Combining dependency, grades, and adjoint logic

Peter Hanukaev

Harley Eades III

phanukaev@augusta.edu

harley.eades@gmail.com

School of Computer and Cyber Sciences

Augusta, Georgia, USA

Abstract

We propose two new dependent type systems. The first, is a dependent graded/linear type system where a graded dependent type system is connected via modal operators to a linear type system in the style of Linear/Non-linear logic. We then generalize this system to support many graded systems connected by many modal operators through the introduction of modes from Adjoint Logic. Finally, we prove several meta-theoretic properties of these two systems including graded substitution.

CCS Concepts: • **Theory of computation** → **Linear logic**; **Type theory**; *Proof theory*; *Categorical semantics*.

Keywords: linear logic, graded types, adjoint logic, dependent types, semantics, resource tracking

1 Introduction

In this paper we consider the combination of dependent types, graded types, and adjoint logic. We propose two new dependent type systems. The first, is a graded version of Krishnaswami et. al's [11] Dependent LNL system, where the dependently typed side is replaced with a graded dependent type system called GRAD [4]. As is the case for LNL style systems there are two systems, a linear one and a non-linear one, connected by a set of modal operators for moving between the two. We then consider how to merge these two systems and support many such modal operators through the introduction of modes from Adjoint Logic [18].

Linear logic [6] and bunched implications [16] are two examples of substructural logics that have been instrumental in the study of type-based resource tracking. In the former, the structural rules for weakening and contraction have been restricted to only be usable when types are annotated with a modal operator called of-course and denoted by $!A$. This restriction enforces that variables can only be used exactly once unless their type is marked by the of-course modality. Thus, if we always annotate our types, then we are essentially programming in a standard functional programming language. Linear types can be given a resource-based semantics which allows the type system to enforce that certain resources must be used or a type error is produced. For example, we can use linear types for the types of file handles which if they are used after they are closed, then a type error is produced. Linear types are also the logical foundation of session type systems which can be used to reason about message passing systems [3].

Bunched implications is the logical foundation of separation logic [15, 16] which is used to reason about imperative programs. Linear logic can be seen a combination of linear types and non-linear types through the of-course modality. Now Bunched implications is a similar combination, but rather than through a modality, linear types and non-linear types coexist in the same system without any modality. Thus, one can program in either one without any annotations. Then both systems interact through the two notions of implication from the linear and non-linear sides.

Linear logic and Bunched implications can be seen as two ends of a spectrum based on how much we need to annotate types. One difficulty with Bunched implications is that it can be difficult to implement, and it is unclear how to generalize it with more than two systems. Benton's Linear/Non-linear logic (LNL) [2] is a compromise between being fully annotated and not needing annotations. LNL combines linear and non-linear types in such a way that one can program using both without the need to annotate non-linear types with the of-course modality, but when one wishes to mix the two fragments one can use two modal operators that allows transporting a non-linear type to the linear fragment, and a linear type to the non-linear fragment. If one composes these two modal operators, then one obtains the of-course modality. Furthermore, we know how to generalize LNL to support more than two systems.

Adjoint Logic [18] generalizes LNL by combining a family of logics with varying degrees of structural rules as opposed to just two fragments of non-linear (allowing weakening and contraction) and linear (no structural rules). This is accomplished by taking a linear base system where all types are annotated with a mode. Each mode is then assigned which structural rules will be allowed in the fragment the mode represents. Finally, one can transport types from one mode to another through modalities similar to the ones found in LNL. This generalization greatly increases the expressiveness of the logic. For example, LNL is easily an instance by taking two modes one allowing both weakening and contraction and one that does not. We could also add a third mode allowing only contraction resulting in combining non-linear logic, relevance logic, and linear logic.

Dependent types allow one to specify and prove properties of programs within the same language they are written [14]. Linear logic has the benefit of affording the ability to specify and prove properties of imperative programs. Krishnaswami et al. [11] show how to integrate dependent types with linear types by generalizing the non-linear fragment of LNL to a dependent type system LNL$_D$ where the modality from the non-linear fragment now transports a dependent type to the linear fragment. Then using this new mixture of dependent and linear types they show how to specify and prove properties of imperative programs in the style of Bunched implications. The modes found in adjoint logic have also been used to design dependent type system similar to LNL$_D$ with more than two fragments, but with an eye towards combining dependent types and a family of modal logics [8] rather than just controlling the existence of structural rules.

Graded types are a rather recent addition to linear types where types are annotated with a resource annotation describing how variables of those types can be used; essentially controlling their dataflow. The type system is parameterized by an ordered semiring whose elements are the grades. The grades on types offer more fine grained control over resource usage. For example, if the ordered semiring is taken to be the natural numbers, then the grade describes exactly the number of times the variable is allowed to be used. Furthermore, graded types also have been shown to be a means of combining linear types with dependent types [1, 4, 13, 17].

Contributions. We combine dependent types, graded types, and the modes of adjoint logic to define a new system capable of combining lots of substructural logics. First, we generalize LNL$_D$ into a new system called DMGL, and then we generalize DMGL by adding modes producing our final system called GLAD. All of our contributions are as follows:

- We replace the dependent type system of LNL$_D$ with GRAD a graded dependent type system. This system called DMGL gives more control over resource usage [4]

producing a graded dependent linear/non-linear system. Then we prove:
  - Substitution for the entire system ensuring that typed graded composition is preserved.
  - Context and type well formedness.
  - Graded contraction and weakening are admissible in the mixed linear/non-linear fragment.
  - Subject reduction for the entire system.
- The previous system has two explicit fragments, but with grading on one side. Now we generalize this system one step further by introducing the modes from adjoint logic. This system is parameterized by a family of modes and preordered semirings where each mode is paired with a potentially different preordered semiring. Then each type is annotated with both a mode and a grade (element of the preordered semiring). Then we prove:
  - Substitution for the entire system ensuring that typed graded composition is preserved.
  - Contraction is admissible in the mixed fragment.

2 A Dependent Mixed Graded and Linear Type System

In this section we present our first type system, which combines the graded dependently typed system GRAD with linear logic. We call this system DMGL (dependent mixed graded linear). As with previously proposed dependently typed graded systems [1, 4, 13], variables in DMGL are annotated by grades drawn from a semiring which captures a computational notion of resource usage.

Definition 2.1 (Grades). DMGL is parametrized by a preordered semiring $(R, 0, 1, +, \cdot, \leq)$, that is R is equipped with a preorder $\leq$ and a semiring structure $(R, 0, 1, +, \cdot)$ such that the operations $(+)$ and $(\cdot)$ are monotonic in both arguments. Elements of R are called *grades* and denoted r, p, q.

Example 2.2 (Variable Re-use). We take $R = \mathbb{N}$ the semiring of natural numbers, with the usual addition and multiplication. In the judgment

$$x :^2 \mathbf{Nat} \vdash \text{if Even}(x) \text{ then } x/2 \text{ else } 3x + 1 : \mathbf{Nat}$$

the annotation of $2 \in \mathbb{N}$, indicates that the variable x is used two times in the computation of the consequent term. This grading was originally introduced by Girard for Bounded Linear Logic [7] and used to characterize polynomial time computation, but has also been used, for example, for automated garbage collection [4]. The preorder we choose on $\mathbb{N}$ is also relevant: It will be used to control the discarding of resources. If we choose the ordinary preorder $\leq$ on $\mathbb{N}$, then we could replace the annotation 2 above by some other integer $k \geq 2$. In this case, the annotation k would mean that the variable x is used up to k times in the computation. On the other hand, choosing the preorder to be the trivial one

with $m \leq n \iff m = n$, would guarantee a usage of two times exactly.

Example 2.3 (Quantitative Semirings). Continuing from the previous example, call a semiring R *quantitative*[1] if it satisfies

  i) $0 \neq 1$
  ii) $r + p = 0 \implies r = p = 0$
  iii) $r \cdot p = 0 \implies r = 0 \vee p = 0$

Choudhury et al. proved of GRAD that a variable graded with 0 in such a semiring is guaranteed to be computationally irrelevant and since our system is based on GRAD, a similar result is expected to hold for DMGL. Therefore, such semirings allow the tracking of computationally relevant vs. irrelevant data. This is particularly relevant in dependently typed programs, as it allows a distinction between variables which are only used in type checking and those which are used in the execution of a program. Examples of such semirings are the natural numbers, and the following two which we elaborate upon in more detail.

The boolean semiring is $R = \{0, 1\}$ with $1 + 1 = 1$. This semiring's tracking of variable usage is coarse grained with 0 meaning computational irrelevance and and 1 representing some usage. We have not yet discussed whether $0 \leq 1$ should hold in this semiring. DMGL features a subusage rule which assert a variable graded with $r \in R$ may also be graded with q, so long as $r \leq q$. If we choose $0 \leq 1$ to be true, the subusage rule will allow us to discard variables graded 1. On the other hand if we choose $0 \leq 1$ to not be true, variables graded 1 are guaranteed to be computationally relevant.

The *none-one-tons* semiring is $R = \{0, 1, \omega\}$ in which we have $1 + 1 = 1 + \omega = \omega$. This semiring offers slightly more fine grained tracking, with 1 now representing linear use, and ω representing unrestricted use. We take $0 \leq \omega$ to allow the discarding of unrestricted variables, and $1 \leq \omega$ to allow promotion of linear variables to the unrestricted case. If we make 1 incomparable by $\leq$ with the other elements, we can guarantee that variables graded 1 are in fact used linearly.

For the remainder of this section, we fix a preordered semiring $(R, 0, 1, +, \cdot, \leq)$.

The syntax of terms and types in DMGL is given in Figure 1 and will be explained throughout the remainder of this section as it becomes relevant.

Definition 2.4. *Grade vectors* are finite lists of grades, and denoted by δ. They have the syntax

$$\delta := \emptyset \mid \delta, r$$

with $\emptyset$ denoting the empty grade vector. We write δ, δ' for the concatenation of grade vectors δ and δ' and use $\vec{0}$ to denote any grade vector consisting of only 0s. We extend the

operations + and $\leq$ to grade vectors of equal length pointwise and define scalar multiplication $r \cdot \delta$ in the obvious way.

$$\emptyset + \emptyset = \emptyset \qquad (\delta, r) + (\delta', r') = \delta + \delta', r + r'$$

$$\emptyset \leq \emptyset \iff \text{True} \qquad (\delta, r) \leq (\delta', r') \iff \delta \leq \delta' \wedge r \leq r'$$

$$r \cdot \emptyset = \emptyset \qquad r \cdot (\delta, q) = r \cdot \delta, r \cdot q$$

The basic structure of DMGL is similar to other mixed linear/non-linear type systems [2, 11, 19]. DMGL consist of two fragments: A purely graded fragment and a mixed fragment. Terms and types are divided into graded and linear as well. Typing judgments in the graded fragment may only have graded hypotheses, and produce *graded terms* which belong to *graded types*. In the mixed fragment, typing judgments produce *linear terms* belonging to *linear types*, but assumptions may consist of both linear and graded formulas. The graded fragment is the dependently typed system GRAD, a type system by Choudhury et al. [4]. That work also provides a more detailed discussion of the rules presented here. In the mixed fragment, linear types may dependent on graded variables but not on linear ones. Because of this, we treat both graded and linear types as graded terms, belonging to type universes Type and Linear respectively. Graded types are denoted X, Y, Z, W and linear types are denoted A, B, C. Their syntax is given as part of the complete syntax of DMGL in Figure 1.

Definition 2.5 (Contexts). Contexts are lists of typing assignments to variables. We use the letters x, y, z for variables. While we make no syntactic distinction between variables assigned to graded or linear types, we do distinguish between *graded* and *linear contexts*, denoted by Δ and Γ respectively and assigning variables to only graded or linear types respectively. The graded fragment is dependently typed, and in the mixed fragment we allow types in the linear context to depend on variables appearing in the graded context. Because of this, our system requires judgments asserting that contexts are well formed. These judgment forms are $\delta \odot \Delta \vdash_G \text{ctx}$ and $\delta \odot \Delta; \Gamma \vdash_M \text{ctx}$ respectively and their rules are given in Figure 2. Here, the formulas $x \notin \text{dom } \Delta$ and $x \notin \text{dom } \Gamma$ indicate that that the variable x is not bound in context Δ or Γ respectively.

The former of the above judgment forms means that Δ is well-formed context and also ensures that the attached grade vector has the same length as Δ. The latter ensures that all types appearing in linear context Γ are well formed over the graded context Δ.

Remark 2.6. Note that in the context extension rule for graded contexts, the grade vector δ is extended by an arbitrary grade r. Because of this it is actually provable that if $\delta \odot \Delta \vdash_G \text{ctx}$, and δ' is any grade vector with the same length as δ, then $\delta' \odot \Delta \vdash_G \text{ctx}$ and similarly for mixed contexts.

---

[1] Here, we follow terminology by Moon et al. [13]

$$X, Y, Z, W := \mathbf{J} \mid (x :^r X) \boxtimes Y \mid X \boxplus Y \mid (x :^r X) \to Y$$
$$\mid \mathcal{G} A \mid \mathsf{Type} \mid \mathsf{Linear}$$
$$t, s := x \mid \mathbf{j} \mid \mathsf{let}\ \mathbf{j} = t_1\ \mathsf{in}\ t_2 \mid (t_1, t_2)$$
$$\mid \mathsf{let}\ (x, y) = t_1\ \mathsf{in}\ t_2 \mid \mathsf{inl}\ t \mid \mathsf{inr}\ t$$
$$\mid \mathsf{case}_q\ t\ \mathsf{of}\ t_1; t_2 \mid \lambda x.t \mid t_1\ t_2 \mid \mathcal{G}\ l \mid X \mid A$$
$$l := x \mid \mathbf{i} \mid \mathsf{let}\ \mathbf{i} = l_1\ \mathsf{in}\ l_2 \mid \lambda x.l \mid l_1\ l_2 \mid (l_1, l_2)$$
$$\mid \mathsf{let}\ (x, y) = l_1\ \mathsf{in}\ l_2 \mid \mathcal{F}\ (t, l)$$
$$\mid \mathsf{let}\ \mathcal{F}\ (x, y) = l_1\ \mathsf{in}\ l_2 \mid \mathcal{G}^{-1}\ t$$
$$A, B, C := \mathbf{I} \mid A \multimap B \mid A \otimes B \mid \mathcal{F}(x :^r X).A$$
$$\Delta := \emptyset \mid \Delta, x : X$$
$$\Gamma := \emptyset \mid \Gamma, x : A$$

Figure 1. Syntax of DMGL

GRADEDCTX-EMPTY
$$\overline{\emptyset \odot \emptyset \vdash_G \mathsf{ctx}}$$

GRADEDCTX-EXTEND
$$\frac{x \notin \mathsf{dom}\ \Delta \quad \delta \odot \Delta \vdash_G \mathsf{ctx} \quad \delta_0 \odot \Delta \vdash_G X : \mathsf{Type}}{\delta, r \odot \Delta, x : X \vdash_G \mathsf{ctx}}$$

MIXEDCTX-EMPTY
$$\frac{\delta \odot \Delta \vdash_G \mathsf{ctx}}{\delta \odot \Delta; \emptyset \vdash_M \mathsf{ctx}}$$

MIXEDCTX-EXTEND
$$\frac{x \notin \mathsf{dom}\ \Delta \quad x \notin \mathsf{dom}\ \Gamma \quad \delta \odot \Delta; \Gamma \vdash_M \mathsf{ctx} \quad \delta_0 \odot \Delta \vdash_G A : \mathsf{Linear}}{\delta \odot \Delta; \Gamma, x : A \vdash_M \mathsf{ctx}}$$
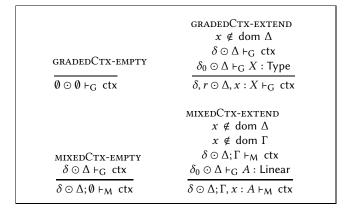
Figure 2. Rules for well formed contexts

Aside from the two type universes, our system only contains two basic types, namely the graded and linear unit types $\mathbf{J}$ and $\mathbf{I}$ respectively. To construct more complex graded types, our system includes coproduct types $X_1 \boxplus X_2$, dependent function types $(x :^r X) \to Y$ and a dependent pair type $(x :^r X) \boxtimes Y$. We explain the roles of the grade annotations in the latter two below. To form more complex linear types, we have the linear function type $A \multimap B$ and tensor product type $A \otimes B$ at our disposal. Lastly, we have the modal operators $\mathcal{F}$ and $\mathcal{G}$, which mediate between the two fragments, transforming linear types into graded ones and vice versa. We give the complete type formation rules in Figure 3.

We now explain the typing rules of our type system in more detail. The typing judgments for the graded and mixed fragment have the forms

$$\delta \odot \Delta \vdash_G t : X \quad \text{and} \quad \delta \odot \Delta; \Gamma \vdash_M l : A$$

respectively. The annotations G and M on the turnstiles indicate whether the judgment is in the graded or mixed fragment. The rules enforce that the length of δ and Δ are equal
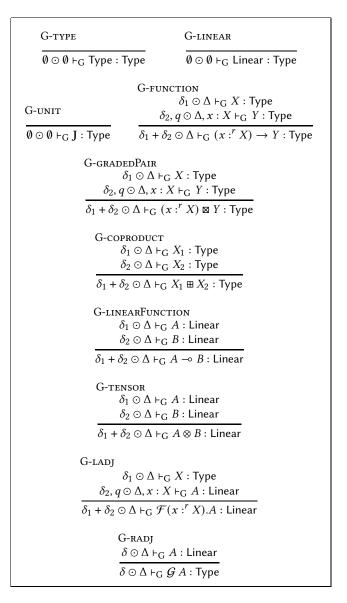
G-TYPE
$$\overline{\emptyset \odot \emptyset \vdash_G \mathsf{Type} : \mathsf{Type}}$$

G-LINEAR
$$\overline{\emptyset \odot \emptyset \vdash_G \mathsf{Linear} : \mathsf{Type}}$$

G-UNIT
$$\overline{\emptyset \odot \emptyset \vdash_G \mathbf{J} : \mathsf{Type}}$$

G-FUNCTION
$$\frac{\delta_1 \odot \Delta \vdash_G X : \mathsf{Type} \quad \delta_2, q \odot \Delta, x : X \vdash_G Y : \mathsf{Type}}{\delta_1 + \delta_2 \odot \Delta \vdash_G (x :^r X) \to Y : \mathsf{Type}}$$

G-GRADEDPAIR
$$\frac{\delta_1 \odot \Delta \vdash_G X : \mathsf{Type} \quad \delta_2, q \odot \Delta, x : X \vdash_G Y : \mathsf{Type}}{\delta_1 + \delta_2 \odot \Delta \vdash_G (x :^r X) \boxtimes Y : \mathsf{Type}}$$

G-COPRODUCT
$$\frac{\delta_1 \odot \Delta \vdash_G X_1 : \mathsf{Type} \quad \delta_2 \odot \Delta \vdash_G X_2 : \mathsf{Type}}{\delta_1 + \delta_2 \odot \Delta \vdash_G X_1 \boxplus X_2 : \mathsf{Type}}$$

G-LINEARFUNCTION
$$\frac{\delta_1 \odot \Delta \vdash_G A : \mathsf{Linear} \quad \delta_2 \odot \Delta \vdash_G B : \mathsf{Linear}}{\delta_1 + \delta_2 \odot \Delta \vdash_G A \multimap B : \mathsf{Linear}}$$

G-TENSOR
$$\frac{\delta_1 \odot \Delta \vdash_G A : \mathsf{Linear} \quad \delta_2 \odot \Delta \vdash_G B : \mathsf{Linear}}{\delta_1 + \delta_2 \odot \Delta \vdash_G A \otimes B : \mathsf{Linear}}$$

G-LADJ
$$\frac{\delta_1 \odot \Delta \vdash_G X : \mathsf{Type} \quad \delta_2, q \odot \Delta, x : X \vdash_G A : \mathsf{Linear}}{\delta_1 + \delta_2 \odot \Delta \vdash_G \mathcal{F}(x :^r X).A : \mathsf{Linear}}$$

G-RADJ
$$\frac{\delta \odot \Delta \vdash_G A : \mathsf{Linear}}{\delta \odot \Delta \vdash_G \mathcal{G} A : \mathsf{Type}}$$

Figure 3. Rules for type formation

in any provable judgment. If $\delta = r_1, \ldots, r_n$ and $\Delta = x_1 : X_1, \ldots, x_n : X_n$, the above judgment forms indicate that variable x_i is used with grade r_i in the construction of the term t (resp. l). Notice that both graded and linear types are themselves graded terms, but the above judgment forms contain no information about the grades used in the construction of the type X (resp. A).

The rule G-WEAK allows weakening, provided the newly added variable is used with grade 0. Similarly, for the variable rule G-VAR we require that the variable in the conclusion of the rule is used exactly with grade 1 and all other variables are used with grade 0. Finally, we include a subusage rule G-SUBUSAGE which asserts that we can make typing judgments with higher grades than necessary. The graded unit type $\mathbf{J}$ has one closed constructor $\mathbf{j}$ and a term of

unit type can be eliminated by pattern matching The graded dependent pair type $(x :^r X) \boxtimes Y$ comes with a grade annotation r. This annotation means that to eliminate a term of the form (t_1, t_2) of this type, the grade at which the first component t_1 of the pair is used must be r times the grade at which t_2 is used.

The coproduct type $X_1 \boxplus X_2$ has the expected left and right injections inl and inr as constructors and its elimination form $\mathsf{case}_q\ t$ of $s_1; s_2$ works by case distinction. The one caveat is that the functions s_1 and s_2, which describe the two cases, must use their input with the same grade q, which we include as annotation on the elimination form.

The dependent function type $(x :^r X) \to Y$ has a grade annotation which indicates at which grade the variable x of type X must be used: Introduction is done via lambda abstraction, with the constraint that the variable that we are abstracting over must be used at grade r. Similarly, if t is of type $(x :^r X) \to Y$ and t' of type X, then the grades used to construct t' are multiplied by r when constructing the application $t\ t'$.

Since term judgments contain no information about the grades used in the type, the grade annotations in the dependent function type $(x :^r X) \to Y$ and $(x :^r X) \boxtimes Y$ do not need to be the same as the grade with which $x : X$ is used in the construction of Y. A similar remark holds for the left adjoint $\mathcal{F}(x :^r X).A$ below.

The mixed fragment behaves like a linear logic, with an additional context of graded variables available. When linear contexts are concatenated, the grade vectors of the shared graded context are added. The mixed fragment features a linear unit type $\mathbf{I}$ with inhabitant $\mathbf{i}$, a linear function type $A \multimap B$ and a pair type $A \otimes B$. The rules the mixed fragment are specified in Figure 5 and Figure 6. Some rules in the mixed fragment feature concatenation of linear contexts. In those cases we assume that variables are renamed to avoid name clashes.

Finally, we discuss the modal operators $\mathcal{F}$ and $\mathcal{G}$. The operator $\mathcal{G}$ takes a linear type A and produces a graded type $\mathcal{G} A$. It's function is analogous to the operators of Benton [2] and Krishnaswami et al. [11]. It transforms linear terms l of type A with no free linear variables into graded terms of type $\mathcal{G} A$. Elimination of terms of type $\mathcal{G} A$ is handled by the operator $\mathcal{G}^{-1}\ t$ which produces terms of type A from terms of type $\mathcal{G} A$. Here, we can see that linearity and grading line up, if l is a linear term with free variable x of linear type A, there is a corresponding term $[\mathcal{G}^{-1}\ y/x]\, l$ with free variable y of type $\mathcal{G} A$ and variable y is graded 1, see Proposition 2.13 below. The operator $\mathcal{F}$ is again similar to that of Kirshnaswami [11]. The type $\mathcal{F}(x :^r X).A$ behaves like a dependent pair type where the first component belongs to graded type X and the second component belongs

G-SUBUSAGE
$$\frac{\delta \odot \Delta \vdash_{\mathsf{G}} t : X \qquad \delta \leq \delta'}{\delta' \odot \Delta \vdash_{\mathsf{G}} t : X}$$

G-WEAK
$$\frac{x \notin \mathrm{dom}\ \Delta \qquad \delta \odot \Delta \vdash_{\mathsf{G}} t : Y \qquad \delta_0 \odot \Delta \vdash_{\mathsf{G}} X : \mathsf{Type}}{\delta, 0 \odot \Delta, x : X \vdash_{\mathsf{G}} t : Y}$$

G-VAR
$$\frac{x \notin \mathrm{dom}\ \Delta \qquad \delta_0 \odot \Delta \vdash_{\mathsf{G}} X : \mathsf{Type}}{\vec{0}, 1 \odot \Delta, x : X \vdash_{\mathsf{G}} x : X}$$

G-CONVERT
$$\frac{\delta \odot \Delta \vdash_{\mathsf{G}} t : X \qquad \delta_0 \odot \Delta \vdash_{\mathsf{G}} Y : \mathsf{Type} \qquad X \equiv Y}{\delta \odot \Delta \vdash_{\mathsf{G}} t : Y}$$

G-UNITINTRO
$$\overline{\emptyset \odot \emptyset \vdash_{\mathsf{G}} \mathbf{j} : \mathbf{J}}$$

G-UNITELIM
$$\frac{\delta' \odot \Delta \vdash_{\mathsf{G}} t' : \mathbf{J} \qquad \delta_0, r_0 \odot \Delta, x : \mathbf{J} \vdash_{\mathsf{G}} X : \mathsf{Type} \qquad \delta \odot \Delta \vdash_{\mathsf{G}} t : [\mathbf{j}/x]X}{\delta + \delta' \odot \Delta \vdash_{\mathsf{G}} \mathsf{let}\ \mathbf{j} = t'\ \mathsf{in}\ t : [t'/x]X}$$

G-GRADEDPAIRINTRO
$$\frac{\delta_0, r_0 \odot \Delta, x : X \vdash_{\mathsf{G}} Y : \mathsf{Type} \qquad \delta_1 \odot \Delta \vdash_{\mathsf{G}} t_1 : X \qquad \delta_2 \odot \Delta \vdash_{\mathsf{G}} t_2 : [t_1/x]Y}{r \cdot \delta_1 + \delta_2 \odot \Delta \vdash_{\mathsf{G}} (t_1, t_2) : (x :^r X) \boxtimes Y}$$

G-GRADEDPAIRELIM
$$\frac{\delta_0, r_0 \odot \Delta, z : (x :^r X) \boxtimes Y \vdash_{\mathsf{G}} W : \mathsf{Type} \qquad \delta_1 \odot \Delta \vdash_{\mathsf{G}} t_1 : (x :^r X) \boxtimes Y \qquad \delta_2, r \cdot q, q \odot \Delta, x : X, y : Y \vdash_{\mathsf{G}} t_2 : [(x, y)/z]\, W}{q \cdot \delta_1 + \delta_2 \odot \Delta \vdash_{\mathsf{G}} \mathsf{let}\ (x, y) = t_1\ \mathsf{in}\ t_2 : [t_1/z]\, W}$$

G-COPRODUCTINL
$$\frac{\delta \odot \Delta \vdash_{\mathsf{G}} t : X_1 \qquad \delta_0 \odot \Delta \vdash_{\mathsf{G}} X_2 : \mathsf{Type}}{\delta \odot \Delta \vdash_{\mathsf{G}} \mathsf{inl}\ t : X_1 \boxplus X_2}$$

G-COPRODUCTINR
$$\frac{\delta_0 \odot \Delta \vdash_{\mathsf{G}} X_1 : \mathsf{Type} \qquad \delta \odot \Delta \vdash_{\mathsf{G}} t : X_2}{\delta \odot \Delta \vdash_{\mathsf{G}} \mathsf{inr}\ t : X_1 \boxplus X_2}$$

G-COPRODUCTELIM
$$\frac{\delta_0, r_0 \odot \Delta, y : X_1 \boxplus X_2 \vdash_{\mathsf{G}} Y : \mathsf{Type} \quad \delta_1 \odot \Delta \vdash_{\mathsf{G}} t : X_1 \boxplus X_2 \quad \delta_2 \odot \Delta \vdash_{\mathsf{G}} s_1 : (x :^q X_1) \to [\mathsf{inl}\ x/y]Y \quad \delta_2 \odot \Delta \vdash_{\mathsf{G}} s_2 : (x :^q X_2) \to [\mathsf{inr}\ x/y]Y}{q \cdot \delta_1 + \delta_2 \odot \Delta \vdash_{\mathsf{G}} \mathsf{case}_q\ t\ \mathsf{of}\ s_1; s_2 : [t/y]Y}$$

G-LAMBDA
$$\frac{\delta, r \odot \Delta, x : X \vdash_{\mathsf{G}} t : Y}{\delta \odot \Delta \vdash_{\mathsf{G}} \lambda x.t : (x :^r X) \to Y}$$

G-APP
$$\frac{\delta_1 \odot \Delta \vdash_{\mathsf{G}} t : (x :^r X) \to Y \qquad \delta_2 \odot \Delta \vdash_{\mathsf{G}} t' : X}{\delta_1 + r \cdot \delta_2 \odot \Delta \vdash_{\mathsf{G}} t\ t' : [t'/x]Y}$$

G-RADJINTRO
$$\frac{\delta \odot \Delta; \emptyset \vdash_{\mathsf{M}} l : A}{\delta \odot \Delta \vdash_{\mathsf{G}} \mathcal{G}\ l : \mathcal{G} A}$$

Figure 4. Graded system type assignment

$$\text{M-}\textsc{id}$$
$$x \notin \text{dom } \Delta$$
$$\delta \odot \Delta \vdash_G A : \text{Linear}$$
$$\overline{\vec{0} \odot \Delta; x : A \vdash_M x : A}$$

$$\text{M-}\textsc{subusage}$$
$$\delta_1 \leq \delta_2$$
$$\delta_1 \odot \Delta; \Gamma \vdash_M l : A$$
$$\overline{\delta_2 \odot \Delta; \Gamma \vdash_M l : A}$$

$$\text{M-}\textsc{weak}$$
$$x \notin \text{dom } \Delta$$
$$x \notin \text{dom } \Gamma$$
$$\delta \odot \Delta; \Gamma \vdash_M l : A$$
$$\delta_0 \odot \Delta \vdash_G X : \text{Type}$$
$$\overline{\delta, 0 \odot \Delta, x : X; \Gamma \vdash_M l : A}$$

$$\text{M-}\textsc{exchange}$$
$$\delta \odot \Delta; \Gamma_1, x : A, y : B, \Gamma_2 \vdash_M l : C$$
$$\overline{\delta \odot \Delta; \Gamma_1, y : B, x : A, \Gamma_2 \vdash_M l : C}$$

$$\text{M-}\textsc{convert}$$
$$\delta \odot \Delta; \Gamma \vdash_M l : A$$
$$\delta_0 \odot \Delta \vdash_G B : \text{Linear}$$
$$A \equiv B$$
$$\overline{\delta \odot \Delta; \Gamma \vdash_M l : B}$$

$$\text{M-}\textsc{unitIntro}$$
$$\overline{\emptyset \odot \emptyset; \emptyset \vdash_M \mathbf{i} : \mathbf{I}}$$

$$\text{M-}\textsc{unitElim}$$
$$\delta_1 \odot \Delta; \Gamma_1, x : \mathbf{I}, \Gamma_3 \vdash_M l_1 : B$$
$$\delta_2 \odot \Delta; \Gamma_2 \vdash_M l_2 : \mathbf{I}$$
$$\overline{\delta_1 + \delta_2 \odot \Delta; \Gamma_1, \Gamma_2, \Gamma_3 \vdash_M \text{let } \mathbf{i} = l_2 \text{ in } l_1 : B}$$

$$\text{M-}\textsc{lambda}$$
$$\delta \odot \Delta; \Gamma, x : A \vdash_M l : B$$
$$\overline{\delta \odot \Delta; \Gamma \vdash_M \lambda x.l : A \multimap B}$$

$$\text{M-}\textsc{app}$$
$$\delta_1 \odot \Delta; \Gamma_1 \vdash_M l_1 : A$$
$$\delta_2 \odot \Delta; \Gamma_2 \vdash_M l_2 : A \multimap B$$
$$\overline{\delta_1 + \delta_2 \odot \Delta; \Gamma_1, \Gamma_2 \vdash_M l_2 \, l_1 : B}$$

Figure 5. Mixed system type assignment

$$\text{M-}\textsc{tensorIntro}$$
$$\delta_1 \odot \Delta; \Gamma_1 \vdash_M l_1 : A$$
$$\delta_2 \odot \Delta; \Gamma_2 \vdash_M l_2 : B$$
$$\overline{\delta_1 + \delta_2 \odot \Delta; \Gamma_1, \Gamma_2 \vdash_M (l_1, l_2) : A \otimes B}$$

$$\text{M-}\textsc{tensorElim}$$
$$\delta_2 \odot \Delta; \Gamma_2 \vdash_M l_1 : A \otimes B$$
$$\delta_1 \odot \Delta; \Gamma_1, x : A, y : B, \Gamma_3 \vdash_M l_2 : C$$
$$\overline{\delta_1 + \delta_2 \odot \Delta; \Gamma_1, \Gamma_2, \Gamma_3 \vdash_M \text{let } (x, y) = l_1 \text{ in } l_2 : C}$$

$$\text{M-}\textsc{ladjIntro}$$
$$\delta_0, r_0 \odot \Delta, x : X \vdash_G A : \text{Linear}$$
$$\delta_1 \odot \Delta \vdash_G t : X$$
$$\delta_2 \odot \Delta; \Gamma \vdash_M l : [t/x]A$$
$$\overline{r \cdot \delta_1 + \delta_2 \odot \Delta; \Gamma \vdash_M \mathcal{F}(t, l) : \mathcal{F}(x :^r X).A}$$

$$\text{M-}\textsc{ladjElim}$$
$$\delta_0 \odot \Delta \vdash_G C : \text{Linear}$$
$$\delta \odot \Delta; \Gamma_2 \vdash_M \text{ctx}$$
$$\delta_1 \odot \Delta; \Gamma_1 \vdash_M l_1 : \mathcal{F}(x :^r X).A$$
$$\delta_2, r \odot \Delta, x : X; \Gamma_2, y : A \vdash_M l_2 : C$$
$$\overline{\delta_1 + \delta_2 \odot \Delta; \Gamma_1, \Gamma_2 \vdash_M \text{let } \mathcal{F}(x, y) = l_1 \text{ in } l_2 : C}$$

$$\text{M-}\textsc{radjElim}$$
$$\delta \odot \Delta \vdash_G t : \mathcal{G} A$$
$$\overline{\delta \odot \Delta; \emptyset \vdash_M \mathcal{G}^{-1} t : A}$$

Figure 6. Mixed system type assignment continued

on graded terms that contains $\rightsquigarrow$ and such that $t_1 \equiv t_2$ implies $[t_1/x]t \equiv [t_2/x]t$ for all terms t, t_1, t_2 and likewise with linear terms. Similarly, we write $\equiv$ for the smallest equivalence relation on linear terms which contains $\rightsquigarrow$ and satisfies the implications $t_1 \equiv t_2 \implies [t_1/x]l \equiv [t_2/x]l$ and $l_1 \equiv l_2 \implies [l_1/x]l \equiv [l_2/x]l$ for all t_1, t_2, l_1, l_2 and l. We will only use the relation $\equiv$ in the type conversion rules, which are standard.

Metatheory

We now turn our attention to the metatheory of DMGL. First we state some well-formedness conditions. Then we show how grading interacts with substitution and linearity. Finally, we show that the full β-reduction rules preserve typing and grading. Proofs of theorems stated in this section can be found in Appendix A.

In provable typing judgments, the context in which the typing occurs is well-formed. Furthermore, terms always have well-formed types.
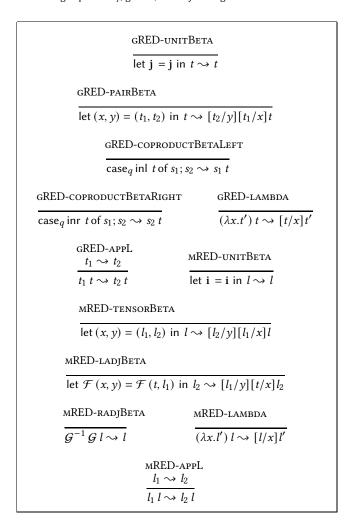
Proposition 2.7. *The following hold by mutual induction:*
  i) If $\delta \odot \Delta \vdash_G t : X$, then $\delta \odot \Delta \vdash_G$ ctx.
  ii) If $\delta \odot \Delta; \Gamma \vdash_M l : A$, then $\delta \odot \Delta \vdash_G$ ctx and $\delta \odot \Delta; \Gamma \vdash_M$ ctx.

to linear type A. Elimination from this type is done by pattern matching let $\mathcal{F}(x, y) = l_1$ in l_2 and the grade annotation r in the type forces the first component to be used with grade r for the eliminator to be invoked.

Figure 7 contains the reduction rules (full β-reduction) for DMGL. Since reduction rules only see the syntactic form of terms, the grades are not involved at all, and hence the reduction rules are as one would expect. The application of a lambda abstraction to some term reduces by substitution, and pattern matching expressions reduce when the term being matched on is exactly of the form of the pattern, in which case the reduction is also by substitution. Finally, the case expression for the coproduct eliminator reduces if the scrutinee was constructed by one of the injections, in which case the reduction works by applying the respective function. We write $\equiv$ for the congruence closure of $\rightsquigarrow$, on graded terms. In other words, $\equiv$ is the smallest equivalence relation

GRED-UNITBETA

$$\frac{}{\text{let } \mathbf{j} = \mathbf{j} \text{ in } t \rightsquigarrow t}$$

GRED-PAIRBETA

$$\frac{}{\text{let } (x, y) = (t_1, t_2) \text{ in } t \rightsquigarrow [t_2/y][t_1/x]t}$$

GRED-COPRODUCTBETALEFT

$$\frac{}{\text{case}_q \text{ inl } t \text{ of } s_1; s_2 \rightsquigarrow s_1 \, t}$$

GRED-COPRODUCTBETARIGHT

$$\frac{}{\text{case}_q \text{ inr } t \text{ of } s_1; s_2 \rightsquigarrow s_2 \, t}$$

GRED-LAMBDA

$$\frac{}{(\lambda x.t') \, t \rightsquigarrow [t/x]t'}$$

GRED-APPL

$$\frac{t_1 \rightsquigarrow t_2}{t_1 \, t \rightsquigarrow t_2 \, t}$$

MRED-UNITBETA

$$\frac{}{\text{let } \mathbf{i} = \mathbf{i} \text{ in } l \rightsquigarrow l}$$

MRED-TENSORBETA

$$\frac{}{\text{let } (x, y) = (l_1, l_2) \text{ in } l \rightsquigarrow [l_2/y][l_1/x]l}$$

MRED-LADJBETA

$$\frac{}{\text{let } \mathcal{F}(x, y) = \mathcal{F}(t, l_1) \text{ in } l_2 \rightsquigarrow [l_1/y][t/x]l_2}$$

MRED-RADJBETA

$$\frac{}{\mathcal{G}^{-1} \, \mathcal{G} \, l \rightsquigarrow l}$$

MRED-LAMBDA

$$\frac{}{(\lambda x.l') \, l \rightsquigarrow [l/x]l'}$$

MRED-APPL

$$\frac{l_1 \rightsquigarrow l_2}{l_1 \, l \rightsquigarrow l_2 \, l}$$

Figure 7. Reduction Rules

Proposition 2.8. *The following hold by mutual induction:*
- *i) If $\delta \odot \Delta \vdash_G t : X$, then $\delta' \odot \Delta \vdash_G X : $ Type for some grade vector δ'.*
- *ii) If $\delta \odot \Delta; \Gamma \vdash_M l : A$, then $\delta' \odot \Delta \vdash_G A : $ Linear for some grade vector δ'.*

Next, we consider substitution. Since we have a graded and a linear fragment we need to state substitution for both fragments, additionally, in the linear fragment substitution is split further into cases where a variable is replaced by a graded or a linear term. Since our system has dependent typing, when a graded term is substituted for a variable we also need to substitute it in part of the context. We therefore make the following definition:

Definition 2.9. Let Δ be a graded context, Γ a linear context, x a term variable and suppose $x \notin \text{dom } \Delta$ and $x \notin \text{dom } \Gamma$. We define $[t/x]\Delta$ and $[t/x]\Gamma$ as follows:

$$[t/x]\emptyset = \emptyset \qquad [t/x](\Delta, y : Y) = [t/x]\Delta, y : [t/x]Y$$

$$[t/x](\Gamma, y : A) = [t/x]\Gamma, y : [t/x]A$$

We use an additional notational convention:

Convention 2.10. We assume that lengths of grade vectors and corresponding contexts match in judgments. For example, when we write $\delta, r, \delta' \odot \Delta, x : X, \Delta' \vdash_G t : Y$, we assume that δ has the same length as Δ and similarly for δ' and Δ'.

We can now state the substitution theorem. Parallel composition is modeled by addition in the semiring, while sequential composition is modeled by multiplication.

Theorem 2.11 (Substitution). *The following hold by mutual induction:*

- *i) (Graded Contexts) If $\delta_0 \odot \Delta \vdash_G t_0 : X$, and $\delta, r, \delta' \odot \Delta, x : X, \Delta' \vdash_G $ ctx then*

$$\delta + r \cdot \delta_0, \delta' \odot \Delta, [t_0/x]\Delta' \vdash_G \text{ ctx}$$

- *ii) (Mixed Contexts) If $\delta_0 \odot \Delta \vdash_G t_0 : X$, and $\delta, r, \delta' \odot \Delta, x : X, \Delta'; \Gamma \vdash_M $ ctx then*

$$\delta + r \cdot \delta_0, \delta' \odot \Delta, [t_0/x]\Delta'; [t_0/x]\Gamma \vdash_M \text{ ctx}$$

- *iii) (Graded) If $\delta_0 \odot \Delta \vdash_G t_0 : X$ and $\delta, r, \delta' \odot \Delta, x : X, \Delta' \vdash_G t : Y$, then*

$$\delta + r \cdot \delta_0, \delta' \odot \Delta, [t_0/x]\Delta' \vdash_G [t_0/x]t : [t_0/x]Y$$

- *iv) (Mixed Graded) If $\delta_0 \odot \Delta \vdash_G t_0 : X$ and $\delta, r, \delta' \odot \Delta, x : X, \Delta'; \Gamma \vdash_M l : A$, then*

$$\delta + r \cdot \delta_0 \odot \Delta, [t_0/x]\Delta'; [t_0/x]\Gamma \vdash_M [t_0/x]l : [t_0/x]A$$

- *v) (Mixed Linear) If $\delta_0 \odot \Delta; \Gamma_0 \vdash_M l_0 : A$ and $\delta \odot \Delta; \Gamma, x : A, \Gamma' \vdash_M l : B$, then*

$$\delta + \delta_0 \odot \Delta; \Gamma, \Gamma_0, \Gamma' \vdash_M [l_0/x]l : B$$

Corollary 2.12 (Contraction). DMGL *admits graded contraction rules:*

- *i) If $\delta, p, q, \delta' \odot \Delta, x : X, y : X, \Delta' \vdash_G t : Y$, then $\delta, p + q, \delta' \odot \Delta, x : X, [x/y]\Delta' \vdash_G [x/y]t : [x/y]Y$.*
- *ii) If $\delta, p, q, \delta' \odot \Delta, x : X, y : X, \Delta'; \Gamma \vdash_M l : A$, then $\delta, p + q, \delta' \odot \Delta, x : X, [x/y]\Delta'; \Gamma \vdash_M [x/y]l : [x/y]A$.*

Proof. Apply Theorem 2.11 to the judgment $\vec{0}, 1 \odot \Delta, x : X \vdash_G x : X$ obtained from the variable rule. □

We now show how the modal operator $\mathcal{G}$ interacts with linearity and grading. The following proposition essentially says that we can embed the linear fragment into the graded one, by making the grades on all variables from the linear fragment equal to 1.

Proposition 2.13. *If $\delta \odot \Delta; \Gamma, x : A \vdash_M l : B$, then $\delta, 1 \odot \Delta, y : \mathcal{G} A; \Gamma \vdash_M [\mathcal{G}^{-1} y/x]l : B$.*

Finally, we state the subject reduction theorem. Reduction does not only preserve typing, it also preserves the grades used in typing judgments. Since most reductions are by substitution, the bulk of the work has already been done in Theorem 2.11.

Theorem 2.14 (Subject Reduction).   *i)* If $\delta \odot \Delta \vdash_G t : X$ and $t \rightsquigarrow t'$, then $\delta \odot \Delta \vdash_G t' : X$.
    ii) If $\delta \odot \Delta; \Gamma \vdash_M l : A$ and $l \rightsquigarrow l'$, then $\delta \odot \Delta; \Gamma \vdash_M l' : A$.

3   A graded type system in the style of adjoint logic

We now present a further generalization of the previous type system, which is inspired by Adjoint Logic [18], and which we cal GLAD. Adjoint Logic offers a smooth way of combining an arbitrary number of substructural logics which are identified by *modes*. Each mode $\mathfrak{m}$ is assigned a set $\sigma(\mathfrak{m}) \subseteq \{W, C\}$ of the structural rules weakening (W) and contraction (C) satisfied by its logic, and the modes are arranged in a preorder such that the map σ is monotone, i.e. if $\mathfrak{m}_1 \leq \mathfrak{m}_2$, then $\sigma(\mathfrak{m}_1) \subseteq \sigma(\mathfrak{m}_2)$. One of the key insights of Adjoint Logic is that judgments

$$A^1_{\mathfrak{m}_1}, \ldots, A^n_{\mathfrak{m}_n} \vdash B_\mathfrak{m}$$

must satisfy $\mathfrak{m}_i \geq \mathfrak{m}$ for each i, where the subscripts indicate the mode a proposition belongs to.

In the system of this section, modes come equipped with a preordered semiring controlling the resource structure. For any one of these semirings, this system is a dependently typed system with the same rules as DMGL. We can control weakening in each of these fragments, but in the presence of dependent types, controlling for contraction becomes difficult. We will discuss the issues of this later.

Definition 3.1. Let R, S be preordered semirings. A morphism of preordered semirings $R \to S$ is a map $f : R \to S$ such that $f(0) = 0$, $f(1) = 1$ and for all $a, b \in R$, we have $f(a + b) = f(a) + f(b)$ and $f(a \cdot b) = f(a) \cdot f(b)$ and $a \leq b \implies f(a) \leq f(b)$. If $f : R \to S$ is a morphism of preordered semirings and $r \in R$ and $s \in S$, then we write $r \cdot s := f(r) \cdot s \in S$.

Definition 3.2. A *mode* $\mathfrak{m}$ is a pair $(R_\mathfrak{m}, \text{Weak}(\mathfrak{m}))$, where $R_\mathfrak{m}$ is a preordered semiring and $\text{Weak}(\mathfrak{m})$ is either true or false. We will write $r : \mathfrak{m}$ to mean $r \in R_\mathfrak{m}$. Modes are denoted by the lowercase fraktur letters $\mathfrak{m}, \mathfrak{n}, \mathfrak{l}$.

Let $\mathfrak{m}$ and $\mathfrak{n}$ be modes and assume that the proposition $\text{Weak}(\mathfrak{m}) \implies \text{Weak}(\mathfrak{n})$ is true. A morphism of modes $\mathfrak{m} \to \mathfrak{n}$ is a morphism $R_\mathfrak{m} \to R_\mathfrak{n}$ of the underlying preordered semirings. There is a category of modes, denoted by Modes.

For the rest of this section, fix a preordered set I and a functor $I \to \text{Modes}$. That is, fix: For each $i \in I$, a mode $\mathfrak{m}_i$, and for each $i \leq j$ in I, a morphism of modes $f_{ij} : \mathfrak{m}_i \to \mathfrak{m}_j$ such that for $i \leq j \leq k$, $f_{jk} \circ f_{ij} = f_{ik}$ holds. We will write $\mathfrak{m}_i \leq \mathfrak{m}_j$ when $i \leq j$. GLAD is parametrized by this data, and we give examples below.

The syntax of types and terms in GLAD is given in Figure 8. It strongly resembles that of DMGL, with a unit type $\mathbf{I}_m$, dependent function and pair types $(x :^{r:\mathfrak{m}} A) \multimap B$ and

---

Types: $A, B, C ::= \text{Type} \mid \mathbf{I}_\mathfrak{m} \mid (x :^{r:\mathfrak{m}} A) \multimap B$
$\mid (x :^{r:\mathfrak{m}} A) \otimes B \mid A \oplus B \mid \uparrow^{\mathfrak{m}_2}_{\mathfrak{m}_1} A$

Terms: $a, b, c ::= x \mid \star_\mathfrak{m} \mid \text{let } \star_\mathfrak{m} = a \text{ in } b \mid \lambda x.a \mid a \, b$
$\mid \text{let } (x, y) = a \text{ in } b \mid \text{inl } a \mid \text{inr } a$
$\mid \text{case}_q \, a \text{ of } b_1; b_2 \mid \uparrow^{\mathfrak{m}_2}_{\mathfrak{m}_1} a \mid \downarrow^{\mathfrak{m}_2}_{\mathfrak{m}_1} a$

Figure 8. Syntax of types and terms in GLAD

$$\text{CTX-EMPTY} \qquad \frac{\begin{array}{c} \text{CTX-EXTEND} \\ r : \mathfrak{m} \\ x \notin \text{dom } \Gamma \\ \delta \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{m} A : \text{Type} \end{array}}{\delta, r \mid \mathcal{M}, \mathfrak{m} \odot \Gamma, x : A \vdash \text{ctx}}$$

$$\frac{}{\emptyset \mid \emptyset \odot \emptyset \vdash \text{ctx}}$$

Figure 9. GLAD well-formed context judgment

$(x :^{r:\mathfrak{m}} A) \otimes B$ and a coproduct type $A \oplus B$. The modal operators $\uparrow^{\mathfrak{m}_2}_{\mathfrak{m}_1} a$ and $\downarrow^{\mathfrak{m}_2}_{\mathfrak{m}_1} a$ take the role of $\mathcal{G}$ and $\mathcal{G}^{-1}$ from before while the role of the modal operator $\mathcal{F}$ is now subsumed by the dependent pair type $(x :^{r:\mathfrak{m}} A) \otimes B$, which allows the first and second component of the pair to belong to different modes. The mode of any type appearing in a valid judgment will always be determined uniquely, and therefore annotations on types to specify their mode are not necessary. We choose to annotate the unit type with its mode, as this will make things easier in the future.

The judgment for well-formed contexts in GLAD has the form

$$\delta \mid \mathcal{M} \odot \Gamma \vdash \text{ctx}.$$

The rules for this judgment are given in Figure 9. In this judgment form, δ is a list of grades, $\mathcal{M}$ is a list of modes, and Γ is a context. If $\delta = (r_1, \ldots, r_n)$, $\mathcal{M} = (\mathfrak{m}_1, \ldots, \mathfrak{m}_n)$ and $\Gamma = x_1 : A_1, \ldots, x_n : A_n$, then the judgment above indicates that $r_i : \mathfrak{m}_i$, that the variable x_i is graded with grade r_i and that the type A_i belongs to mode $\mathfrak{m}_i$. The same is true for typing judgments:

$$\delta \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{m} a : A.$$

The annotation $\mathfrak{m}$ on the turnstile indicates that the judgment is made in mode $\mathfrak{m}$, and that a and A belong to mode $\mathfrak{m}$, therefore supporting the structural rules allowed by $\mathfrak{m}$. As in adjoint logic, we demand that in such a judgment, we have $\mathfrak{m} \leq \mathcal{M}$, that is $\mathfrak{m} \leq \mathfrak{n}$ for each mode $\mathfrak{n}$ appearing in $\mathcal{M}$. This property is enforced by the typing rules: If we assume that all premise judgments of a rule satisfy this property, then the conclusion necessarily does, too.

We give the full rules for type formation in Figure 10 and those for typing in Figure 11 and Figure 12. The rules strongly resemble the ones of DMGL, so we will omit most explanations, focusing primarily on the differences. GLAD has more

GLAD-TYPE

$$\overline{\emptyset \mid \emptyset \odot \emptyset \vdash_\mathfrak{m} \text{Type} : \text{Type}}$$

GLAD-UNIT

$$\overline{\emptyset \mid \emptyset \odot \emptyset \vdash_\mathfrak{m} \mathbf{I}_\mathfrak{m} : \text{Type}}$$

GLAD-FUNCTION

$$\frac{\delta, q_0 \mid \mathcal{M}, \mathfrak{m} \odot \Gamma, x : A \vdash_\mathfrak{n} B : \text{Type}}{\delta \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{n} (x :^{r:\mathfrak{m}} A) \multimap B : \text{Type}}$$

GLAD-TENSOR

$$\frac{\delta, q_0 \mid \mathcal{M}, \mathfrak{m} \odot \Gamma, x : A \vdash_\mathfrak{n} B : \text{Type}}{\delta \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{n} (x :^{r:\mathfrak{m}} A) \otimes B : \text{Type}}$$

GLAD-COPRODUCT

$$\frac{\delta_1 \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{m} A : \text{Type} \quad \delta_2 \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{m} B : \text{Type}}{\delta_1 + \delta_2 \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{m} A \oplus B : \text{Type}}$$

GLAD-SHIFTUP

$$\frac{\mathfrak{m}_1 \le \mathfrak{m}_2 \quad \mathfrak{m}_2 \le \mathcal{M} \quad \delta \mid \mathcal{M} \odot \Gamma \vdash_{\mathfrak{m}_1} A : \text{Type}}{\delta \mid \mathcal{M} \odot \Gamma \vdash_{\mathfrak{m}_2} \uparrow_{\mathfrak{m}_1}^{\mathfrak{m}_2} A : \text{Type}}$$

Figure 10. GLAD type formation rules

general control over weakening: We may add unused variables of mode $\mathfrak{m}$ to the context, so long as they are graded with the grade 0 and the mode $\mathfrak{m}$ allows weakening. If $\mathfrak{m}$ is a mode for which Weak($\mathfrak{m}$) is true, the functionality of the preorder on $R_\mathfrak{m}$ is overloaded in the following way: It captures both the subusaging relation and gives control over grades which may be computationally irrelevant. For nonzero elements of $R_\mathfrak{m}$, the preorder captures the subusaging behavior, allowing to use a higher grade of resources than necessary to construct a term. However, for elements q with $0 \le q$, the preorder captures that variables graded with q may be discarded.

Some of the rules feature a scalar multiplication $r \cdot \delta$. There are two things to note here: First, the vector δ is a list grades coming from potentially different semirings, but this doesn't affect the definition of scalar multiplication as entrywise multiplication with r. Second, let $\mathfrak{m}_i$ be the mode of the i-th entry of δ and $\mathfrak{m}$ the mode of r. Observe that the rules where scalar multiplication occurs guarantee that $\mathfrak{m} \le \mathfrak{m}_i$ for each i and therefore the scalar multiplication is indeed well-defined according to Definition 3.1. Similarly, some rules feature addition of grade vectors $\delta + \delta'$. Notice that whenever this is the case, the annotation by mode vectors $\mathcal{M}$ forces the i-th entries of δ and δ' to belong to the same mode, ensuring that the sums of grade vectors are well-defined. Finally, the subusage rule features the preorder relation $\delta_1 \le \delta_2$ on grade vectors. This is defined in the natural way as componentwise $\le$, with the additional condition that for each i, the i-th component of δ_1 and δ_2 must be from the same mode.

GLAD-VAR

$$\frac{x \notin \text{dom}\,\Gamma \quad \text{Weak}(\mathcal{M}) \quad \delta \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{m} A : \text{Type}}{\vec{0}, 1 \mid \mathcal{M}, \mathfrak{m} \odot \Gamma, x : A \vdash_\mathfrak{m} x : A}$$

GLAD-WEAK

$$\frac{\text{Weak}(\mathfrak{l}) \quad \mathfrak{m} \le \mathfrak{l} \quad x \notin \text{dom}\,\Gamma \quad \delta \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{m} a : A \quad \delta_0 \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{l} B : \text{Type}}{\delta, 0 \mid \mathcal{M}, \mathfrak{l} \odot \Gamma, x : B \vdash_\mathfrak{m} a : A}$$

GLAD-SUBUSAGE

$$\frac{\delta_1 \le \delta_2 \quad \delta_1 \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{m} a : A}{\delta_2 \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{m} a : A}$$

GLAD-UNITINTRO

$$\overline{\emptyset \mid \emptyset \odot \emptyset \vdash_\mathfrak{m} \star_\mathfrak{m} : \mathbf{I}_\mathfrak{m}}$$

GLAD-UNITELIM

$$\frac{\delta' \mid \mathcal{M} \odot \Gamma \vdash_{\mathfrak{m}_1} b : \mathbf{I}_{\mathfrak{m}_1} \quad \delta_0, r_0 \mid \mathcal{M}, \mathfrak{m}_1 \odot \Gamma, x : \mathbf{I}_{\mathfrak{m}_1} \vdash_{\mathfrak{m}_2} A : \text{Type} \quad \delta \mid \mathcal{M} \odot \Gamma \vdash_{\mathfrak{m}_2} a : [\star_{\mathfrak{m}_1}/x]A}{\delta + \delta' \mid \mathcal{M} \odot \Gamma \vdash_{\mathfrak{m}_2} \text{let } \star_{\mathfrak{m}_1} = b \text{ in } a : [b/x]A}$$

GLAD-LAMBDA

$$\frac{\delta, r \mid \mathcal{M}, \mathfrak{m} \odot \Gamma, x : A \vdash_\mathfrak{n} b : B}{\delta \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{n} \lambda x.b : (x :^{r:\mathfrak{m}} A) \multimap B}$$

GLAD-APP

$$\frac{\delta_1 \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{m} a : A \quad \delta_2 \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{n} c : (x :^{r:\mathfrak{m}} A) \multimap B}{\delta_2 + r \cdot \delta_1 \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{n} c\, a : [a/x]B}$$

Figure 11. GLAD typing rules

The dependent pair type $(x :^{r:\mathfrak{m}} A) \otimes B$ now carries an additional mode annotation, indicating that type A belongs to mode $\mathfrak{m}$ and that $r : \mathfrak{m}$. Due to the construction of the dependent pair type, if $\delta \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{n} (x :^{r:\mathfrak{m}} A) \otimes B : \text{Type}$, then B and $(x :^{r:\mathfrak{m}} A) \otimes B$ belong to mode $\mathfrak{m}$. If $\mathfrak{n}$ and $\mathfrak{m}$ are the same mode, then we recover ordinary versions of the dependent pair type belonging to mode $\mathfrak{m}$. On the other hand, if the modes are distinct, the first component of the pair is "moved" from the higher mode $\mathfrak{m}$ to the lower mode $\mathfrak{n}$. This is the way the left adjoint $\mathcal{F}(x :^r X).A$ functions in DMGL. Since the dependent pair and left adjoint $\mathcal{F}$ have the same introduction and elimination rules (mutandis mutatis), we treat them as special instances of the same type in GLAD, subsuming both functionalities.

The dependent function type $(x :^{r:\mathfrak{m}} A) \multimap B$ also carries a mode annotation now. This is because we allow the modes of A and B to be distinct, and the annotation indicates the mode of A. Like with the dependent pair type, if B has mode $\mathfrak{n}$, then so does $(x :^{r:\mathfrak{m}} A) \multimap B$ and we necessarily have $\mathfrak{m} \ge \mathfrak{n}$. A similar construction exists in LNL$_D$ [11], where

GLAD-TENSORINTRO
$$\frac{\delta, q_0 \mid \mathcal{M}, \mathfrak{m} \odot \Gamma, x \colon A \vdash_\mathfrak{n} B \colon \text{Type} \qquad \delta_1 \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{m} a \colon A \qquad \delta_2 \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{n} b \colon [a/x]B}{r \cdot \delta_1 + \delta_2 \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{n} (a, b) \colon (x \colon^{r \colon \mathfrak{m}} A) \otimes B}$$

GLAD-TENSORELIM
$$\frac{\delta_0, r_0 \mid \mathcal{M}, \mathfrak{n} \odot \Gamma, z \colon (x \colon^{r \colon \mathfrak{m}} A) \otimes B \vdash_I C \colon \text{Type} \qquad \delta_1 \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{n} a \colon (x \colon^{r \colon \mathfrak{m}} A) \otimes B \qquad \delta_2, q \cdot r, q \mid \mathcal{M}, \mathfrak{m}, \mathfrak{n} \odot \Gamma, x \colon A, y \colon B \vdash_I c \colon [(x, y)/z]C}{q \cdot \delta_1 + \delta_2 \mid \mathcal{M} \odot \Gamma \vdash_I \text{let } (x, y) = a \text{ in } c \colon [a/z]C}$$

GLAD-INL
$$\frac{\delta_0 \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{m} A \oplus B \colon \text{Type} \qquad \delta \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{m} a \colon A}{\delta \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{m} \text{inl } a \colon A \oplus B}$$

GLAD-INR
$$\frac{\delta_0 \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{m} A \oplus B \colon \text{Type} \qquad \delta \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{m} b \colon B}{\delta \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{m} \text{inr } b \colon A \oplus B}$$

GLAD-COPRODUCTELIM
$$\frac{\delta_0, r_0 \mid \mathcal{M}, \mathfrak{m} \odot \Gamma, z \colon A \oplus B \vdash_\mathfrak{n} C \colon \text{Type} \qquad \delta_1 \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{m} a \colon A \oplus B \qquad \delta_2 \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{n} c_1 \colon (x \colon^{q \colon \mathfrak{m}} A) \multimap [\text{inl } x/z]C \qquad \delta_2 \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{n} c_2 \colon (x \colon^{q \colon \mathfrak{m}} B) \multimap [\text{inl } x/z]C}{q \cdot \delta_1 + \delta_2 \mid \mathcal{M} \odot \Gamma \vdash_\mathfrak{n} \text{case}_q \, a \text{ of } c_1; c_2 \colon [a/z]C}$$

GLAD-RAISE
$$\frac{\mathfrak{m}_1 \le \mathfrak{m}_2 \qquad \mathfrak{m}_2 \le \mathcal{M} \qquad \delta \mid \mathcal{M} \odot \Gamma \vdash_{\mathfrak{m}_1} a \colon A}{\delta \mid \mathcal{M} \odot \Gamma \vdash_{\mathfrak{m}_2} \uparrow_{\mathfrak{m}_1}^{\mathfrak{m}_2} a \colon \uparrow_{\mathfrak{m}_1}^{\mathfrak{m}_2} A}$$

GLAD-UNRAISE
$$\frac{\delta \mid \mathcal{M} \odot \Gamma \vdash_{\mathfrak{m}_2} a \colon \uparrow_{\mathfrak{m}_1}^{\mathfrak{m}_2} A}{\delta \mid \mathcal{M} \odot \Gamma \vdash_{\mathfrak{m}_1} \downarrow_{\mathfrak{m}_1}^{\mathfrak{m}_2} a \colon A}$$

Figure 12. GLAD typing rules continued

there are two dependent function types, one between intuitionistic types and one whose functions take intuitionistic arguments and produce linear terms.

Metatheory

We discuss some metatheory of GLAD, and return to the point of controlling contraction in GLAD. Substitution holds in GLAD. We use Convention 2.10 again, but extend it to also imply that lists of modes $\mathcal{M}$ have the same length as the corresponding grade vectors and contexts.

Theorem 3.3 (GLAD substitution). *The following hold by mutual induction:*

  i) (Contexts) If $\delta, r, \delta' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x \colon A, \Gamma' \vdash \text{ctx}$ and $\delta_0 \mid \mathcal{M} \odot \Gamma \vdash_{\mathfrak{m}_0} a_0 \colon A$ then

$$\delta + r \cdot \delta_0, \delta' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma' \vdash \text{ctx}$$

  ii) (Terms) If $\delta, r, \delta' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x \colon A, \Gamma' \vdash_\mathfrak{m} b \colon B$ and $\delta_0 \mid \mathcal{M} \odot \Gamma \vdash_{\mathfrak{m}_0} a_0 \colon A$, then

$$\delta + r \cdot \delta_0, \delta' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma' \vdash_\mathfrak{m} [a_0/x]b \colon [a_0/x]B$$

The proof is by induction and given in Appendix B. As for DMGL we obtain a graded contraction rule as a corollary.

Corollary 3.4 (GLAD contraction). *If*

$$\delta, r_1, r_2, \delta' \mid \mathcal{M}, \mathfrak{m}, \mathfrak{m}, \mathcal{M}' \odot \Gamma, x \colon A, y \colon A, \Gamma' \vdash_\mathfrak{n} b \colon B,$$

then

$$\delta, r_1 + r_2, \delta' \mid \mathcal{M}, \mathfrak{m}, \mathcal{M}' \odot \Gamma, x \colon A, [x/y]\Gamma' \vdash_\mathfrak{n} [x/y]b \colon [x/y]B.$$

Proof. Apply substitution with the judgment

$$\vec{0}, 1 \mid \mathcal{M}, \mathfrak{m} \odot \Gamma, x \colon A \vdash_\mathfrak{m} x \colon A$$

obtained from the GLAD-VAR rule. $\square$

Upon closer inspection, the thing that makes contraction work in our system, is the form of the substitution theorem. In its statement the grades δ_0 for constructing $a : A$ are added to the ones used in the construction of $b : B$. In other words the substitution theorem contains a contraction implicitly. In fact, contraction is implicitly included in the rules of GLAD: For example in the rule GLAD-UNITELIM, the grade vectors δ and δ' are added. This is implicitly a contraction, as we are adding grades instead of concatenating contexts.

But concatenating contexts is not well-suited to the dependently typed setting. When we concatenate contexts in a simply-typed system which has control over contraction, there is a renaming of the variables in the context to avoid name clashes. While such a renaming is possible in a dependently typed system, it immediately becomes very difficult to type any terms as the variables occurring in a term may also be part of its type. For example consider the rule GLAD-APP: If we opted to rename variables, in order to know that the application $c \, a$ is well typed, we would need to remember the fact that the domain type of c and the type of a were equal prior to the renaming. It is not clear to us at the moment how to handle this renaming and to how to incorporate it with grades. We leave an investigation of this for future work.

Examples

In this section, we give some example instantiations of GLAD and show how we may recover existing graded and mixed systems.

Example 3.5 (Recovering DMGL). We explore the relationship between DMGL and GLAD. Let R be a semiring. We ask the following question: How closely can we approximate DMGL graded by R using modes of GLAD? A natural approach to solving this, is to take GLAD with two modes, one with the semiring R and one with a semiring that captures linearity. But since GLAD admits contraction, while the linear fragment of DMGL does not, the latter semiring cannot

exist. Because of this, we can only hope to recover a version of DMGL where the linear fragment is graded as in Example 2.2. This turns out to work: We take as modes L (linear) and G (graded) with $L \leq G$ and set $R_G = R$ and $R_L = \mathbb{N}$ with the trivial preorder $n \leq m \iff n = m$. We also take Weak(G) = True and Weak(L) = False. There exists a unique morphism of modes $\phi \colon L \to G$. GLAD instantiated with this data produces a system with two fragments, one graded by R and which has the same rules as the graded fragment of DMGL. The other fragment corresponds to the mixed fragment of DMGL and has assumptions graded by R as well as by $\mathbb{N}$, with the assumptions graded by $\mathbb{N}$ behaving in a way that's similar to Bounded Linear Logic.

Example 3.6 (Recovering LNL$_D$ [11]). Let U be the unrestricted mode where $R_U = 0$, the trivial semiring with $0 = 1$ and Weak(U) = True. Furthermore, let L be the mode with R_L the none-one-tons semiring of Example 2.3 with the reflexive preorder relation and Weak(L) = False. Variables in mode U have no grading information attached to them, and therefore can be used intuitionistically. On the other hand, variables in mode L are used linearly by default and may not be weakened or duplicated. We have a unique morphism of modes $L \to U$. Instantiating GLAD with this data allows us to recover a system similar to LNL$_D$.

Example 3.7. In this example we consider two modes W, R with the semiring for both modes being the none-one-tons semiring of Example 2.3. We set Weak(W) = True and Weak(R) = False. We chose the preorder generated by $0 \leq \omega$ for W and the reflexive preorder for R. There is now a morphism of modes $R \to W$ which is the identity morphism on the underlying semirings. The mode W admits weakening, for variables used with grades 0 or ω, while the mode R does not. In other words, R is a relevance logic.

Similar to the *of course* modality ! of linear logic, and its decomposition into two adjoints F and G in LNL, we can use the mode shifting operators of GLAD to introduce irrelevantly used variables to the mode R in a controlled way.

Example 3.8. Consider the semiring Var = $\{\sim\sim, \uparrow\uparrow, \downarrow\downarrow, ??\}$, with the preorder generated by $\sim\sim \leq \uparrow\uparrow, \downarrow\downarrow \leq ??$, with $0 = ??$, $1 = \uparrow\uparrow$, addition defined by $a + b = \inf(a, b)$ the greatest lower bound on $\{a, b\}$ and multiplication determined by the equations

$$\downarrow\downarrow \cdot \downarrow\downarrow = \uparrow\uparrow,$$
$$\sim\sim \cdot \downarrow\downarrow = \sim\sim \cdot \sim\sim = \sim\sim$$

and the requirement that multiplication is commutative. This is the *variance* semiring introduced by Wood and Atkey [20] and it allows to track whether a term depends on a variable covariantly ($\uparrow\uparrow$), contravariantly ($\downarrow\downarrow$), invariantly ($\sim\sim$), or if there are no guarantees (??). We define the mode V to have R_V = Var and Weak V = True.

We add two more modes: L with $R_L = \mathbb{N}$ and M with R_M the none-one-tons semiring. We take the preorders on

these semirings to be the trivial reflexive preorders. Furthermore, we set Weak L = False and Weak M = True. There are unique morphisms of modes $L \to M \to V$.

4 Discussion, Future Work, Related Work
Related Work

Closely related to GLAD is the framework of Licata et al. [12]. Their system is a simply typed linear sequent calculus equipped with a mode theory where every formula and judgment is annotated with a mode that constrains the structural rules allowed within the context. The modes found in GLAD are not as elaborate as the mode theory found in their system. Their modes are generic and have morphisms between them, but our modes are specifically semirings. In addition, GLAD is dependently typed.

GLAD is based on GRAD of Choudhury et al. [4]. GRAD is essentially identical to the graded side of DMGL with the addition of the modal operators. However, GLAD differs substantially from GRAD in that the former now supports multiple semirings and a theory of modes.

The Graded Modal Dependent Type Theory (GMDTT) of Moon et al. [13] is very similar to the graded side of DMGL and our second system GLAD, but GMDTT strives to track resource usage in types as well as terms where GLAD and DMGL only tracks resource usage in terms. In addition, GMDTT is not based on the theory of adjoints and modal operators in line with LNL and Adjoint Logic.

Gratzer et al. [8] propose modal dependent type theory which uses modes to support the embedding of a family of modal logics. Their mode theory is similar to the one found in GLAD, but our system's goal is to relate graded type systems and theirs is to relate modal logics.

Simply Typed Version with Control over Contraction

We presented GLAD as a dependently typed system with grading and modes, similar to adjoint logic. The fact that GLAD is dependently typed makes it difficult to control for contraction in a manner that's similar to adjoint logic, and we have given an argument for why this is the case. It appears that the difficulties with controlling graded contraction disappear if one considers a simply typed system instead. As a next step, we will investigate a simply typed version of GLAD in which we can control for contraction. We will take the approach of equipping a mode $\mathfrak{m}$ with a subset $\text{Cont}(\mathfrak{m}) \subseteq R_{\mathfrak{m}}$ which is closed under addition, but may also need to satisfy other algebraic properties. We can then introduce an explicit graded contraction rule such as

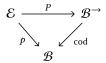$$\frac{r, q \in \text{Cont}(\mathfrak{m}) \quad \delta, r, q, \delta' \mid \mathcal{M}, \mathfrak{m}, \mathfrak{m}, \mathcal{M}' \odot \Gamma, x : A, y : A, \Gamma' \vdash_{\mathfrak{n}} b : B}{\delta, r + q, \delta' \mid \mathcal{M}, \mathfrak{m}, \mathcal{M}' \odot \Gamma, x : A, \Gamma' \vdash_{\mathfrak{n}} [x/y]b : B}$$

Categorical Semantics

Categorical semantics for dependent graded type systems are not well explored at the time of writing. The only approach know to us is presented for Atkey's QTT [1]. However, Katsumata [10] has developed a general approach to the categorical semantics using graded linear exponential comonads and formulates the coherence conditions on such comonads in a compact way using double categories. In our preliminary considerations on the categorical semantics of GLAD, we have recovered Katsumata's approach exactly.

A common approach to categorical semantics of dependent type theory is through categories with families [5] and this is also the approach taken by Atkey via quantitative categories with families (QCwF's). A similar approach is to use comprehension categories [9]. We believe the latter to be slightly nicer, as the category of comprehension categories embeds into the 2-category of cartesian fibrations over a base category $\mathcal{B}$ and this 2-category enjoys nice properties. Furthermore, a comprehension category can be very compactly described as morphism of cartesian fibrations

$$\mathcal{E} \xrightarrow{\;\;P\;\;} \mathcal{B}^{\rightarrow}$$
$$p \searrow \qquad \swarrow \text{cod}$$
$$\mathcal{B}$$

with $\mathcal{B}$ cartesian closed, $\mathcal{B}^{\rightarrow}$ the category of arrows in $\mathcal{B}$ and cod the codomain fibration. In this regard, one minor criticism we have of Atkey's QCwF's, is that they do not (or at least are not know to) arise as an instantiation of a more general categorical concept, like CwF's do with fibrations.

In category theory, it is often helpful to formulate specific concepts as instances of more general ones. Our goal for the future is to combine the general categorical pictures provided by Katsumata's graded linear exponential comonads and comprehension categories to develop categorical semantics for graded dependent type theory.

Conclusion

In the present work we have presented two graded dependent type systems. The first was a obtained by replacing the dependent fragment of LNL$_D$ with the graded dependent type system GRAD. The second type system is a further generalization of the first, allowing different assumptions to be graded by grades coming from different semirings. This system resembles adjoint logic in its structure, and employs a similar construct of *modes*. We proved meta-theoretic properties of these systems: For the former we proved substitution and presented a reduction relation which we showed to preserve grading and types. The latter system was proven to admit substitution and full graded contraction.

Acknowledgments

References

[1] Robert Atkey. Syntax and semantics of quantitative type theory. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*. ACM, jul 2018.

[2] P. N. Benton. A mixed linear and non-linear logic: Proofs, terms and models. In *Computer Science Logic*, pages 121–135. Springer Berlin Heidelberg, 1995.

[3] Luís Caires and Frank Pfenning. Session types as intuitionistic linear propositions. In Paul Gastin and François Laroussinie, editors, *CONCUR 2010 - Concurrency Theory*, volume 6269 of *Lecture Notes in Computer Science*, pages 222–236. Springer Berlin Heidelberg, 2010.

[4] Pritam Choudhury, Harley Eades III, Richard A. Eisenberg, and Stephanie Weirich. A graded dependent type system with a usage-aware semantics. *Proceedings of the ACM on Programming Languages*, 5(POPL):1–32, jan 2021.

[5] Peter Dybjer. Internal type theory. In Stefano Berardi and Mario Coppo, editors, *Types for Proofs and Programs*, pages 120–134, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[6] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987.

[7] Jean-Yves Girard, Andre Scedrov, and Philip J. Scott. Bounded linear logic: a modular approach to polynomial-time computability. *Theoretical Computer Science*, 97(1):1–66, 1992.

[8] Daniel Gratzer, G. A. Kavvos, Andreas Nuyts, and Lars Birkedal. Multimodal dependent type theory. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '20, page 492–506, New York, NY, USA, 2020. Association for Computing Machinery.

[9] Bartholomeus Paulus Franciscus Jacobs. *Categorical type theory*. Nijmegen, 1991. PhD thesis.

[10] Shin-ya Katsumata. A double category theoretic analysis of graded linear exponential comonads. In *Lecture Notes in Computer Science*, pages 110–127. Springer International Publishing, 2018.

[11] Neelakantan R. Krishnaswami, Pierre Pradic, and Nick Benton. Integrating linear and dependent types. *ACM SIGPLAN Notices*, 50(1):17–30, may 2015.

[12] Daniel R. Licata, Michael Shulman, and Mitchell Riley. A Fibrational Framework for Substructural and Modal Logics. In Dale Miller, editor, *2nd International Conference on Formal Structures for Computation and Deduction (FSCD 2017)*, volume 84 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 25:1–25:22, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[13] Benjamin Moon, Harley Eades III, and Dominic Orchard. Graded modal dependent type theory. In *Programming Languages and Systems*, pages 462–490. Springer International Publishing, 2021.

[14] Bengt Nordstrom, Kent Petersson, and Jan M. Smith. *Programming in Martin-Löf's Type Theory: An Introduction*. Oxford University Press, USA, July 1990.

[15] Peter O'Hearn, John Reynolds, and Hongseok Yang. Local reasoning about programs that alter data structures. In Laurent Fribourg, editor, *Computer Science Logic*, pages 1–19, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[16] Peter W. O'Hearn and David J. Pym. The logic of bunched implications. *The Bulletin of Symbolic Logic*, 5(2):215–244, 1999.

[17] Dominic Orchard, Vilem-Benjamin Liepelt, and Harley Eades III. Quantitative program reasoning with graded modal types. *Proc. ACM Program. Lang.*, 3(ICFP), jul 2019.

[18] Klaas Pruiksma, William Chargin, Frank Pfenning, and Jason Reed. Adjoint logic. Unpublished Draft:

http://www.cs.cmu.edu/ fp/papers/adjoint18b.pdf, 2018.

[19] Victoria Vollmer, Harley Eades III, Daniel Marshall, and Dominic Orchard. A mixed linear and graded logic: Proofs, terms and models. forthcoming.

[20] James Wood and Robert Atkey. A framework for substructural type systems. In Ilya Sergey, editor, *Programming Languages and Systems*, pages 376–402, Cham, 2022. Springer International Publishing.

A   Metatheory of DMGL

Proposition 2.7. *The following hold by mutual induction:*

  i) If $\delta \odot \Delta \vdash_G t : X$, then $\delta \odot \Delta \vdash_G$ ctx.

  ii) If $\delta \odot \Delta; \Gamma \vdash_M l : A$, then $\delta \odot \Delta \vdash_G$ ctx and $\delta \odot \Delta; \Gamma \vdash_M$ ctx.

Lemma A.1. *Let δ, δ' be grade vectors of the same length.*

  i) If $\delta \odot \Delta \vdash_G$ ctx, then $\delta' \odot \Delta \vdash_G$ ctx.

  ii) If $\delta \odot \Delta; \Gamma \vdash_M$ ctx, then $\delta' \odot \Delta; \Gamma \vdash_M$ ctx.

Proof.   i) By induction on the derivation of $\delta \odot \Delta \vdash_G$ ctx. Notice that in the rule for graded context extension

$$\frac{x \notin \text{dom } \Delta}{\delta \odot \Delta \vdash_G \text{ ctx} \qquad \delta'_0 \odot \Delta \vdash_G X : \text{Type}}{\delta, r \odot \Delta, x : X \vdash_G \text{ ctx}}$$

the grade r by which we the grade vector is extended is arbitrary. So, in the derivation of $\delta \odot \Delta \vdash_G$ ctx "build up" δ' instead of δ.

  ii) Similarly to i), the grade vector is not relevant to extending the linear section of a mixed context.   □

Lemma A.2. *If $\delta \odot \Delta \vdash_G$ ctx then $\delta \odot \Delta; \Gamma \vdash_M$ ctx is provable if and only if for each type A occurring in Γ, the judgment $\delta' \odot \Delta \vdash_G A : \text{Linear}$ is provable for some δ', and all variables in Γ are distinct and do not occur in Δ.*

Proof. This is easily seen by induction on the length of Γ using the rule MIXEDCTX-EXTEND.   □

Proof of Proposition 2.7. Straightforward, by mutual induction on the derivations of $\delta \odot \Delta \vdash_G t : X$ and $\delta \odot \Delta; \Gamma \vdash_M l : A$ respectively. For the graded fragment, most of the work is done by Lemma A.1, and Lemma A.2 takes care of the mixed fragment.   □

Theorem 2.11 (Substitution). *The following hold by mutual induction:*

  i) (Graded Contexts) If $\delta_0 \odot \Delta \vdash_G t_0 : X$, and $\delta, r, \delta' \odot \Delta, x : X, \Delta' \vdash_G$ ctx then

$$\delta + r \cdot \delta_0, \delta' \odot \Delta, [t_0/x]\Delta' \vdash_G \text{ ctx}$$

  ii) (Mixed Contexts) If $\delta_0 \odot \Delta \vdash_G t_0 : X$, and $\delta, r, \delta' \odot \Delta, x : X, \Delta'; \Gamma \vdash_M$ ctx then

$$\delta + r \cdot \delta_0, \delta' \odot \Delta, [t_0/x]\Delta'; [t_0/x]\Gamma \vdash_M \text{ ctx}$$

  iii) (Graded) If $\delta_0 \odot \Delta \vdash_G t_0 : X$ and $\delta, r, \delta' \odot \Delta, x : X, \Delta' \vdash_G t : Y$, then

$$\delta + r \cdot \delta_0, \delta' \odot \Delta, [t_0/x]\Delta' \vdash_G [t_0/x]t : [t_0/x]Y$$

  iv) (Mixed Graded) If $\delta_0 \odot \Delta \vdash_G t_0 : X$ and $\delta, r, \delta' \odot \Delta, x : X, \Delta'; \Gamma \vdash_M l : A$, then

$$\delta + r \cdot \delta_0 \odot \Delta, [t_0/x]\Delta'; [t_0/x]\Gamma \vdash_M [t_0/x]l : [t_0/x]A$$

  v) (Mixed Linear) If $\delta_0 \odot \Delta; \Gamma_0 \vdash_M l_0 : A$ and $\delta \odot \Delta; \Gamma, x : A, \Gamma' \vdash_M l : B$, then

$$\delta + \delta_0 \odot \Delta; \Gamma, \Gamma_0, \Gamma' \vdash_M [l_0/x]l : B$$

Proof. The proof is by induction over the derivation of the second judgment. For each rule, we need to pattern-match the resulting judgment with the pattern in the statement of the theorem, which we will do at the beginning of each case.

Case G-SUBUSAGE

$$\frac{\delta_1, r_1, \delta'_1 \le \delta_2, r_2, \delta'_2 \qquad \delta_1, r_1, \delta'_1 \odot \Delta, x : X, \Delta' \vdash_G t : Y}{\delta_2, r_2, \delta'_2 \odot \Delta, x : X, \Delta' \vdash_G t : Y} \text{ G-SUBUSAGE}$$

By induction we know that $\delta_1 + r_1 \cdot \delta_0, \delta'_1 \odot \Delta, [t_0/x]\Delta' \vdash_G [t_0/x]t : [t_0/x]Y$ and since $+, \cdot$ are monotonic, we have $\delta_1 + r_1 \cdot \delta_0 \le \delta_2 + r_2 \cdot \delta_0$ and hence also $\delta_1 + r_1 \cdot \delta_0, \delta'_1 \le \delta_2 + r_2 \cdot \delta_0, \delta'_2$. Applying G-SUBUSAGE concludes this case.

Case G-WEAK We need to match the judgment $\delta, 0 \odot \Delta, z : Z \vdash_G t : Y$ with the pattern $\delta, r, \delta' \odot \Delta, x : X, \Delta' \vdash_G t : Y$. There are two cases. In the first case we have $\Delta' = \emptyset$ and $X = Z$. Then our judgment was obtained as follows

$$\frac{x \notin \text{dom } \Delta \qquad \delta \odot \Delta \vdash_G t : Y \\ \delta'_0 \odot \Delta \vdash_G X : \text{Type}}{\delta, 0 \odot \Delta, x : X \vdash_G t : Y} \text{ G-WEAK}$$

and we must prove $\delta + 0 \cdot \delta_0 \odot \Delta \vdash_G [t_0/x]x : [t_0/x]Y$. But we have $\delta \odot \Delta \vdash_G t : Y$, and therefore know that the variable x does not occur freely in t or Y. It follows that we need to prove $\delta \odot \Delta \vdash_G t : Y$, which we know by assumption.

In the second case we have $\Delta' \ne \emptyset$. In this case the judgment was obtained by

$$\frac{z \notin \text{dom}(\Delta, x : X, \Delta') \qquad \delta, r, \delta' \odot \Delta, x : X, \Delta' \vdash_G t : Y \\ \delta_1, r_1, \delta'_1 \odot \Delta, x : X, \Delta' \vdash_G Z : \text{Type}}{\delta, r, \delta', 0 \odot \Delta, x : X, \Delta', z : Z \vdash_G t : Y}$$

We need to prove

$$\delta + r \cdot \delta_0, \delta', 0 \odot \Delta, [t_0/x]\Delta', z : [t_0/x]Z \vdash_G [t_0/x]t : [t_0/x]Y.$$

From the inductive hypothesis we have

$$\delta + r \cdot \delta_0, \delta' \odot \Delta, [t_0/x]\Delta' \vdash_G [t_0/x]t : [t_0/x]Y$$

and $\delta_1 + r_1 \cdot \delta_0, \delta'_1 \odot \Delta, [t_0/x]\Delta' \vdash_G [t_0/x]Z : \text{Type}$

and applying G-WEAK to these judgments produces the desired judgment.

Case G-CONVERT Immediate by induction and the fact that "≡" is a congruence relation, i.e. $[t_0/x]Y \equiv [t_0/x]Y'$ follows from $Y \equiv Y'$.

Case G-VAR We need to pattern-match the judgment $\vec{0}, 1 \odot \Delta, z : Z \vdash_G z : Z$ with the pattern $\delta, r, \delta' \odot \Delta, x : X, \Delta' \vdash_G t : Y$. There are two possibilities to do this. In the first, $\Delta' = \emptyset$

and $X = Y = Z$. In this case the judgment we are performing induction on was obtained thus

$$\frac{x \notin \text{dom } \Delta \qquad \delta_1 \odot \Delta \vdash_G X : \text{Type}}{\vec{0}, 1 \odot \Delta, x : X \vdash_G x : X}$$

and we must prove $\vec{0} + 1 \cdot \delta_0 \odot \Delta \vdash_G [t/x]x : [t/x]X$. From $\delta_1 \odot \Delta \vdash_G X : \text{Type}$ and $x \notin \text{dom } \Delta$, we know that x is not free in X. Therefore, we need to prove $\delta_0 \odot \Delta \vdash_G t : X$, which is an assumption.

In the second case, the judgment we are performing induction on was obtained as

$$\frac{\begin{array}{c} z \notin \text{dom } (\Delta, x : X, \Delta') \\ \delta, r, \delta' \odot \Delta, x : X, \Delta' \vdash_G Z : \text{Type} \end{array}}{\vec{0}, 0, \vec{0}, 1 \odot \Delta, x : X, \Delta', z : Z \vdash_G z : Z}$$

and we need to prove $\vec{0}, \vec{0}, 1 \odot \Delta, [t_0/z]\Delta', z : [t_0/x]Z \vdash_G [t_0/x]z : [t_0/x]Z$. But this follows from the inductive hypothesis $\delta + r \cdot \delta_0, \delta' \odot \Delta, \Delta' \vdash_G [t_0/x]Z : \text{Type}$ the variable rule and the fact that $z = [t_0/x]z$.

Cases G-TYPE, G-LINEAR, G-UNIT, G-UNITINTRO Trivial.
Case G-UNITELIM

$$\frac{\begin{array}{c} \delta_1, r_1, \delta'_1 \odot \Delta, x : X, \Delta' \vdash_G t_1 : J \\ \delta_2, r_2, \delta'_2, q_2 \odot \Delta, x : X, \Delta', x' : J \vdash_G X' : \text{Type} \\ \delta_3, r_3, \delta'_3 \odot \Delta, x : X, \Delta' \vdash_G t_3 : [j/x]X' \end{array}}{\begin{array}{c} \delta_1 + \delta_3, r_1 + r_3, \delta'_1 + \delta'_3 \odot \Delta, x : X, \Delta' \\ \vdash_G \text{let } j = t_1 \text{ in } t_3 : [t_1/x']X' \end{array}}$$

By induction we have

$$\delta_1 + r_1 \cdot \delta_0, \delta'_1 \odot \Delta, [t_0/x]\Delta' \vdash_G [t_0/x]t_1 : J$$

$$\delta_2 + r_2 \cdot \delta_0, \delta'_2, q_2 \odot \Delta, [t_0/x]\Delta', x' : J \vdash_G [t_0/x]X' : \text{Type}$$

$$\delta_3 + r_3 \cdot \delta_0, \delta'_3 \odot \Delta, [t_0/x]\Delta' \vdash_G [t_0/x]t_3 : [t_0/x][j/x']X'$$

Applying G-UNITELIM to these judgments yields

$$\begin{array}{c} \delta_1 + r_1 \cdot \delta_0 + \delta_3 + r_3 \cdot \delta_0, \delta'_1 + \delta'_3 \odot \Delta, [t_0/x]\Delta' \\ \vdash_G \text{let } j = [t_0/x]t_1 \text{ in } [t_0/x]t_3 : [t_0/x][j/x']X' \end{array}$$

Since j has no free variables, the terms $[t_0/x](\text{let } j = t_1 \text{ in } t_3)$ and let $j = [t_0/x]t_1$ in $[t_0/x]t_3$ are equal, which concludes this case.
Case G-RADJINTRO

$$\frac{\delta, r, \delta' \odot \Delta, x : X, \Delta'; \emptyset \vdash_M l : A}{\delta, r, \delta' \odot \Delta, x : X, \Delta' \vdash_G \mathcal{G} l : \mathcal{G} A}$$

Applying the induction hypothesis to the premise of the rule, we get

$$\delta, r \cdot \delta_0, \delta' \odot \Delta, [t_0/x]\Delta'; \emptyset \vdash_M [t_0/x]l : [t_0/x]A$$

Applying G-RADJINTRO to this judgment, concludes this case, once we observe that $[t_0/x](\mathcal{G} l)$ is equal to $\mathcal{G} [t_0/x]l$ and similarly for A instead of l.

The remaining cases in the graded fragment follow analogously and are therefore omitted.

Case M-ID In the mixed graded case we have

$$\frac{\begin{array}{c} y \notin \text{dom } (\Delta, x : X, \Delta') \\ \delta, r, \delta' \odot \Delta, x : X, \Delta' \vdash_G A : \text{Linear} \end{array}}{\vec{0}, 0, \vec{0} \odot \Delta, x : X, \Delta'; y : A \vdash_M y : A}$$

and we need to prove $\vec{0}, \vec{0} \odot \Delta, [t_0/x]\Delta'; y : [t_0/x]A \vdash_M y : [t_0/x]A$ which follows immediately from the inductive hypothesis and the M-ID rule. There is nothing to do for the mixed linear case.
Case M-SUBUSAGE Both the mixed linear and mixed graded cases are analogous to the G-SUBUSAGE case.
Case M-WEAK The mixed graded case is analogous to the purely graded case. For the mixed linear case, the rule M-WEAK pattern matches as

$$\frac{\delta \odot \Delta; \Gamma, y : A, \Gamma' \vdash_M l : B \qquad \delta' \odot \Delta \vdash_G X : \text{Type}}{\delta, 0 \odot \Delta, x : X; \Gamma, y : A, \Gamma' \vdash_M l : B}$$

By induction on the first hypothesis we have

$$\delta \odot \Delta; \Gamma, \Gamma_0, \Gamma' \vdash_M [l_0/y]l : B$$

Weakening this by X produces the desired result.
Case M-EXCHANGE The mixed graded case is straightforward and we omit the details. For the mixed linear case, consider the rule

$$\frac{\delta \odot \Delta; \Gamma, x : B_1, y : B_2, \Gamma' \vdash_M l : C}{\delta \odot \Delta; \Gamma, y : B_2, x : B_1, \Gamma' \vdash_M l : C} \text{M-EXCHANGE}$$

where we assume that the type A appears somewhere in the context $\Gamma, x : B_1, y : B_2, \Gamma'$. The cases where A appears in Γ or Γ' are straightforward. The cases that remain are that A is B_1 or B_2. These cases are parallel, so we only treat the case that A is B_2. The inductive hypothesis yields

$$\delta \odot \Delta; \Gamma, x : B_1, \Gamma_0, \Gamma' \vdash_M [l_0/y]l : C$$

and we want to show that

$$\delta \odot \Delta; \Gamma, \Gamma_0, x : B_1, \Gamma' \vdash_M [l_0/y]l : C$$

This is done by repeatedly applying the exchange rule to move B_1 past all types in Γ_0.
M-TENSORINTRO For the mixed graded case we have

$$\frac{\begin{array}{c} \delta_1, r_1, \delta'_1 \odot \Delta, x : X, \Delta'; \Gamma_1 \vdash_M l_1 : A \\ \delta_2, r_2, \delta'_2 \odot \Delta, x : X, \Delta'; \Gamma_2 \vdash_M l_2 : B \end{array}}{(\delta_1, r_1, \delta'_1) + (\delta_2, r_2, \delta'_2) \odot \Delta, x : X, \Delta'; \Gamma_1, \Gamma_2 \vdash_M (l_1, l_2) : A \otimes B}$$

By the inductive hypothesis we have the judgments

$$\delta_1 + r_1 \cdot \delta_0, \delta'_1 \odot \Delta, [t_0/x]\Delta'; [t_0/x]\Gamma_1 \vdash_M [t_0/x]l_1 : [t_0/x]A$$

$$\delta_2 + r_2 \cdot \delta_0, \delta'_2 \odot \Delta, [t_0/x]\Delta'; [t_0/x]\Gamma_2 \vdash_M [t_0/x]l_2 : [t_0/x]B$$

Applying M-TENSORINTRO yields

$$\begin{array}{c} (\delta_1 + \delta_2) + (r_1 + r_2) \cdot \delta_0, \delta'_1 + \delta'_2 \\ \odot \Delta, [t_0/x]\Delta'; [t_0/x]\Gamma_1, [t_0/x]\Gamma_2 \\ \vdash_M ([t_0/x]l_1, [t_0/x]l_2) : ([t_0/x]A) \otimes ([t_0/x]B) \end{array}$$

We conclude by noting that the terms $([t_0/x]l_1, [t_0/x]l_2)$ and $[t_0/x](l_1, l_2)$ are equal, and similarly for the types $([t_0/x]A) \otimes ([t_0/x]B)$ and $[t_0/x](A \otimes B)$.

For the mixed linear case of this rule we have

$$\frac{\delta_1 \odot \Delta; \Gamma_1 \vdash_M l_1 : B_1 \qquad \delta_2 \odot \Delta; \Gamma_2 \vdash_M l_2 : B_2}{\delta_1 + \delta_2 \odot \Delta; \Gamma_1, \Gamma_2 \vdash_M (l_1, l_2) : B_1 \otimes B_2}$$

and we assume that $x : A$ occurs somewhere in the linear context Γ_1, Γ_2. There are two symmetric cases, depending on whether $x : A$ occurs in Γ_1 or Γ_2. We consider the case where $x : A$ occurs in Γ_1, i.e. Γ_1 has the form $\Gamma'_1, x : A, \Gamma''_1$. From the inductive hypothesis we no get $\delta_1 \odot \Delta; \Gamma'_1, \Gamma''_1 \vdash_M [l_0/x]l_1 : B_1$. Applying M-TENSORINTRO now yields $\delta_1 + \delta_2 \odot \Delta; \Gamma'_1, \Gamma''_1, \Gamma_2 \vdash_M ([l_0/x]l_1, l_2) : B_1 \otimes B_2$. Since we rename variables when concatenating linear contexts, the terms l_1 and l_2 have disjoint sets of free variables, and it follows that the terms $([l_0/x]l_1, l_2)$ and $[l_0/x](l_1, l_2)$ are equal. This concludes this case.

M-TENSORELIM For the mixed graded case we have

$$\frac{\delta_2, r_2, \delta'_2 \odot \Delta, x : X, \Delta'; \Gamma_2 \vdash_M l_1 : B_1 \otimes B_2 \qquad \delta_1, r_1, \delta'_1 \odot \Delta, x : X, \Delta'; \Gamma_1, y_1 : B_1, y_2 : B_2, \Gamma_3 \vdash_M l_2 : C}{\begin{array}{c}(\delta_1, r_1, \delta'_1) + (\delta_2, r_2, \delta'_2) \odot \Delta, x : X, \Delta'; \Gamma_1, \Gamma_2, \Gamma_3 \\ \vdash_M \text{let } (y_1, y_2) = l_1 \text{ in } l_2 : C\end{array}}$$

and we need to prove

$$(\delta_1 + \delta_2) + (r_1 + r_2) \cdot \delta_0, \delta'_1 + \delta'_2$$
$$\odot \Delta, [t_0/x]\Delta'; [t_0/x](\Gamma_1, \Gamma_2, \Gamma_3)$$
$$\vdash_M [t_0/x](\text{let } (y_1, y_2) = l_1 \text{ in } l_2) : [t_0/x]C$$

By the inductive hypothesis we have the judgments

$$\delta_2 + r_2 \cdot \delta_0, \delta'_2 \odot \Delta, [t_0/x]\Delta'; [t_0/x]\Gamma_2$$
$$\vdash_M [t_0/x]l_1 : [t_0/x]B_1 \otimes [t_0/x]B_2$$

$$\delta_1 + r_1 \cdot \delta_0, \delta'_1 \odot \Delta, [t_0/x]\Delta'$$
$$; [t_0/x](\Gamma_1, y_1 : B_1, y_2 : B_2, \Gamma_3)$$
$$\vdash_M [t_0/x]l_2 : [t_0/x]C$$

from which the desired judgment follows.

For the mixed linear case, we have

$$\frac{\delta_2 \odot \Delta; \Gamma_2 \vdash_M l_1 : B_1 \otimes B_2 \qquad \delta_1 \odot \Delta; \Gamma_1, y_1 : B_1, y_2 : B_2, \Gamma_3 \vdash_M l_2 : C}{\delta_1 + \delta_2 \odot \Delta; \Gamma_1, \Gamma_2, \Gamma_3 \vdash_M \text{let } (y_1, y_2) = l_1 \text{ in } l_2 : C}$$

We assume $x : A$ occurs somewhere in the context $\Gamma_1, \Gamma_2, \Gamma_3$. We have three cases, depending on whether $x : A$ occurs in Γ_1, Γ_2 or Γ_3. In the first case, where $x : A$ occurs in Γ_1, the second assumption of the rule above has the form

$$\delta_1 \odot \Delta; \Gamma'_1, x : A, \Gamma''_1, y_1 : B, y_2 : B, \Gamma_2 \vdash_M l_2 : C$$

By induction we have

$$\delta_1 \odot \Delta; \Gamma'_1, \Gamma''_1, y_1 : B, y_2 : B, \Gamma_2 \vdash_M [l_0/x]l_2 : C$$

and applying the rule M-TENSORELIM to this judgment together with the other assumption of the rule above produces

the judgment we need to prove. The other two cases proceed analogously.

The other cases for the mixed fragment proceed analogously. We highlight the cases involving the adjoints $\mathcal{G}$ and $\mathcal{F}$, as they are nonstandard.

Case M-RADJELIM The mixed linear case is trivial. For the mixed graded case:

$$\frac{\delta, r, \delta' \odot \Delta, x : X, \Delta' \vdash_G t : \mathcal{G}A}{\delta, r, \delta' \odot \Delta, x : X, \Delta'; \emptyset \vdash_M \mathcal{G}^{-1}t : A}$$

As in the previous case, we have that the terms $[t_0/x]\mathcal{G}^{-1}t$ and $\mathcal{G}^{-1}[t_0/x]t$ are equal, and $[t_0/x](\mathcal{G}A)$ and $\mathcal{G}[t_0/x]A$ are also equal. So, by applying the induction hypothesis to the the premise of the rule above and then applying M-RADJELIM we get

$$\frac{\delta + r \cdot \delta_0 + \delta' \odot \Delta, [t_0/x]\Delta' \vdash_G [t_0/x]t : \mathcal{G}[t_0/x]A}{\delta + r \cdot \delta_0 + \delta' \odot \Delta, [t_0/x]\Delta'; \emptyset \vdash_M \mathcal{G}^{-1}[t_0/x]t : [t_0/x]A}$$

Observe that the terms $\mathcal{G}^{-1}[t_0/x]t$ and $[t_0/x]\mathcal{G}^{-1}t$ are equal. This produces the desired result.

Case M-LADJINTRO In the mixed linear case we have

$$\frac{\delta_3, r_3 \odot \Delta, x : X \vdash_G A : \text{Linear} \quad \delta_1 \odot \Delta \vdash_G t : X \quad \delta_2 \odot \Delta; \Gamma, y : B, \Gamma' \vdash_M l : [t/x]A}{r \cdot \delta_1 + \delta_2 \odot \Delta; \Gamma, y : B, \Gamma' \vdash_M \mathcal{F}(t, l) : \mathcal{F}(x :^r X).A}$$

Keeping the first two premises unchanged, and applying the induction hypothesis to the third premise, we can use the rule M-LADJINTRO as follows

$$\frac{\delta_3, r_3 \odot \Delta, x : X \vdash_G A : \text{Linear} \quad \delta_1 \odot \Delta \vdash_G t : X \quad \delta_0 + \delta_2 \odot \Delta; \Gamma, \Gamma_0, \Gamma' \vdash_M [l_0/y]l : [t/x]A}{r \cdot \delta_1 + \delta_2 \odot \Delta; \Gamma, \Gamma_0, \Gamma' \vdash_M \mathcal{F}(t, [l_0/y]l) : \mathcal{F}(x :^r X).A}$$

To conclude, we only need to observe that $\mathcal{F}(t, [l_0/y]l)$ is equal to $[l_0/y]\mathcal{F}(t, l)$ which is clear since y cannot occur freely in t.

The mixed graded case is more interesting:

$$\frac{\delta_3, r_3, \delta'_3, r'_3 \odot \Delta, y : Y, \Delta', x : X \vdash_G A : \text{Linear} \quad \delta_1, r_1, \delta'_1 \odot \Delta, y : Y, \Delta' \vdash_G t : X \quad \delta_2, r_2, \delta'_2 \odot \Delta, y : Y, \Delta'; \Gamma \vdash_M l : [t/x]A}{\begin{array}{c}r \cdot (\delta_1, r_1, \delta'_1) + (\delta_2, r_2, \delta'_2) \odot \Delta, y : Y, \Delta'; \Gamma \\ \vdash_M \mathcal{F}(t, l) : \mathcal{F}(x :^r X).A\end{array}}$$

applying the induction hypothesis to all premises, we get the following judgments:

$$\delta_3, r_3, \delta'_3 \odot \Delta, [t_0/y]\Delta', x : [t_0/y]X \vdash_G [t_0/y]A : \text{Linear}$$

$$r_1 \cdot \delta_0 + \delta_1, \delta'_1 \odot \Delta, [t_0/y]\Delta' \vdash_G [t_0/y]t : [t_0/y]X$$

$$r_2 \cdot \delta_0 + \delta_2, \delta'_2 \odot \Delta, [t_0/y]\Delta'; [t_0/y]\Gamma \vdash_M [t_0/y]l : [t_0/y][t/x]A$$

We cannot immediately apply the rule M-LADJINTRO to these sequents, as the third one does not have the right form. We observe that $[t_0/y][t/x]A$ is equal to $[[t_0/y]t/x][t_0/y]A$.

Substituting this into the third sequent puts it into the right form to apply M-ʟᴀᴅᴊIɴᴛʀᴏ and we obtain

$$\frac{\begin{array}{c} \delta_3, r_3, \delta_3' \odot \Delta, [t_0/y]\Delta', x : [t_0/y]X \vdash_G [t_0/y]A : \text{Linear} \\ r_1 \cdot \delta_0 + \delta_1, \delta_1' \odot \Delta, [t_0/y]\Delta' \vdash_G [t_0/y]t : [t_0/y]X \\ r_2 \cdot \delta_0 + \delta_2, \delta_2' \odot \Delta, [t_0/y]\Delta'; [t_0/y]\Gamma \\ \vdash_M [t_0/y]l : [[t_0/y]t/x][t_0/y]A \end{array}}{\begin{array}{c} r \cdot (r_1 \cdot \delta_0 + \delta_1, \delta_1') + (r_2 \cdot \delta_0 + \delta_2, \delta_2') \odot \Delta, [t_0/y]\Delta'; [t_0/y]\Gamma \\ \vdash_M [t_0/y]l : [[t_0/y]t/x][t_0/y]A \end{array}}$$

Applying the above equality in the conclusion, we obtain the desired result.

Case M-ʟᴀᴅᴊEʟɪᴍ The mixed graded case is straightforward. In the mixed linear case, we assume that we have

$$\frac{\delta_3 \odot \Delta \vdash_G C : \text{Linear} \qquad \delta_1 \odot \Delta; \Gamma_1 \vdash_M l_1 : \mathcal{F}(x :^q X).B \qquad \delta_2, q \odot \Delta, x : X; \Gamma_2, y : B \vdash_M l_2 : C}{\delta_1 + \delta_2 \odot \Delta; \Gamma_1, \Gamma_2 \vdash_M \text{let } \mathcal{F}(x, y) = l_1 \text{ in } l_2 : C}$$

We assume that the type assignment $z : A$ occurs in Γ_1, Γ_2, the linear context of the conclusion of this rule. There are now two subcases:

Case. $z : A$ occurs in Γ_1. Then Γ_1 is of the form $\Gamma_1', z : A, \Gamma_1''$ and by the inductive hypothesis we obtain $\delta_0 + \delta_1 \odot \Delta; \Gamma_1', \Gamma_0, \Gamma_1'' \vdash_M [l_0/z]l_1 : \mathcal{F}(x :^q X).A$.

Case. $z : A$ occurs in Γ_2. Then Γ_2 is of the form $\Gamma_2', z : A, \Gamma_2''$ and by the inductive hypothesis we obtain $\delta_0 + \delta_2 \odot \Delta; \Gamma_2', \Gamma_0, \Gamma_2'' \vdash_M [l_0/z]l_2 : C$.

In either case we conclude immediately, by applying the rule M-ʟᴀᴅᴊEʟɪᴍ to the newly obtained judgment and the two judgments to which we did not apply the inductive hypothesis.

The cases concerned with judgments about well-fromedness of graded and mixed contexts are straightforward. □

Proposition 2.13. *If $\delta \odot \Delta; \Gamma, x : A \vdash_M l : B$, then $\delta, 1 \odot \Delta, y : \mathcal{G} A; \Gamma \vdash_M [\mathcal{G}^{-1} y/x]l : B$.*

Proof. We have a derivation

$$\frac{\dfrac{\vdots}{\vec{0}, 1 \odot \Delta, y : \mathcal{G} A \vdash_G y : \mathcal{G} A} \text{ G-ᴠᴀʀ}}{\vec{0}, 1 \odot \Delta, y : \mathcal{G} A; \emptyset \vdash_M \mathcal{G}^{-1} y : A} \text{ M-ʀᴀᴅᴊEʟɪᴍ}$$

Weakening the assumed sequent by $\mathcal{G} A$ we get

$$\delta, 0 \odot \Delta, y : \mathcal{G} A; \Gamma, x : A \vdash_M l : B$$

Now by substituion, we get

$$(\vec{0}, 1) + (\delta, 0) \odot \Delta, y : \mathcal{G} A; \Gamma \vdash_M [\mathcal{G}^{-1} y/x]l : B$$

as desired. □

In the following we will write

$$\delta \odot \Delta \vdash_G X \equiv Y : \text{Type}$$

to mean that the following hold simultaneously:

1. $\delta \odot \Delta \vdash_G X : \text{Type}$

2. $\delta \odot \Delta \vdash_G Y : \text{Type}$
3. $X \equiv Y$

Lemma A.3 (Lambda Inversion). *If $\delta \odot \Delta \vdash_G (\lambda x.t) : T$, then there exist X, Y, r and δ_0 such that $\delta_0 \odot \Delta \vdash_G T \equiv (x :^r X) \to Y : \text{Type}$ and $\delta, r \odot \Delta, x : X \vdash_G t : Y$.*

Proof. By induction on the derivation of $\delta \odot \Delta \vdash_G (\lambda x.t) : T$.
Case G-ʟᴀᴍʙᴅᴀ There is nothing to do in this case.
Case G-ꜱᴜʙᴜꜱᴀɢᴇ Inverting the subusage rule, we get $\delta' \odot \Delta \vdash_G (\lambda x.t) : T$ for some δ' with $\delta' \le \delta$. By the inductive hypothesis we get $\delta', r \odot \Delta, x : X \vdash_G t : Y$ and by applying subusage we get $\delta', r \odot \Delta, x : X \vdash_G t : Y$ as desired. $\delta_0 \odot \Delta \vdash_G T \equiv (x :^r X) \to Y : \text{Type}$ also follows immediately from the inductive hypothesis.
Case G-ᴡᴇᴀᴋ We have that $\Delta = \Delta', z : Z$ and $\delta = \delta', 0$ for some Δ', δ' and Z. Furthermore we have $\delta' \odot \Delta' \vdash_G (\lambda x.t) : T$. By inductive hypothesis we have $\delta_0 \odot \Delta' \vdash_G T \equiv (x :^r X) \to Y : \text{Type}$ and $\delta', r \odot \Delta', x : X \vdash_G t : Y$. Since Z is a well-formed type in context Δ' it is also well-formed in context $\Delta', x : X$ and hence we may weaken by Z, obtaining $\delta', r, 0 \odot \Delta', x : X, z : Z \vdash_G t : Y$. Similarly X is well formed in context Δ' and hence by exchange we get $\delta', 0, r \odot \Delta', z : Z, x : X \vdash_G t : Y$ as desired.
Case G-ᴄᴏɴᴠᴇʀᴛ Immediate since the relation "$\equiv$" is transitive. □

Lemma A.4 (Unit Inversion). *If $\delta \odot \Delta \vdash_G \mathbf{j} : T$ then we have $\vec{0} \le \delta$ and $\delta_0 \odot \Delta \vdash_G T \equiv \mathbf{J} : \text{Type}$ for some δ_0.*

Proof. By induction on the derivation of $\delta \odot \Delta \vdash_G \mathbf{j} : T$.
Case G-ᴜɴɪᴛ There is nothing to prove in this case.
Case G-ꜱᴜʙᴜꜱᴀɢᴇ Inverting the subusage rule, we get $\delta' \odot \Delta \vdash_G \mathbf{j} : T$ with $\delta' \le \delta$. By inductive hypothesis we have $\delta_0 \odot \Delta \vdash_G T \equiv \mathbf{J} : \text{Type}$ and $\vec{0} \le \delta'$. Since $\le$ is transitive, we obtain $\vec{0} \le \delta$.
Case G-ᴡᴇᴀᴋ We have $\Delta = \Delta', x : X$ and $\delta = \delta', 0$. By inductive hypothesis we have $\vec{0} \le \delta'$ and $\delta_0 \odot \Delta' \vdash_G T \equiv \mathbf{J} : \text{Type}$ weakening the latter by X yields the second claim. The first claim follows since $\vec{0}, 0 \le \delta', 0$.
Case G-ᴄᴏɴᴠᴇʀᴛ By inductive hypothesis and the fact that the relation "$\equiv$" is transitive. □

Lemma A.5 (Pair Inversion). *If $\delta \odot \Delta \vdash_G (t_1, t_2) : T$, then*

i) $\delta_0' \odot \Delta \vdash_G X : \text{Type}$ *and* $\delta_0'', r_0'' \odot \Delta, x : X \vdash_G Y : \text{Type}$
ii) $\delta_0 \odot \Delta \vdash_G T \equiv (x :^r X) \boxtimes Y : \text{Type}$
iii) $\delta_1 \odot \Delta \vdash_G t_1 : X$
iv) $\delta_2 \odot \Delta \vdash_G t_2 : [t_1/x]Y$
v) $r \cdot \delta_1 + \delta_2 \le \delta$

for some $X, Y, \delta_0, \delta_0', \delta_0'', r_0'', \delta_1, \delta_2$.

Proof. By induction on the derivation of $\delta \odot \Delta \vdash_G (t_1, t_2) : T$.
Case G-ɢʀᴀᴅᴇᴅPᴀɪʀIɴᴛʀᴏ In this case, there is nothing to prove.
Case G-ᴡᴇᴀᴋ This is similar to the proofs of the previous inversion lemmas: Everything follows by applying weakening to the respective points in the inductive hypothesis.

Case G-subusage Again, similar to the previous inversion lemmas and using the fact that $\leq$ is transitive in point v). Note that we cannot state point v) as $r \cdot \delta_1 + \delta_2 = \delta$.

Case G-convert By inductive hypothesis the fact that "$\equiv$" is transitive.   □

Lemma A.6 (Coproduct Inversion). *If $\delta \odot \Delta \vdash_G$ inl $t : T$, then $\delta_0 \odot \Delta \vdash_G T \equiv X_1 \boxplus X_2 :$ Type and $\delta_0 \odot \Delta \vdash_G t : X_1$ for appropriate X_1, X_2 and δ_0. A symmetrical statement holds for* inr t.

Proof. By induction on the derivation of $\delta \odot \Delta \vdash_G$ inl $t : T$.

Case G-coproductInl There is nothing to prove in this case.

Case G-weak In this case we have $\Delta = \Delta', y : Y$ and $\delta = \delta', 0$ and $\delta' \odot \Delta' \vdash_G$ inl $t : T$. By the inductive hypothesis we have $\delta' \odot \Delta' \vdash_G t : X_1$ and $\delta'_0 \odot \Delta' \vdash_G T \equiv X_1 \boxplus X_2 :$ Type. Weakening these by Y yields the desired result.

Case G-subusage In this case we have $\delta' \odot \Delta \vdash_G$ inl $t : T$ for some δ' with $\delta' \leq \delta$. By the inductive hypothesis we get $\delta_0 \odot \Delta \vdash_G T \equiv X_1 \boxplus X_2 :$ Type and $\delta' \odot \Delta \vdash_G t : X_1$. Applying subusage to the latter judgment yields $\delta \odot \Delta \vdash_G t : X_1$.

Case G-convert Follows from the inductive hypothesis and transitivity of "$\equiv$".   □

Theorem 2.14 (Subject Reduction).   i) *If $\delta \odot \Delta \vdash_G t : X$ and $t \rightsquigarrow t'$, then $\delta \odot \Delta \vdash_G t' : X$.*
  ii) *If $\delta \odot \Delta; \Gamma \vdash_M l : A$ and $l \rightsquigarrow l'$, then $\delta \odot \Delta; \Gamma \vdash_M l' : A$.*

Proof. By induction on the derivation of $\delta \odot \Delta \vdash_G t : X$. We will omit those rules that cannot produce a term which can be reduced.

Case G-subusage

$$\frac{\delta \odot \Delta \vdash_G t : X \qquad \delta \leq \delta'}{\delta' \odot \Delta \vdash_G t : X}$$

By the inductive hypothesis we have $\delta \odot \Delta \vdash_G t' : X$ and by subusage we conclude $\delta' \odot \Delta \vdash_G t' : X$.

Case G-weak

$$\frac{\delta \odot \Delta \vdash_G t : X \qquad \delta_0 \odot \Delta \vdash_G Y : \mathsf{Type} \qquad x \notin \mathrm{dom}\ \Delta}{\delta, 0 \odot \Delta, y : Y \vdash_G t : X}$$

By induction we have $\delta \odot \Delta \vdash_G t' : X$. Weakening this judgment by Y produces the desired result.

Case G-convert

$$\frac{\begin{array}{c}\delta \odot \Delta \vdash_G t : X \\ \delta_0 \odot \Delta \vdash_G Y : \mathsf{Type} \qquad \delta'_0 \odot \Delta \vdash_G X \equiv Y : \mathsf{Type}\end{array}}{\delta \odot \Delta \vdash_G t : Y}$$

We get $\delta \odot \Delta \vdash_G t' : X$ by induction. Applying conversion yields $\delta \odot \Delta \vdash_G t' : Y$, as desired.

Case G-unitElim The term constructed by this rule can only be reduced if it is of the form

$$\frac{\delta_0, r_0 \odot \Delta, x : \mathbf{J} \vdash_G X : \mathsf{Type}}{\delta' \odot \Delta \vdash_G \mathbf{j} : \mathbf{J} \qquad \delta \odot \Delta \vdash_G t : [\mathbf{j}/x]X}{\delta + \delta' \odot \Delta \vdash_G \mathsf{let}\ \mathbf{j} = \mathbf{j}\ \mathsf{in}\ t : [\mathbf{j}/x]X}$$

There is only one reduction that can be applied here, namely let $\mathbf{j} = \mathbf{j}$ in $t \rightsquigarrow t$, so we want to prove $\delta + \delta' \odot \Delta \vdash_G t : [\mathbf{j}/x]X$. By Lemma A.4 we have $\vec{0} \leq \delta'$. By assumption we have $\delta \odot \Delta \vdash_G t : [\mathbf{j}/x]X$. By subusaging we now obtain $\delta + \delta' \odot \Delta \vdash_G t : [\mathbf{j}/x]X$.

Case G-gradedPairElim Again, this can only be reduced if the constructed term looks as follows:

$$\frac{\begin{array}{c}\delta_0, r_0 \odot \Delta, z : (x :^r X) \boxtimes Y \vdash_G W : \mathsf{Type} \\ \delta_1 \odot \Delta \vdash_G (t_1, t_2) : (x :^r X) \boxtimes Y \\ \delta_2, r \cdot q, q \odot \Delta, x : X, y : Y \vdash_G t : [(x, y)/z]W\end{array}}{q \cdot \delta_1 + \delta_2 \odot \Delta \vdash_G \mathsf{let}\ (x, y) = (t_1, t_2)\ \mathsf{in}\ t : [(t_1, t_2)/z]W}$$

We want to prove

$$q \cdot \delta_1 + \delta_2 \odot \Delta \vdash_G [t_1/x][t_2/y]t : [(t_1, t_2)/z]W.$$

Consider the second of the assumptions in the above rule. By Lemma A.5 we see that there are δ'_1, δ''_1 such that $\delta'_1 \odot \Delta \vdash_G t_1 : X$, $\delta''_1 \odot \Delta \vdash_G t_2 : [t_1/x]Y$ and $r \cdot \delta'_1 + \delta''_1 = \delta_1$. Weakening the latter of the two judgments produces $\delta''_1, 0 \odot \Delta, x' : X \vdash_G t_2 : [t_1/x]X$. Now applying substitution to the third assumption twice yields

$$\delta_2, r \cdot q, q \odot \Delta, x : X, y : Y \vdash_G t : [(x, y)/z]W$$

substitute t_2 for y

$$\delta_2 + q \cdot \delta''_1, r \cdot q \odot \Delta, x : X \vdash_G [t_2/y]t : [(x, t_2)/z]W$$

substitute t_1 for x

$$\delta_2 + q \cdot \delta''_1 + r \cdot q \cdot \delta'_1 \odot \Delta \vdash_G [t_1/x][t_2/y]t : [(t_1, t_2)/z]W$$

So, we are done.

Case G-coproductElim In this case there are two reductions possible, one where the constructed term looks thus

$$\frac{\begin{array}{c}1 \leq q \\ \delta_0, r_0 \odot \Delta, y : X_1 \boxplus X_2 \vdash_G Y : \mathsf{Type} \\ \delta_1 \odot \Delta \vdash_G \mathsf{inl}\ t : X_1 \boxplus X_2 \\ \delta_2, q \odot \Delta \vdash_G s_1 : (x :^q X_1) \rightarrow [\mathsf{inl}\ x/y]Y \\ \delta_2, q \odot \Delta \vdash_G s_2 : (x :^q X_2) \rightarrow [\mathsf{inr}\ x/y]Y\end{array}}{q \cdot \delta_1 + \delta_2 \odot \Delta \vdash_G \mathsf{case}_q\ \mathsf{inl}\ t\ \mathsf{of}\ s_1; s_2 : [\mathsf{inl}\ t/y]Y}$$

and a symmetrical version for inr instead of inl. We will only treat the inl case. Want to show

$$q \cdot \delta_1 + \delta_2 \odot \Delta \vdash_G s_1\ t : [\mathsf{inl}\ t/y]Y$$

By Lemma A.6 we get $\delta_1 \odot \Delta \vdash_G t : X_1$. Then, by G-app we get $q \cdot \delta_1 + \delta_2 \odot \Delta \vdash_G s_1\ t : [t/x][\mathsf{inl}\ x/y]Y$, and since $[t/x][\mathsf{inl}\ x/y]Y$ is equal to $[\mathsf{inl}\ t/y]Y$ we are done.

Case G-app In this case there are two possible reductions. The first is by the reduction rule gRED-lambda. In this case the constructed term must look as follows:

$$\frac{\begin{array}{c}\delta_0, r_0 \odot \Delta, x : X \vdash_G Y : \mathsf{Type} \\ \delta_1 \odot \Delta \vdash_G (\lambda x.t') : (x :^r X) \rightarrow Y \\ \delta_2 \odot \Delta \vdash_G t : X\end{array}}{\delta_1 + r \cdot \delta_2 \odot \Delta \vdash_G (\lambda x.t')\ t : [t/x]Y}$$

By Lemma A.3 we get $\delta_2, r \odot \Delta, x : X \vdash_G t' : Y$. By substitution we obtain $\delta_1 + r \cdot \delta_2 \odot \Delta \vdash_G [t/x]t' : [t/x]Y$ as desired.

The second possible reduction is by the rule GRED-APPL. The application of the rule G-APP is thus:

$$\frac{\begin{array}{c} \delta_0, r_0 \odot \Delta, x : X \vdash_G Y : \mathsf{Type} \\ \delta_1 \odot \Delta \vdash_G t_1 : (x :^r X) \to Y \\ \delta_2 \odot \Delta \vdash_G t : X \end{array}}{\delta_1 + r \cdot \delta_2 \odot \Delta \vdash_G t_1\ t : [t/x]Y}$$

We have by assumption that $t_1 \leadsto t_2$ for some t_2. By the inductive hypothesis we know that $\delta_1 \odot \Delta \vdash_G t_2 : (x :^r X) \to Y$ and by applying G-APP, we get $\delta_1 + r \cdot \delta_2 \odot \Delta \vdash_G t_2\ t : [t/x]Y$ as desired.

The cases for the mixed reduction are similar and require similar inversion lemmas as the ones stated above. We omit these cases. □

B   Metatheory of GLAD

Theorem 3.3 (GLAD substitution). *The following hold by mutual induction:*

i) (Contexts) If $\delta, r, \delta' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x: A, \Gamma' \vdash \mathsf{ctx}$ and $\delta_0 \mid \mathcal{M} \odot \Gamma \vdash_{\mathfrak{m}_0} a_0 : A$ then

$$\delta + r \cdot \delta_0, \delta' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma' \vdash \mathsf{ctx}$$

ii) (Terms) If $\delta, r, \delta' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x: A, \Gamma' \vdash_{\mathfrak{m}} b : B$ and $\delta_0 \mid \mathcal{M} \odot \Gamma \vdash_{\mathfrak{m}_0} a_0 : A$, then

$$\delta + r \cdot \delta_0, \delta' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma' \vdash_{\mathfrak{m}} [a_0/x]b : [a_0/x]B$$

Proof. The proof is by induction on the derivation of the first sequent in each point.

Case GLAD-VAR We need to match the judgment

$$\vec{0}, 1 \mid \mathcal{M}, \mathfrak{m} \odot \Gamma, y: B \vdash_{\mathfrak{m}} y: B$$

with the pattern

$$\delta, r, \delta' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x: A, \Gamma' \vdash_{\mathfrak{m}} b: B$$

There are two cases. In the first, we have $\delta' = \emptyset$, and therefore $A = B$. i.e. the judgment we are performing induction on was derived as

$$\frac{x \notin \mathrm{dom}\,\Gamma \qquad \delta \mid \mathcal{M} \odot \Gamma \vdash_{\mathfrak{m}_0} A: \mathsf{Type}}{\vec{0}, 1 \mid \mathcal{M}, \mathfrak{m}_0 \odot \Gamma, x: A \vdash_{\mathfrak{m}_0} x: A}$$

and we need to prove

$$\vec{0} + 1 \cdot \delta_0 \mid \mathcal{M} \odot \Gamma \vdash_{\mathfrak{m}_0} [a_0/x]x: [a_0/x]A$$

But this simplifies to the assumption $\delta_0 \mid \mathcal{M} \odot \Gamma \vdash_{\mathfrak{m}_0} a_0: A$ once we observe that the variable x is not free in A, and hence $[a_0/x]A = A$.

In the second case we have $\delta' \neq \emptyset$. Then our derivation has the form

$$\frac{\begin{array}{c} y \notin \mathrm{dom}(\Gamma, x: A, \Gamma') \\ \delta, r, \delta' \mid \mathcal{M}, \mathfrak{m}, \mathcal{M}' \odot \Gamma, x: A, \Gamma' \vdash_{\mathfrak{n}} B: \mathsf{Type} \end{array}}{\vec{0}, 0, \vec{0}, 1 \mid \mathcal{M}, \mathfrak{m}, \mathcal{M}', \mathfrak{n} \odot \Gamma, x: A, \Gamma', y: B \vdash_{\mathfrak{n}} y: B}$$

and we need to prove that

$$\vec{0}, \vec{0}, 1 \mid \mathcal{M}, \mathcal{M}', \mathfrak{n} \odot \Gamma, [a_0/x]\Gamma', y: [a_0/x]B \vdash_{\mathfrak{n}} y: [a_0/x]B$$

which follows from the inductive hypothesis

$$\delta + r \cdot \delta_0, \delta' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma' \vdash_{\mathfrak{n}} [a_0/x]B: \mathsf{Type}$$

and an application of the rule GLAD-VAR.

Case GLAD-WEAK We need to match the judgment

$$\delta, 0 \mid \mathcal{M}, \mathfrak{l} \odot \Gamma, y: C \vdash_{\mathfrak{m}} b: B$$

with the pattern

$$\delta, r, \delta' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x: A, \Gamma' \vdash_{\mathfrak{m}} b: B$$

Two cases are possible. In the first, we have $\delta' = \emptyset$ and therefore and therefore $A = C$ our derivation has the form

$$\frac{\begin{array}{ccc} \mathsf{Weak}(\mathfrak{l}) & \mathfrak{m} \leq \mathfrak{m}_0 & x \notin \mathrm{dom}\,\Gamma \\ \delta \mid \mathcal{M} \odot \Gamma \vdash_{\mathfrak{m}} b: B & \delta_1 \mid \mathcal{M} \odot \Gamma \vdash_{\mathfrak{m}_0} A: \mathsf{Type} \end{array}}{\delta, 0 \mid \mathcal{M}, \mathfrak{m}_0 \odot \Gamma, x: A \vdash_{\mathfrak{m}} b: B}$$

and we need to prove

$$\delta + 0 \cdot \delta_0 \mid \mathcal{M} \odot \Gamma \vdash_{\mathfrak{m}} [a_0/x]b: [a_0/x]B$$

which is an assumption since x is not a free variable in b or B.

In the second case we have $\delta' \neq \emptyset$. In this case, our derivation has the form

$$\frac{\begin{array}{ccc} \mathsf{Weak}(\mathfrak{l}) & \mathfrak{m} \leq \mathfrak{l} & y \notin \mathrm{dom}(\Gamma, x: A, \Gamma') \\ \delta, r, \delta' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x: A, \Gamma' \vdash_{\mathfrak{m}} b: B \\ \delta_1, r_1, \delta_1' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x: A, \Gamma' \vdash_{\mathfrak{l}} C: \mathsf{Type} \end{array}}{\delta, r, \delta', 0 \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}', \mathfrak{l} \odot \Gamma, x: A, \Gamma', y: C \vdash_{\mathfrak{m}} b: B}$$

and we need to prove

$$\delta + r \cdot \delta_0, \delta', 0 \mid \mathcal{M}, \mathcal{M}', \mathfrak{l}$$
$$\odot \Gamma, [a_0/x]\Gamma', y: [a_0/x]C \vdash_{\mathfrak{m}} [a_0/x]b : [a_0/x]B$$

This follows from the inductive hypotheses

$$\delta + r \cdot \delta_0, \delta' \mid \mathcal{M}, \mathcal{M}'$$
$$\odot \Gamma, [a_0/x]\Gamma' \vdash_{\mathfrak{m}} [a_0/x]b : [a_0/x]B$$
and
$$\delta_1 + r \cdot \delta_0, \delta_1' \mid \mathcal{M}, \mathcal{M}'$$
$$\odot \Gamma, [a_0/x]\Gamma' \vdash_{\mathfrak{l}} [a_0/x]C : \mathsf{Type}$$

by applying the rule GLAD-WEAK.

Cases GLAD-UNIT, GLAD-UNITINTRO Trivial.

Case GLAD-UNITELIM Our derivation has the form

$$\frac{\begin{array}{c} \delta_1, r_1, \delta_1' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x: A, \Gamma' \vdash_{\mathfrak{m}_1} c: \mathfrak{l}_{\mathfrak{m}_1} \\ \delta_3, r_3, \delta_3', q \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}', \mathfrak{m}_1 \odot \Gamma, x: A, \Gamma', y: \mathfrak{l}_{\mathfrak{m}_1} \vdash_{\mathfrak{m}_2} B: \mathsf{Type} \\ \delta_2, r_2, \delta_2' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x: A, \Gamma' \vdash_{\mathfrak{m}_2} b: [\star_{\mathfrak{m}_1}/y]B \end{array}}{\begin{array}{c} \delta_1 + \delta_2, r_1 + r_2, \delta_1' + \delta_2' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x: A, \Gamma' \\ \vdash_{\mathfrak{m}_1} \mathsf{let}\ \star_{\mathfrak{m}_1} = c\ \mathsf{in}\ b : [c/y]B \end{array}}$$

and we need to prove

$$\delta_1 + \delta_2 + (r_1 + r_2) \cdot \delta_0, \delta_1' + \delta_2' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma'$$
$$\vdash_{\mathfrak{m}_1} [a_0/x](\mathsf{let}\ \star_{\mathfrak{m}_1} = c\ \mathsf{in}\ b) : [a_0/x][c/y]B$$

We have the inductive hypotheses

$$\delta_1 + r_1 \cdot \delta_0, \delta_1' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma' \vdash_{\mathfrak{m}_1} [a_0/x]c : [a_0/x]C$$

$$\delta_2 + r_2 \cdot \delta_0, \delta_2' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma'$$
$$\vdash_{\mathfrak{m}_2} [a_0/x]b : [a_0/x][\star_{\mathfrak{m}_1}/y]B$$

$$\delta_3 + r_3 \cdot \delta_0, \delta_3' \mid \mathcal{M}, \mathcal{M}', \mathfrak{m}_1 \odot \Gamma, [a_0/x]\Gamma', y \colon \mathbf{I}_{\mathfrak{m}_1}$$
$$\vdash_{\mathfrak{m}_2} [a_0/x]B : \mathsf{Type}$$

In the second hypothesis, the type $[a_0/x][\star_{\mathfrak{m}_1}/y]B$ is equal to $[\star_{\mathfrak{m}_1}/y][a_0/x]B$ since y is not free in a_0 and $\star_{\mathfrak{m}_1}$ has no free variables. Furthermore, the types $[a_0/x][c/y]B$ and $[[a_0/x]c/y][a_0/x]B$ are equal. Because of this, we may apply the rule GLAD-UNITELIM to the three inductive hypotheses to obtain the desired judgment.

Case GLAD-FUNCTION Our derivation has the form

$$\frac{\delta, r, \delta', q_0 \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}', \mathfrak{m} \odot \Gamma, x \colon A, \Gamma', y \colon B \vdash_{\mathfrak{n}} C : \mathsf{Type}}{\delta, r, \delta', q_0 \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x \colon A, \Gamma' \vdash_{\mathfrak{n}} (y \colon^{q : \mathfrak{m}} B) \multimap C : \mathsf{Type}}$$

and we need to prove

$$\delta + r \cdot \delta_0, \delta' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma'$$
$$\vdash_{\mathfrak{n}} [a_0/x]((y \colon^{q : \mathfrak{m}} B) \multimap C) : \mathsf{Type}$$

Since the types $[a_0/x]((y \colon^{q : \mathfrak{m}} B) \multimap C)$ and $(y \colon^{q : \mathfrak{m}} [a_0/x]B) \multimap [a_0/x]C$ are equal, this follows from the inductive hypothesis

$$\delta + r \cdot \delta_0, \delta', q_0 \mid \mathcal{M}, \mathcal{M}', \mathfrak{m}$$
$$\odot \Gamma, [a_0/x]\Gamma', y \colon [a_0/x]B \vdash_n [a_0/x]C : \mathsf{Type}$$

Case GLAD-LAMBDA Our derivation has the form

$$\frac{\delta, r, \delta', q \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}', \mathfrak{m} \odot \Gamma, x \colon A, \Gamma', y \colon B \vdash_{\mathfrak{n}} c : C}{\delta, r, \delta' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x \colon A, \Gamma' \vdash_{\mathfrak{n}} \lambda y.c : (y \colon^{q : \mathfrak{m}} B) \multimap C}$$

We need to prove

$$\delta + r \cdot \delta_0, \delta' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma' \vdash_n$$
$$[a_0/x](\lambda y.c) : (y \colon^{q : \mathfrak{m}} [a_0/x]B) \multimap [a_0/x]C$$

But this follows immediately from the inductive hypothesis

$$\delta + r \cdot \delta_0, \delta', q \mid \mathcal{M}, \mathcal{M}', \mathfrak{m} \odot \Gamma, [a_0/x]\Gamma', y \colon [a_0/x]B \vdash_n$$
$$[a_0/x]c : [a_0/x]C$$

and the fact that the terms $[a_0/x](\lambda y.c)$ and $\lambda y.[a_0/x]c$ are equal.

Case GLAD-APP The derivation has the form

$$\frac{\begin{array}{c}\delta_1, r_1, \delta_1' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x \colon A, \Gamma' \vdash_{\mathfrak{m}} b : B \\ \delta_2, r_2, \delta_2' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x \colon A, \Gamma' \vdash_{\mathfrak{n}} c : (y \colon^{q : \mathfrak{m}} B) \multimap C\end{array}}{\begin{array}{c}(\delta_1, r_1, \delta_1') + q \cdot (\delta_2, r_2, \delta_2') \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x \colon A, \Gamma' \\ \vdash_{\mathfrak{n}} c \, b : [b/y]C.\end{array}}$$

and we need to show that

$$\delta_1 + q \cdot \delta_2 + (r_1 + q \cdot r_2) \cdot \delta_0, \delta_1' + q \cdot \delta_2' \mid \mathcal{M}, \mathcal{M}'$$
$$\odot \Gamma, [a_0/x]\Gamma' \vdash_{\mathfrak{n}} [a_0/x](c \, b) : [a_0/x][b/y]C$$

The inductive hypotheses are

$$\delta_1 + r_1 \cdot \delta_0, \delta_1' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma' \vdash_{\mathfrak{m}} [a_0/x]b : [a_0/x]B$$

$$\delta_2 + r_2 \cdot \delta_0, \delta_2' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma'$$
$$\vdash_{\mathfrak{n}} [a_0/x]c : (y \colon^{q : \mathfrak{m}} [a_0/x]B) \multimap [a_0/x]C$$

The types $[a_0/x][b/y]C$ and $[[a_0/x]b/y][a_0/x]C$ are equal. Therefore we may apply the GLAD-APP rule to the inductive hypotheses to obtain

$$\delta_1 + \delta_2 + (r_1 + q \cdot r_2) \cdot \delta_0, \delta_1' + q \cdot \delta_2' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma'$$
$$\vdash_{\mathfrak{m}} [a_0/x]c \, [a_0/x]b : [[a_0/x]b/y][a_0/x]C$$

This is the desired judgment once we replace the type $[[a_0/x]b/y][a_0/x]C$ by $[a_0/x][b/y]C$ since we observed this equality before.

Case GLAD-TENSORINTRO Our derivation has the form

$$\frac{\begin{array}{c}\delta_3, r_3, \delta_3' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x \colon A, \Gamma', y \colon B \vdash_{\mathfrak{n}} C : \mathsf{Type} \\ \delta_1, r_1, \delta_1' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x \colon A, \Gamma' \vdash_{\mathfrak{m}} b : B \\ \delta_2, r_2, \delta_2' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x \colon A, \Gamma' \vdash_{\mathfrak{n}} c : [b/y]C\end{array}}{\begin{array}{c}q \cdot (\delta_1, r_1, r_1') + (\delta_2, r_2, \delta_2') \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \\ \odot \Gamma, x \colon A, \Gamma' \vdash_{\mathfrak{n}} (b, c) : (y \colon^{q : \mathfrak{m}} B) \otimes C\end{array}}$$

and we need to prove

$$q \cdot \delta_1 + \delta_2 + (q \cdot r_1 + r_2) \cdot \delta_0, q \cdot \delta_1' + \delta_2' \mid \mathcal{M}, \mathcal{M}'$$
$$\odot \Gamma, [a_0/x]\Gamma' \vdash_{\mathfrak{n}} [a_0/x](b, c) : [a_0/x]((y \colon^{q : \mathfrak{m}} B) \otimes C)$$

We have the inductive hypotheses

$$\delta_3 + r_3 \cdot \delta_0, \delta_3' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma' \vdash_{\mathfrak{n}} [a_0/x]C : \mathsf{Type}$$

$$\delta_1 + r_1 \cdot \delta_0, \delta_1' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma' \vdash_{\mathfrak{n}} [a_0/x]b : [a_0/x]B$$

$$\delta_2 + r_2 \cdot \delta_0, \delta_2' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma'$$
$$\vdash_{\mathfrak{n}} [a_0/x]c : [a_0/x][b/y]C$$

The types $[a_0/x][b/y]C$ and $[[a_0/x]b/y][a_0/x]C$ are equal. We therefore may apply GLAD-TENSORINTRO to the three inductive hypotheses and obtain

$$q \cdot (\delta_1 + r_1 \cdot \delta_0, \delta_1') + (\delta_2 + r_2 \cdot \delta_0, \delta_2') \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma'$$
$$\vdash_{\mathfrak{n}} ([a_0/x]b, [a_0/x]c) : (y \colon^{q : \mathfrak{m}} [a_0/x]B) \otimes [a_0/x]C$$

which is the desired judgment.

Case GLAD-TENSORELIM Our derivationhas the form

$$\frac{\begin{array}{c}\delta_3, r_3, \delta_3', r_3' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}', \mathfrak{m}_2 \\ \odot \Gamma, x \colon A, \Gamma', z \colon (y \colon^{q : \mathfrak{m}_1} B_1) \otimes B_2 \vdash_{\mathrm{I}} C : \mathsf{Type} \\ \\ \delta_1, r_1, \delta_1' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x \colon A, \Gamma' \vdash_{\mathfrak{m}_2} b : (y \colon^{q : \mathfrak{m}_1} B_1) \otimes B_2 \\ \\ \delta_2, r_2, \delta_2', s \cdot q, s \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}', \mathfrak{m}_1, \mathfrak{m}_2 \\ \odot \Gamma, x \colon A, \Gamma', y_1 \colon B_1, y_2 \colon B_2 \vdash_{\mathrm{I}} c : [(y_1, y_2)/z]C\end{array}}{\begin{array}{c}s \cdot (\delta_1, r_1, \delta_1') + (\delta_2, r_2, \delta_2') \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x \colon A, \Gamma' \\ \vdash_{\mathrm{I}} \mathsf{let}\ (y_1, y_2) = b\ \mathsf{in}\ c : [b/z]C\end{array}}$$

We need to prove

$$s \cdot \delta_1 + \delta_2 + (s \cdot r_1 + r_2) \cdot \delta_0, s \cdot \delta_1' + \delta_2' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma'$$
$$\vdash_{\mathrm{I}} [a_0/x](\mathsf{let}\ (y_1, y_2) = b\ \mathsf{in}\ c) : [a_0/x][b/z]C$$

The inductive hypotheses are

$$\delta_3 + r_3 \cdot \delta_0, \delta_3', r_3' \mid \mathcal{M}, \mathcal{M}', \mathfrak{m}_2$$
$$\odot \Gamma, [a_0/x]\Gamma', z \colon (y :^{q \cdot \mathfrak{m}_1} [a_0/z]B_1) \otimes [a_0/x]B_2$$
$$\vdash_I [a_0/x]C \colon \mathsf{Type}$$
$$\delta_1 + r_1 \cdot \delta_0, \delta_1' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma'$$
$$\vdash_{\mathfrak{m}2} [a_0/x]b \colon (y :^{q \cdot \mathfrak{m}_1} [a_0/x]B_1) \otimes [a_0/x]B_2$$
$$\delta_2 + r_2 \cdot \delta_0, \delta_2', s \cdot q, s \mid \mathcal{M}, \mathcal{M}', \mathfrak{m}_1, \mathfrak{m}_2$$
$$\odot \Gamma, [a_0/x]\Gamma', y_1 \colon [a_0/x]B_1, y_2 \colon [a_0/x]B_2$$
$$\vdash_I [a_0/x]c \colon [a_0/x][(y_1, y_2)/z]C$$

The types $[a_0/x][(y_1, y_2)/z]C$ and $[(y_1, y_2)/z][a_0/x]C$ are equal. We may therefore apply GLAD-TENSORELIM to the three inductive hypotheses and obtain

$$s \cdot (\delta_1 + r_1 \cdot \delta_0, \delta_1') + (\delta_2 + r_2 \cdot \delta_0, \delta_2') \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma'$$
$$\vdash_I \mathsf{let}\ (y_1, y_2) = [a_0/x]b\ \mathsf{in}\ [a_0/x]c \colon [[a_0/x]b/z][a_0/x]C$$

which is the desired judgment once we observe that the types $[[a_0/x]b/z][a_0/x]C$ and $[a_0/x][b/z]C$ are equal.
Case GLAD-RAISE Our derivation has the form

$$\frac{\mathfrak{m}_1 \leq \mathfrak{m}_2 \qquad \mathfrak{m}_2 \leq (\mathcal{M}, \mathfrak{m}_0, \mathcal{M}') \qquad \delta, r, \delta' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x \colon A, \Gamma' \vdash_{\mathfrak{m}_1} b \colon B}{\delta, r, \delta' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x \colon A, \Gamma' \vdash_{\mathfrak{m}_2} \uparrow_{\mathfrak{m}_1}^{\mathfrak{m}_2} b \colon\ \uparrow_{\mathfrak{m}_1}^{\mathfrak{m}_2} B}$$

We need to show

$$\delta + r \cdot \delta_0, \delta' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma'$$
$$\vdash_{\mathfrak{m}_2} \uparrow_{\mathfrak{m}_1}^{\mathfrak{m}_2} [a_0/x]b \colon \uparrow_{\mathfrak{m}_1}^{\mathfrak{m}_2} [a_0/x]B$$

The inductive hypothesis is

$$\delta + r \cdot \delta_0, \delta' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma'$$
$$\vdash_{\mathfrak{m}_1} [a_0/x]b \colon [a_0/x]B$$

Furthermore, $\mathfrak{m}_2 \leq (\mathcal{M}, \mathcal{M}')$ follows from $\mathfrak{m}_2 \leq (\mathcal{M}, \mathfrak{m}_0, \mathcal{M}')$. Therefore we can apply GLAD-RAISE to obtain the desired result.
Case GLAD-UNRAISE In this case, our derivation has the form

$$\frac{\delta, r, \delta' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x \colon A, \Gamma' \vdash_{\mathfrak{m}_2} b \colon\ \uparrow_{\mathfrak{m}_1}^{\mathfrak{m}_2} B}{\delta, r, \delta' \mid \mathcal{M}, \mathfrak{m}_0, \mathcal{M}' \odot \Gamma, x \colon A, \Gamma' \vdash_{\mathfrak{m}_1} \downarrow_{\mathfrak{m}_1}^{\mathfrak{m}_2} b \colon B}$$

We need to show

$$\delta + r \cdot \delta_0, \delta' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma'$$
$$\vdash_{\mathfrak{m}_1} \downarrow_{\mathfrak{m}_1}^{\mathfrak{m}_2} [a_0/x]b \colon [a_0/x]B$$

which follows immediately from the inductive hypothesis

$$\delta + r \cdot \delta_0, \delta' \mid \mathcal{M}, \mathcal{M}' \odot \Gamma, [a_0/x]\Gamma'$$
$$\vdash_{\mathfrak{m}_2} [a_0/x]b \colon \uparrow_{\mathfrak{m}_1}^{\mathfrak{m}_2} [a_0/x]B$$

by applying GLAD-UNRAISE.

The remaining cases are similar and we omit them. □