Gamma'_1, \Gamma'_2 \vdash X \otimes Y} \text{ } (\otimes R)}
 \end{aligned}$$

The weight of each proof keeps $|\pi'| + |\kappa_1| + |\kappa_2| + 1$ in this reduction step.

Case 4. (L \multimap)

$$\begin{aligned}
 &\frac{\frac{\frac{\vdash \pi_1}{\Delta \mid \Gamma_1 \vdash X} \quad \frac{\vdash \pi_2}{\Delta \mid \Gamma_2, Y \vdash A}}{\Delta \mid \Gamma_1, \Gamma_2, X \multimap Y \vdash A} \text{ } (\multimap L) \quad \frac{\vdash \kappa}{\Delta \mid \Gamma', A \vdash B}}{\Delta \mid \Gamma_1, \Gamma_2, \Gamma', \Delta, X \multimap Y \vdash B} \text{ (Cut)} \\
 &\rightarrow \frac{\frac{\vdash \pi_1}{\Delta \mid \Gamma_1 \vdash X} \quad \frac{\frac{\frac{\vdash \pi_2}{\Delta \mid \Gamma_2, Y \vdash A} \quad \frac{\vdash \kappa}{\Delta \mid \Gamma', A \vdash B}}{\Delta \mid \Gamma_2, \Gamma', Y \vdash B} \text{ (Cut)}}{\Delta \mid \Gamma_1, \Gamma_2, \Gamma', X \multimap Y \vdash A} \text{ } (\multimap L)}
 \end{aligned}$$

The weight of the proof keeps $|\pi_1| + |\pi_2| + |\kappa| + 1$ in this reduction step.

Case 5. (R \multimap -L)

$$\begin{aligned}
 &\frac{\frac{\frac{\vdash \pi'}{\Delta \mid \Gamma \vdash A} \quad \frac{\frac{\frac{\vdash \kappa_1}{\Delta \mid \Gamma'_1, A \vdash X} \quad \frac{\vdash \kappa_2}{\Delta \mid \Gamma'_2, Y \vdash B}}{\Delta \mid \Gamma'_1, \Gamma'_2, A, X \multimap Y \vdash B} \text{ } (\multimap L)}{\Delta \mid \Gamma, \Gamma'_1, \Gamma'_2, X \multimap Y \vdash B} \text{ (Cut)}}{\rightarrow \frac{\frac{\frac{\vdash \pi'}{\Delta \mid \Gamma \vdash A} \quad \frac{\vdash \kappa_1}{\Delta \mid \Gamma'_1, A \vdash X}}{\Delta \mid \Gamma, \Gamma'_1 \vdash X} \text{ (Cut)} \quad \frac{\vdash \kappa_2}{\Delta \mid \Gamma'_2, Y \vdash B}}{\Delta \mid \Gamma, \Gamma'_1, \Gamma'_2, X \multimap Y \vdash B} \text{ } (\multimap L)} \\
 &\frac{\frac{\vdash \pi'}{\Delta \mid \Gamma \vdash A} \quad \frac{\frac{\frac{\frac{\vdash \kappa_1}{\Delta \mid \Gamma'_1 \vdash X} \quad \frac{\vdash \kappa_2}{\Delta \mid \Gamma'_2, A, Y \vdash B}}{\Delta \mid \Gamma'_1, \Gamma'_2, A, X \multimap Y \vdash B} \text{ } (\multimap L)}{\Delta \mid \Gamma, \Gamma'_1, \Gamma'_2, X \multimap Y \vdash B} \text{ (Cut)}}
 \end{aligned}$$

$$\rightarrow \frac{\frac{\frac{\dot{\kappa}_1}{\Delta \mid \Gamma'_1 \vdash X} \quad \frac{\frac{\dot{\pi}'}{\Delta \mid \Gamma \vdash A} \quad \frac{\dot{\kappa}_2}{\Delta \mid \Gamma'_2, A, Y \vdash B}}{\Delta \mid \Gamma, \Gamma'_2, Y \vdash B} \text{ (Cut)}}{\Delta \mid \Gamma, \Gamma'_1, \Gamma'_2, X \multimap Y \vdash B} \text{ (-}\circ\text{L)}}{\Delta \mid \Gamma, \Gamma'_1, \Gamma'_2, X \multimap Y \vdash B} \text{ (-}\circ\text{L)}$$

The weight of each proof keeps $|\pi'| + |\kappa_1| + |\kappa_2| + 1$ in this reduction step.

Case 6. (R \multimap -R)

$$\frac{\frac{\frac{\dot{\pi}_1}{\Delta \mid \Gamma_1 \vdash A} \quad \frac{\dot{\pi}_2}{\Delta \mid \Gamma_2, A, X \vdash Y}}{\Delta \mid \Gamma_2, A \vdash X \multimap Y} \text{ (-}\circ\text{R)}}{\Delta \mid \Gamma_1, \Gamma_2 \vdash X \multimap Y} \text{ (Cut)}}{\Delta \mid \Gamma_1, \Gamma_2 \vdash X \multimap Y} \text{ (-}\circ\text{R)}$$

$$\rightarrow \frac{\frac{\frac{\dot{\pi}_1}{\Delta \mid \Gamma_1 \vdash A} \quad \frac{\dot{\pi}_2}{\Delta \mid \Gamma_2, A, X \vdash Y}}{\Delta \mid \Gamma_1, \Gamma_2, X \vdash Y} \text{ (Cut)}}{\Delta \mid \Gamma_1, \Gamma_2 \vdash X \multimap Y} \text{ (-}\circ\text{R)}$$

The weight of the proof keeps $|\pi_1| + |\pi_2| + 1$ in this reduction step.

Case 7. (S \otimes)

$$\frac{\frac{\frac{\dot{\pi}_1}{\Delta \mid \Gamma_1 \vdash X} \quad \frac{\dot{\pi}_2}{\Delta \mid \Gamma_2 \vdash Y}}{\Delta \mid \Gamma_1, \Gamma_2 \vdash X \otimes Y} \text{ (}\otimes\text{R)}}{\Delta \mid \Gamma_1, \Gamma_2, \Delta \vdash B} \text{ (Cut)} \quad \frac{\frac{\dot{\kappa}}{\Delta \mid \Gamma', X, Y \vdash B}}{\Delta \mid \Gamma', X \otimes Y \vdash B} \text{ (}\otimes\text{L)}}{\Delta \mid \Gamma_1, \Gamma_2, \Delta \vdash B} \text{ (Cut)}}{\Delta \mid \Gamma_1, \Gamma_2 \vdash B} \text{ (Cut)}$$

$$\rightarrow \frac{\frac{\frac{\dot{\pi}_1}{\Delta \mid \Gamma_1 \vdash X} \quad \frac{\dot{\pi}_2}{\Delta \mid \Gamma_2 \vdash Y}}{\Delta \mid \Gamma_2, \Delta, X \vdash B} \text{ (Cut)}}{\Delta \mid \Gamma_1, \Gamma_2 \vdash B} \text{ (Cut)}}{\Delta \mid \Gamma_1, \Gamma_2 \vdash B} \text{ (Cut)}$$

The weight of the proof decreases from $|\pi_1| + |\pi_2| + |\kappa| + 2$ to $|\pi_1| + |\pi_2| + |\kappa|$.

Case 8. (S \multimap)

$$\frac{\frac{\frac{\dot{\pi}'}{\Delta \mid \Gamma, X \vdash Y}}{\Delta \mid \Gamma \vdash X \multimap Y} \text{ (-}\circ\text{R)}}{\Delta \mid \Gamma, \Gamma'_1, \Gamma'_2 \vdash B} \text{ (Cut)} \quad \frac{\frac{\frac{\dot{\kappa}_1}{\Delta \mid \Gamma'_1 \vdash X} \quad \frac{\dot{\kappa}_2}{\Delta \mid \Gamma'_2, Y \vdash B}}{\Delta \mid \Gamma'_1, \Gamma'_2, X \multimap Y \vdash B} \text{ (-}\circ\text{L)}}{\Delta \mid \Gamma, \Gamma'_1, \Gamma'_2 \vdash B} \text{ (Cut)}}{\Delta \mid \Gamma, \Gamma'_1, \Gamma'_2 \vdash B} \text{ (Cut)}$$

$$\rightarrow \frac{\frac{\frac{\dot{\kappa}_1}{\Delta \mid \Gamma'_1 \vdash X} \quad \frac{\frac{\dot{\pi}'}{\Delta \mid \Gamma, X \vdash Y} \quad \frac{\dot{\kappa}_2}{\Delta \mid \Gamma'_2, Y \vdash B}}{\Delta \mid \Gamma, \Gamma'_2, X \vdash B} \text{ (Cut)}}{\Delta \mid \Gamma, \Gamma'_1, \Gamma'_2 \vdash B} \text{ (Cut)}}{\Delta \mid \Gamma, \Gamma'_1, \Gamma'_2 \vdash B} \text{ (Cut)}$$

The weight of the proof decreases from $|\pi'| + |\kappa_1| + |\kappa_2| + 2$ to $|\pi'| + |\kappa_1| + |\kappa_2|$.

Case 9. (W)

$$\begin{array}{c}
 \frac{\frac{\frac{\Delta \mid \Gamma_1 \quad \vdash A}{\vdash \pi_1} \text{ (!W)}}{\Delta \mid \Gamma_1, !_0 X \vdash A} \quad \frac{\Delta \mid \Gamma_2, A \vdash B}{\vdash \pi_2} \text{ (Cut)}}{\Delta \mid \Gamma_1, \Gamma_2, !_0 X \vdash B} \\
 \longrightarrow \frac{\frac{\frac{\Delta \mid \Gamma_1 \vdash A \quad \Delta \mid \Gamma_2, A \vdash B}{\vdash \pi_1} \text{ (Cut)}}{\Delta \mid \Gamma_1, \Gamma_2 \quad \vdash B} \text{ (!W)}}{\Delta \mid \Gamma_1, \Gamma_2, !_0 X \vdash B} \\
 \\
 \frac{\frac{\frac{\Delta \mid \Gamma_1 \vdash A \quad \frac{\frac{\Delta \mid \Gamma_2 \quad, A \vdash B}{\vdash \pi_2} \text{ (!W)}}{\Delta \mid \Gamma_2, !_0 X, A \vdash B} \text{ (Cut)}}{\Delta \mid \Gamma_1, \Gamma_2, !_0 X \vdash B} \\
 \longrightarrow \frac{\frac{\frac{\Delta \mid \Gamma_1 \vdash A \quad \Delta \mid \Gamma_2, A \vdash B}{\vdash \pi_1} \text{ (Cut)}}{\Delta \mid \Gamma_1, \Gamma_2 \quad \vdash B} \text{ (!W)}}{\Delta \mid \Gamma_1, \Gamma_2, !_0 X \vdash B}
 \end{array}$$

The weight of the proof keeps $|\pi_1| + |\pi_2| + 1$ in this reduction step.

Case 10. (D)

$$\begin{array}{c}
 \frac{\frac{\frac{\Delta \mid \Gamma_1, \quad X \vdash A}{\vdash \pi_1} \text{ (!D)}}{\Delta \mid \Gamma_1, !_{\text{id}} X \vdash A} \quad \frac{\Delta \mid \Gamma_2, A \vdash B}{\vdash \pi_2} \text{ (Cut)}}{\Delta \mid \Gamma_1, \Gamma_2, !_{\text{id}} X \vdash B} \\
 \longrightarrow \frac{\frac{\frac{\Delta \mid \Gamma_1 \vdash A \quad \Delta \mid \Gamma_2, X, A \vdash B}{\vdash \pi_1} \text{ (Cut)}}{\Delta \mid \Gamma_1, \Gamma_2, \quad X \vdash B} \text{ (!D)}}{\Delta \mid \Gamma_1, \Gamma_2, !_{\text{id}} X \vdash B} \\
 \\
 \frac{\frac{\frac{\Delta \mid \Gamma_1 \vdash A \quad \frac{\frac{\Delta \mid \Gamma_2, \quad X, A \vdash B}{\vdash \pi_2} \text{ (!D)}}{\Delta \mid \Gamma_2, !_{\text{id}} X, A \vdash B} \text{ (Cut)}}{\Delta \mid \Gamma_1, \Gamma_2, !_{\text{id}} X \vdash B} \\
 \longrightarrow \frac{\frac{\frac{\Delta \mid \Gamma_1, X \vdash A \quad \Delta \mid \Gamma_2, A \vdash B}{\vdash \pi_1} \text{ (Cut)}}{\Delta \mid \Gamma_1, \Gamma_2, \quad X \vdash B} \text{ (!D)}}{\Delta \mid \Gamma_1, \Gamma_2, !_{\text{id}} X \vdash B}
 \end{array}$$

The weight of the proof keeps $|\pi_1| + |\pi_2| + 1$ in this reduction step.

Case 11. (F)

$$\begin{array}{c}
\frac{\frac{\frac{\Delta \mid \Gamma_1, !_g X \vdash A}{\Delta \mid \Gamma_1, !_f X \vdash A} (\text{!F})}{\Delta \mid \Gamma_1, \Gamma_2, !_f X \vdash B} (\text{Cut})}{\Delta \mid \Gamma_1, \Gamma_2, !_f X \vdash B} \\
\rightarrow \frac{\frac{\frac{\frac{\Delta \mid \Gamma_1, !_g X \vdash A}{\Delta \mid \Gamma_1, \Gamma_2, !_g X \vdash B} (\text{!F})}{\Delta \mid \Gamma_1, \Gamma_2, !_f X \vdash B} (\text{!F})}{\Delta \mid \Gamma_1, \Gamma_2, !_f X \vdash B} (\text{Cut})}{\Delta \mid \Gamma_1, \Gamma_2, !_f X \vdash B} \\
\rightarrow \frac{\frac{\frac{\frac{\Delta \mid \Gamma_2, !_g X, A \vdash B}{\Delta \mid \Gamma_2, !_f X, A \vdash B} (\text{!F})}{\Delta \mid \Gamma_1, \Gamma_2, !_f X \vdash B} (\text{Cut})}{\Delta \mid \Gamma_1, \Gamma_2, !_f X \vdash B} \\
\rightarrow \frac{\frac{\frac{\frac{\Delta \mid \Gamma_1 \vdash A}{\Delta \mid \Gamma_1, \Gamma_2, !_g X \vdash B} (\text{!F})}{\Delta \mid \Gamma_1, \Gamma_2, !_f X \vdash B} (\text{!F})}{\Delta \mid \Gamma_1, \Gamma_2, !_f X \vdash B} (\text{Cut})}{\Delta \mid \Gamma_1, \Gamma_2, !_f X \vdash B}
\end{array}$$

The weight of the proof keeps $|\pi_1| + |\pi_2| + 1$ in this reduction step.

Case 12. (C)

$$\begin{array}{c}
\frac{\frac{\frac{\frac{\Delta \mid \Gamma_1, !_{f_1} X, !_{f_2} X \vdash A}{\Delta \mid \Gamma_1, !_{f_1+f_2} X \vdash A} (\text{!C})}{\Delta \mid \Gamma_1, \Gamma_2, !_{f_1+f_2} X \vdash B} (\text{Cut})}{\Delta \mid \Gamma_1, \Gamma_2, !_{f_1+f_2} X \vdash B} \\
\rightarrow \frac{\frac{\frac{\frac{\frac{\Delta \mid \Gamma_1, !_{f_1} X, !_{f_2} X \vdash A}{\Delta \mid \Gamma_1, \Gamma_2, !_{f_1} X, !_{f_2} X \vdash B} (\text{!C})}{\Delta \mid \Gamma_1, \Gamma_2, !_{f_1+f_2} X \vdash B} (\text{!C})}{\Delta \mid \Gamma_1, \Gamma_2, !_{f_1+f_2} X \vdash B} (\text{Cut})}{\Delta \mid \Gamma_1, \Gamma_2, !_{f_1+f_2} X \vdash B} \\
\rightarrow \frac{\frac{\frac{\frac{\frac{\Delta \mid \Gamma_2, !_{f_1} X, !_{f_2} X, A \vdash B}{\Delta \mid \Gamma_2, !_{f_1+f_2} X, A \vdash B} (\text{!C})}{\Delta \mid \Gamma_1, \Gamma_2, !_{f_1+f_2} X \vdash B} (\text{Cut})}{\Delta \mid \Gamma_1, \Gamma_2, !_{f_1+f_2} X \vdash B} \\
\rightarrow \frac{\frac{\frac{\frac{\frac{\Delta \mid \Gamma_2, !_{f_1} X, !_{f_2} X, A \vdash B}{\Delta \mid \Gamma_1, \Gamma_2, !_{f_1} X, !_{f_2} X \vdash B} (\text{!C})}{\Delta \mid \Gamma_1, \Gamma_2, !_{f_1+f_2} X \vdash B} (\text{!C})}{\Delta \mid \Gamma_1, \Gamma_2, !_{f_1+f_2} X \vdash B} (\text{Cut})}{\Delta \mid \Gamma_1, \Gamma_2, !_{f_1+f_2} X \vdash B}
\end{array}$$

The weight of the proof keeps $|\pi_1| + |\pi_2| + 2$ in this reduction step.

Case 13. (!W)

$$\frac{\frac{\frac{\dot{\pi}_1}{\Delta' \mid \emptyset \vdash A} \text{ (P!)} \quad \frac{\frac{\dot{\pi}_2}{\Delta \mid \Gamma \vdash B} \text{ (!W)}}{\Delta \mid \Gamma, !_0 A \vdash B} \text{ (Cut)}}{\Delta \mid \emptyset \vdash !_0 A} \text{ (P!)} \quad \frac{\dot{\pi}_2}{\Delta \mid \Gamma \vdash B}}{\Delta \mid \Gamma \vdash B} \text{ (Cut)} \longrightarrow \frac{\dot{\pi}_2}{\Delta \mid \Gamma \vdash B}$$

The weight of the proof decreases from $|\pi_2| + 2$ to $|\pi_2|$ in this reduction step.

Case 14. (!D)

$$\frac{\frac{\frac{\dot{\pi}_1}{\Delta \mid \emptyset \vdash A} \text{ (P!)} \quad \frac{\frac{\dot{\pi}_2}{\Delta \mid \Gamma, A \vdash B} \text{ (!D)}}{\Delta \mid \Gamma, !_\text{id} A \vdash B} \text{ (Cut)}}{\Delta \mid \emptyset \vdash !_\text{id} A} \text{ (P!)} \quad \frac{\dot{\pi}_2}{\Delta \mid \Gamma, !_\text{id} A \vdash B}}{\Delta \mid \Gamma \vdash B} \text{ (Cut)} \longrightarrow \frac{\frac{\dot{\pi}_1}{\Delta \mid \emptyset \vdash A} \quad \frac{\dot{\pi}_2}{\Delta \mid \Gamma, A \vdash B}}{\Delta \mid \Gamma \vdash B} \text{ (Cut)}$$

The weight of the proof decreases from $|\pi_1| + |\pi_2| + 3$ to $|\pi_1| + |\pi_2|$.

Case 15. (!F) Let $f, g \in \mathbf{Idx}(\Delta, \Delta')$ and there is $\sigma \in \mathbf{Idx}(\Delta, \Delta')(f, g)$ (resp. $f, g \in \mathbf{Idx}_a(\Delta, \Delta')$ and $\sigma \in \mathbf{Idx}_a(\Delta, \Delta')(f, g)$)

$$\frac{\frac{\frac{\dot{\pi}_1}{\Delta' \mid \emptyset \vdash A} \text{ (P!)} \quad \frac{\frac{\dot{\pi}_2}{\Delta \mid \Gamma, !_g A \vdash B} \text{ (!F)}}{\Delta \mid \Gamma, !_f A \vdash B} \text{ (Cut)}}{\Delta \mid \emptyset \vdash !_f A} \text{ (P!)} \quad \frac{\dot{\pi}_2}{\Delta \mid \Gamma, !_f A \vdash B}}{\Delta \mid \Gamma \vdash B} \text{ (Cut)} \longrightarrow \frac{\frac{\frac{\dot{\pi}_1}{\Delta' \mid \emptyset \vdash A} \text{ (P!)} \quad \frac{\dot{\pi}_2}{\Delta \mid \Gamma, !_g A \vdash B}}{\Delta \mid \emptyset \vdash !_g A} \text{ (P!)} \quad \frac{\dot{\pi}_2}{\Delta \mid \Gamma, !_g A \vdash B}}{\Delta \mid \Gamma \vdash B} \text{ (Cut)}$$

When $\sigma : f \rightarrow g$ is a morphism in $\mathbf{Idx}(\Delta, \Delta')$, $f(\delta)$ and $g(\delta)$ (for $\delta \in \Delta$) are just permutation of each other. Therefore $\sum_{\gamma \in f(\delta)}$ and $\sum_{\gamma \in g(\delta)}$ are the same. On the other hand, when σ is in $\mathbf{Idx}_a(\Delta, \Delta')$, it holds $\sum_{\gamma \in f(\delta)} \geq \sum_{\gamma \in g(\delta)}$ from the definition.

The weight of proof (for $\delta \in \Delta$) decreases from $\sum_{\gamma \in g(\delta)} (|\pi_1|(\gamma) + 1) + |\pi_2|(\delta) + 2$ to $\sum_{\gamma \in f(\delta)} (|\pi_1|(\gamma) + 1) + |\pi_2|(\delta) + 1$.

Case 16. (!C)

$$\frac{\frac{\frac{\dot{\pi}_1}{\Delta' \mid \emptyset \vdash A} \text{ (P!)} \quad \frac{\frac{\dot{\pi}_2}{\Delta \mid \Gamma, !_f A, !_f A \vdash B} \text{ (!C)}}{\Delta \mid \Gamma, !_f A \vdash B} \text{ (Cut)}}{\Delta \mid \emptyset \vdash !_f A} \text{ (P!)} \quad \frac{\dot{\pi}_2}{\Delta \mid \Gamma, !_f A \vdash B}}{\Delta \mid \Gamma \vdash B} \text{ (Cut)} \longrightarrow \frac{\frac{\dot{\pi}_1}{\Delta' \mid \emptyset \vdash A} \quad \frac{\dot{\pi}_2}{\Delta \mid \Gamma, !_f A \vdash B}}{\Delta \mid \Gamma \vdash B} \text{ (Cut)}$$

$$\frac{\frac{\frac{\vdots\pi_1}{\Delta' \mid \emptyset \vdash A} \text{ (P!)} \quad \frac{\frac{\vdots\pi_1}{\Delta' \mid \emptyset \vdash A} \text{ (P!)} \quad \frac{\vdots\pi_2}{\Delta \mid \Gamma, !_{f_1} A, !_{f_2} A \vdash B} \text{ (Cut)}}{\Delta \mid \emptyset \vdash !_{f_2} A} \text{ (P!)} \quad \frac{\Delta \mid \Gamma, !_{f_1} A \vdash B}{\Delta \mid \Gamma, !_{f_1} A \vdash B} \text{ (Cut)}}{\Delta \mid \Gamma \vdash B} \text{ (Cut)}$$

The weight of the proof for $\delta \in \Delta$ decreases from $\sum_{\gamma \in f_1 + f_2(\delta)} (|\pi_1|(\gamma) + 1) + |\pi_2|(\delta) + 2$ to $\sum_{\gamma \in f_1(\delta)} (|\pi_1|(\gamma) + 1) + \sum_{\gamma \in f_2(\delta)} (|\pi_1|(\gamma) + 1) + |\pi_2|(\delta)$ in this reduction step.

Case 17. (IP)

$$\frac{\frac{\frac{\vdots\pi_1}{\Delta'' \mid \emptyset \vdash A} \text{ (P!)} \quad \frac{\frac{\vdots\pi_2}{\Delta' \mid \dots, !_{\beta_\ell} B_\ell, !_g A \vdash B} \text{ (P!)} \quad \frac{\vdots\pi_2}{\Delta \mid \dots, !_{\beta_\ell \bullet f} B_\ell, !_{g \bullet f} A \vdash !_f B} \text{ (P!)}}{\Delta \mid \emptyset \vdash !_{g \bullet f} A} \text{ (P!)} \quad \frac{\Delta' \mid \dots, !_{\beta_\ell \bullet f} B_\ell, !_{g \bullet f} A \vdash !_f B}{\Delta \mid \dots, !_{\beta_\ell \bullet f} B_\ell \vdash !_f B} \text{ (Cut)}}{\Delta \mid \dots, !_{\beta_\ell \bullet f} B_\ell \vdash !_f B} \text{ (Cut)}$$

$$\longrightarrow \frac{\frac{\frac{\vdots\pi_1}{\Delta'' \mid \emptyset \vdash A} \text{ (P!)} \quad \frac{\vdots\pi_2}{\Delta' \mid \dots, !_{\beta_\ell} B_\ell, !_g A \vdash B} \text{ (Cut)}}{\Delta' \mid \emptyset \vdash !_g A} \text{ (P!)} \quad \frac{\Delta' \mid \dots, !_{\beta_\ell} B_\ell, !_g A \vdash B}{\Delta' \mid \dots, !_{\beta_\ell} B_\ell \vdash B} \text{ (P!)}}{\Delta \mid \dots, !_{\beta_\ell \bullet f} B_\ell \vdash !_f B} \text{ (P!)} \text{ (Cut)}$$

For each $\delta \in \Delta$, the weight of proof decreases from

$$\sum_{\gamma \in g \bullet f(\delta)} (|\pi_1|(\gamma) + 1) + \sum_{\gamma' \in f(\delta)} (|\pi_2|(\gamma') + 2(\ell + 1) + 1) + \ell + 3$$

to

$$\sum_{\gamma \in f(\delta)} \left(\sum_{\gamma' \in g(\gamma)} (|\pi_1|(\gamma') + 1) + |\pi_2|(\gamma) + 2\ell + 2 \right) + \ell + 1$$

B Proofs for Section 3 (Translation from Constrained BLL)

Proposition B.1. *For a well-formed formula A of QBLL, $V_{\mathcal{C}}(F) \vdash [A]^{(F; \mathcal{C})}$.*

Proof. Recall we supposed it exists an atomic $[a] \in \mathcal{A}(\mathbb{N}^n)$ of GBLL for each atomic a of QBLL with arity $n (= \text{ar}(a))$.

1. For an atomic $a(p_1, \dots, p_n)$, a map from $V_{\mathcal{C}}(F) \rightarrow \mathbb{N}^n$ is given as the restriction of $\langle p_1, \dots, p_n \rangle$ to $V_{\mathcal{C}}(F)$. Soon we have $V_{\mathcal{C}}(F) \vdash [a] \star \langle p_1, \dots, p_n \rangle|_{V_{\mathcal{C}}(F)}$.
2. For a formula $A \otimes B$, from hypothesis of induction, we obtain $V_{\mathcal{C}}(F) \vdash [A]^{(F; \mathcal{C})}$ and $V_{\mathcal{C}}(F) \vdash [B]^{(F; \mathcal{C})}$. Then we have $V_{\mathcal{C}}(F) \vdash [A \otimes B]^{(F; \mathcal{C})}$.
3. For a formula $A \multimap B$, we obtain $V_{\mathcal{C}}(F) \vdash [A \multimap B]^{(F; \mathcal{C})}$ from $V_{\mathcal{C}}(F) \vdash [A]^{(F; \mathcal{C})}$ and from $V_{\mathcal{C}}(F) \vdash [B]^{(F; \mathcal{C})}$.
4. For a formula $!_{x < p} A$, we obtain $V_{\mathcal{C} \cup \{x < p\}}(F \cup \{x\}) \vdash [A]^{(F \cup \{x\}; \mathcal{C} \cup \{x < p\})}$ from hypothesis of induction. Then we have $V_{\mathcal{C}}(F) \vdash !_{[x.p]_{(F, \mathcal{C})}} [A]^{(F \cup \{x\}; \mathcal{C} \cup \{x < p\})}$.

C Proofs for Section 4 (Categorical Semantics for GBL)

C.1 Semantics of GBL

Proposition 4.1. *For any $r : \Delta \rightarrow \Delta'$ and well-formed formula $\Delta' \vdash A$, we have $\llbracket \Delta \vdash A|_r \rrbracket = Cr \llbracket \Delta' \vdash A \rrbracket$.*

Proof. The crucial cases are atomic formulas and ! formulas.

$$\begin{aligned} \llbracket \Delta \vdash a \star (r \circ r') \rrbracket &= C(r \circ r')[a] = CrCr'[a] = Cr \llbracket \Delta' \vdash a \rrbracket \\ \llbracket \Delta \vdash !_{f \bullet Jr} A \rrbracket &= D(f \bullet Jr) \llbracket \Delta' \vdash A \rrbracket = Cr(Df \llbracket \Delta' \vdash A \rrbracket) = Cr \llbracket \Delta \vdash !_f A \rrbracket. \end{aligned}$$

□

Proposition 4.2. $\llbracket \Delta \vdash !_{Jr \bullet f} A \rrbracket = \llbracket \Delta' \vdash !_f(A|_r) \rrbracket$.

Proof. We have the following equality:

$$\llbracket \Delta \vdash !_{Jr \bullet f} A \rrbracket = D(Jr \bullet f)(\llbracket \Delta'' \vdash A \rrbracket) = Df(Cr \llbracket \Delta'' \vdash A \rrbracket) = \llbracket \Delta' \vdash !_f(A|_r) \rrbracket.$$

□

Theorem 4.1. *For a proof $\pi \triangleright \Delta \mid \Gamma \vdash A$, if π has a reducible cut and reduces into π' by a reduction step, then $\llbracket \pi \rrbracket = \llbracket \pi' \rrbracket$ in $C\Delta$.*

Proof. We will discuss according to cases in section A. For the first eight cases, because the category $C\Delta$ is symmetric monoidal closed, we can verify the soundness. Next, the first case of (W), (D), (F) and (C) are trivial. For the rest cases, it is shown in Figure 2.

C.2 Construction of an Indexed Linear Exponential Comonad

In the following proofs, we use some specific natural isomorphisms:

$$\begin{aligned} r_{A,B}^l : [i \in l]A(i) \otimes [i \in l]B(i) &\xrightarrow{\sim} [i \in l](A \dot{\otimes} B)(i) \\ \text{wk}_A : [i \in ()]A(i) &\xrightarrow{\cong} I, \quad \text{ct}_A^{l,l'} : [i \in l + l']A(i) \xrightarrow{\sim} [i \in l]A(i) \otimes [i \in l']A(i) \end{aligned}$$

Proposition 4.3. *For $\Delta \xrightarrow{f} \Delta'$ in Idx and $l = i_1 \cdots i_k \in \mathcal{S}\Delta$, let $f(l)$ denote $f(i_1) \cdots f(i_k)$. Then it holds $\mathbb{T}(f(l), A) \simeq \mathbb{T}(l, \mathbb{T}(f(-), A))$ and this isomorphism is natural for A .*

Proof. Let $f(i_j) = (i_{j1} \cdots i_{jm_j})$. Then

$$\begin{aligned} \mathbb{T}(f(l), A) &\simeq A(i_{11}) \otimes \cdots \otimes A(i_{1m_1}) \otimes \cdots \otimes A(i_{n1}) \otimes \cdots \otimes A(i_{nm_n}) \\ &\simeq \mathbb{T}(f(i_1), A) \otimes \cdots \otimes \mathbb{T}(f(i_n), A) \\ &\simeq \mathbb{T}(l, \mathbb{T}(f(-), A)) \end{aligned}$$

clearly this equation is natural for A .

□

Proposition C.1. For $l = (l_1 \cdots l_k) \in \mathcal{S}\Delta$ and $A \in \Delta \pitchfork \mathbb{C}$, we define a morphism $\gamma_A^l : \mathbb{T}(l, V \circ A) \rightarrow V\mathbb{T}(l, A)$ to be the multiple composition of $m_{A,B}$:

$$VA(l_1) \otimes \cdots \otimes VA(l_k) \rightarrow V(A(l_1) \otimes \cdots \otimes A(l_k)).$$

This morphism is monoidal natural on l and A .

Proof. For every $l \in \mathcal{S}\Delta$ and $A \in \Delta \pitchfork \mathbb{C}$, γ_A^l is natural for A because each morphism $m_{A(l_1) \otimes \cdots \otimes A(l_j), A(l_{j+1})}^V$ is natural. On the other hand, since \mathbb{C} is symmetric monoidal and V is lax monoidal functor, γ_A^l is natural for $l \in \mathcal{S}\Delta$. Moreover from each definition the following diagram commutes hence γ_A^l is monoidal natural for l and A .

$$\begin{array}{ccc} \mathbb{T}(l, V \circ A) \otimes \mathbb{T}(l, V \circ B) & \xrightarrow{\gamma_A^l \otimes \gamma_B^l} & V\mathbb{T}(l, A) \otimes V\mathbb{T}(l, B) \\ \downarrow r_{V \circ A, V \circ B}^l & & \downarrow m_{\mathbb{T}(l, A), \mathbb{T}(l, B)}^V \\ \mathbb{T}(l, (V \circ A) \dot{\otimes} (V \circ B)) & & V(\mathbb{T}(l, A) \otimes \mathbb{T}(l, B)) \\ \downarrow \mathbb{T}(l, m_{A(-), B(-)}^V) & & \downarrow Vr_{A, B}^l \\ \mathbb{T}(l, V \circ (A \dot{\otimes} B)) & \xrightarrow{\gamma_{A \dot{\otimes} B}^l} & V\mathbb{T}(l, A \dot{\otimes} B) \end{array}$$

$$\begin{array}{ccc} \mathbb{T}(l, V \circ A) \otimes \mathbb{T}(l', V \circ A) & \xrightarrow{\gamma_A^l \otimes \gamma_A^{l'}} & V\mathbb{T}(l, A) \otimes V\mathbb{T}(l', A) \\ \downarrow \text{ct}_A^{-1} & & \downarrow m_{\mathbb{T}(l, A), \mathbb{T}(l', A)}^V \\ \mathbb{T}(l+l', V \circ A) & \xrightarrow{\gamma_A^{l+l'}} & V(\mathbb{T}(l, A) \otimes \mathbb{T}(l', A)) \\ & & \downarrow V \text{ct}_A^{-1} \end{array}$$

□

Theorem 4.2. The symmetric colax monoidal functor D defined by (4.1) with monoidal natural transformations ϵ, δ defined by (4.2) and (4.3) determines an ILEC over \mathcal{C} .

Proof. This theorem is proved by Prop. C.2, C.3 and C.4.

Proposition C.2. The following diagrams commute for every $\Delta \xrightarrow{f} \Delta' \xrightarrow{g} \Delta'' \xrightarrow{h} \Delta'''$ in \mathbf{Idx} .

$$\begin{array}{ccc} Df \xrightarrow{\delta_{f, \text{id}}} \text{Did}_\Delta \circ Df & D(h \bullet g \bullet f) \xrightarrow{\delta_{h, g \bullet f}} D(g \bullet f) \circ Dh \\ \downarrow \delta_{\text{id}, f} & \downarrow \delta_{h \bullet g, f} & \downarrow \delta_{g, f \circ Dh} \\ Df \circ \text{Did}_{\Delta'} \xrightarrow{Df \circ \epsilon} Df & Df \circ D(h \bullet g) \xrightarrow{Df \circ \delta_{h, g}} Df \circ Dg \circ Dh \end{array}$$

Proof. In this proof, we write $[i \in l]A(i)$ for $\top(l, A)$. First, for $f : \Delta \rightarrow \Delta'$,

$$\begin{aligned}
 \epsilon_{DfA} \circ \delta_{f, \text{id}; A} : DfA &= [j \in (-)][k \in f(j)]VA(k) \\
 &\xrightarrow{[j \in (-)][k \in f(j)]\delta_{A(k)}} [j \in (-)][k \in f(j)]VVA(k) \\
 &\xrightarrow{[j \in (-)]\gamma_{V \circ A}^{f(j)}} [j \in (-)]V[k \in f(j)]VA(k) \\
 &\xrightarrow{\epsilon_{DfA}} DfA \\
 Df(\epsilon_A) \circ \delta_{\text{id}, f; A} : DfA &= [j \in f(-)][k \in (j)]VA(k) \\
 &\xrightarrow{[j \in f(-)][k \in (j)]\delta_{A(k)}} [j \in f(-)][k \in (j)]VVA(k) \\
 &\xrightarrow{[j \in f(-)]\gamma_{V \circ A}^{(j)}} [j \in f(-)]V[k \in (j)]VA(k) \\
 &\xrightarrow{\sim} [j \in f(-)]V(D\text{id}_{\Delta'}A(j)) \\
 &\xrightarrow{[j \in f(-)]V\epsilon_{A; j}} [j \in f(-)]VA(j) \\
 &\xrightarrow{\sim} DfA
 \end{aligned}$$

Then it holds $(\epsilon \circ Df)(\delta_{f, \text{id}}) = (Df \circ \epsilon)(\delta_{\text{id}, f}) = \text{id}_{Df}$. On the other hand, for $\Delta \xrightarrow{f} \Delta' \xrightarrow{g} \Delta'' \xrightarrow{h} \Delta'''$, each composite are shown that:

$$\begin{aligned}
 \delta_{g, f; DhA} \circ \delta_{h, g \bullet f} : D(h \bullet g \bullet f)A &\xrightarrow{\sim} [k \in (g \bullet f)(-)][l \in h(k)]VA(l) \\
 &\xrightarrow{[k \in (g \bullet f)(-)][l \in h(k)]\delta_{A(l)}} [k \in (g \bullet f)(-)][l \in h(k)]VVA(l) \\
 &\xrightarrow{[k \in (g \bullet f)(-)]\gamma_{V \circ A}^{h(k)}} [k \in (g \bullet f)(-)]V[l \in h(k)]VA(l) \\
 &\xrightarrow{\sim} [j \in f(-)][k \in g(j)]V(DhA)(k) \\
 &\xrightarrow{[j \in f(-)][k \in g(j)]\delta_{DhA(k)}} [j \in f(-)][k \in g(j)]VV(DhA)(k) \\
 &\xrightarrow{[j \in (-)]\gamma_{V \circ (DhA)}^{g(j)}} [j \in f(-)]V[k \in g(j)]V(DhA)(k) \\
 &= Df(Dg(DhA)) \\
 Df\delta_{h, g; A} \circ \delta_{h \bullet g, f; A} : D(h \bullet g \bullet f)A &\xrightarrow{\sim} [j \in f(-)][l \in (h \bullet g)(j)]VA(l) \\
 &\xrightarrow{[j \in f(-)][l \in (h \bullet g)(j)]\delta_{A(l)}} [j \in f(-)][l \in (h \bullet g)(j)]VVA(l) \\
 &\xrightarrow{[j \in f(-)]\gamma_{V \circ A}^{(h \bullet g)(j)}} [j \in f(-)]V[l \in (h \bullet g)(j)]VA(l) \\
 &\xrightarrow{\sim} Df([k \in g(-)][l \in h(k)]VA(l)) \\
 &\xrightarrow{Df([k \in g(-)][l \in h(k)]\delta_{A(l)})} Df([k \in g(-)][l \in h(k)]VVA(l)) \\
 &\xrightarrow{Df([k \in g(-)]\gamma_{V \circ A}^{h(k)})} Df([k \in g(-)]V[l \in h(k)]VA(l)) \\
 &= Df(Dg(DhA))
 \end{aligned}$$

Therefore $\delta_{g, f}\delta_{h, g \bullet f} = (Df \circ \delta_{h, g})\delta_{h \bullet g, f}$. \square

Lemma C.1. For $f, g \in \mathbf{Idx}(\Delta, \Delta')$, $l, l' \in S\Delta$ and for $A \in \Delta'$, the following diagrams commute:

$$\begin{array}{ccccc}
\mathbb{T}(f+l+g(l), A) & \xlongequal{\quad} & \mathbb{T}((f+g)(l), A) & & \mathbb{T}(\cdot, A) \xlongequal{\quad} \mathbb{T}(0(l), A) \\
\downarrow & & \downarrow & & \downarrow & \downarrow \\
\mathbb{T}(f(l), A) \otimes \mathbb{T}(g(l), A) & & \mathbb{T}(l, \mathbb{T}((f+g)(-), A)) & & \mathbb{T}(l, \mathbb{T}(0(-), A)) \\
\downarrow & & \downarrow & & \downarrow & \downarrow \\
\mathbb{T}(l, \mathbb{T}(f(-), A)) \otimes \mathbb{T}(l, \mathbb{T}(g(-), A)) & \longrightarrow & \mathbb{T}(l, \mathbb{T}(f(-), A) \dot{\otimes} \mathbb{T}(g(-), A)) & & I \longrightarrow \mathbb{T}(l, l)
\end{array}$$

$$\begin{array}{ccccc}
\mathbb{T}(f(l)+f(l'), A) & \xlongequal{\quad} & \mathbb{T}(f(l+l'), A) & & \mathbb{T}(\cdot, A) \xlongequal{\quad} \mathbb{T}(f(\cdot), A) \\
\downarrow & & \downarrow & & \downarrow & \downarrow \\
\mathbb{T}(f(l), A) \otimes \mathbb{T}(f(l'), A) & & & & \mathbb{T}(\cdot, \mathbb{T}(f(-), A)) \\
\downarrow & & \downarrow & & \downarrow & \downarrow \\
\mathbb{T}(l, \mathbb{T}(f(-), A)) \otimes \mathbb{T}(l', \mathbb{T}(f(-), A)) & \longrightarrow & \mathbb{T}(l+l', \mathbb{T}(f(-), A)) & & I \xlongequal{\quad} I
\end{array}$$

Proof. It follows from the definition of folding product. \square

Proposition C.3. The following diagrams commute:

$$\begin{array}{ccccc}
D(f \bullet h + g \bullet h)A & \xlongequal{\quad} & D((f+g) \bullet h)A & & D0A \xlongequal{\quad} D(0 \bullet h)A \\
\downarrow & & \downarrow & & \downarrow & \downarrow \\
D(f \bullet h)A \dot{\otimes} D(g \bullet h)A & & Dh(D(f+g)A) & & Dh(D0A) \\
\downarrow & & \downarrow & & \downarrow & \downarrow \\
Dh(DfA) \dot{\otimes} Dh(DgA) & \longrightarrow & Dh(DfA \dot{\otimes} DgA) & & I \longrightarrow DhI
\end{array}$$

$$\begin{array}{ccccc}
D(h \bullet f + h \bullet g)A & \longrightarrow & D(h \bullet (f+g))A & & D0A \xlongequal{\quad} D(h \bullet 0)A \\
\downarrow & & \downarrow & & \downarrow & \downarrow \\
D(h \bullet f)A \dot{\otimes} D(h \bullet g)A & & D(f+g)(DhA) & & D0(DhA) \\
\downarrow & & \downarrow & & \downarrow & \downarrow \\
Df(DhA) \dot{\otimes} Dg(DhA) & \xlongequal{\quad} & (Df \dot{\otimes} Dg)(DhA) & & I \xlongequal{\quad} I
\end{array}$$

Proof. From lemma C.1, we obtain the following diagrams for each i :

$$\begin{array}{ccccc}
D(f \bullet h + g \bullet h)A(i) & \xlongequal{\quad} & D((f+g) \bullet h)A(i) & & \\
\downarrow & & \downarrow & & \\
D(f \bullet h)A(i) \otimes D(g \bullet h)A(i) & & \mathbb{T}(h(i), D(f+g)A) & \longrightarrow & Dh(D(f+g)A)(i) \\
\downarrow & & \downarrow & & \downarrow \\
\mathbb{T}(h(i), DfA) \otimes \mathbb{T}(h(i), DgA) & \longrightarrow & \mathbb{T}(h(i), DfA \dot{\otimes} DgA) & & \\
\downarrow & & \searrow & & \downarrow \\
Dh(DfA)(i) \otimes Dh(DgA)(i) & \longrightarrow & Dh(DfA \dot{\otimes} DgA)(i) & &
\end{array}$$

$$\begin{array}{c}
D0A(i) \equiv D(0 \bullet h)A(i) \\
\downarrow \qquad \qquad \downarrow \\
I \longrightarrow \mathbb{T}(h(i), l) \longrightarrow Dh l \\
\qquad \qquad \qquad \downarrow \qquad \qquad \downarrow \\
\qquad \qquad \qquad \mathbb{T}(h(i), D0A) \longrightarrow Dh(D0A)(i) \\
\qquad \qquad \qquad \downarrow \\
\qquad \qquad \qquad Dh l
\end{array}$$

$$\begin{array}{c}
D(h \bullet f + h \bullet g)A(i) \equiv D(h \bullet (f+g))A(i) \\
\downarrow \qquad \qquad \qquad \downarrow \\
D(h \bullet f)A(i) \otimes D(g \bullet f)A(i) \qquad \qquad \qquad D(h \bullet 0)A(i) \\
\downarrow \qquad \qquad \qquad \downarrow \qquad \qquad \qquad \downarrow \\
\mathbb{T}(f(i), DhA) \otimes \mathbb{T}(g(i), DhA) \longrightarrow \mathbb{T}(f(i)+g(i), DhA) \qquad \qquad \qquad \mathbb{T}(0(i), DhA) \longrightarrow D0(DhA)(i) \\
\downarrow \qquad \qquad \qquad \downarrow \qquad \qquad \qquad \downarrow \\
Df(DhA)(i) \otimes Dg(DhA)(i) \longrightarrow D(f+g)(DhA)(i) \qquad \qquad \qquad I \equiv I \equiv I \\
\searrow \qquad \qquad \qquad \downarrow \\
(Df \otimes Dg)(DhA)(i)
\end{array}$$

□

Proposition C.4. $Cr' \circ Df \circ Cr = D(Jr \bullet f \bullet Jr')$ holds for every morphism f in \mathbf{Idx} and morphisms r, r' in \mathbf{Set} .

Proof. First, note that $((r \pitchfork \mathbb{C})A)(j) = A(r(j))$ for a map r in \mathbf{Set} . This equation also holds when A is replaced by a family of morphisms f . We prove the goal separately.

$$\begin{aligned}
D(Jr \bullet f)A(k) &= \mathbb{T}((Jr \bullet f)k, V \circ A) \\
&= [j \in f(k)]V(A(r(j))) \\
&= \mathbb{T}(f(k), V \circ (r \pitchfork A)) \\
&= Df(r \pitchfork A)(k). \\
D(f \bullet Jr)A(k) &= [j \in f(r(k))]V(Aj) \\
&= (DfA)(r \pitchfork \mathbb{C})(k).
\end{aligned}$$

Again, these equations also extend to morphisms. We have therefore proved the functor equality

$$D(Jr \bullet f \bullet Jr') = (r' \pitchfork \mathbb{C}) \circ Df \circ (r \pitchfork \mathbb{C}).$$

□

$$\begin{array}{ccc}
\begin{array}{ccc}
\llbracket \Gamma_1, \Gamma_2, !_0 X \rrbracket \xrightarrow{\llbracket \Gamma_1 \vdash A \rrbracket} \llbracket \Gamma_2, A, !_0 X \rrbracket & & \llbracket \Gamma_1, \Gamma_2, !_{\text{id}} X \rrbracket \xrightarrow{\llbracket \Gamma_1 \vdash A \rrbracket} \llbracket \Gamma_2, !_{\text{id}} X, A \rrbracket \\
\downarrow \text{id} \otimes w & \downarrow \text{id} \otimes w & \downarrow \text{id} \otimes \epsilon & \downarrow \text{id} \otimes \epsilon \\
\llbracket \Gamma_1, \Gamma_2 \rrbracket \xrightarrow{\llbracket \Gamma_1 \vdash A \rrbracket} \llbracket \Gamma_2, A \rrbracket & \longrightarrow & \llbracket \Gamma_1, \Gamma_2, X \rrbracket \xrightarrow{\llbracket \Gamma_1 \vdash A \rrbracket} \llbracket \Gamma_2, X, A \rrbracket & \longrightarrow & \llbracket B \rrbracket
\end{array} \\
\\
\begin{array}{ccc}
\llbracket \Gamma_1, \Gamma_2, !_f X \rrbracket \xrightarrow{\llbracket \Gamma_1 \vdash A \rrbracket} \llbracket \Gamma_2, !_f X, A \rrbracket & & \llbracket \Gamma_1, \Gamma_2, !_{f_1+f_2} X \rrbracket \xrightarrow{\llbracket \Gamma_1 \vdash A \rrbracket} \llbracket \Gamma_2, !_{f_1+f_2} X, A \rrbracket \\
\downarrow \text{id} \otimes D\sigma & \downarrow \text{id} \otimes D\sigma & \downarrow \simeq \text{oid} \otimes c_{[X]} & \downarrow \simeq \text{oid} \otimes c_{[X]} \\
\llbracket \Gamma_1, \Gamma_2, !_g X \rrbracket \xrightarrow{\llbracket \Gamma_1 \vdash A \rrbracket} \llbracket \Gamma_2, !_g X, A \rrbracket & \longrightarrow & \llbracket \Gamma_1, \Gamma_2, !_{f_1} X, !_{f_2} X \rrbracket \xrightarrow{\llbracket \Gamma_1 \vdash A \rrbracket} \llbracket \Gamma_2, !_{f_1} X, !_{f_2} X, A \rrbracket & \longrightarrow & \llbracket B \rrbracket
\end{array} \\
\\
\begin{array}{ccc}
\llbracket \Gamma \rrbracket \xrightarrow{m_0} \llbracket \Gamma \rrbracket \otimes D0I \xrightarrow{D0[\llbracket \emptyset \vdash A \rrbracket]} \llbracket \Gamma, !_0 A \rrbracket & \llbracket \Gamma \rrbracket \xrightarrow{m_{\text{id}}} \llbracket \Gamma \rrbracket \otimes \text{Did}I \xrightarrow{\text{Did}[\llbracket \emptyset \vdash A \rrbracket]} \llbracket \Gamma, !_{\text{id}} A \rrbracket \\
\parallel & \downarrow w_I & \downarrow w_A & \searrow \sim & \downarrow \epsilon_I & \downarrow \epsilon_{[A]} \\
\llbracket \Gamma \rrbracket & \xleftarrow{\sim} \llbracket \Gamma \rrbracket \otimes I & \xlongequal{\quad} \llbracket \Gamma \rrbracket \otimes I & & \llbracket \Gamma \rrbracket \otimes I & \xrightarrow{[\llbracket \emptyset \vdash A \rrbracket]} \llbracket \Gamma, A \rrbracket \\
\downarrow \llbracket \Gamma \vdash B \rrbracket & & & & & \downarrow \llbracket \Gamma, A \vdash B \rrbracket \\
\llbracket B \rrbracket & & & & & \llbracket B \rrbracket
\end{array} \\
\\
\begin{array}{ccc}
\llbracket \Gamma \rrbracket \xrightarrow{m_{f_1+f_2}} \llbracket \Gamma \rrbracket \otimes D(f_1+f_2)I \xrightarrow{D(f_1+f_2)[\llbracket \emptyset \vdash A \rrbracket]} \llbracket \Gamma, !_{f_1+f_2} A \rrbracket \\
\downarrow m_{f_1} & \downarrow c_{f_1+f_2} & \downarrow c_{f_1+f_2} \\
\llbracket \Gamma \rrbracket \otimes Df_1I \xrightarrow{m_{f_2}} \llbracket \Gamma \rrbracket \otimes (Df_1I \otimes Df_2I) \xrightarrow{(Df_1 \otimes Df_2)[\llbracket \emptyset \vdash A \rrbracket]} \llbracket \Gamma \rrbracket \otimes (\llbracket !_{f_1} A \rrbracket \otimes \llbracket !_{f_2} A \rrbracket) \\
\downarrow Df_1[\llbracket \emptyset \vdash A \rrbracket] & \downarrow Df_1[\llbracket \emptyset \vdash A \rrbracket] & \downarrow \wr \\
\llbracket \Gamma, !_{f_1} A \rrbracket \xrightarrow{m_{f_2}} \llbracket \Gamma, !_{f_1} A \rrbracket \otimes Df_2I \xrightarrow{Df_2[\llbracket \emptyset \vdash A \rrbracket]} \llbracket \Gamma, !_{f_1} A, !_{f_2} A \rrbracket \xrightarrow{\llbracket \Gamma, !_{f_1} A, !_{f_2} A \vdash B \rrbracket} \llbracket B \rrbracket
\end{array} \\
\\
\begin{array}{ccc}
\otimes_i \llbracket !_{\beta_i \bullet f} B_i \rrbracket \xrightarrow{m_{g \bullet f}} (\otimes_i \llbracket !_{\beta_i \bullet f} B_i \rrbracket) \otimes D(g \bullet f)I \xrightarrow{D(g \bullet f)[\llbracket \emptyset \vdash A \rrbracket]} (\otimes_i \llbracket !_{\beta_i \bullet f} B_i \rrbracket) \otimes \llbracket !_{g \bullet f} A \rrbracket \\
\downarrow \otimes_i \delta_{\beta_i, f} & \downarrow (\otimes_i \delta_{\beta_i, f}) \otimes \delta_{g, f} & \downarrow (\otimes_i \delta_{\beta_i, f}) \otimes \delta_{g, f} \\
\otimes_i Df \llbracket !_{\beta_i} B_i \rrbracket \xrightarrow{Df m_g \circ m_f} (\otimes_i Df \llbracket !_{\beta_i} B_i \rrbracket) \otimes Df DgI \xrightarrow{Df Dg[\llbracket \emptyset \vdash A \rrbracket]} (\otimes_i Df \llbracket !_{\beta_i} B_i \rrbracket) \otimes Df DgA \\
\downarrow & \downarrow & \downarrow \\
Df (\otimes_i \llbracket !_{\beta_i} B_i \rrbracket) \xrightarrow{Df m_g} Df (\otimes_i \llbracket !_{\beta_i} B_i \rrbracket \otimes DgI) \xrightarrow{Df Dg[\llbracket \emptyset \vdash A \rrbracket]} Df (\otimes_i \llbracket !_{\beta_i} B_i \rrbracket \otimes \llbracket !_{g \bullet f} A \rrbracket) \\
\downarrow Df [\dots, !_{\beta_\ell} B_\ell, !_{g \bullet f} A \vdash B] \\
\llbracket !_{f \bullet f} B \rrbracket
\end{array}
\end{array}$$

Fig. 2. Commutativity for cases of cut reduction