A Mixed Linear and Graded Logic: Proofs, Terms, and Models

Victoria Vollmer
School of Computing, University of Kent, UK

Danielle Marshall
School of Computing, University of Kent, UK

Harley Eades III
Computer Science, Augusta University, GA, USA

Dominic Orchard
School of Computing, University of Kent, UK
Department of Computer Science and Technology, University of Cambridge, UK

Abstract

Graded modal logics generalise standard modal logics via families of modalities indexed by an algebraic structure whose operations mediate between the different modalities. The graded "of-course" modality $!_r$ captures how many times a proposition is used and has an analogous interpretation to the of-course modality from linear logic; the of-course modality from linear logic can be modelled by a linear exponential comonad and graded of-course can be modelled by a graded linear exponential comonad. Benton showed in his seminal paper on Linear/Non-Linear logic that the of-course modality can be split into two modalities connecting intuitionistic logic with linear logic, forming a symmetric monoidal adjunction. Later, Fujii et al. demonstrated that every graded comonad can be decomposed into an adjunction and a "strict action". We give a similar result to Benton, leveraging Fujii et al.'s decomposition, showing that graded modalities can be split into two modalities connecting a graded logic with a graded linear logic. We propose a sequent calculus, its proof theory and categorical model, and a natural deduction system which we show is isomorphic to the sequent calculus system. Interestingly, our system can also be understood as Linear/Non-Linear logic composed with an action that adds the grading, further illuminating the shared principles between linear logic and a class of graded modal logics.

2012 ACM Subject Classification Theory of computation → Logic

Keywords and phrases linear logic, graded modal logic, adjoint decomposition

Digital Object Identifier 10.4230/LIPIcs.CSL.2025.32

Related Version *Full version with appendices*: `https://arxiv.org/abs/2401.17199` [40]

Funding This work was supported in part by the EPSRC grant EP/T013516/1 (*Verifying Resource-like Data Use in Programs via Types*). The second author received support through Schmidt Sciences, LLC.

Acknowledgements We thank all the anonymous reviewers of this, and previous versions, of this paper. We are also grateful for discussions with Peter Hanukaev and helpful comments from Paulo Torrens on a draft of this manuscript.

1 Introduction

Intuitionistic logic has a central role in the foundations of programming language theory, providing a logical basis for type theories and type systems, and other program reasoning principles. A significant amount of the expressivity of proof systems for intuitionistic logic (both natural deduction and sequent calculus forms) lies within the structure of the

hypotheses – the *context*. Probing the foundations of this part of the logic has, perhaps surprisingly, yielded the very fertile field of *substructural logics* [39] including influential logics such as linear logic [14] and its variants, and the Lambek calculus [25].

By restricting the manipulation of hypotheses in the context we typically arrive at logics which align more closely with physical reality, where propositions are instead "resources" that cannot necessarily be copied, discarded, or reordered. Such restricted logics have been used to construct type systems for safely manipulating values that should be treated in a resourceful way, such as file handlers, pointers to mutable memory, or channels [43, 42].

However, such pervasive restrictions often hamper expressivity and thus some substructural logics then seek to carefully control the reintroduction of structural rules. For example, linear logic provides the ! modality ("of course") for reintroducing weakening and contraction of propositions, which linear logic otherwise prohibits. However, this modality is coarse-grained: for those propositions under the modality, it re-enables all the structural rules that have been removed in linear logic. *Subexponentials* instead aim to be more fine-grained, offering families of modalities capturing specific structural rules [8, 22]. The related notion of *grading* [12, 31] gives an alternate view, providing an indexed family of modalities whose indices are subject to an algebra which accounts for any structural rules applied: structural rules are "counted" by the algebra (whose operations mirror the shape of structural rules). Bounded Linear Logic [15] is a special case where the family of modalities $!_n A$ uses indices n which are natural numbers (or polynomial terms over naturals) counting the upper bound on usage of the proposition A. Various systems generalise this approach to arbitrary semirings to capture data-flow properties [1, 2, 5, 12, 13, 24, 28, 31, 33, 34, 36]. Such graded systems annotate hypotheses/variables in the context with elements of the semiring ("grades") denoting their usage, e.g., $x :_0 A \vdash t : B$ types a term t which does not use x and $y :_{1+1} A \vdash t' : B$ types a term t' in which y is used in two different subterms once each, accounted for by the semiring addition. A graded modality *internalises* the semiring grade, causing a multiplication to the grades of any captured dependencies when the graded modality is introduced, e.g., $y :_{0*(1+1)} A \vdash \Box t' : \Box_0 B$.

We seek here to further understand the underlying structure of graded modal logics by following an "adjoint resolution" approach à la Benton's seminal "A Mixed Linear and Non-Linear Logic: Proofs, Terms and Models" at CSL 1994 [3]. Benton showed that the exponential modality of linear logic (modelled by a comonad) can be decomposed into an adjunction, defining a pair of "adjoint" logics (a linear logic and a non-linear intuitionistic, or "Cartesian", logic) which embed into each other [3]. This provides a beautiful reduction of the core features of linear logic and its non-linearity modality. *Adjoint logic* applies the same idea but to subexponentials [37, 38]. We follow the same scheme, via the adjoint decomposition of graded modalities which generalise linear logic's ! and which are traditionally modelled by graded exponential comonads [5, 6, 12, 13, 24, 34]. Whilst Benton's work has a pair of adjoint modalities mediating between the two sublogics, we have a pair of a modality Lin and a *graded modality* Grd_r. We give a categorical model, showing that these are captured by an LNL-like adjunction paired with a "strict action" for incorporating the grading, following the Fujii-Katsumata-Melliès adjoint decomposition of graded (co)monads [10, 24]. The result is a pair of logics which serve to explain and clarify the relationship between linearity and grading. We call our system Mixed Graded/Linear (mGL) Logic.

This pair of logics also shines light on a relationship between two styles of graded system in the literature: those which take linear types as their basis augmented with a graded modality [6, 12, 31] versus those with no base notion of linearity where grading is pervasive, tracking all substructurality [1, 2, 4, 7, 28, 30, 34]. Our linear fragment is analogous to the

former whilst our graded fragment is analogous to the latter. The mutual embedding shows that these two styles of graded logics have a similar relationship to the adjoint relationship of intuitionistic logic and linear logic.

Aside from the internal motivation of better understanding the relationship between grading and linearity, an external motivation for this work is that it can provide a basis for flexible, safe programming with resources. By separating out the linear fragment from an intuitionistic graded fragment, one could avoid the strictures of linearity for working with standard data types which need not be linear, working only in the linear fragment for handling resources like file handles. The mutual embedding would allow the programmer to move smoothly between these two subcalculi, as seen also in other adjunction-based calculi, e.g., for concurrent programming [35]. The focus here however is on the core theory rather than developing these applications yet.

Since our focus is on the relationship between grading and linearity, we consider the semiring-graded modalities that generalise linear logic's !. Other flavours of graded modality (e.g., graded monads for capturing side effecting behaviour [23, 32]) are not considered here.

Roadmap

Section 2 defines a pair of sequent calculi, the mixed fragment MS, which has both linear and graded assumptions, and the graded fragment GS, which has only graded assumptions and no function arrow. As described above, these calculi have a mutual embedding via modalities between the two. Section 3 considers the categorical model of mGL leveraging recent work on the adjoint resolution of graded comonads [10, 24]. Section 4 provides the natural deduction formulation of the calculus, which is proved equivalent to the sequent calculus version. Section 5 discusses how this work gives a view on the landscape of graded systems in the literature and considers other related work and future applications.

A version of this paper with the appendices providing full proof details can be found on the arXiv [40].

2 Mixed Graded/Linear Logic: Proofs and Terms

We present first a sequent calculus for Mixed Graded/Linear logic, which comes in the form of a term assignment. Figure 1 collects the term syntax for reference; it will also be used in Section 4 for the natural deduction formulation. The syntax is explained with reference to its associated proof rules in the next section.

$$
\begin{array}{llll}
(\mathsf{GS/GT}) & t ::= & x \mid \mathsf{j} \mid \mathsf{let}\,\mathsf{j} = t_1\,\mathsf{in}\,t_2 & \\
\textit{graded} & \mid & (t_1, t_2) \mid \mathsf{let}\,(x,y) = t_1\,\mathsf{in}\,t_2 & \\
& \mid & \mathsf{Lin}\; l & \\
\\
(\mathsf{MS/MT}) & l ::= & x \mid \mathsf{i} \mid \mathsf{let}\,\mathsf{i} = l_1\,\mathsf{in}\,l_2 & \\
\textit{linear} & \mid & (l_1, l_2) \mid \mathsf{let}\,(x,y) = l_1\,\mathsf{in}\,l_2 & \\
& \mid & \lambda x.l \mid l_1\; l_2 & \\
& \mid & \mathsf{Grd}\; r\; t \mid \mathsf{let}\,\mathsf{Grd}\; r\; x = l_1\,\mathsf{in}\,l_2 & \\
& \mid & \mathsf{Unlin}\; z & \\
& \mid & \mathsf{let}\,\mathsf{j} = z\,\mathsf{in}\,l \mid \mathsf{let}\,(x,y) = z\,\mathsf{in}\,l &
\end{array}
$$

Variables are ranged over by x, y, z in both fragments.

Terms are mostly grouped above with introduction forms followed by elimination forms, though note that in the last two lines of syntax for l there are additional eliminators: for the linear modality (Unlin), for units j, and for tensors coming from the graded context.

Figure 1 Collected term syntax.

Benton's approach has two proof systems [3]: one system of linear propositions (the L of LNL) with two contexts for linear and non-linear propositions respectively, and one system of non-linear propositions (the NL in LNL). We generalise this approach to the graded setting by replacing the non-linear parts with *graded* notions. Thus, our system (mGL) has two analogous proof systems: one of linear propositions with two contexts for linear and graded propositions, with judgments subscripted as $\vdash_{\mathsf{MS}}$ (for "Mixed (linear/graded) Sequent"), and one system of graded propositions, with judgments subscripted as $\vdash_{\mathsf{GS}}$ ("Graded Sequent").

The syntax of Benton's propositions is split into two, "*conventional*" (i.e., Cartesian / non-linear) and *linear* [3]. The syntax of our propositions is analogously split into two, *graded* and *linear*:

$$\begin{array}{ll}\text{(Graded)} & X, Y, Z ::= \mathsf{J} \mid X \boxtimes Y \mid \mathsf{Lin}\, A \\ \text{(Linear)} & A, B, C ::= \mathsf{I} \mid A \otimes B \mid A \multimap B \mid \mathsf{Grd}_r\, X\end{array}$$

where J and I are unit types, and $\boxtimes$ and $\otimes$ are tensor (product) operators in their respective domains. In the case of the linear domain, the product is the standard multiplicative conjunction. The Lin modality encapsulates a linear proposition as a graded proposition, and the Grd_r modality encapsulates a graded proposition (at grade r, whose structure is defined below) as a linear proposition. Thus, the two logics are interconnected by $\mathsf{Grd}_r\, X$ and $\mathsf{Lin}\, A$. Using these two modalities we will later define graded modalities $\Box_r A$ as $\mathsf{Grd}_r\, (\mathsf{Lin}\, A)$, similarly to how the of-course modality $!A$ can be defined in LNL logic as the composition of two adjoint modalities.

▶ **Definition 1.** *Grades (ranged over by r, s) are drawn from a semiring parameterizing the system $(\mathcal{R}, 1, *, 0, +, \leq)$ with preorder $(\mathcal{R}, \leq)$ such that both $*$ and $+$ are monotonic wrt $\leq$.*

The semiring governs the structural rules: the additive part of the semiring is involved in weakening and contraction, and the multiplicative part in usage and composition. Various concrete examples of interesting semirings are given at the end of Subsection 2.1.

Section 4 develops an equivalent natural deduction formulation of mGL. We then show that the natural deduction and the sequent calculus are interderivable without modifying the term witnessing a derivation. Thus, any semantic model of one is a model of the other. We opt to focus on the sequent calculus form for now without loss of generality.

2.1 Sequent Calculus

We first define contexts used in the judgments:

▶ **Definition 2** (Graded contexts)**.** *Suppose $(\mathcal{R}, 1, *, 0, +, \leq)$ is a preordered semiring (Def. 1). Then* grade vectors *δ are sequences of $\mathcal{R}$, contexts Δ are sequences of graded formulas X, and contexts Γ are sequences of linear formulas:*

$$\delta := \emptyset \mid \delta, r \qquad \Delta := \emptyset \mid \Delta, x : X \qquad \Gamma := \emptyset \mid \Gamma, x : A$$

The comma operator is overloaded for sequence concatenation, i.e., we can write δ_1, δ_2 and Δ_1, Δ_2, which further requires that Δ_1 and Δ_2 are disjoint contexts.

A graded context *$\delta \odot \Delta$ is a pairing of a grade vector and a context defined as follows:*

$$\emptyset \odot \emptyset = \emptyset \qquad (\delta, r) \odot (\Delta, x : X) = (\delta \odot \Delta), x : (r \odot X)$$

where $r \odot X$ pairs a formula with a grade r capturing (by the rules of the system) how the formula X (named x) is used to form a judgment.

We lift the operations of semirings to grade vectors, forming a semimodule, with the pointwise addition and scalar multiplication defined in a standard way:

$$\emptyset + \emptyset = \emptyset \qquad\qquad r * \emptyset = \emptyset$$
$$(\delta_1, r_1) + (\delta_2, r_2) = (\delta_1 + \delta_2), (r_1 + r_2) \qquad\qquad r * (\delta, s) = (r * \delta), (r * s)$$

Addition of grade vectors requires the vectors to be of the same length.

The judgment form for our fully graded logic $\delta \odot \Delta \vdash_{\mathsf{GS}} t : X$ captures a concluding proposition X under the graded context of assumptions $\delta \odot \Delta$. Mixed graded/linear logic judgments $\delta \odot \Delta; \Gamma \vdash_{\mathsf{MS}} l : A$ are similar but also have a context Γ of linear assumptions which, being linear, do not have a corresponding grade vector.

The two judgments $\vdash_{\mathsf{GS}}$ and $\vdash_{\mathsf{MS}}$ (also called *sub-logics* or *fragments*) are defined by mutual induction. We present conceptually related rules from both systems side-by-side where possible, or one-after-the-other, in the order GS then MS.

The identity (axiom) rules are:

$$\textsc{id}_{\mathsf{GS}}\ \frac{}{1 \odot x : X \vdash_{\mathsf{GS}} x : X} \qquad\qquad \textsc{id}_{\mathsf{MS}}\ \frac{}{\emptyset \odot \emptyset; x : A \vdash_{\mathsf{MS}} x : A}$$

The multiplicative identity 1 is the "default" grade for formulas in the graded logic GS (left), in the sense that we can think of the right-hand side of the judgment as also implicitly having grade 1. The graded identity rule says that a graded formula that is used must have the default grade. For example, in the natural number semiring $(\mathbb{N}, 1, *, 0, +, =)$ the multiplicative identity $1 \in \mathbb{N}$ captures linear usage. The mixed identity rule types linear assumption use, requiring just a singleton linear context (forcing a lack of weakening). It also requires that there are no graded formulas in context – the graded context is empty $\emptyset$.

The "cut" rules are:

$$\textsc{cut}_{\mathsf{GS}}\ \frac{\delta_2 \odot \Delta_2 \vdash_{\mathsf{GS}} t_1 : X \qquad (\delta_1, r, \delta_3) \odot (\Delta_1, x : X, \Delta_3) \vdash_{\mathsf{GS}} t_2 : Y}{(\delta_1, r * \delta_2, \delta_3) \odot (\Delta_1, \Delta_2, \Delta_3) \vdash_{\mathsf{GS}} [t_1/x]t_2 : Y}$$

$$\textsc{cut}_{\mathsf{MS}}\ \frac{\delta_2 \odot \Delta_2; \Gamma_2 \vdash_{\mathsf{MS}} l_1 : A \qquad \delta_1 \odot \Delta_1; (\Gamma_1, x : A, \Gamma_3) \vdash_{\mathsf{MS}} l_2 : B}{(\delta_1, \delta_2) \odot (\Delta_1, \Delta_2); (\Gamma_1, \Gamma_2, \Gamma_3) \vdash_{\mathsf{MS}} [l_1/x]l_2 : B} \qquad \textsc{gcut}_{\mathsf{MS}}\ \frac{\delta_2 \odot \Delta_2 \vdash_{\mathsf{GS}} t : X \qquad (\delta_1, r, \delta_3) \odot (\Delta_1, x : X, \Delta_3); \Gamma \vdash_{\mathsf{MS}} l : B}{(\delta_1, r * \delta_2, \delta_3) \odot (\Delta_1, \Delta_2, \Delta_3); \Gamma \vdash_{\mathsf{MS}} [t/x]l : B}$$

The $\textsc{cut}_{\mathsf{GS}}$ rule provides a cut through a graded proposition X of grade r in the receiving context (second premise). Thus, the resulting term uses semiring multiplication (lifted to contexts, Def. 2) to capture sequential usage, scaling the grade vector δ_2 of the cut term t_1 by r. The $\textsc{cut}_{\mathsf{MS}}$ rules provides a cut through a linear proposition A and has no effect on the graded contexts. However, MS has a further cut rule $\textsc{gcut}_{\mathsf{MS}}$ for graded propositions in its graded context, incurring a scaling similarly to $\textsc{cut}_{\mathsf{GS}}$. This pattern occurs throughout: operations applied to the graded context in GS have a sister rule in MS applying the same operation in the MS graded context.

Both sub-logics have free use of exchange:

$$\textsc{ex}_{\mathsf{GS}}\ \frac{(\delta_1, r_1, r_2, \delta_2) \odot (\Delta_1, x : X, y : Y, \Delta_2) \vdash_{\mathsf{GS}} t : Z}{(\delta_1, r_2, r_1, \delta_2) \odot (\Delta_1, y : Y, x : X, \Delta_2) \vdash_{\mathsf{GS}} t : Z}$$

$$\textsc{ex}_{\mathsf{MS}}\ \frac{\delta \odot \Delta; (\Gamma_1, x : A, y : B, \Gamma_2) \vdash_{\mathsf{MS}} l : C}{\delta \odot \Delta; (\Gamma_1, y : B, x : A, \Gamma_2) \vdash_{\mathsf{MS}} l : C} \qquad \textsc{gex}_{\mathsf{MS}}\ \frac{(\delta_1, r_1, r_2, \delta_2) \odot (\Delta_1, x : X, y : Y, \Delta_2); \Gamma \vdash_{\mathsf{MS}} l : B}{(\delta_1, r_2, r_1, \delta_2) \odot (\Delta_1, x : Y, y : X, \Delta_2); \Gamma \vdash_{\mathsf{MS}} l : B}$$

Exchanging graded propositions simultaneously exchanges their grades in the grade vector.

We can use weakening and contraction in the graded system and the mixed system within the graded contexts, with the semiring's 0 representing weakened hypotheses and the grades of contracted hypotheses combined via semiring addition $+$:

$$\textsc{weak}_{\mathsf{GS}}\ \frac{(\delta_1, \delta_2) \odot (\Delta_1, \Delta_2) \vdash_{\mathsf{GS}} t : Y}{(\delta_1, 0, \delta_2) \odot (\Delta_1, x : X, \Delta_2) \vdash_{\mathsf{GS}} t : Y} \qquad \textsc{cont}_{\mathsf{GS}}\ \frac{(\delta_1, r_1, r_2, \delta_2) \odot (\Delta_1, x : X, y : X, \Delta_2) \vdash_{\mathsf{GS}} t : Y}{(\delta_1, r_1 + r_2, \delta_2) \odot (\Delta_1, x : X, \Delta_2) \vdash_{\mathsf{GS}} [x/y]t : Y}$$

$$\textsc{weak}_{\mathsf{MS}}\ \frac{(\delta_1, \delta_2) \odot (\Delta_1, \Delta_2); \Gamma \vdash_{\mathsf{MS}} l : B}{(\delta_1, 0, \delta_2) \odot (\Delta_1, x : X, \Delta_2); \Gamma \vdash_{\mathsf{MS}} l : B} \qquad \textsc{cont}_{\mathsf{MS}}\ \frac{(\delta_1, r_1, r_2, \delta_2) \odot (\Delta_1, x : X, y : X, \Delta_2); \Gamma \vdash_{\mathsf{MS}} l : B}{(\delta_1, r_1 + r_2, \delta_2) \odot (\Delta_1, x : X, \Delta_2); \Gamma \vdash_{\mathsf{MS}} [x/y]l : B}$$

The left and right rules for units for the graded and mixed logics are akin to linear logic:

$$\textsc{unit}^{\mathsf{J}}_{L}\ \frac{(\delta_1, \delta_2) \odot (\Delta_1, \Delta_2) \vdash_{\mathsf{GS}} t : X}{(\delta_1, r, \delta_2) \odot (\Delta_1, x : \mathsf{J}, \Delta_2) \vdash_{\mathsf{GS}} \mathsf{let\,j} = x \,\mathsf{in}\, t : X} \qquad \textsc{unit}^{\mathsf{J}}_{R}\ \frac{}{\emptyset \odot \emptyset \vdash_{\mathsf{GS}} \mathsf{j} : \mathsf{J}}$$

$$\textsc{unit}^{\mathsf{I}}_{L}\ \frac{\delta \odot \Delta; (\Gamma_1, \Gamma_2) \vdash_{\mathsf{MS}} l : A}{\delta \odot \Delta; (\Gamma_1, x : \mathsf{I}, \Gamma_2) \vdash_{\mathsf{MS}} \mathsf{let\,i} = x \,\mathsf{in}\, l : A} \qquad \textsc{unit}^{\mathsf{I}}_{R}\ \frac{}{\emptyset \odot \emptyset; \emptyset \vdash_{\mathsf{MS}} \mathsf{i} : \mathsf{I}}$$

$$\textsc{unit}^{\mathsf{J-MS}}_{L}\ \frac{(\delta_1, \delta_2) \odot (\Delta_1, \Delta_2); \Gamma \vdash_{\mathsf{MS}} l : A}{(\delta_1, r, \delta_2) \odot (\Delta_1, z : \mathsf{J}, \Delta_2); \Gamma \vdash_{\mathsf{MS}} \mathsf{let\,j} = z \,\mathsf{in}\, l : A}$$

Thus, in GS, we can eliminate a graded unit j at an arbitrary grade r, whereas the linear unit i in MS gets eliminated from the linear context. The additional left rule ($\textsc{unit}^{\mathsf{J-MS}}_{L}$) for MS again similarly eliminates graded units J in the graded context.

Tensor products are then eliminated in each fragment as follows:

$$\boxtimes_L\ \frac{(\delta_1, r, r, \delta_2) \odot (\Delta_1, x : X, y : Y, \Delta_2) \vdash_{\mathsf{GS}} t : Z}{(\delta_1, r, \delta_2) \odot (\Delta_1, z : X \boxtimes Y, \Delta_2) \vdash_{\mathsf{GS}} \mathsf{let}\,(x, y) = z \,\mathsf{in}\, t : Z} \qquad \boxtimes_R\ \frac{\delta_1 \odot \Delta_1 \vdash_{\mathsf{GS}} t_1 : X \quad \delta_2 \odot \Delta_2 \vdash_{\mathsf{GS}} t_2 : Y}{(\delta_1, \delta_2) \odot (\Delta_1, \Delta_2) \vdash_{\mathsf{GS}} (t_1, t_2) : X \boxtimes Y}$$

$$\otimes_L\ \frac{\delta \odot \Delta; (\Gamma_1, x : A, y : B, \Gamma_2) \vdash_{\mathsf{MS}} l : C}{\delta \odot \Delta; (\Gamma_1, z : A \otimes B, \Gamma_2) \vdash_{\mathsf{MS}} \mathsf{let}\,(x, y) = z \,\mathsf{in}\, l : C} \qquad \otimes_R\ \frac{\delta_1 \odot \Delta_1; \Gamma_1 \vdash_{\mathsf{MS}} l_1 : A \quad \delta_2 \odot \Delta_2; \Gamma_2 \vdash_{\mathsf{MS}} l_2 : B}{(\delta_1, \delta_2) \odot (\Delta_1, \Delta_2); (\Gamma_1, \Gamma_2) \vdash_{\mathsf{MS}} (l_1, l_2) : A \otimes B}$$

$$\boxtimes_{L-\mathsf{MS}}\ \frac{(\delta_1, r, r, \delta_2) \odot (\Delta_1, x : X, y : Y, \Delta_2); \Gamma \vdash_{\mathsf{MS}} l : A}{(\delta_1, r, \delta_2) \odot (\Delta_1, z : X \boxtimes Y, \Delta_2); \Gamma \vdash_{\mathsf{MS}} \mathsf{let}\,(x, y) = z \,\mathsf{in}\, l : A}$$

The left rule for $\boxtimes$ eliminates from the graded context at any grade r, where the components of the tensor product both inherit this grade in the premise. Reading instead top-down, the graded tensor product requires that both components are graded with the same grade; this is similar to linear products, where both components are linear.

Note that Benton has two left rules for (non-linear) tensor products, in the "projection" style. We instead must use the pattern matching style for the soundness of grading so that each component is bound to a variable with the same grade.

Only the mixed linear-graded system has implication, and only on linear propositions, thus we have $\multimap$ in MS with left and right rules:

$$\multimap_L \quad \frac{\delta_2 \odot \Delta_2; \Gamma_2 \vdash_{\mathsf{MS}} l_2 : A \qquad \delta_1 \odot \Delta_1; (\Gamma_1, x : B, \Gamma_3) \vdash_{\mathsf{MS}} l_1 : C}{(\delta_1, \delta_2) \odot (\Delta_1, \Delta_2); (\Gamma_1, z : A \multimap B, \Gamma_2, \Gamma_3) \vdash_{\mathsf{MS}} [z\, l_2/x] l_1 : C} \qquad \multimap_R \quad \frac{\delta \odot \Delta; (\Gamma, x : A) \vdash_{\mathsf{MS}} l : B}{\delta \odot \Delta; \Gamma \vdash_{\mathsf{MS}} \lambda x.l : A \multimap B}$$

In Lemma 4, we recover a graded implication through the modal operators of the system in the same way that Melliès did for (ungraded) LNL logic [29].

We now consider the modal operators Lin and Grd_r which connect the two sub-logics.

The right rule for the Lin modality transports a linear formula from the linear system MS into the graded system GS where it can be reasoned with non-linearly as accounted for by grading. The corresponding left rule is akin to *dereliction* from linear logic, enabling a linear assumption $x : A$ to treated as a (renamed) graded assumption $z : \mathsf{Lin}\, A$ at grade 1:

$$\mathsf{Lin}_L \quad \frac{\delta \odot \Delta; (x : A, \Gamma) \vdash_{\mathsf{MS}} l : B}{(\delta, 1) \odot (\Delta, z : \mathsf{Lin}\, A); \Gamma \vdash_{\mathsf{MS}} [\mathsf{Unlin}\, z/x] l : B} \qquad \mathsf{Lin}_R \quad \frac{\delta \odot \Delta; \emptyset \vdash_{\mathsf{MS}} l : B}{\delta \odot \Delta \vdash_{\mathsf{GS}} \mathsf{Lin}\, l : \mathsf{Lin}\, B}$$

The other modal operator Grd, or rather the family of modal operators Grd_r, transports a graded formula with its grade into the linear system where it can be reasoned with linearly:

$$\mathsf{Grd}_L \quad \frac{(\delta, r) \odot (\Delta, x : X); \Gamma \vdash_{\mathsf{MS}} l : C}{\delta \odot \Delta; (z : \mathsf{Grd}_r\, X, \Gamma) \vdash_{\mathsf{MS}} \mathsf{let}\, \mathsf{Grd}\, r\, x = z\, \mathsf{in}\, l : C} \qquad \mathsf{Grd}_R \quad \frac{\delta \odot \Delta \vdash_{\mathsf{GS}} t : X}{r * \delta \odot \Delta; \emptyset \vdash_{\mathsf{MS}} \mathsf{Grd}\, r\, t : \mathsf{Grd}_r\, X}$$

The right rule is akin to *promotion* for Grd_r where we subsequently scale the graded context by the grade r. The left rule "unboxes" a graded modality $\mathsf{Grd}_r\, X$ providing access to the X formula "inside", graded at r.

Perhaps the most remarkable property of these modal operators is that they decompose semiring-graded necessity modalities into $\Box_r A = \mathsf{Grd}_r\, (\mathsf{Lin}\, A)$ [24] within the mixed system. In fact, their introduction and elimination rules are derivable:

▶ **Lemma 3** (mGL Graded Necessity Modality). *The following are derivable:*

$$\Box_E \quad \frac{\delta_2 \odot \Delta_2; \Gamma_2 \vdash_{\mathsf{MS}} l_1 : \Box_r A \qquad (\delta_1, r, \delta_3) \odot (\Delta_1, x : \mathsf{Lin}\, A, \Delta_3); \Gamma_1 \vdash_{\mathsf{MS}} l_2 : B}{(\delta_1, \delta_2, \delta_3) \odot (\Delta_1, \Delta_2, \Delta_3); (\Gamma_1, \Gamma_2) \vdash_{\mathsf{MS}} \mathsf{let}\, \mathsf{Grd}\, r\, x = l_1\, \mathsf{in}\, l_2 : B} \qquad \Box_i \quad \frac{\delta \odot \Delta; \emptyset \vdash_{\mathsf{MS}} l : A}{(r * \delta) \odot \Delta; \emptyset \vdash_{\mathsf{MS}} \mathsf{Grd}\, r\, (\mathsf{Lin}\, l) : \Box_r A}$$

Proof. The elimination rule follows by applying Grd_L to the second premise and then applying cut with the first premise. The introduction rule follows by Lin_R then Grd_R. ◀

The previous lemma reveals a lot about the structure of existing graded type systems. First, graded hypotheses, usually denoted with their grade $x :_r A$ in the literature (e.g. [1, 31]), are graded hypotheses $r \odot \mathsf{Lin}\, A$ where the linear formula has been transported into the graded system. Second, the restriction to only graded variables in the promotion rule is very explicit in the introduction rule $\Box_i$ above. Third, the elimination rule (here $\Box_E$) is really the left rule for Grd followed by a cut. Thus, graded type systems, whilst typically of a natural deduction form, incorporate a little of the flavour of sequent calculi in the rules for graded modalities because of the integrated cut.

Using the modal operators we can derive a graded implication of the form $\mathsf{Grd}_r\, X \multimap A$:

▶ **Lemma 4** (mGL Graded Implication). *The following rules are derivable:*

$$\multimap_{GL}\ \frac{\begin{array}{c}\delta_2 \odot \Delta_2 \vdash_{\mathsf{GS}} t : X \\ (\delta_1, \delta_3) \odot (\Delta_1, \Delta_3); (\Gamma_1, x : A, \Gamma_2) \vdash_{\mathsf{MS}} l : B\end{array}}{(\delta_1, r * \delta_2, \delta_3) \odot (\Delta_1, \Delta_2, \Delta_3); (\Gamma_1, z : \mathsf{Grd}_r\, X \multimap A, \Gamma_2) \vdash_{\mathsf{MS}} [z\,(\mathsf{Grd}\, r\, t)/x]l : B}$$

$$\multimap_{GR}\ \frac{(\delta, r) \odot (\Delta, x : X); \Gamma \vdash_{\mathsf{MS}} l : A}{\delta \odot \Delta; \Gamma \vdash_{\mathsf{MS}} \lambda y.(\mathsf{let}\ \mathsf{Grd}\, r\, x = y\, \mathsf{in}\, l) : \mathsf{Grd}_r\, X \multimap A}$$

Proof. The left rule follows by applying Grd_R to the first premise, and then $\multimap_L$ using the second premise. The right rule follows by applying Grd_L to the premise and then $\multimap_R$. ◀

The final rules are the approximation rules for grades:

$$\mathrm{SUB}_{\mathsf{GS}}\ \frac{\delta_1 \odot \Delta \vdash_{\mathsf{GS}} t : X \quad \delta_1 \leq \delta_2}{\delta_2 \odot \Delta \vdash_{\mathsf{GS}} t : X} \qquad \mathrm{SUB}_{\mathsf{MS}}\ \frac{\delta_1 \odot \Delta; \Gamma \vdash_{\mathsf{MS}} l : B \quad \delta_1 \leq \delta_2}{\delta_2 \odot \Delta; \Gamma \vdash_{\mathsf{MS}} l : B}$$

Approximation allows for the abstraction of grades along an ordering. For example, in the semiring $(\mathbb{N}, 1, *, 0, +, \leq)$ where the order is the usual ordering on natural numbers, then a grade r stands for "at-most r", generalising the notion of "affine" usage tracking. Disabling the ordering by forcing it to be true only in the case of reflexive pairs (i.e., an equality relation) results in exact usage tracking. This demonstrates that having a notion of approximation results in a more general usage tracking framework.

▶ **Example 5** (Derivation). As an example derivation in the natural numbers semiring $(\mathbb{N}, 1, *, 0, +, \leq)$, for "affine" usage tracking, the following copies an assumption to make a pair in the graded fragment, then uses an approximation, and then transports the pair into the linear fragment, scaling its grades further:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{1 \odot x : X \vdash_{\mathsf{GS}} x : X}\,\mathrm{ID}_{\mathsf{GS}} \quad \dfrac{}{1 \odot y : X \vdash_{\mathsf{GS}} y : X}\,\mathrm{ID}_{\mathsf{GS}}}{1, 1 \odot x : X, y : X \vdash_{\mathsf{GS}} (x, y) : X \boxtimes X}\,\boxtimes_R}{2 \odot x : X \vdash_{\mathsf{GS}} [x/y](x, y) : X \boxtimes X \quad 2 \leq 3}\,\mathrm{CONT}_{\mathsf{GS}}}{3 \odot x : X \vdash_{\mathsf{GS}} (x, x) : X \boxtimes X}\,\mathrm{SUB}_{\mathsf{GS}}}{2 * 3 \odot x : X; \emptyset \vdash_{\mathsf{MS}} \mathsf{Grd}\, 2\, (x, x) : \mathsf{Grd}_2\, (X \boxtimes X)}\,\mathsf{Grd}_R$$

▶ **Example 6** (None-One-Tons [28]). The semiring over $\{0, 1, \omega\}$ with $0 \leq \omega$ and $1 \leq \omega$, where $+$ and $*$ are saturating at ω, can be used to distinguish between linear and various non-linear uses: assigning $r = 1$ to linear usage, $r = 0$ to non-usage (when a resource is discarded), and $r = \omega$ to arbitrary usage.

Note, however, that even with the above semiring we are unable to exactly represent the exponential modality ! from linear logic via some particular grade r within the graded logic. This is because no matter which grade we choose, we are able to "push" the grade into the tensor product using this graded tensor elimination ($\boxtimes_L$), allowing derivation of $\mathsf{Grd}_r\,(X \boxtimes Y) \multimap (\mathsf{Grd}_r\,X \otimes \mathsf{Grd}_r\,Y)$, and yet in linear logic it is not possible to derive $!(A \otimes B) \multimap !A \otimes !B$. Therefore, our logic cannot reduce to Benton's LNL logic simply by taking the Cartesian (trivial) semiring, as one might expect at first glance. This quality is typical of *graded base* systems, so reconciling these with linear logic requires some additional structure on the semiring [18] (though this is not the focus here).

On the other hand, notice that we have another way to represent graded products: as linear products wrapped in the derived graded modality, or $\Box_r(A \otimes B)$. Importantly, here it is not possible to "push" the grade "through" the tensor as we can for the graded product; we cannot derive $(\Box_r A) \otimes (\Box_r B)$. This representation of graded products thus has behaviour more typical of a *linear base* graded type system, with our combined logic again giving us a clearer understanding of the relationship between these contrasting styles.

▶ **Example 7** (Security levels)**.** Information-Flow Control properties can be tracked by instantiating the semiring with a lattice of security levels [12], e.g., with $(\{\mathsf{Lo} \leq \mathsf{Hi}\}, \mathsf{Lo}, \wedge, \mathsf{Hi}, \vee)$ where Hi-graded inputs are treated as irrelevant: we cannot depend on any high-security inputs when building a low-security graded output $\mathsf{Grd}_{\mathsf{Lo}}A$.

▶ **Example 8** (Sensitivity)**.** The real number semiring $(\mathbb{R}, 1, *, 0, +, \leq)$ can be leveraged to capture a notion of numerical sensitivity in programs/logic [11, 9], where a program is k-sensitive (for $k \in \mathbb{R}$) in a variable if a change ϵ in its inputs to x produces at most a change of $k\epsilon$ in the output of the program. This instantiation of the system tracks sensitivities as grades where additional dependent-type-based mechanisms are needed to lift program values into the types, e.g., $\mathsf{scale} : (k : \mathbb{R}) \to \mathsf{Grd}_k\mathbb{R} \to \mathbb{R}$.

2.2 Metatheory

mGL enjoys a rich metatheory. First, it satisfies cut elimination, for which we give the full proof. The proof of cut reduction requires a generalization of the graded cut rules to graded *multicut* rules in order to accommodate the structural rule of graded contraction.[1]

Thus, throughout the cut elimination proof we use the following graded multicut rules:

MCUT

$$\frac{\begin{array}{c}\delta_2 \odot \Delta_2 \vdash_{\mathsf{GS}} t_1 : X \\ (\delta_1, \delta, \delta_3) \odot (\Delta_1, x^n : X^n, \Delta_3) \vdash_{\mathsf{GS}} t_2 : Y\end{array}}{(\delta_1, (\delta \circledast [{\delta_2}^n]), \delta_3) \odot (\Delta_1, \Delta_2, \Delta_3) \vdash_{\mathsf{GS}} [t_1, \ldots, t_1/x_1, \ldots, x_n]t_2 : Y}$$

GMCUT

$$\frac{\begin{array}{c}\delta_2 \odot \Delta_2 \vdash_{\mathsf{GS}} t : X \\ (\delta_1, \delta, \delta_3) \odot (\Delta_1, X^n, \Delta_3); \Gamma \vdash_{\mathsf{MS}} l : B\end{array}}{(\delta_1, (\delta \circledast [{\delta_2}^n]), \delta_3) \odot (\Delta_1, \Delta_2, \Delta_3); \Gamma \vdash_{\mathsf{MS}} [t, \ldots, t/x_1, \ldots, x_n]l : B}$$

Both rules compute the contraction of the n hypotheses involved in the multicut on the cut-formula X. To do this we use *row-vector matrix multiplication*. We denote the matrix consisting of n-copies of the row vector δ_2 by $[\delta_2^n]$. Then row-vector multiplication is:

[1] Whilst cut reduction can be proved for intuitionistic sequent calculus without multicut [41], we use the standard multicut approach as it relates well to the categorical models developed later.

$$\delta \circledast [{\delta_2}^n] = \bigoplus_{k=1}^{n} (\delta(k) * \delta_2)$$

where the $*$ on the right is the scalar multiplication derived from the semiring, $\bigoplus$ is the pointwise addition of vectors, and where $\delta(k)$ is the k-th element of the vector δ. This computes the usages of the hypotheses in Δ_2 as the multiplication of a matrix of size $1 \times n$ with a matrix of size $n \times |\Delta_2|$ to yield a matrix of size $1 \times |\Delta_2|$.

We now proceed with the proof of cut elimination. The *rank* $\mathsf{Rank}(X)$ and $\mathsf{Rank}(A)$ of a formula is the height of the input formula's syntax tree where constants are of rank 0. The *cut rank* $\mathsf{CutRank}\,(\Pi)$ of a derivation Π of some judgment is defined to be one more than the maximum rank of the cut formula's in Π, and 0 if Π is cut free. The *depth* $\mathsf{Depth}(\Pi)$ of a derivation Π is the length of the longest path in the proof tree of Π, and hence, the depth of an axiom is 0. We prove cut-elimination without term annotations on the rules, in keeping with traditional proofs.

▶ **Lemma 9** (Cut Reduction for mGL).

1. *(Graded) If* Π_1 *is a proof of* $\delta_2 \odot \Delta_2 \vdash_{\mathsf{GS}} X$ *and* Π_2 *is a proof of* $(\delta_1, \delta, \delta_3) \odot (\Delta_1, X^n, \Delta_3) \vdash_{\mathsf{GS}} Y$ *with* $\mathsf{CutRank}\,(\Pi_1), \mathsf{CutRank}\,(\Pi_2) \leq \mathsf{Rank}(X)$*, then there exists a proof* Π *of* $(\delta_1, \delta \circledast [{\delta_2}^n], \delta_3) \odot (\Delta_1, \Delta_2, \Delta_3) \vdash_{\mathsf{GS}} Y$ *with* $\mathsf{CutRank}\,(\Pi) \leq \mathsf{Rank}(X)$.
2. *(Graded/Mixed) If* Π_1 *is a proof of* $\delta_2 \odot \Delta_2 \vdash_{\mathsf{GS}} X$ *and* Π_2 *is a proof of* $(\delta_1, \delta, \delta_3) \odot (\Delta_1, X^n, \Delta_3); \Gamma \vdash_{\mathsf{MS}} B$ *with* $\mathsf{CutRank}\,(\Pi_1), \mathsf{CutRank}\,(\Pi_2) \leq \mathsf{Rank}(X)$*, then there exists a proof* Π *of* $(\delta_1, \delta \circledast [{\delta_2}^n], \delta_3) \odot (\Delta_1, \Delta_2, \Delta_3); \Gamma \vdash_{\mathsf{MS}} B$ *with* $\mathsf{CutRank}\,(\Pi) \leq \mathsf{Rank}(X)$.
3. *(Mixed) If* Π_1 *is a proof of* $\delta_2 \odot \Delta_2; \Gamma_2 \vdash_{\mathsf{MS}} A$ *and* Π_2 *is a proof of* $\delta_1 \odot \Delta_1; (\Gamma_1, A, \Gamma_3) \vdash_{\mathsf{MS}} B$ *with* $\mathsf{CutRank}\,(\Pi_1), \mathsf{CutRank}\,(\Pi_2) \leq \mathsf{Rank}(A)$*, then there exists a proof* Π *of* $(\delta_1, \delta_2) \odot (\Delta_1, \Delta_2); (\Gamma_1, \Gamma_2, \Gamma_3) \vdash_{\mathsf{MS}} B$ *with* $\mathsf{CutRank}\,(\Pi) \leq \mathsf{Rank}(A)$.

Proof. By mutual induction on $\mathsf{Depth}(\Pi_1) + \mathsf{Depth}(\Pi_2)$ (see Appendix C.1 [40] for proof). ◀

▶ **Lemma 10** (Decreasing Order of mGL). *If* Π *is a proof of* $\delta \odot \Delta \vdash_{\mathsf{GS}} X$ *or* $\delta \odot \Delta; \Gamma \vdash_{\mathsf{MS}} A$ *with* $\mathsf{CutRank}\,(\Pi) > 0$*, then there is a proof* Π' *of* $\delta' \odot \Delta \vdash_{\mathsf{GS}} X$ *or* $\delta' \odot \Delta; \Gamma \vdash_{\mathsf{MS}} A$ *with* $\delta \leq \delta'$ *and* $\mathsf{CutRank}\,(\Pi') < \mathsf{CutRank}\,(\Pi)$.

Proof. By induction on $\mathsf{Depth}(\Pi)$ (see Appendix C.2 [40] for proof). ◀

▶ **Theorem 11** (Cut Elimination of mGL). *If* Π *is a proof of* $\delta \odot \Delta \vdash_{\mathsf{GS}} X$ *or* $\delta \odot \Delta; \Gamma \vdash_{\mathsf{MS}} A$ *with* $\mathsf{CutRank}\,(\Pi) > 0$*, then there is an algorithm which yields a cut-free proof* Π' *of* $\delta \odot \Delta \vdash_{\mathsf{GS}} X$ *or* $\delta \odot \Delta; \Gamma \vdash_{\mathsf{MS}} A$ *respectively.*

Proof. Follows immediately by induction on $\mathsf{CutRank}\,(\Pi)$ and the previous lemma. ◀

▶ **Lemma 12** (Subformula property).

1. *(Graded) Every formula occurring in a cut-free proof* Π *of a judgment,* $\delta \odot \Delta \vdash_{\mathsf{GS}} X$*, consists of subformulas of the formulas occurring in* $\delta \odot \Delta \vdash_{\mathsf{GS}} X$.
2. *(Mixed) Every formula occurring in a cut-free proof* Π *of a judgment,* $\delta \odot \Delta; \Gamma \vdash_{\mathsf{MS}} A$*, consists of subformulas of the formulas occurring in* $\delta \odot \Delta; \Gamma \vdash_{\mathsf{MS}} A$.

Proof. By induction on Π (See Appendix C.3 [40] for proof). ◀

Lastly, we define an equational theory for mGL:

▶ **Definition 13** (Equational theory $\equiv$, subset). *An equational theory on derivations accounts for equalities between proofs of the same sequent arising from the graded structure (where the terms are the same but the structure of the proof tree differs), as well as cut elimination, i.e., in* GS*, if cut elimination on derivation* Π_1 *of* $\delta \odot \Delta \vdash_{\mathsf{GS}} t : X$ *yields the cut-free derivation of* Π_2 *for* $\delta \odot \Delta \vdash_{\mathsf{GS}} t' : X$ *then the equational theory has* $\Pi_1 \equiv \Pi_2$*, and similarly for* MS*.*

As a sample of two equations from the GS *fragment, the following shows an equation leveraging the commutativity of contraction, and another on the interaction between weakening and contraction leveraging the left-unit of semiring addition:*

$$\dfrac{\dfrac{\delta_1, r, \delta_2 \odot \Delta_1, X, \Delta_2 \vdash_{\mathsf{GS}} t : Y}{\delta_1, 0, r, \delta_2 \odot \Delta_1, X, X, \Delta_2 \vdash_{\mathsf{GS}} t : Y}\, \mathit{WEAK}_{\mathsf{GS}}}{\delta_1, 0 + r, \delta_2 \odot \Delta_1, X, \Delta_2 \vdash_{\mathsf{GS}} t : Y}\, \mathit{CONT}_{\mathsf{GS}} \equiv \delta_1, r, \delta_2 \odot \Delta_1, X, \Delta_2 \vdash_{\mathsf{GS}} t : Y \quad (\textsc{contr-unitL})$$

$$\dfrac{\dfrac{\delta_1, r, s, \delta_2 \odot \Delta_1, X, X, \Delta_2 \vdash_{\mathsf{GS}} t : Y}{\delta_1, s, r, \delta_2 \odot \Delta_1, X, X, \Delta_2 \vdash_{\mathsf{GS}} t : Y}\, \mathit{EX}_{\mathsf{GS}}}{\delta_1, s + r, \delta_2 \odot \Delta_1, X, \Delta_2 \vdash_{\mathsf{GS}} t : Y}\, \mathit{CONT}_{\mathsf{GS}} \equiv \dfrac{\delta_1, r, s, \delta_2 \odot \Delta_1, X, X, \Delta_2 \vdash_{\mathsf{GS}} t : Y}{\delta_1, r + s, \delta_2 \odot \Delta_1, X, X, \Delta_2 \vdash_{\mathsf{GS}} t : Y}\, \mathit{CONT}_{\mathsf{GS}} \quad (\textsc{contr-sym})$$

Appendix A.1 [40] gives the full definition of the equational theory.

▶ Remark 14 ("$\beta\eta$-equalities" and "Triangle identities" via cut reduction). One might wonder where β-equalities are in the above equational theory, e.g., that $(\lambda x.l)l'$ in MS is equal to the cut $[l'/x]l$. Such β-equalities are provided by the cut elimination procedure, which reduces away interacting pairs of right and left formulas (the principal vs. principal cases).

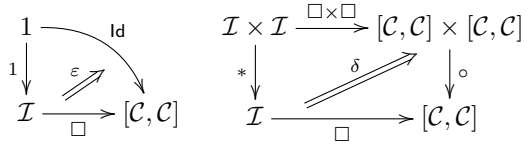
Similarly, η-equalities are equivalent to the identity expansion part of cut elimination procedure (where the cut of an identity axiom is transformed into an interacting left and right pair, with identity axioms expanded towards the leaves).

The internal derivations for the graded equivalent of the "triangle identities' (that one usually has associated with an adjunction) are also handled in the cut elimination procedure. The main feature of the derivations for both identities is that after one step the left and right rules for the modal operators match up. This leads to consecutive principal vs. principal cases where rules for the interacting left and right pairs in the two subproofs are removed by the reduction step.

3 Model

We detail a denotational model for mGL which is based on an adjoint decomposition of graded comonads. We introduce key definitions as needed.

A *graded comonad* can be summarised as a colax monoidal functor $\Box : \mathcal{I} \longrightarrow [\mathcal{C}, \mathcal{C}]$ where $\mathcal{I}$ is a preordered monoid $(\mathcal{I}, 1, *, \leq)$ treated as a monoidal category and $[\mathcal{C}, \mathcal{C}]$ is the category of endofunctors on $\mathcal{C}$ [33, 34]. Colax monoidality of $\Box$ means that the laws of a monoidal functor become 2-cells, providing the graded comonad operations:



which are thus natural transformations $\varepsilon_A : \Box_1 A \to A.$ and $\delta_{r,s,A} : \Box_{(r*s)} A \to \Box_r(\Box_s A)$.

Fujii et al. [10] gave a formal theory for graded monads, which can be easily dualised to graded comonads, showing that in an analogous way to an ordinary comonad, every graded comonad can be decomposed into an adjunction $\mathsf{Mny} \dashv \mathsf{Lin} : \mathcal{M} \longrightarrow \mathcal{C}$ and (key to *graded* comonads) a monoidal action $\odot : \mathcal{R} \times \mathcal{C} \longrightarrow \mathcal{C}$, and thus vice versa:

▶ **Lemma 15.** *(Resolution of a graded comonad [24, 10]) An adjunction* $\mathsf{L} \dashv \mathsf{R} : \mathcal{M} \longrightarrow \mathcal{C}$ *and a strict monoidal action* $\odot : \mathcal{R} \times \mathcal{C} \longrightarrow \mathcal{C}$ *together induce a graded comonad over the family of endofunctors defined by* $\Box_r = \mathsf{L}(r \odot (\mathsf{R}-)) : \mathcal{M} \longrightarrow \mathcal{M}$.

Along with some additional structure relating to substructurality (see below), this result provides a model of mGL with $\mathcal{C}$ providing a model for GS derivations, $\mathcal{M}$ providing a model for MS derivations, the type constructor Grd_r transporting from GS to MS modelled by $\mathsf{L}(r \odot -) : \mathcal{C} \to \mathcal{M}$, and type constructor Lin transporting MS to GS modelled by $\mathsf{R} : \mathcal{M} \to \mathcal{C}$.

However we need additional structure for the (sub)structural behaviour of our logic. In the literature on graded modal type theories, graded comonads are extended to *graded exponential comonads* (sometimes called *graded linear exponential comonads* [24]) defined as a colax monoidal functor $\Box : \mathcal{R} \longrightarrow [\mathcal{M}, \mathcal{M}]$ where $\mathcal{R}$ is a preordered semiring $(\mathcal{R}, 1, *, 0, +, \leq)$ (viewed as a category), $[\mathcal{M}, \mathcal{M}]$ is the category of symmetric lax monoidal endofunctors on a symmetric monoidal category $\mathcal{M}$, and $\Box$ has additional symmetric lax monoidal structure for the additional monoidality of $\mathcal{R}$ and $[\mathcal{M}, \mathcal{M}]$ [12]. This additional structure provides natural transformations $w_A : \Box_0 A \longrightarrow 1$ and $c_{r,s,A} : \Box_{(r+s)} A \longrightarrow (\Box_r A) \otimes (\Box_s A)$ capturing (graded) weakening and contraction, subject to comonoidal coherence conditions. This additional structure can be induced by the adjoint decomposition given an *exponential action*:

▶ **Definition 16** (Exponential action)**.** *Given a preordered semiring* $(\mathcal{R}, 1, *, 0, +, \leq)$ *and a symmetric monoidal category* $(\mathcal{C}, \mathsf{J}, \boxtimes)$*, we say that a bifunctor* $\odot : \mathcal{R} \times C \longrightarrow \mathcal{C}$ *is*

1. *a* strict action *(a* strict graded comonad*), if it satisfies the following equalities:*

$$\begin{aligned} \varepsilon_X &: \quad 1 \odot X = X \\ \delta_{X,r,s} &: (r * s) \odot X = r \odot (s \odot X) \end{aligned}$$

   Note that we treat these equalities as strict natural transformations named ε *and* δ*;*
2. symmetric lax monoidal *in the second argument if it has:*

$$\begin{aligned} m_{\mathsf{J},r} &: \quad \mathsf{J} \to r \odot \mathsf{J} \\ m_{\boxtimes,r,X,Y} &: (r \odot X) \boxtimes (r \odot Y) \to r \odot (X \boxtimes Y) \end{aligned}$$

   where m_{J} *is the unit of* $m_{\boxtimes}$ *and* $m_{\boxtimes}$ *is associative and commutative up to isomorphism;*
3. symmetric colax monoidal *between* $(\mathcal{R}, 0, +, \leq)$ *and* $(\mathcal{C}, \mathsf{J}, \boxtimes)$ *in the first argument if it has natural transformations:*

$$\begin{aligned} \mathsf{weak}_X &: \quad 0 \odot X \to \mathsf{J} \\ \mathsf{contr}_{r,s,X} &: (r + s) \odot X \to (r \odot X) \boxtimes (s \odot X) \end{aligned}$$

   where weak *is the unit of* contr*, e.g.* $\rho_{r \odot X} \circ (id \boxtimes \mathsf{weak}_X) \circ \mathsf{contr}_{r,0,X} = id$ *with right unitor* ρ*, and* contr *is associative and commutative, i.e., that* $\mathsf{contr}_{r,s,X} = c \circ \mathsf{contr}_{s,r,X}$*. Furthermore, these natural transformations must be preserved by the strict action and monoidal structure as described by the standard additional equations in Figure 2.*

If we have all of the above properties then we refer to $\odot$ as an *exponential action*. This terminology recalls the *exponential action* of Brunel et al. [6] which is the same as the above but where strictness is instead laxness in their definition. Our definition is also similar to linear exponential graded comonads (see e.g., [24, 12]), but here the graded comonad is uncurried (in the form of an action) and has equalities for its natural transformations (strictness).

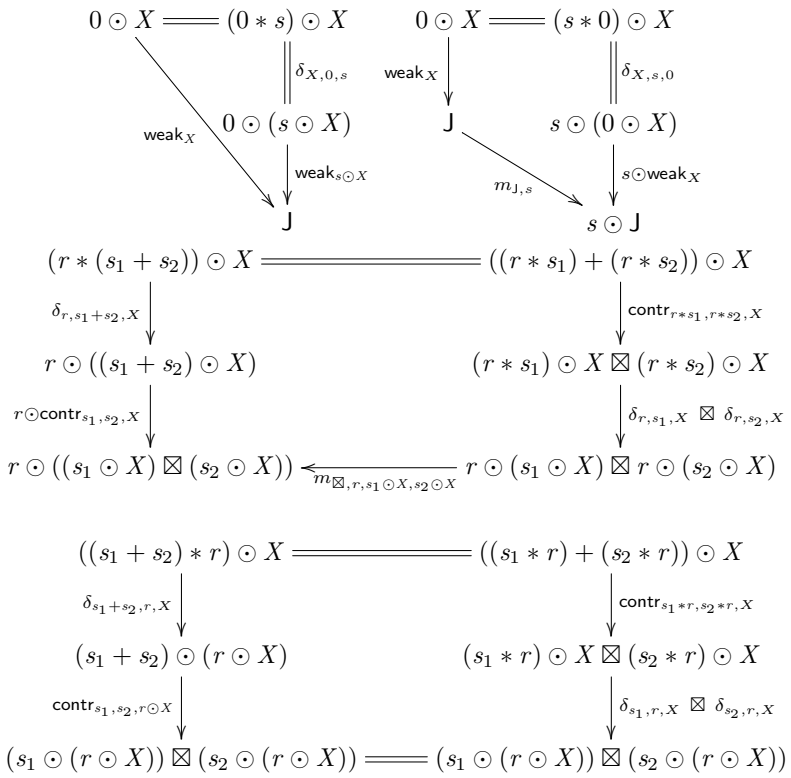


Figure 2 Further equations of a strict exponential action, interacting the colax symmetric monoidal structure, strict action, and (strict) monoidality.

We define a *strict exponential action* to be an exponential action as above but where the monoidal structure m_{J} and $m_{\boxtimes}$ is also strict, where for clarity (in the appendix) we sometimes orient the equality as a morphism, where in the opposite direction we denote these morphisms by $n_{\mathsf{J},r}$ and $n_{\boxtimes,r,X,Y}$ respectively. Strictness of the monoidal structure is needed for soundness of our model.

We now give the definition of the model of mGL, where we now use the opposite category $\mathcal{R}^{\mathsf{op}}$ to capture the correct polarity of the approximation rules.

▶ **Definition 17** (Mixed Graded/Linear model). *Suppose* $(\mathcal{C}, \mathsf{J}, \boxtimes)$ *and* $(\mathcal{M}, \mathsf{I}, \otimes)$ *are symmetric monoidal categories, where* $\mathcal{M}$ *is symmetric monoidal closed (with exponents* $\multimap$*), and* $(\mathcal{R}, 1, *, 0, +, \leq)$ *is a preordered semiring. Then a* Mixed Graded/Linear model *is a symmetric monoidal adjunction* $\mathsf{Mny} \dashv \mathsf{Lin} : \mathcal{M} \longrightarrow \mathcal{C}$ *along with an exponential action* $\odot : \mathcal{R}^{\mathsf{op}} \times \mathcal{C} \longrightarrow \mathcal{C}$.

Thus an mGL model is essentially an LNL model with a strict action. However, whilst Benton's LNL models are initially stated to require that $\mathcal{M}$ is Cartesian closed, he goes on to show that Cartesian properties are induced for the Eilenberg-Moore category of !-coalgebras for a symmetric monoidal category [3]. In our setting, the Cartesian structure is not needed since the MS logic is a mix of graded and linear logic, rather than Cartesian and linear logic. That is, graded propositions do not have arbitrary weakening and contraction, but instead these structural rules are controlled by grades (and corresponding underlying categorical structure [12, 24]). Therefore, a symmetric monoidal closed $\mathcal{M}$ suffices.

From Definition 17, we define our denotational model of mGL:

▶ **Definition 18** (Interpretation of Mixed Graded/Linear Logic.). *Given a Mixed Graded/Linear model (Def. 17) (with* $\mathsf{Mny} \dashv \mathsf{Lin} : \mathcal{M} \longrightarrow \mathcal{C}$ *and* $\odot : \mathcal{R}^{\mathsf{op}} \times \mathcal{C} \longrightarrow \mathcal{C}$*), we interpret by two mutually defined interpretations* $[\![-]\!]^{\mathsf{GS}}$ *and* $[\![-]\!]^{\mathsf{MS}}$ *on types and proofs (derivations):*

- *For every* GS *type* X *there is an object* $[\![X]\!]^{\mathsf{GS}} \in \mathcal{C}$ *and for every* MS *type* A *there is an object* $[\![A]\!]^{\mathsf{MS}} \in \mathcal{M}$*, mutually defined inductively as:*

$$\begin{array}{rlrl} [\![\mathsf{J}]\!]^{\mathsf{GS}} &= \mathsf{J} & [\![\mathsf{I}]\!]^{\mathsf{MS}} &= \mathsf{I} \\ [\![X \boxtimes Y]\!]^{\mathsf{GS}} &= [\![X]\!]^{\mathsf{GS}} \boxtimes [\![Y]\!]^{\mathsf{GS}} & [\![A \otimes B]\!]^{\mathsf{MS}} &= [\![A]\!]^{\mathsf{MS}} \otimes [\![B]\!]^{\mathsf{MS}} \\ [\![\mathsf{Lin}\ A]\!]^{\mathsf{GS}} &= \mathsf{Lin}[\![A]\!]^{\mathsf{MS}} & [\![A \multimap B]\!]^{\mathsf{MS}} &= [\![A]\!]^{\mathsf{MS}} \multimap [\![B]\!]^{\mathsf{MS}} \\ & & [\![\mathsf{Grd}_r\ X]\!]^{\mathsf{MS}} &= \mathsf{Mny}(r \odot [\![X]\!]^{\mathsf{GS}}) \end{array}$$

- *For every proof* Π *of a* GS *sequent* $(r_1, \ldots, r_n) \odot (x_1 : X_1, \ldots, x_n : X_n) \vdash_{\mathsf{GS}} t : X$ *there is a morphism in the category* $\mathcal{C}$*:*

$$[\![\Pi]\!]^{\mathsf{GS}} : (r_1 \odot [\![X_1]\!]^{\mathsf{GS}}) \boxtimes \ldots \boxtimes (r_n \odot [\![X_n]\!]^{\mathsf{GS}}) \longrightarrow [\![X]\!]$$

  (where an empty context is interpreted as $\emptyset^{\mathsf{GS}} = \mathsf{J}$*).*

- *For every proof* Π *of an* MS *sequent* $(r_1, \ldots, r_n) \odot (x_1 : X_1, \ldots, x_n : X_n); y_1 : A_1, \ldots, y_m : A_m \vdash_{\mathsf{MS}} l : B$ *there is a morphism in the category* $\mathcal{M}$*:*

$$[\![\Pi]\!]^{\mathsf{MS}} : \mathsf{Mny}(r_1 \odot [\![X_1]\!]^{\mathsf{GS}}) \otimes \ldots \otimes \mathsf{Mny}(r_n \odot [\![X_n]\!]^{\mathsf{GS}}) \otimes [\![A_1]\!]^{\mathsf{MS}} \otimes \ldots \otimes [\![A_m]\!]^{\mathsf{MS}} \longrightarrow [\![B]\!]^{\mathsf{MS}}$$

  (where an empty MS *context is interpreted as* $\emptyset^{\mathsf{MS}} = \mathsf{I}$*).*

Appendix C.4 [40] gives the full definition of the interpretation, including intermediate derivations from the mGL *model.*

Finally, we have our soundness and completeness theorems:

▶ **Theorem 19** (Soundness of Mixed Graded/Linear Logic models). *Suppose a mixed graded/-linear model as above. Then for derivation* Π_1 *of* $\delta \odot \Delta \vdash_{\mathsf{GS}} t_1 : X$ *and derivation* Π_2 *of* $\delta \odot \Delta \vdash_{\mathsf{GS}} t_2 : X$ *then if* $\Pi_1 \equiv \Pi_2$ *then* $[\![\Pi_1]\!] = [\![\Pi_2]\!]$*.*

Similarly for Π_1 *of* $\delta \odot \Delta; \Gamma \vdash_{\mathsf{MS}} l_1 : A$ *and derivation* Π_2 *of* $\delta \odot \Delta; \Gamma \vdash_{\mathsf{MS}} l_2 : A$ *then if* $\Pi_1 \equiv \Pi_2$ *then* $[\![\Pi_1]\!] = [\![\Pi_2]\!]$*.*

Proof. This proof holds by mutual induction. For the details see Appendix C.5 [40]. ◀

▶ **Theorem 20** (Completeness of Mixed Graded/Linear Logic models). *For derivations* Π_1*,* Π_2 *(of either* GS *or* MS*) if* $[\![\Pi_1]\!] = [\![\Pi_2]\!]$ *in all mixed graded/linear models, then* $\Pi_1 \equiv \Pi_2$*.*

Proof. This is a standard proof, where we build a generic model based on the syntax and the equational theory. For the details see Appendix C.6 [40]. ◀

4 Natural Deduction

We now develop a natural deduction formulation of mGL. Whilst sequent calculus judgments were denoted $\vdash_{\mathsf{MS}}$ and $\vdash_{\mathsf{GS}}$, natural deduction judgments are correspondingly $\vdash_{\mathsf{MT}}$ and $\vdash_{\mathsf{GT}}$.

The syntax for terms is identical to the sequent calculus, collected in Figure 1. Appendix B [40] gives the introduction and elimination rules and structural rules for mGL's natural deduction formulation. The unit constructors are j and i. Tensor products in both systems are denoted by pairs of terms with corresponding let-expressions for eliminators. The graded modal introduction form $\mathsf{Lin}\ l$ operates on mixed terms, dual to $\mathsf{Grd}\ r\ t$ which operates on graded terms. The mixed syntax includes abstraction $\lambda x.l$ and function application $l_1\ l_2$. The most interesting aspect is the rules for the modal operators:

$$\mathsf{Lin}_I\ \frac{\delta \odot \Delta; \emptyset \vdash_{\mathsf{MT}} l : B}{\delta \odot \Delta \vdash_{\mathsf{GT}} \mathsf{Lin}\, l : \mathsf{Lin}\, B} \qquad \mathsf{Lin}_E\ \frac{\delta \odot \Delta \vdash_{\mathsf{GT}} t : \mathsf{Lin}\, A}{\delta \odot \Delta; \emptyset \vdash_{\mathsf{MT}} \mathsf{Unlin}\, t : A}$$

$$\mathsf{Grd}_I\ \frac{\delta \odot \Delta \vdash_{\mathsf{GT}} t : X}{r * \delta \odot \Delta; \emptyset \vdash_{\mathsf{MT}} \mathsf{Grd}\, r\, t : \mathsf{Grd}_r\, X} \qquad \mathsf{Grd}_E\ \frac{\delta_2 \odot \Delta_2; \Gamma_2 \vdash_{\mathsf{MT}} l_1 : \mathsf{Grd}_r\, X \quad (\delta_1, r, \delta_3) \odot (\Delta_1, x : X, \Delta_3); \Gamma_1 \vdash_{\mathsf{MT}} l_2 : B}{(\delta_1, \delta_2, \delta_3) \odot (\Delta_1, \Delta_2, \Delta_3); (\Gamma_1, \Gamma_2) \vdash_{\mathsf{MT}} \mathsf{let}\, \mathsf{Grd}\, r\, x = l_1\, \mathsf{in}\, l_2 : B}$$

In the sequent calculus presented in Section 2, the right rule for Lin is in the graded subsystem, but the left rule is in the mixed subsystem. A similar idea arises here, the introduction rule for Lin (rule Lin_I) is in the graded subsystem and the elimination rule (rule Lin_E) is in the mixed subsystem. Introducing Grd_r formulas (rule Grd_I) has the effect of scaling the input grades by r. The elimination rule for Grd_r (rule Grd_E) is a pattern match on the form of l_1. Since Lin and Grd are the decomposition of graded modalities (Section 3), the form of the elimination rule for Grd_r is defined in a way which resembles that of elimination rules for graded modalities in other natural deduction-based type systems [31].

This formulation also has explicit graded structural rules:

$$\text{WEAK}\ \frac{(\delta_1, \delta_2) \odot (\Delta_1, \Delta_2) \vdash_{\mathsf{GT}} t : Y}{(\delta_1, 0, \delta_2) \odot (\Delta_1, x : X, \Delta_2) \vdash_{\mathsf{GT}} t : Y} \qquad \text{CONT}\ \frac{(\delta_1, r_1, r_2, \delta_2) \odot (\Delta_1, x : X, y : X, \Delta_2) \vdash_{\mathsf{GT}} t : Y}{(\delta_1, r_1 + r_2, \delta_2) \odot (\Delta_1, x : X, \Delta_2) \vdash_{\mathsf{GT}} [x/y]t : Y}$$

$$\text{EX}\ \frac{(\delta_1, r_1, r_2, \delta_2) \odot (\Delta_1, x : X, y : Y, \Delta_2) \vdash_{\mathsf{GT}} t : Z}{(\delta_1, r_2, r_1, \delta_2) \odot (\Delta_1, y : Y, x : X, \Delta_2) \vdash_{\mathsf{GT}} t : Z}$$

In the transition from sequent calculus to natural deduction, left rules transform into elimination rules, and as a result the additional graded left rules in the mixed sequent calculus are no longer explicitly part of the system, but can be derived. We go on to prove that the sequent calculus of Section 2.1 is equivalent to the natural deduction system.

We give two main results related to the natural deduction system; the first is substitution for typing. Note that this reuses the row-vector multiplication operation of Section 2.2.

▶ **Lemma 21** (Substitution for $\vdash_{\mathsf{GT}}$ and $\vdash_{\mathsf{MT}}$). *The following hold by mutual induction:*

1. *(Graded) If* $\delta_2 \odot \Delta_2 \vdash_{\mathsf{GT}} t_1 : X$ *and* $(\delta_1, \delta, \delta_3) \odot (\Delta_1, x^n : X^n, \Delta_3) \vdash_{\mathsf{GT}} t_2 : Y$*, then* $(\delta_1, \delta \circledast [{\delta_2}^n], \delta_3) \odot (\Delta_1, \Delta_2, \Delta_3) \vdash_{\mathsf{GT}} [t_1, \ldots, t_1/x_1, \ldots, x_n]t_2 : Y$.
2. *(Graded/Mixed) If* $\delta_2 \odot \Delta_2 \vdash_{\mathsf{GT}} t : X$ *and* $(\delta_1, \delta, \delta_3) \odot (\Delta_1, x^n : X^n, \Delta_3); \Gamma \vdash_{\mathsf{MT}} l : B$*, then* $(\delta_1, \delta \circledast [{\delta_2}^n], \delta_3) \odot (\Delta_1, \Delta_2, \Delta_3); \Gamma \vdash_{\mathsf{MT}} [t, \ldots, t/x_1, \ldots, x_n]l : B$.
3. *(Mixed) If* $\delta_2 \odot \Delta_2; \Gamma_2 \vdash_{\mathsf{MT}} l_1 : A$ *and* $\delta_1 \odot \Delta_1; (\Gamma_1, x : A, \Gamma_3) \vdash_{\mathsf{MT}} l_2 : B$*, then* $(\delta_1, \delta_2) \odot (\Delta_1, \Delta_2); (\Gamma_1, \Gamma_2, \Gamma_3) \vdash_{\mathsf{MT}} [l_1/x]l_2 : B$.

Proof. By mutual induction on the second assumed derivation (see Appendix C.7 [40]). ◀

Since we have an explicit structural rule for contraction (above and listed in Appendix B [40]), the substitution lemma on the graded fragment is formalized as multi-substitution. Otherwise, its proof is a fairly standard substitution proof for graded systems (e.g., as in [31]). Lastly, the natural deduction system is interderivable with the sequent calculus, which we establish such that the term witnessing the derivations does not change between systems:

▶ **Theorem 22** (Sequent calculus and natural deduction interderivability). $\delta \odot \Delta \vdash_{\mathsf{GS}} t : X \Leftrightarrow \delta \odot \Delta \vdash_{\mathsf{GT}} t : X$ *and* $\delta \odot \Delta; \Gamma \vdash_{\mathsf{MS}} l : A \Leftrightarrow \delta \odot \Delta; \Gamma \vdash_{\mathsf{MT}} l : A$.

Proof. By mutual induction on the assumed derivations (Appendix C.8 and C.9 [40]). The sequent calculus to natural deduction direction requires the substitution lemma above. ◀

The implication of the previous result is that we only need a semantic model of one of the two systems, and the other can be modelled using the same interpretation of terms. We chose to model the sequent calculus form directly.

5 Discussion

5.1 Relating linear base vs. graded base calculi

A major thread of graded type systems in the literature starts with a linear logic base and then generalises the ! modality to a semiring-graded modality atop a linear logic, e.g., the systems of Brunel et al. [6], Gaboardi et al. [12], Orchard et al. [31], and others [11, 18]. Often these systems are presented with a single context containing both linear and graded propositions [12, 31]. Overall, these approaches have a common core which is isomorphic to the natural deduction MT fragment shown here with the (natural deduction analogue of the) derived $\Box_r$ graded modality of Lemma 3 as part of their definition (i.e., not derived). We refer to this style of graded type system as the *linear base* style.

A contrasting approach has no base notion of linearity, but instead has pervasive grading tracking substructurality, i.e., no linear assumptions, every assumption has a grade, and function arrows come equipped with a grade describing the usage of their input in the function (e.g., written $A \xrightarrow{r} B$). Such systems include the coeffect calculi of Petricek et al. [33, 34], the general graded modal system of Bernardy et al. [1], and several others [2, 4, 7, 28, 30]. The GT fragment of our system here corresponds to a common subset of these approaches: a subset without function arrows and without a graded modality, since there is no graded modality that lives in the GT side ($\Box_i$ is derived into MT). Hughes et al. also develop a program synthesis technique for graded base systems, where grades are used to prune the search space [19]; its synthesis calculus formulation resembles closely GS.

Our work thus shows the relationship between the linear base and graded base style, namely that there is a mutual embedding between these two approaches which generates the graded modality in the linear base (Lemma 3). Exploring this in more depth is further work. For instance, it is unclear what is needed to realise a graded comonadic modality in GT that arises from the embedding (or a different embedding), and how this could interact with a graded function arrow in GS or GT. Pursuing this line of work would help to explain the relationship between the two dominant styles of graded system in the literature, which seem strongly related, and their relative expressive power. Nonetheless, by following Benton's programme and giving it a graded rendering here, we can already see here the close connection between these two styles of graded system.

5.2 Related work on adjoint logics

Pruiksma et al. formalized a general way to add and remove structural rules from a logic through adjunctions [37]. Their work is similar to ours as it relates logics through adjoint decompositions based on modal operators to control structural rules. Their formulation with "modes of truth" resembles our work with grades; however, modes of truth lack the algebraic properties graded formulations depend on and instead have a very relational flavor. Building on this work of Pruiksma et al., Jang et al. develop a natural deduction formulation

of adjoint logic [21]. They leverage this to give a functional language able to reason about resource properties like strictness and erasure. Similar reasoning can be developed ontop of our natural deduction formulation here, though this is left as further work.

A question is whether grading can be unified with the adjoint logic approach. Eades and Orchard sketched a unification based on generalising semiring operations to relations rather than functions, with predicates classifying unit values [20]. Hanukaev et al. develop this idea further, introducing a dependent type system based on a similar structure as the logics here but using a generalised notion of grading that combines the modes of adjoint logic [16]. They prove that their system is well-formed syntactically, but do not introduce any semantic model. Our logic mGL can be seen as an instantiation of their system, but the categorical model given here could potentially be generalised into a model of their system.

5.3 Further work

Practical implementation to leverage linear/grading separation

The separation of the mixed system (MS/MT) from the purely graded fragment (GS/GT) (which acts more as a standard intuitionistic system) can provide a basis for a programming language design. In such a language, the restrictions of linearity could be used only for handling data that needs to be linear, such as file handles or channels. However, for data types which need not be linear, e.g., primitive types like integers, characters, or structures over them, the graded fragment could be used without having to confront linearity constraints. The mutual embedding would allow the programmer to move smoothly between these two subcalculi. Similar ideas are discussed for the polarized extension of SILL for concurrent programming [35]. The implementation could borrow ideas from the Granule programming language, which already provides a mature and feature rich implementation of a linear-base style graded type system [31]. Instead, an mGL-inspired implementation could be based on the natural deduction term calculus with the modalities mediating between the two judgments. Exploring this application, perhaps as an extension to Granule, is future work.

Other generalisations

In LNL, the adjunction can be followed in the opposite direction to derive a monad $?A = \mathsf{Lin}(\mathsf{Mny}A)$. However, in mGL we do not get a graded monad by composing $\mathsf{Lin}\,(\mathsf{Grd}_r\,X)$ since the adjoint resolution of a graded monad has a strict action on the other side (on $\mathcal{M}$ in the model). Exploring a calculus with a pair of actions to allow both graded monads and graded comonads is further work.

Uniqueness typing

Recent work has demonstrated that *uniqueness* is a closely related but distinct concept to linearity [27]; uniqueness logic [17] is substructural in much the same way as linear logic, but provides a monadic modality for enabling the structural rules in contrast to linear logic's comonadic ! modality. Building an adjoint model for uniqueness or a calculus which unifies uniqueness and linearity [27] would be interesting further work, and this could potentially be extended to more recent systems which develop graded notions of uniqueness [26].

References

1 Andreas Abel and Jean-Philippe Bernardy. A unified view of modalities in type systems. *Proc. ACM Program. Lang.*, 4(ICFP):90:1–90:28, 2020. doi:10.1145/3408972.

2 Robert Atkey. Syntax and Semantics of Quantitative Type Theory. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 56–65. ACM, 2018. doi:10.1145/3209108.3209189.

3 P. N. Benton. A mixed linear and non-linear logic: Proofs, terms and models (extended abstract). In Leszek Pacholski and Jerzy Tiuryn, editors, *Computer Science Logic, 8th International Workshop, CSL '94, Kazimierz, Poland, September 25-30, 1994, Selected Papers*, volume 933 of *Lecture Notes in Computer Science*, pages 121–135. Springer, 1994. doi:10.1007/BFB0022251.

4 Jean-Philippe Bernardy, Mathieu Boespflug, Ryan R. Newton, Simon Peyton Jones, and Arnaud Spiwack. Linear Haskell: practical linearity in a higher-order polymorphic language. *Proc. ACM Program. Lang.*, 2(POPL):5:1–5:29, 2018. doi:10.1145/3158093.

5 Flavien Breuvart and Michele Pagani. Modelling coeffects in the relational semantics of linear logic. In *24th EACSL Annual Conference on Computer Science Logic, CSL 2015, September 7-10, 2015, Berlin, Germany*, pages 567–581, 2015. doi:10.4230/LIPIcs.CSL.2015.567.

6 Aloïs Brunel, Marco Gaboardi, Damiano Mazza, and Steve Zdancewic. A core quantitative coeffect calculus. In Zhong Shao, editor, *Programming Languages and Systems - 23rd European Symposium on Programming, ESOP 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014, Proceedings*, volume 8410 of *Lecture Notes in Computer Science*, pages 351–370. Springer, 2014. doi:10.1007/978-3-642-54833-8_19.

7 Pritam Choudhury, Harley Eades III, Richard A. Eisenberg, and Stephanie Weirich. A graded dependent type system with a usage-aware semantics. *Proc. ACM Program. Lang.*, 5(POPL):1–32, 2021. doi:10.1145/3434331.

8 Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. The Structure of Exponentials: Uncovering the Dynamics of Linear Logic Proofs. In Georg Gottlob, Alexander Leitsch, and Daniele Mundici, editors, *Computational Logic and Proof Theory, Third Kurt Gödel Colloquium, KGC'93, Brno, Czech Republic, August 24-27, 1993, Proceedings*, volume 713 of *Lecture Notes in Computer Science*, pages 159–171. Springer, 1993. doi:10.1007/BFB0022564.

9 Loris D'Antoni, Marco Gaboardi, Emilio Jesús Gallego Arias, Andreas Haeberlen, and Benjamin C. Pierce. Sensitivity analysis using type-based constraints. In *Proceedings of the 1st Annual Workshop on Functional Programming Concepts in Domain-Specific Languages, FPCDSL@ICFP 2013, Boston, Massachusetts, USA, September 22, 2013*, pages 43–50, 2013. doi:10.1145/2505351.2505353.

10 Soichiro Fujii, Shin-ya Katsumata, and Paul-André Melliès. Towards a Formal Theory of Graded Monads. In Bart Jacobs and Christof Löding, editors, *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*, volume 9634 of *Lecture Notes in Computer Science*, pages 513–530. Springer, 2016. doi:10.1007/978-3-662-49630-5_30.

11 Marco Gaboardi, Andreas Haeberlen, Justin Hsu, Arjun Narayan, and Benjamin C. Pierce. Linear dependent types for differential privacy. In *The 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '13, Rome, Italy - January 23 - 25, 2013*, pages 357–370, 2013. doi:10.1145/2429069.2429113.

12 Marco Gaboardi, Shin-ya Katsumata, Dominic A. Orchard, Flavien Breuvart, and Tarmo Uustalu. Combining effects and coeffects via grading. In *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming, ICFP 2016, Nara, Japan, September 18-22, 2016*, pages 476–489, 2016. doi:10.1145/2951913.2951939.

13 Dan R. Ghica and Alex I. Smith. Bounded linear types in a resource semiring. In Zhong Shao, editor, *Programming Languages and Systems - 23rd European Symposium on Programming, ESOP 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014, Proceedings*, volume 8410 of *Lecture Notes in Computer Science*, pages 331–350. Springer, 2014. doi:10.1007/978-3-642-54833-8_18.

14 Jean-Yves Girard. Linear logic. *Theor. Comput. Sci.*, 50:1–102, 1987. doi:10.1016/0304-3975(87)90045-4.

15 Jean-Yves Girard, Andre Scedrov, and Philip J. Scott. Bounded linear logic: A modular approach to polynomial-time computability. *Theor. Comput. Sci.*, 97(1):1–66, 1992. doi:10.1016/0304-3975(92)90386-T.

16 Peter Hanukaev and Harley Eades III. Combining dependency, grades, and adjoint logic. In *Proceedings of the 8th ACM SIGPLAN International Workshop on Type-Driven Development*, TyDe 2023, pages 58–70, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3609027.3609408.

17 Dana Harrington. Uniqueness logic. *Theor. Comput. Sci.*, 354(1):24–41, 2006. doi:10.1016/j.tcs.2005.11.006.

18 Jack Hughes, Danielle Marshall, James Wood, and Dominic Orchard. Linear Exponentials as Graded Modal Types. In *5th International Workshop on Trends in Linear Logic and Applications (TLLA 2021)*, Rome (virtual), Italy, June 2021. URL: https://hal-lirmm.ccsd.cnrs.fr/lirmm-03271465.

19 Jack Hughes and Dominic Orchard. Program synthesis from graded types. In Stephanie Weirich, editor, *Programming Languages and Systems - 33rd European Symposium on Programming, ESOP 2024, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2024, Luxembourg City, Luxembourg, April 6-11, 2024, Proceedings, Part I*, volume 14576 of *Lecture Notes in Computer Science*, pages 83–112. Springer, 2024. doi:10.1007/978-3-031-57262-3_4.

20 Harley Eades III and Dominic Orchard. Grading adjoint logic. *CoRR*, abs/2006.08854, 2020. arXiv:2006.08854.

21 Junyoung Jang, Sophia Roshal, Frank Pfenning, and Brigitte Pientka. Adjoint Natural Deduction. In Jakob Rehof, editor, *9th International Conference on Formal Structures for Computation and Deduction, FSCD 2024, July 10-13, 2024, Tallinn, Estonia*, volume 299 of *LIPIcs*, pages 15:1–15:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.FSCD.2024.15.

22 Max I. Kanovich, Stepan L. Kuznetsov, Vivek Nigam, and Andre Scedrov. Soft subexponentials and multiplexing. In Nicolas Peltier and Viorica Sofronie-Stokkermans, editors, *Automated Reasoning - 10th International Joint Conference, IJCAR 2020, Paris, France, July 1-4, 2020, Proceedings, Part I*, volume 12166 of *Lecture Notes in Computer Science*, pages 500–517. Springer, 2020. doi:10.1007/978-3-030-51074-9_29.

23 Shin-ya Katsumata. Parametric effect monads and semantics of effect systems. In Suresh Jagannathan and Peter Sewell, editors, *The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20-21, 2014*, pages 633–646. ACM, 2014. doi:10.1145/2535838.2535846.

24 Shin-ya Katsumata. A double category theoretic analysis of graded linear exponential comonads. In Christel Baier and Ugo Dal Lago, editors, *Foundations of Software Science and Computation Structures - 21st International Conference, FOSSACS 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings*, volume 10803 of *Lecture Notes in Computer Science*, pages 110–127. Springer, 2018. doi:10.1007/978-3-319-89366-2_6.

25 Joachim Lambek. The mathematics of sentence structure. *The American Mathematical Monthly*, 65(3):154–170, 1958.

26 Danielle Marshall and Dominic Orchard. Functional Ownership through Fractional Uniqueness. *Proc. ACM Program. Lang.*, 8(OOPSLA1):1040–1070, 2024. doi:10.1145/3649848.

27 Danielle Marshall, Michael Vollmer, and Dominic Orchard. Linearity and Uniqueness: An Entente Cordiale. In Ilya Sergey, editor, *Programming Languages and Systems - 31st European Symposium on Programming, ESOP 2022, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2022, Munich, Germany, April 2-7, 2022, Proceedings*, volume 13240 of *Lecture Notes in Computer Science*, pages 346–375. Springer, 2022. doi:10.1007/978-3-030-99336-8_13.

28 Conor McBride. I Got Plenty o' Nuttin'. In Sam Lindley, Conor McBride, Philip W. Trinder, and Donald Sannella, editors, *A List of Successes That Can Change the World - Essays Dedicated to Philip Wadler on the Occasion of His 60th Birthday*, volume 9600 of *Lecture Notes in Computer Science*, pages 207–233. Springer, 2016. doi:10.1007/978-3-319-30936-1_12.

29 Paul-André Melliès. Categorical semantics of linear logic. In Pierre-Louis Curien, Hugo Herbelin, Jean-Louis Krivine, and Paul-André Melliès, editors, *Interactive Models of Computation and Program Behaviour*. Panoramas et Synthèses 27, Société Mathématique de France, 2009.

30 Benjamin Moon, Harley Eades III, and Dominic Orchard. Graded modal dependent type theory. In *Programming Languages and Systems - 30th European Symposium on Programming, ESOP 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27 - April 1, 2021, Proceedings*, pages 462–490, 2021. doi:10.1007/978-3-030-72019-3_17.

31 Dominic Orchard, Vilem-Benjamin Liepelt, and Harley Eades III. Quantitative program reasoning with graded modal types. *Proc. ACM Program. Lang.*, 3(ICFP):110:1–110:30, 2019. doi:10.1145/3341714.

32 Dominic A. Orchard, Tomas Petricek, and Alan Mycroft. The semantic marriage of monads and effects. *CoRR*, abs/1401.5391, 2014. arXiv:1401.5391.

33 Tomas Petricek, Dominic A. Orchard, and Alan Mycroft. Coeffects: Unified static analysis of context-dependence. In *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part II*, pages 385–397, 2013. doi:10.1007/978-3-642-39212-2_35.

34 Tomas Petricek, Dominic A. Orchard, and Alan Mycroft. Coeffects: a calculus of context-dependent computation. In *Proceedings of the 19th ACM SIGPLAN international conference on Functional programming, Gothenburg, Sweden, September 1-3, 2014*, pages 123–135, 2014. doi:10.1145/2628136.2628160.

35 Frank Pfenning and Dennis Griffith. Polarized substructural session types. In Andrew M. Pitts, editor, *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*, volume 9034 of *Lecture Notes in Computer Science*, pages 3–22. Springer, 2015. doi:10.1007/978-3-662-46678-0_1.

36 Elaine Pimentel, Carlos Olarte, and Vivek Nigam. Process-as-formula interpretation: A substructural multimodal view (invited talk). In Naoki Kobayashi, editor, *6th International Conference on Formal Structures for Computation and Deduction, FSCD 2021, July 17-24, 2021, Buenos Aires, Argentina (Virtual Conference)*, volume 195 of *LIPIcs*, pages 3:1–3:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.FSCD.2021.3.

37 Klaas Pruiksma, William Chargin, Frank Pfenning, and Jason Reed. Adjoint logic. Unpublished Draft: http://www.cs.cmu.edu/~fp/papers/adjoint18b.pdf, 2018.

38 Klaas Pruiksma and Frank Pfenning. A message-passing interpretation of adjoint logic. *Journal of Logical and Algebraic Methods in Programming*, page 100637, 2020.

39 Greg Restall. *An introduction to substructural logics*. Routledge, 2002.

40 Victoria Vollmer, Danielle Marshall, Harley Eades III, and Dominic Orchard. A Mixed Linear and Graded Logic: Proofs, Terms, and Models. *CoRR*, abs/2401.17199, 2024. doi:10.48550/arXiv.2401.17199.

41 Jan von Plato. A proof of Gentzen's Hauptsatz without multicut. *Arch. Math. Log.*, 40(1):9–18, 2001. doi:10.1007/S001530050170.

42 Philip Wadler. Linear types can change the world! In Manfred Broy and Cliff B. Jones, editors, *Programming concepts and methods: Proceedings of the IFIP Working Group 2.2, 2.3 Working Conference on Programming Concepts and Methods, Sea of Galilee, Israel, 2-5 April, 1990*, page 561. North-Holland, 1990.

43 David Walker. Substructural type systems. *Advanced topics in types and programming languages*, pages 3–44, 2005.