Graded Modal Dependent Type Theory

Benjamin Moon[1(✉)], Harley Eades III[2], and Dominic Orchard[1]

[1] University of Kent, Canterbury, UK
{bgm4,d.a.orchard}@kent.ac.uk
[2] Augusta University, Augusta, USA
harley.eades@gmail.com

Abstract. Graded type theories are an emerging paradigm for augmenting the reasoning power of types with parameterizable, fine-grained analyses of program properties. There have been many such theories in recent years which equip a type theory with quantitative dataflow tracking, usually via a semiring-like structure which provides analysis on variables (often called 'quantitative' or 'coeffect' theories). We present Graded Modal Dependent Type Theory (Grtt for short), which equips a dependent type theory with a general, parameterizable analysis of the flow of data, both in and between computational terms and types. In this theory, it is possible to study, restrict, and reason about data use in programs and types, enabling, for example, parametric quantifiers and linearity to be captured in a dependent setting. We propose Grtt, study its metatheory, and explore various case studies of its use in reasoning about programs and studying other type theories. We have implemented the theory and highlight the interesting details, including showing an application of grading to optimising the type checking procedure itself.

1 Introduction

The difference between simply-typed, polymorphically-typed, and dependently-typed languages can be characterised by the *dataflow* permitted by each type theory. In each, dataflow can be enacted by *substituting* a term for occurrences of a variable in another term, the scope of which is delineated by a binder. In the simply-typed λ-calculus, data can only flow in 'computational' terms; computations and types are separate syntactic categories, with variables, bindings (λ), and substitution—and thus dataflow—only at the computational level. In contrast, polymorphic calculi like System F [26,51] permit dataflow within types, via type quantification ($\forall$), and a limited form of dataflow from computations to types, via type abstraction (Λ) and type application. Dependently-typed calculi (e.g., [14,40,41,42]) break down the barrier between computations and types further: variables are bound simultaneously in types and computations, such that data can flow both to computations and types via dependent functions (Π) and application. This pervasive dataflow enables the Curry-Howard correspondence to be leveraged for program reasoning and theorem proving [58]. However, unrestricted dataflow between computations and types can impede reasoning and can interact poorly with other type theoretic ideas.

Firstly, System F allows *parametric reasoning* and notions of representation independence [52,56], but this is lost in general in dependently-typed languages when quantifying over higher-kinded types [44] (rather than just 'small' types [7,36]). Furthermore, unrestricted dataflow impedes efficient compilation as compilers do not know, from the types alone, where a term is actually needed. Additional static analyses are needed to recover dataflow information for optimisation and reasoning. For example, a term shown to be used only for type checking (not flowing to the computational 'run time' level) can be erased [9]. Thus, dependent theories do not expose the distinction between proof relevant and irrelevant terms, requiring extensions to capture irrelevance [4,49,50]. Whilst unrestricted dataflow between computations and terms has its benefits, the permissive nature of dependent types can hide useful information. This permissiveness also interacts poorly with other type theories which seek to deliberately restrict dataflow, notably *linear types*.

Linear types allow data to be treated as a 'resource' which must be consumed exactly once: linearly-typed values are restricted to linear dataflow [27,57,59]. Reasoning about resourceful data has been exploited by several languages, e.g., ATS [53], Alms [55], Clean [18], Granule [45], and Linear Haskell [8]. However, linear dataflow is rare in a dependently-typed setting. Consider typing the body of the polymorphic identity function in Martin-Löf type theory:

$$a : \mathsf{Type}, x : a \vdash x : a$$

This judgment uses a twice (typing x in the context and the subject of the judgment) and x once in the term but not at all in the type. There have been various attempts to meaningfully reconcile linear and dependent types [12,15,37,39] usually by keeping them separate, allowing types to depend only on non-linear variables. All such theories cannot distinguish variables used for computation from those used purely for type formation, which could be erased at runtime.

Recent work by McBride [43], refined by Atkey [6], generalises ideas from 'coeffect analyses' (variable usage analyses, like that of Petricek et al. [48]) to a dependently-typed setting to reconcile the ubiquitous flow of data in dependent types with the restricted dataflow of linearity. This approach, called Quantitative Type Theory (QTT), types the above example as:

$$a \overset{0}{:} \mathsf{Type}, x \overset{1}{:} a \vdash x \overset{1}{:} a$$

The annotation 0 on a explains that we can use a to form a type, but we cannot, or do not, use it at the term level, thus it can be erased at runtime. The cornerstone of QTT's approach is that dataflow of a term to the type level counts as 0 use, so arbitrary type-level use is allowed whilst still permitting quantitative analysis of computation-level dataflow. Whilst this gives a useful way to relate linear and dependent types, it cannot however reason about dataflow at the type-level (all type-level usage counts as 0). Thus, for example, QTT cannot express that a variable is used just computationally but not at all in types.

In an extended abstract, Abel proposes a generalisation of QTT to track variable use in both types and computations [2], suggesting that tracking in types enables type checking optimisations and increased expressivity. We develop a core dependent type theory along the same lines, using the paradigm of *grading*: graded systems augment types with additional information, capturing the structure of programs [23,45]. We therefore name our approach *Graded Modal Dependent Type Theory* (GrTT for short). Our type theory is parameterised by a semiring which, like other coeffect and quantitative approaches [3,6,10,25,43,48,60], describes dataflow through a program, but in *both types and computations equally*, remedying QTT's inability to track type-level use. We extend Abel's initial idea by presenting a rich language, including dependent tensors, a complete metatheory, and a *graded modality* which aids the practical use of this approach (e.g., enabling functions to use components of data non-uniformly). The result is a calculus which extends the power of existing non-dependent graded languages, like Granule [45], to a dependent setting.

We begin with the definition of GrTT in Section 2, before demonstrating the power of GrTT through case studies in Section 3, where we show how to use grading to restrict GrTT terms to simply-typed reasoning, parametric reasoning (regaining universal quantification smoothly within a dependent theory), existential types, and linear types. The calculus can be instantiated to different kinds of dataflow reasoning: we show an example application to information-flow security. We then show the metatheory of GrTT in Section 4: admissibility of graded structural rules, substitution, type preservation, and strong normalisation.

We implemented a prototype language based on GrTT called **Gerty**.[3] We briefly mention its syntax in Section 2.5 for use in examples. Later, Section 5 describes how the formal definition of GrTT is implemented as a bidirectional type checking algorithm, interfacing with an SMT solver to solve constraints over grades. Furthermore, Abel conjectured that a quantitative dependent theory could enable usage-based optimisation of type-checking itself [2], which would assist dependently-typed programming at scale. We validate this claim in Section 5 showing a grade-directed optimisation to **Gerty**'s type checker.

Section 6 discusses next steps for increasing the expressive power of GrTT. Full proofs and details are provided in Appendix A.2.

Gerty has some similarity to Granule [45]: both are functional languages with graded types. However, Granule has a linearly typed core and no dependent types (only indexed types), thus has no need for resource tracking at the type level (type indices are not subject to tracking and their syntax is restricted).

2 GrTT: Graded Modal Dependent Type Theory

GrTT augments a standard presentation of dependent type theory with 'grades' (elements of a semiring) which account for how variables are used, i.e., their *dataflow*. Whilst existing work uses grades to describe usage only in computational terms (e.g. [10]), GrTT incorporates additional grades to account for how variables are used in types. We introduce here the syntax and typing, and briefly show the syntax of the implementation. Section 4 describes its metatheory.

[3] https://github.com/granule-project/gerty/releases/tag/esop2021

2.1 Syntax

The syntax of GRTT is that of a standard Martin-Löf type theory, with the addition of a *graded modality* and grade annotations on function and tensor binders. Throughout, s and r range over grades, which are elements of a semiring $(\mathcal{R}, *, 1, +, 0)$. It is instructive to instantiate this semiring to the natural number semiring $(\mathbb{N}, \times, 1, +, 0)$, which captures the exact number of times variables are used. We appeal to this example in descriptions here.

GRTT has a single syntactic sort for computations and types:

$$
\begin{array}{llll}
(\mathit{terms}) & t, A, B, C ::= x & \mid \mathsf{Type}_l & \\
& \mid (x :_{(s,r)} A) \to B & \mid \lambda x.t & \mid t_1\, t_2 \\
& \mid (x :_r A) \otimes B & \mid (t_1, t_2) & \mid \mathsf{let}\,(x, y) = t_1\,\mathsf{in}\,t_2 \\
& \mid \Box_s A & \mid \Box t & \mid \mathsf{let}\,\Box x = t_1\,\mathsf{in}\,t_2 \\
(\mathit{levels}) & l ::= 0 \mid \mathsf{suc}\ l \mid l_1 \sqcup l_2 & &
\end{array}
$$

Terms include variables and a constructor for an inductive hierarchy of universes, annotated by a level l. Dependent function types are annotated with a pair of grades s and r, with s capturing how x is used in the body of the inhabiting function and r capturing how x is used in the codomain B. Dependent tensors have a single grade r, which describes how the first element is used in the typing of the second. The graded modal type operator $\Box_s A$ 'packages' a term and its dependencies so that values of type A can be used with grade s in the future. Graded modal types are introduced via *promotion* $\Box t$ and eliminated via $\mathsf{let}\,\Box x = t_1\,\mathsf{in}\,t_2$. The following sections explain the semantics of each piece of syntax with respect to its typing. We typically use A and B to connote terms used as types.

2.2 Typing Judgments, Contexts, and Grading

Typing judgments are written in either of the following two equivalent forms:

$$
(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A \qquad\qquad \begin{pmatrix} \Delta \\ \sigma_1 \\ \sigma_2 \end{pmatrix} \odot \Gamma \vdash t : A
$$

The 'horizontal' syntax (left) is used most often, with the equivalent 'vertical' form (right) used for clarity in some places. Ignoring the part to the left of $\odot$, typing judgments and their rules are essentially those of Martin-Löf type theory (with the addition of the modality) where Γ ranges over usual dependently-typed typing *contexts*. The left of $\odot$ provides the grading information, where σ and Δ range over *grade vectors* and *context grade vectors* respectively, of the form:

$$
\begin{array}{ccc}
(\mathit{contexts}) & (\mathit{grade\ vectors}) & (\mathit{context\ grade\ vectors}) \\
\Gamma ::= \emptyset \mid \Gamma, x : A & \sigma ::= \emptyset \mid \sigma, s & \Delta ::= \emptyset \mid \Delta, \sigma
\end{array}
$$

A grade vector σ is a vector of semiring elements, and a context vector Δ is a vector of grade vectors. We write $(s_1, \ldots, s_n)$ to denote an n-vector and likewise for context grade vectors. We omit parentheses when this would not cause ambiguity. Throughout, a comma is used to concatenate vectors and disjoint contexts, and to extend vectors with a single grade, grade vector, or typing assumption.

For a judgment $(\Delta \mid \sigma_s \mid \sigma_r) \odot \Gamma \vdash t : A$ the vectors Γ, Δ, σ_s, and σ_r are all of equal size. Given a typing assumption $y : B$ at index i in Γ, the grade $\sigma_s[i] \in \mathcal{R}$ denotes the use of y in t (the *subject* of the judgment), the grade $\sigma_r[i] \in \mathcal{R}$ denotes the use of y in A (the *subject's type*), and $\Delta[i] \in \mathcal{R}^i$ (of size i) describes how assumptions prior to y are used to form y's type, B.

Consider the following example, which types the body of a function that takes two arguments of type a, and returns only the first:

$$\begin{pmatrix} (),(1),(1,0) \\ 0,1,0 \\ 1,0,0 \end{pmatrix} \odot a : \mathsf{Type}_l, x : a, y : a \vdash x : a$$

Let the context grade vector be called Δ. Then, $\Delta[0] = ()$ (empty vector) explains that there are no assumptions that are used to type a in the context, as Type_l is a closed term and the first assumption. $\Delta[1] = (1)$ explains that the first assumption a is used (grade 1) in the typing of x in the context, and $\Delta[2] = (1, 0)$, explains that a is used once in the typing of y in the context, and x is unused in the typing of y. The subject grade vector $\sigma_s = (0, 1, 0)$ explains that a is unused in the subject, x is used once, and y is unused. Finally, the subject type vector $\sigma_r = (1, 0, 0)$ explains that a appears once in the subject's type (which is just a), and x and y are unused in the formation of the subject's type.

To aid reading, recall that standard typing rules typically have the form *context* $\vdash$ *subject* : *subject-type*, the order of which is reflected by $(\Delta \mid \sigma_s \mid \sigma_r) \odot \ldots$ giving the context, subject, and subject-type grading respectively.

Well-formed Contexts The relation $\Delta \odot \Gamma \vdash$ identifies a context Γ as well-formed with respect to context grade vector Δ, defined by the following rules:

$$\frac{}{\emptyset \odot \emptyset \vdash}\ \textsc{wf}\emptyset \qquad \frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l}{\Delta, \sigma \odot \Gamma, x : A \vdash}\ \textsc{wfExt}$$

Unlike typing, well-formedness does not need to include subject and subject-type grade vectors, as it considers only the well-formedness of the assumptions in a context with respect to prior assumptions in the context. The $\textsc{wf}\emptyset$ rule states that the empty context is well-formed with an empty context grade vector as there are no assumptions to account for. The $\textsc{wfExt}$ rule states that given A is a type under the assumptions in Γ, with σ accounting for the usage of Γ variables in A, and Δ accounting for usage within Γ, then we can form the well-formed context $\Gamma, x : A$ by extending Δ with σ to account for the usage of A in forming the context. The notation $\mathbf{0}$ denotes a vector for which each element is the semiring 0. Note that the well-formedness $\Delta \odot \Gamma \vdash$ is inherent from the premise of $\textsc{wfExt}$ due to the following lemma:

Lemma 1 (Typing contexts are well-formed). *If* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$ *then* $\Delta \odot \Gamma \vdash$.

2.3 Typing Rules

We examine the typing rules of GRTT one at a time. The rules are collected in Appendix A.1.

Variables are introduced as follows:

$$\frac{(\Delta_1, \sigma, \Delta_2) \odot \Gamma_1, x : A, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \sigma, \Delta_2 \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \sigma, 0, \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash x : A} \textsc{Var}$$

The premise identifies $\Gamma_1, x : A, \Gamma_2$ as well-formed under the context grade vector $\Delta_1, \sigma, \Delta_2$. By the size condition $|\Delta_1| = |\Gamma_1|$, we are able to identify σ as capturing the usage of the variables Γ_1 in forming A. This information is used in the conclusion, capturing type-level variable usage as $\sigma, 0, \mathbf{0}$, which describes that Γ_1 is used according to σ in the subject's type (A), and that the x and the variables of Γ_2 are used with grade 0. For subject usage, we annotate the first zero vector with a size $|\Delta_1|$, allowing us to single out x as being the only assumption used with grade 1 in the subject; all other assumptions are used with grade 0.

For example, typing the body of the polymorphic identity ends with VAR:

$$\frac{\dfrac{\dots}{((), (1)) \odot a : \mathsf{Type}, x : a \vdash} \textsc{wfExt} \qquad |(())| = |a : \mathsf{Type}|}{(((), (1)) \mid 0, 1 \mid 1, 0) \odot a : \mathsf{Type}, x : a \vdash x : a} \textsc{Var}$$

The premise implies that $((), 1, 0) \odot a : \mathsf{Type} \vdash a : \mathsf{Type}$ by the following lemma:

Lemma 2 (Typing an assumption in a well-formed context). *If $\Delta_1, \sigma, \Delta_2 \odot \Gamma_1, x : A, \Gamma_2 \vdash$ with $|\Delta_1| = |\Gamma_1|$, then $(\Delta_1 \mid \sigma \mid \mathbf{0}) \odot \Gamma_1 \vdash A : \mathsf{Type}_l$ for some l.*

In the conclusion of VAR, the typing $((), 1, 0) \odot a : \mathsf{Type} \vdash a : \mathsf{Type}$ is 'distributed' to the typing of x in the context and to the formation the subject's type. Thus subject grade $(0, 1)$ corresponds to the absence of a from the subject and the presence of x, and subject-type grade $(1, 0)$ corresponds to the presence of a in the subject's type (a), and the absence of x.

Typing universes are formed as follows:

$$\frac{\Delta \odot \Gamma \vdash}{(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}} \mathsf{Type}$$

We use an inductive hierarchy of universes [46] with ordering $<$ such that $l < \mathsf{suc}\ l$. Universes can be formed under any well-formed context, with every assumption graded with 0 subject and subject-type use, capturing the absence of any assumptions from the universes, which are closed forms.

Functions Function types $(x :_{(s,r)} A) \to B$ are annotated with two grades: explaining that x is used with grade s in the body of the inhabiting function and with grade r in B. Function types have the following formation rule:

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : \mathsf{Type}_{l_1 \sqcup l_2}} \to$$

The usage of the dependencies of A and B (excepting x) are given by σ_1 and σ_2 in the premises (in the 'subject' position) which are combined as $\sigma_1 + \sigma_2$ (via pointwise vector addition using the $+$ of the semiring), which serves to *contract* the dependencies of the two types. The usage of x in B is captured by r, and then internalised to the binder in the conclusion of the rule. An arbitrary grade for s is allowed here as there is no information on how x is used in an inhabiting function body. Function terms are then typed by the following rule:

$$\frac{(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \quad (\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t : B}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t : (x :_{(s,r)} A) \to B}\ \lambda_i$$

The second premise types the body of the λ-term, showing that s captures the usage of x in t and r captures the usage of x in B; the subject and subject-type grades of x are then internalised as annotations on the function type's binder.

Dependent functions are eliminated through application:

$$\frac{\begin{array}{l}(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t_1 : (x :_{(s,r)} A) \to B \quad (\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : A\end{array}}{(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash t_1\, t_2 : [t_2/x]B}\ \lambda_e$$

where $*$ is the scalar multiplication of a vector, using the semiring multiplication. Given a function t_1 which uses its parameter with grade s to compute and with grade r in the typing of the result, we can apply it to a term t_2, provided that we have the resources required to form t_2 scaled by s at the subject level and by r at the subject-type level, since t_2 is substituted into the return type B. This scaling behaviour is akin to that used in coeffect calculi [25,48], QTT [6,43] and Linear Haskell [8], but scalar multiplication happens here at both the subject and subject-type level. The use of variables in A is accounted for by σ_1 as explained in the third premise, but these usages are not present in the resulting application since A no longer appears in the types or the terms.

Consider the constant function $\lambda x.\lambda y.x : (x :_{(1,0)} A) \to (y :_{(0,0)} B) \to A$ (for some A and B). Here the resources required for the second parameter will always be scaled by 0, which is absorbing, meaning that anything passed as the second argument has 0 subject and subject-type use. This example begins to show some of the power of grading—the grades capture the program structure at all levels.

Tensors The rule for forming dependent tensor types is as follows:

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_r A) \otimes B : \mathsf{Type}_l}\ \otimes$$

This rule is almost identical to function type formation $\to$ but with only a single grade r on the binder, since x is only bound in B (the type of the second component), and not computationally. For 'quantitative' semirings, where 0 really means unused (see Section 3), $(x :_0 A) \otimes B$ is then a product $A \times B$.

Dependent tensors are introduced as follows:

$$\frac{\begin{array}{l}(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma \vdash t_1 : A \qquad (\Delta \mid \sigma_4 \mid \sigma_3 + r * \sigma_2) \odot \Gamma \vdash t_2 : [t_1/x]B\end{array}}{(\Delta \mid \sigma_2 + \sigma_4 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash (t_1, t_2) : (x :_r A) \otimes B} \otimes_i$$

In the typing premise for t_2, occurrences of x are replaced with t_1 in the type, ensuring that the type of the second component (t_2) is calculated using the first component (t_1). The resources for t_1 in this substitution are scaled by r, accounting for the existing usage of x in B. In the conclusion, we see the resources for the two components (and their types) combined via the semiring addition.

Finally, tensors are eliminated with the following rule:

$$\frac{\begin{array}{l}(\Delta \mid \sigma_3 \mid \sigma_1 + \sigma_2) \odot \Gamma \vdash t_1 : (x :_r A) \otimes B \\ (\Delta, (\sigma_1 + \sigma_2) \mid \sigma_5, r' \mid \mathbf{0}) \odot \Gamma, z : (x :_r A) \otimes B \vdash C : \mathsf{Type}_l \\ (\Delta, \sigma_1, (\sigma_2, r) \mid \sigma_4, s, s \mid \sigma_5, r', r') \odot \Gamma, x : A, y : B \vdash t_2 : [(x, y)/z]C\end{array}}{(\Delta \mid \sigma_4 + s * \sigma_3 \mid \sigma_5 + r' * \sigma_3) \odot \Gamma \vdash \mathsf{let}\,(x, y) = t_1 \,\mathsf{in}\, t_2 : [t_1/z]C} \otimes_e$$

As this is a dependent eliminator, we allow the result type C to depend upon the value of the tensor as a whole, bound as z in the second premise with grade r', into which is substituted our actual tensor term t_1 in the conclusion.

Eliminating a tensor (t_1) requires that we consider each component (x and y) is used with the same grade s in the resulting expression t_2, and that we scale the resources of t_1 by s. This is because we cannot inspect t_1 itself, and semiring addition is not injective (preventing us from splitting the grades required to form t_1). This prevents forming certain functions (e.g., projections) under some semirings, but this can be overcome by the introduction of *graded modalities*.

Graded Modality Graded binders alone do not allow different parts of a value to be used differently, e.g., computing the length of a list ignores the elements, projecting from a pair discards one component. We therefore introduce a *graded modality* (à la [10,45]) allowing us to capture the notion of local inspection on data and internalising usage information into types. A type $\Box_s A$ denotes terms of type A that are used with grade s. Type formation and introduction rules are:

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A : \mathsf{Type}_l} \Box \qquad \frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A}{(\Delta \mid s * \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box t : \Box_s A} \Box_i$$

To form a term of type $\Box_s A$, we 'promote' a term t of type A by requiring that we can use the resources used to form t (σ_1) according to grade s. This 'promotion' resembles that of other graded modal systems (e.g., [3,10,23,45]), but the elimination needs to also account for type usage due to dependent elimination.

We can see promotion $\Box_i$ as capturing t for later use according to grade s. Thus, when eliminating a term of type $\Box_s A$, we must consider how the 'unboxed' term is used with grade s, as per the following dependent eliminator:

$$\frac{\begin{array}{l}(\Delta, \sigma_2 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, z : \Box_s A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : \Box_s A \quad (\Delta, \sigma_2 \mid \sigma_3, s \mid \sigma_4, (s * r)) \odot \Gamma, x : A \vdash t_2 : [\Box x/z]B\end{array}}{(\Delta \mid \sigma_1 + \sigma_3 \mid \sigma_4 + r * \sigma_1) \odot \Gamma \vdash \mathsf{let}\,\Box x = t_1 \,\mathsf{in}\, t_2 : [t_1/z]B} \Box_e$$

This rule can be understood as a kind of 'cut', connecting a 'capability' to use a term of type A according to grade s with the requirement that $x : A$ is used according to grade s as a dependency of t_2. Since we are in a dependently-typed setting, we also substitute t_1 into the type level such that B can depend on t_1 according to grade r which then causes the dependencies of t_1 (σ_1) to be scaled-up by r and added to the subject-type grading.

Equality, Conversion, and Subtyping A key part of dependent type theories is a notion of term equality and type conversion [33]. GRTT term equality is via judgments $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A$ equating terms t_1 and t_2 of type A. Equality includes full congruences as well as $\beta\eta$-equality for functions, tensors, and graded modalities, of which the latter are:

$$\frac{\begin{array}{l}(\Delta, \sigma_2 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, z : \Box_s A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : A \qquad (\Delta, \sigma_2 \mid \sigma_3, s \mid \sigma_4, (s * r)) \odot \Gamma, x : A \vdash t_2 : [\Box x/z]B\end{array}}{(\Delta \mid \sigma_3 + s * \sigma_1 \mid \sigma_4 + s * r * \sigma_1) \odot \Gamma \vdash (\mathsf{let}\, \Box x = \Box t_1 \,\mathsf{in}\, t_2) = [t_1/x]t_2 : [\Box t_1/z]B} \textsc{Eq}_{\Box_c}$$

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : \Box_s A}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t = (\mathsf{let}\, \Box x = t \,\mathsf{in}\, \Box x) : \Box_s A} \textsc{Eq}_{\Box_u}$$

A subtyping relation ($(\Delta \mid \sigma) \odot \Gamma \vdash A \leq B$) subsumes equality, adding ordering of universe levels. *Type conversion* allows re-typing terms based on the judgment:

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash A \leq B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : B} \textsc{Conv}$$

The full rules for equality and subtyping are in Appendix A.1.

2.4 Operational Semantics

As with other graded modal calculi (e.g., [3,10,23]), the core calculus of GRTT has a Call-by-Name small-step operational semantics with reductions $t \rightsquigarrow t'$. The rules are standard, with the addition of the β-rule for the graded modality:

$$\mathsf{let}\, \Box x = \Box t_1 \,\mathsf{in}\, t_2 \rightsquigarrow [t_1/x]t_2 \qquad (\beta\Box)$$

Type preservation and normalisation are considered in Section 4.

2.5 Implementation and Examples

To explore our theory, we provide an implementation, **Gerty**. Section 5 describes how the declarative definition of the type theory is implemented as a bidirectional type checking algorithm. We briefly mention the syntax here for use in later examples. The following is the polymorphic identity function in **Gerty**:

```
id : (a : (.0, .2) Type 0) -> (x : (.1, .0) a) -> a
id = \a -> \x -> x
```

The syntax resembles the theory, where grading terms `.n` are syntactic sugar for a unary encoding of grades in terms of 0 and repeated addition of 1, e.g., `.2` = `(.0 + .1) + .1`. This syntax can be used for grade terms of any semiring, which can be resolved to particular built-in semirings at other points of type checking.

The following shows first projection on (non-dependent) pairs, using the graded modality (at grade 0 here) to give fine-grained usage on compound data:

```
fst : (a : (.0, .2) Type 0) (b : (.0, .1) Type 0) -> <a * [.0] b> -> a
fst = \a b p -> case p of <x, y> -> let [z] = y in x
```

The implementation adds various built-in semirings, some syntactic sugar, and extras such as: a singleton *unit* type, extensions of the theory to semirings with a pre-ordering (discussed further in Section 6), and some implicit resolution. Anywhere a grade is expected, an underscore can be supplied to indicate that **Gerty** should try to resolve the grade implicitly. Grades may also be omitted from binders (see above in `fst`), in which case they are treated as implicits. Currently, implicits are handled by generating existentially quantified grade variables, and using SMT to solve the necessary constraints (see Section 5).

So far we have considered the natural numbers semiring providing an analysis of usage. We come back to this and similar examples in Section 3. To show another kind of example, we consider a lattice semiring of privacy levels (appearing elsewhere [3,23,45]) which enforces information-flow control, akin to DCC [1]. Differently to DCC, dataflow is tracked through variable dependencies, rather than through the results of computations in the monadic style of DCC.

Definition 1. [Security levels] Let $\mathcal{R} = \texttt{Lo} \leq \texttt{Hi}$ be a set of labels with $0 = \texttt{Hi}$ and $1 = \texttt{Lo}$, semiring addition as the meet and multiplication as join. Here, $1 = \texttt{Lo}$ treats the base notion of dataflow as being in the low security (public) domain. Variables graded with `Hi` must then be unused, or guarded by a graded modality. This semiring is primitive in **Gerty**; we can express the following example:

```
idLo : (a : (.0, .2) Type 0) -> (x : (Lo, Hi) a) -> a
idLo = \a -> \x -> x
-- The following is rejected as ill-typed
leak : (a : (.0, .2) Type 0) -> (x : (Hi, Hi) a) -> a
leak = \a -> \x -> idLo a x
```

The first definition is well-typed, but the second yields a typing error originating from the application in its body:

```
At subject stage got the following mismatched grades:
 For 'x' expected Hi but got .1
```

where grade `1` is `Lo` here. Thus we can use this abstract label semiring as a way of restricting flow of data between regions (*cf.* region typing systems [31,54]). Note that the ordering is not leveraged here other than in the lattice operations.

3 Case Studies

We now demonstrate GRTT via several cases studies that focus the reasoning power of dependent types via grading. Since grading in GRTT serves to explain dataflow, we can characterise subsets of GRTT that correspond to various type theories. We demonstrate the approach with simple types, parametric polymorphism, and linearity. In each case study, we restrict GRTT to a subset by *a characterisation of the grades*, rather than by, say, placing detailed syntactic restrictions or employing meta-level operations or predicates that restrict syntax (as one might do for example to map a subset of Martin-Löf type theory into the simply-typed λ-calculus by restriction to closed types, requiring deep inspection of type terms). Since this restriction is only on grades, we can harness the specific reasoning power of particular calculi from within the language itself, simply by specifications on grades. In the context of an implementation like **Gerty**, this amounts to using type signatures to restrict dataflow.

This section shows the power of tracking dataflow in types via grades, going beyond QTT [6] and GRAD [13]. For 'quantitative' semirings, a 0 type-grade means that we can recover simply-typed reasoning (Section 3.3) and distinguish computational functions from type-parameter functions for parametric reasoning (Section 3.4), embedding a grade-restricted subset of GRTT into System F.

Section 5 returns to a case study that builds on the implementation.

3.1 Recovering Martin-Löf Type Theory

When the semiring parameterising GRTT is the singleton semiring (i.e., any semiring where $1 = 0$), we have an isomorphism $\square_r A \cong A$, and grade annotations become redundant, as all grades are equal. All vectors and grades on binders may then be omitted, and we can write typing judgments as $\Gamma \vdash t : A$, giving rise to a standard Martin-Löf type theory as a special case of GRTT.

3.2 Determining Usage via Quantitative Semirings

Unlike existing systems, we can use the fine-grained grading to *guarantee* the relevance or irrelevance of assumptions in types. To do this we must consider a subset of semirings $(\mathcal{R}, *, 1, +, 0)$ called *quantitative* semirings, satisfying:

$$\begin{aligned} &\text{(zero-unique)} && 1 \neq 0; \\ &\text{(positivity)} && \forall r, s.\ r + s = 0 \implies r = 0 \wedge s = 0; \\ &\text{(zero-product)} && \forall r, s.\ r * s = 0 \implies r = 0 \vee s = 0. \end{aligned}$$

These axioms[4] ensure that a 0-grade in a quantitative semiring represents irrelevant variable use. This notion has recently been proved for computational use

[4] Atkey requires *positivity* and *zero-product* for all semirings parameterising QTT [6] (as does Abel [2]). Atkey imposes this for admissibility of substitution. We need not place this restriction on GRTT to have substitution in general (Sec. 4.1).

by Choudhury et al. [13] via a heap-based semantics for grading (on computations) and the same result applies here. Conversely, in a quantitative semiring any grade other than 0 denotes relevance. From this, we can *directly* encode non-dependent tensors and arrows: in $(x :_0 A) \otimes B$ the grade 0 captures that x cannot have any computational content in B, and likewise for $(x :_{(s,0)} A) \to B$ the grade 0 explains that x cannot have any computational content in B, but may have computational use according to s in the inhabiting function. Thus, the grade 0 here describes that elimination forms *cannot* ever inspect the variable during normalisation. Additionally, quantitative semirings can be used for encoding simply-typed and polymorphic reasoning.

Example 1. Some quantitative semirings are:

- (*Exact usage*) $(\mathbb{N}, \times, 1, +, 0)$;
- (*0-1*) The semiring over $\mathcal{R} = \{0, 1\}$ with $1 + 1 = 1$ which describes relevant *vs.* irrelevant dependencies, but no further information.
- (*None-One-Tons* [43]) The semiring on $\mathcal{R} = \{0, 1, \infty\}$ is more fine-grained than 0-1, where ∞ represents more than 1 usage, with $1 + 1 = \infty = 1 + \infty$.

3.3 Simply-typed Reasoning

As discussed in Section 1, the simply-typed λ-calculus (STLC) can be distinguished from dependently-typed calculi via the restriction of dataflow: in simple types, data can only flow at the computational level, with no dataflow within, into, or from types. We can thus view a Grtt function as simply typed when its variable is irrelevant in the type, e.g., $(x :_{(s,0)} A) \to B$ for quantitative semirings. We define a subset of Grtt restricted to simply-typed reasoning:

Definition 2. [Simply-typed Grtt] For a quantitative semiring, the following predicate $\textsc{Stlc}(-)$ determines a subset of simply-typed Grtt programs:

$$\textsc{Stlc}((\emptyset \mid \emptyset \mid \emptyset) \odot \emptyset \vdash t : A)$$
$$\textsc{Stlc}((\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A) \implies \textsc{Stlc}((\Delta, \mathbf{0} \mid \sigma_1, s \mid \sigma_2, 0) \odot \Gamma, x : B \vdash t : A)$$

That is, all subject-type grades are 0 (thus function types are of the form $(x :_{(s,0)} A) \to B$). A similar predicate is defined on well-formed contexts (elided), restricting context grades of well-formed contexts to only zero grading vectors.

Under the restriction of Definition 2, a subset of Grtt terms embeds into the simply-typed λ-calculus in a sound and complete way. Since STLC does not have a notion of tensor or modality, this is omitted from the encoding:

$$[\![x]\!] = x \quad [\![\lambda x.t]\!] = \lambda x.[\![t]\!] \quad [\![t_1\, t_2]\!] = [\![t_1]\!][\![t_2]\!] \quad [\![(x :_{(s,0)} A) \to B]\!]_\tau = [\![A]\!]_\tau \to [\![B]\!]_\tau$$

Variable contexts of Grtt are interpreted by point-wise applying $[\![-]\!]_\tau$ to typing assumptions. We then get the following preservation of typing into the simply-typed λ-calculus, and soundness and completeness of this encoding:

Lemma 3 (Soundness of typing). *Given a derivation of* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$ *such that* $\textsc{Stlc}((\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A)$ *then* $[\![\Gamma]\!]_\tau \vdash [\![t]\!] : [\![A]\!]_\tau$ *in STLC.*

Theorem 1 (Soundness and completeness of the embedding). *Given* $\textsc{Stlc}((\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A)$ *and* $[\![(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A]\!]$ *then for CBN reduction* $\leadsto^{\text{STLC}}$ *in simply-typed* λ*-calculus:*

$$\begin{array}{rl} \textit{(soundness)}\ \forall t'.\ \textit{if}\ t \leadsto t' & \textit{then}\ [\![t]\!] \leadsto^{\text{STLC}} [\![t']\!] \\ \textit{(completeness)}\ \forall t_a.\ \textit{if}\ [\![t]\!] \leadsto^{\text{STLC}} t_a & \textit{then}\ \exists t'.\ t \leadsto t' \ \wedge\ [\![t']\!] \equiv_{\beta\eta} t_a \end{array}$$

Thus, we capture simply-typed reasoning just by restricting type grades to 0 for quantitative semirings. We consider quantitative semirings again for parametric reasoning, but first recall issues with parametricity and dependent types.

3.4 Recovering Parametricity via Grading

One powerful feature of grading in a dependent type setting is the ability to recover parametricity from dependent function types. Consider the following type of functions in System F (we borrow this example from Nuyts et al. [44]):

$$\mathsf{RI}\ A\ B \ \triangleq\ \forall\gamma.(\gamma \to A) \to (\gamma \to B)$$

Due to parametricity, we get the following notion of *representation independence* in System F: for a function $f : \mathsf{RI}\ A\ B$, some type γ', and terms $h : \gamma' \to A$ and $c : \gamma'$, then we know that f can only use c by applying $h\ c$. Subsequently, $\mathsf{RI}\ A\ B \cong A \to B$ by parametricity [51], defined uniquely as:

$$\begin{array}{ll} \textit{iso} : \mathsf{RI}\ A\ B \to (A \to B) & \textit{iso}^{-1} : (A \to B) \to \mathsf{RI}\ A\ B \\ \textit{iso}\ f = f\ A\ (\textit{id}\ A) & \textit{iso}^{-1}\ g = \Lambda\gamma.\ \lambda h.\ \lambda(c : \gamma).\ g(h\,c) \end{array}$$

In a dependently-typed language, one might seek to replace System F's universal quantifier with Π-types, i.e.

$$\mathsf{RI}'\ A\ B \ \triangleq\ (\gamma : \mathsf{Type}) \to (\gamma \to A) \to (\gamma \to B)$$

However, we can no longer reason parametrically about the inhabitants of such types (we cannot prove that $\mathsf{RI}'\ A\ B \cong A \to B$) as the free interaction of types and computational terms allows us to give the following non-parametric element of $\mathsf{RI}'\ A\ B$ over 'large' type instances:

$$\textit{leak} = \lambda\gamma.\ \lambda h.\ \lambda c.\ \gamma : \mathsf{RI}'\ A\ \mathsf{Type}$$

Instead of applying $h\ c$, the above "leaks" the type parameter γ. Grtt can recover universal quantification, and hence parametric reasoning, by using grading to restrict the data-flow capabilities of a Π-type. We can refine representation independence to the following:

$$\mathsf{RI}''\ A\ B \ \triangleq\ (\gamma :_{(0,2)} \mathsf{Type}) \to (h :_{(s_1,0)} (x :_{(s_2,0)} \gamma) \to A) \to (c :_{(s_3,0)} \gamma) \to B$$

for some grades s_1, s_2, and s_3, and with shorthand $2 = 1 + 1$.

If we look at the definition of *leak* above, we see that γ is used in the body of the function and thus requires usage 1, so *leak* cannot inhabit $\mathsf{RI}''\ A\ \mathsf{Type}$. Instead, *leak* would be typed differently as:

$$leak : (\gamma :_{(1,2)} \mathsf{Type}) \to (h :_{(0,0)} (x :_{(s,0)} \gamma) \to A) \to (c :_{(0,0)} \gamma) \to \mathsf{Type}$$

The problematic behaviour (that the type parameter γ is returned by the inner function) is exposed by the subject grade 1 on the binder of γ. We can thus define a graded universal quantification from a graded Π-typed:

$$\forall_r(\gamma : A).B \triangleq (\gamma :_{(0,r)} A) \to B \tag{1}$$

This denotes that the type parameter γ can appear freely in B described by grade r, but is irrelevant in the body of any corresponding λ-abstraction. This is akin to the work of Nuyts et al. who develop a system with several modalities for regaining parametricity within a dependent type theory [44]. Note however that parametricity is recovered for us here as one of many possible options coming from systematically specialising the grading.

Capturing Existential Types With the ability to capture universal quantifier, we can similarly define existentials (allowing, e.g., abstraction [11]). We define the existential type via a Church-encoding as follows:

$$\exists_r(x : A).B \triangleq \forall_2(C : \mathsf{Type}_l).(f :_{(1,0)} \forall_r(x : A).(b :_{(s,0)} B) \to C) \to C$$

Embedding into Stratified System F We show that parametricity is regained here (and thus eqn. (1) really behaves as a universal quantifier and not a general Π-type) by showing that we can embed a subset of GRTT into System F, based solely on a classification of the grades. We follow a similar approach to Section 3.3 for simply-typed reasoning but rather than defining a purely syntactic encoding (and then proving it type sound) our encoding is type directed since we embed GRTT functions of type $(x :_{(0,r)} \mathsf{Type}_l) \to B$ as universal types in System F with corresponding type abstractions (Λ) as their inhabitants. Since GRTT employs a predicative hierarchy of universes, we target Stratified System F (hereafter SSF) since it includes the analogous inductive hierarchy of kinds [38]. We use the formulation of Eades and Stump [21] with terms t_s and types T:

$$t_s ::= x \mid \lambda(x : T).t_s \mid t_s\, t'_s \mid \Lambda(X : K).t_s \mid t_s\,[T] \qquad T ::= X \mid T \to T' \mid \forall(X : K).T$$

with kinds $K ::= \star_l$ where $l \in \mathbb{N}$ providing the stratified kind hierarchy. Capitalised variables X are System F type variables and $t_s\,[T]$ is type application. Contexts may contain both type and computational variables, and so free-variable type assumptions may have dependencies, akin to dependent type systems. Kinding is via judgments $\Gamma \vdash T : \star_l$ and typing via $\Gamma \vdash t : T$.

We define a type directed encoding on a subset of GRTT typing derivations characterised by the following predicate:

$$\begin{aligned}
&\textsc{Ssf}((\emptyset \mid \emptyset \mid \emptyset) \odot \emptyset \vdash t : A)\\
&\textsc{Ssf}((\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A) \implies \textsc{Ssf}((\Delta, \mathbf{0} \mid \sigma_1, 0 \mid \sigma_2, r) \odot \Gamma, x : \mathsf{Type}_l \vdash t : A)\\
&\textsc{Ssf}((\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A) \wedge \mathsf{Type}_l \not\in^{+ve} B\\
&\qquad\qquad\qquad\qquad\qquad \implies \textsc{Ssf}((\Delta, \sigma_3 \mid \sigma_1, s \mid \sigma_2, 0) \odot \Gamma, x : B \vdash t : A)
\end{aligned}$$

By $\mathsf{Type}_l \not\in^{+ve} B$ we mean Type_l is not a positive subterm of B, avoiding higher-order typing terms (e.g., type constructors) which do not exist in SSF.

Under this restriction, we give a type-directed encoding mapping derivations of GRTT to SSF: given a GRTT derivation of judgment $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$ we have that $\exists t_s$ (an SSF term) such that there is a derivation of judgment $[\![\Gamma]\!] \vdash t_s : [\![A]\!]_\tau$ in SSF where we interpret a subset of GRTT terms A as types:

$$\begin{aligned}
[\![x]\!]_\tau &= x\\
[\![\mathsf{Type}_l]\!]_\tau &= \star_l\\
[\![(x :_{(0,r)} \mathsf{Type}_l) \to B]\!]_\tau &= \forall x : \star_l . [\![B]\!]_\tau \quad \textit{where } \mathsf{Type}_l \not\in^{+ve} B\\
[\![(x :_{(s,0)} A) \to B]\!]_\tau &= [\![A]\!]_\tau \to [\![B]\!]_\tau \quad \textit{where } \mathsf{Type}_l \not\in^{+ve} A, B
\end{aligned}$$

Thus, dependent functions with Type parameters that are computationally irrelevant (subject grade 0) map to $\forall$ types, and dependent functions with parameters irrelevant in types (subject-type grade 0) map to regular function types. We elide the full details but sketch key parts where functions and applications are translated inductively (where Ty_l is shorthand for Type_l):

$$[\![\frac{(\Delta, \sigma_1 \mid \sigma_2, 0 \mid \sigma_3, r) \odot \Gamma, x : \mathsf{Ty}_l \vdash t : B}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t : (x :_{(0,r)} \mathsf{Ty}_l) \to B}]\!] = \frac{[\![\Gamma]\!], x : \star_l \vdash t_s : [\![B]\!]_\tau}{[\![\Gamma]\!] \vdash \Lambda(x : \star_l).t_s : \forall x : \star_l . [\![B]\!]_\tau}$$

$$[\![\frac{(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, 0) \odot \Gamma, x : A \vdash t : B}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t : (x :_{(s,0)} A) \to B}]\!] = \frac{[\![\Gamma]\!], x : [\![A]\!]_\tau \vdash t_s : [\![B]\!]_\tau}{[\![\Gamma]\!] \vdash \lambda(x : [\![A]\!]_\tau).t_s : [\![A]\!]_\tau \to [\![B]\!]_\tau}$$

$$[\![\frac{\begin{array}{c}(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t_1 : (x :_{(0,r)} \mathsf{Ty}_l) \to B\\ (\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : \mathsf{Ty}_l\end{array}}{(\Delta \mid \sigma_2 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash t_1\, t_2 : [t_2/x]B}]\!] = \frac{\begin{array}{c}[\![\Gamma]\!] \vdash t_s : \forall (x : \star_l).[\![B]\!]_\tau\\ [\![\Gamma]\!] \vdash T : \star_l\end{array}}{[\![\Gamma]\!] \vdash t_s[T] : [T/x][\![B]\!]_\tau}$$

$$[\![\frac{\begin{array}{c}(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t_1 : (x :_{(s,0)} A) \to B\\ (\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : A\end{array}}{(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3) \odot \Gamma \vdash t_1\, t_2 : [t_2/x]B}]\!] = \frac{\begin{array}{c}[\![\Gamma]\!] \vdash t_s : [\![A]\!]_\tau \to [\![B]\!]_\tau\\ [\![\Gamma]\!] \vdash t'_s : [\![A]\!]_\tau\end{array}}{[\![\Gamma]\!] \vdash t_s\ t'_s : [t'_s/x][\![B]\!]_\tau}$$

In the last case, note the presence of $[t'_s/x][\![B]\!]_\tau$. Reasoning under the context of the encoding, this is proven equivalent to $[\![B]\!]_\tau$ since the subject type grade is 0 and therefore use of x in B is irrelevant.

Theorem 2 (Soundness and completeness of SSF embedding). *Given* $\textsc{Ssf}((\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A)$ *and* t_a *in SSF where* $[\![(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A]\!] = [\![\Gamma]\!] \vdash t_s : [\![A]\!]_\tau$ *then for CBN reduction* $\rightsquigarrow^{\textsc{Ssf}}$ *in Stratified System F:*

$$\begin{aligned}
&\textit{(soundness)}\ \forall t'.\ t \rightsquigarrow t' \implies \exists t'_s . t_s \rightsquigarrow^{\textsc{Ssf}} t'_s\\
&\qquad\qquad \wedge\ [\![(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t' : A]\!] = [\![\Gamma]\!] \vdash t'_s : [\![A]\!]_\tau\\
&\textit{(completeness)}\ \forall t'_s .\ t_s \rightsquigarrow^{\textsc{Ssf}} t'_s \implies \exists t'.t \rightsquigarrow t'\\
&\qquad\qquad \wedge\ [\![(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t' : A]\!] = [\![\Gamma]\!] \vdash t'_s : [\![A]\!]_\tau
\end{aligned}$$

Thus, we can capture parametricity in GRTT via the judicious use of 0 grading (at either the type or computational level) for quantitative semirings. This embedding is not possible from QTT since QTT variables graded with 0 may be used arbitrarily in the types; the embedding here relies on GRTT's 0 type-grade capturing abscence in types for quantitative semirings.

3.5 Graded Modal Types and Non-dependent Linear Types

GRTT can embed the reasoning present in other graded modal type theories (which often have a linear base), for example the explicit semiring-graded necessity modality found in coeffect calculi [10,23] and Granule [45]. We can recover the axioms of a graded necessity modality (usually modelled by an exponential graded comonad [23]). For example, in **Gerty** the following are well typed:

```
counit : (a : (.0, .2) Type) -> (z : (.1 , .0) [.1] a) -> a
counit = \a z -> case z of [y] -> y
comult : (a : (.0, .2) Type) -> (z : (.1 , .0) [.6] a) -> [.2] ([.3] a)
comult = \a z -> case z of [y] -> [[y]]
```

corresponding to $\varepsilon : \Box_1 A \to A$ and $\delta_{r,s} : \Box_{r*s} A \to \Box_r(\Box_s A)$: operations of graded necessity / graded comonads. Since we cannot use arbitrary terms for grades in the implementation, we have picked some particular grades here for `comult`. First-class grading is future work, discussed in Section 6.

Linear functions can be captured as $A \multimap B \triangleq (x :_{(1,r)} A) \to B$ for an exact usage semiring. It is straightforward to characterise a subset of GRTT programs that maps to the linear λ-calculus akin to the encodings above. Thus, GRTT provides a suitable basis for studying both linear and non-linear theories alike.

4 Metatheory

We now study GRTT's metatheory. We first explain how substitution presents itself in the theory, and how type preservation follows from a relationship between equality and reduction. We then show admissibility of graded structural rules for contraction, exchange, and weakening, and strong normalization.

4.1 Substitution

We introducing substitution for well-formed contexts and then typing.

Lemma 4 (Substitution for well-formed contexts). *If the following hold:*

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ *and* *2.* $(\Delta, \sigma_1, \Delta') \odot \Gamma_1, x : A, \Gamma_2 \vdash$

Then: $\Delta, (\Delta' \backslash |\Delta| + (\Delta' / |\Delta|) * \sigma_2) \odot \Gamma_1, [t/x]\Gamma_2 \vdash$

That is, given $\Gamma_1, x : A, \Gamma_2$ is well-formed, we can cut out x by substituting t for x in Γ_2, accounting for the new usage in the context grade vectors. The usage of Γ_1 in t is given by σ_2, and the usage in A by σ_1. When substituting, Δ remains the same, as Γ_1 is unchanged. However, to account for the usage in $[t/x]\Gamma_2$, we have to form a new context grade vector $\Delta' \backslash |\Delta| + (\Delta' / |\Delta|) * \sigma_2$.

The operation $\Delta' \backslash |\Delta|$ (pronounced 'discard') removes grades corresponding to x, by removing the grade at index $|\Delta|$ from each grade vector in Δ'. Everything previously used in the typing of x in the context must now be distributed across $[t/x]\Gamma_2$, which is done by adding on $(\Delta' / |\Delta|) * \sigma_2$, which uses $\Delta' / |\Delta|$ (pronounced 'choose') to produce a vector of grades, which correspond to the grades cut out in $\Delta' \backslash |\Delta|$. The multiplication of $(\Delta' / |\Delta|) * \sigma_2$ produces a context grade vector by scaling σ_2 by each element of $(\Delta' / |\Delta|)$. When adding vectors, if the sizes of the vectors are different, then the shorter vector is right-padded with zeroes. Thus $\Delta' \backslash |\Delta| + (\Delta' / |\Delta|) * \sigma_2$ can be read as *'Δ' without the grades corresponding to x, plus the usage of t scaled by the prior usage of x'*.

For example, given typing $((), (1) \mid 0, 1 \mid 1, 0) \odot a : \mathsf{Type}, y : a \vdash y : a$ and well-formed context $((), (1), (1, 0), (0, 0, 2)) \odot a : \mathsf{Type}, y : a, x : a, z : t' \vdash$, where t' uses x twice, we can substitute y for x. Therefore, let $\Gamma_1 = a : \mathsf{Type}, y : a$ thus $|\Gamma_1| = 2$ and $\Gamma_2 = z : x$ and $\Delta' = ((0, 0, 2))$ and $\sigma_1 = 1, 0$ and $\sigma_2 = 0, 1$. Then the context grade of the substitution $[y/x]\Gamma_2$ is calculated as:

$$((0,0,2)) \backslash |\Gamma_1| = ((0,0)) \qquad (((0,1,2)) / |\Gamma_1|) * \sigma_2 = (2) * (0,1) = ((0,2))$$

Thus the resulting judgment is $((), (1), (0, 2)) \odot a : \mathsf{Type}, y : a, z : [y/x]t' \vdash$.

Lemma 5 (Substitution for typing). *If the following premises hold:*

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$
2. $(\Delta, \sigma_1, \Delta' \mid \sigma_3, s, \sigma_4 \mid \sigma_5, r, \sigma_6) \odot \Gamma_1, x : A, \Gamma_2 \vdash t' : B$
3. $|\sigma_3| = |\sigma_5| = |\Gamma_1|$

Then $\begin{pmatrix} \Delta, (\Delta' \backslash |\Delta| + (\Delta' / |\Delta|) * \sigma_2) \\ (\sigma_3 + s * \sigma_2), \sigma_4 \\ (\sigma_5 + r * \sigma_2), \sigma_6 \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]t' : [t/x]B$.

As with substitution for well-formed contexts, we account for the replacement of x with t in Γ_2 by 'cutting out' x from the context grade vectors, and adding on the grades required to form t, scaled by the grades that described x's usage. We additionally must account for the altered subject and subject-type usage. We do this in a similar manner, by taking, for example, the usage of Γ_1 in the subject (σ_3), and adding on the grades required to form t, scaled by the grade with which x was previously used (s). Subject-type grades are calculated similarly.

4.2 Type Preservation

Lemma 6. *Reduction implies equality If $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : A$ and $t_1 \rightsquigarrow t_2$, then $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A$.*

Lemma 7. *Equality inversion If $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A$, then $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : A$ and $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_2 : A$.*

Lemma 8. *Type preservation If* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$ *and* $t \leadsto t'$*, then* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t' : A$.

Proof. By Lemma 6 we have $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t = t' : A$, and therefore by Lemma 7 we have $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t' : A$, as required.

4.3 Structural Rules

We now consider the structural rules of *contraction*, *exchange*, and *weakening*.

Lemma 9 (Contraction). *The following rule is admissible:*

$$\frac{\begin{pmatrix} \Delta_1,\sigma_1,(\sigma_1,0),\Delta_2 \\ \sigma_2,s_1,s_2,\sigma_3 \\ \sigma_4,r_1,r_2,\sigma_5 \end{pmatrix} \odot \Gamma_1, x : A, y : A, \Gamma_2 \vdash t : B \quad |\Delta_1| = |\sigma_2| = |\sigma_4| = |\Gamma_1|}{\begin{pmatrix} \Delta_1,\sigma_1,\mathsf{contr}(|\Delta_1|;\Delta_2) \\ \sigma_2,(s_1+s_2),\sigma_3 \\ \sigma_4,(r_1+r_2),\sigma_5 \end{pmatrix} \odot \Gamma_1, z : A, [z, z/x, y]\Gamma_2 \vdash [z, z/x, y]t : [z, z/x, y]B} \textsc{Contr}$$

The operation $\mathsf{contr}(\pi; \Delta)$ contracts the elements at index π and $\pi + 1$ for each vector in Δ by combining them with the semiring addition, defined $\mathsf{contr}(\pi; \Delta) = \Delta\backslash(\pi+1) + \Delta/(\pi+1) * (\mathbf{0}^\pi, 1)$. Admissibility follows from the semiring addition, which serves to contract dependencies, being threaded throughout the rules.

Lemma 10 (Exchange). *The following rule is admissible:*

$$\frac{\begin{matrix} x \notin \mathsf{FV}(B) \\ |\Delta_1| = |\sigma_3| = |\sigma_5| = |\Gamma_1| \quad \begin{pmatrix} \Delta_1,\sigma_1,(\sigma_2,0),\Delta_2 \\ \sigma_3,s_1,s_2,\sigma_4 \\ \sigma_5,r_1,r_2,\sigma_6 \end{pmatrix} \odot \Gamma_1, x : A, y : B, \Gamma_2 \vdash t : C \end{matrix}}{\begin{pmatrix} \Delta_1,\sigma_2,(\sigma_1,0),\mathsf{exch}(|\Delta_1|;\Delta_2) \\ \sigma_3,s_2,s_1,\sigma_4 \\ \sigma_5,r_2,r_1,\sigma_6 \end{pmatrix} \odot \Gamma_1, y : B, x : A, \Gamma_2 \vdash t : C} \textsc{Exc}$$

Notice that if you strip away the vector fragment and sizing premise, this is exactly the form of exchange we would expect in a dependent type theory: if x and y are assumptions in a context typing $t : C$, and the type of y does not depend upon x, then we can type $t : C$ when we swap the order of x and y.

The action on grade vectors is simple: we swap the grades associated with each of the variables. For the context grade vector however, we must do two things: first, we capture the formation of A with σ_1, and the formation of B with $\sigma_1, 0$ (indicating x being used with grade 0 in B), then swap these around, cutting the final grade from $\sigma_2, 0$, and adding 0 to the end of σ_1 to ensure correct sizing. Next, the operation $\mathsf{exch}(|\Delta_1| ; \Delta_2)$ swaps the element at index $|\Delta_1|$ (i.e., that corresponding to usage of x) with the element at index $|\Delta_1| + 1$ (corresponding to y) for every vector in Δ_2; this exchange operation ensures that usage in the trailing context is reordered appropriately.

Lemma 11 (Weakening). *The following rule is admissible:*

$$\frac{\begin{matrix} (\Delta_1, \Delta_2 \mid \sigma_1, \sigma_1' \mid \sigma_2, \sigma_2') \odot \Gamma_1, \Gamma_2 \vdash t : B \\ (\Delta_1 \mid \sigma_3 \mid \boldsymbol{0}) \odot \Gamma_1 \vdash A : \mathsf{Type}_l \qquad |\sigma_1| = |\sigma_2| = |\Gamma_1| \end{matrix}}{(\Delta_1, \sigma_3, \mathsf{ins}(|\Delta_1| ; 0; \Delta_2) \mid \sigma_1, 0, \sigma_1' \mid \sigma_2, 0, \sigma_2') \odot \Gamma_1, x : A, \Gamma_2 \vdash t : B} \textsc{Weak}$$

Weakening introduces irrelevant assumptions to a context. We do this by capturing the usage in the formation of the assumption's type with σ_3 to preserve the well-formedness of the context. We then indicate irrelevance of the assumption by grading with 0 in appropriate places. The operation $\mathsf{ins}(\pi; s; \Delta)$ inserts the element s at index π for each σ in Δ, such that all elements preceding index π (in σ) keep their positions, and every element at index π or greater (in σ) will be shifted one index later in the new vector. The 0 grades in the subject and subject-type grade vector positions correspond to the absence of the irrelevant assumption from the subject and subject's type.

4.4 Strong Normalization

We adapt Geuvers' strong normalization proof for the Calculus of Constructions (CC) [24] to a fragment of GRTT (called $\text{GRTT}^{\{0,1\}}$) restricted to two universe levels and without variables of type Type_1. This results in a less expressive system than full GRTT when it comes to higher kinds, but this is orthogonal to the main idea here of grading. We briefly overview the strong normalization proof; details can be found in Appendix A.2. Note this strong normalization result is with respect to β-reduction only (our semantics does not include η-reduction).

We use the proof technique of saturated sets, based on the reducibility candidates of Girard [29]. While $\text{GRTT}^{\{0,1\}}$ has a collapsed syntax we use judgments to break typing up into stages. We use these sets to match on whether a term is a kind, type, constructor, or a function (we will refer to these as terms).

Definition 3. Typing can be broken up into the following stages:

$$\begin{array}{lcl}
\mathsf{Kind} & := & \{A \mid \exists \Delta, \sigma_1, \Gamma.(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_1\} \\
\mathsf{Type} & := & \{A \mid \exists \Delta, \sigma_1, \Gamma.(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_0\} \\
\mathsf{Con} & := & \{t \mid \exists \Delta, \sigma_1, \sigma_2, \Gamma, A.(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A \wedge (\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_1\} \\
\mathsf{Term} & := & \{t \mid \exists \Delta, \sigma_1, \sigma_2, \Gamma, A.(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A \wedge (\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_0\}
\end{array}$$

Lemma 12 (Classification). *We have* $\mathsf{Kind} \cap \mathsf{Type} = \emptyset$ *and* $\mathsf{Con} \cap \mathsf{Term} = \emptyset$.

The classification lemma states that we can safely case split over kinds and types, or constructors and terms without fear of an overlap occurring.

Saturated sets are essentially collections of strongly normalizing terms that are closed under β-reduction. The intuition behind this proof is that every typable program ends up in some saturated set, and hence, is strongly normalizing.

Definition 4. [Base terms and saturated terms] Informally, the set of base terms $\mathcal{B}$ is inductively defined from variables and Type_0 and Type_1, and compound terms over base $\mathcal{B}$ and strongly normalising terms SN.

A set of terms X is *saturated* if $X \subset \mathsf{SN}$, $\mathcal{B} \subset X$, and if $\mathsf{red_k}\, t \in X$ and $t \in \mathsf{SN}$, then $t \in X$. Thus saturated sets are closed under strongly normalizing terms with a *key redex*, denoted $\mathsf{red_k}\, t$, which are redexes or a redex at the head of an elimination form. SAT denotes the collection of saturated sets.

Lemma 13 **(SN saturated).** *All saturated sets are non-empty;* SN *is saturated.*

Since $\text{GRTT}^{\{0,1\}}$ allows computation in types as well as in types, we separate the interpretations for kinds and types, where the former is a set of the latter.

Definition 5. For $A \in \mathsf{Kind}$, the kind interpretation, $\mathcal{K}[\![A]\!]$, is defined:

$$
\begin{array}{ll}
\mathcal{K}[\![\mathsf{Type}_0]\!] = \mathsf{SAT} & \mathcal{K}[\![(x :_{(s,r)} A) \to B]\!] = \{f \mid f : \mathcal{K}[\![A]\!] \to \mathcal{K}[\![B]\!]\}, \text{ if } A, B \in \mathsf{Kind} \\
\mathcal{K}[\![\Box_s A]\!] \;= \mathcal{K}[\![A]\!] & \mathcal{K}[\![(x :_{(s,r)} A) \to B]\!] = \mathcal{K}[\![A]\!], \text{ if } A \in \mathsf{Kind}, B \in \mathsf{Type} \\
 & \mathcal{K}[\![(x :_{(s,r)} A) \to B]\!] = \mathcal{K}[\![B]\!], \text{ if } A \in \mathsf{Type}, B \in \mathsf{Kind} \\
 & \mathcal{K}[\![(x :_{r} A) \otimes B]\!] \quad\; = \mathcal{K}[\![A]\!] \times \mathcal{K}[\![B]\!], \text{ if } A, B \in \mathsf{Kind} \\
 & \mathcal{K}[\![(x :_{r} A) \otimes B]\!] \quad\; = \mathcal{K}[\![A]\!], \text{ if } A \in \mathsf{Kind}, B \in \mathsf{Type} \\
 & \mathcal{K}[\![(x :_{r} A) \otimes B]\!] \quad\; = \mathcal{K}[\![B]\!], \text{ if } A \in \mathsf{Type}, B \in \mathsf{Kind}
\end{array}
$$

Next we define the interpretation of types, which requires the interpretation to be parametric on an interpretation of type variables called a type evaluation. This is necessary to make the interpretation well-founded (first realized by Girard [29]).

Definition 6. *Type valuations*, $\Delta \odot \Gamma \models \varepsilon$, are defined as follows:

$$
\frac{}{\emptyset \odot \emptyset \models \emptyset}\,\text{E}
\qquad
\frac{X \in \mathcal{K}[\![A]\!] \qquad \Delta \odot \Gamma \models \varepsilon \qquad (\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_1}{(\Delta, \sigma) \odot (\Gamma, x : A) \models \varepsilon[x \mapsto X]}\,\text{Ty}
\qquad
\frac{\Delta \odot \Gamma \models \varepsilon \qquad (\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_0}{(\Delta, \sigma) \odot (\Gamma, x : A) \models \varepsilon}\,\text{Tm}
$$

Type valuations ignore term variables (rule Tm), in fact, the interpretations of both types and kinds ignores them because we are defining sets of terms over types, and thus terms in types do not contribute to the definition of these sets. However as these interpretations define sets of open terms we must carry a graded context around where necessary. Thus, type valuations are with respect to a well-formed graded context $\Delta \odot \Gamma$. We now outline the type interpretation.

Definition 7. For type valuation $\Delta \odot \Gamma \models \varepsilon$ and a type $A \in (\mathsf{Kind} \cup \mathsf{Type} \cup \mathsf{Con})$ with A typable in $\Delta \odot \Gamma$, the interpretation of types $[\![A]\!]_\varepsilon$ is defined inductively. For brevity, we list just a few illustrative cases, including modalities and some function cases; the complete definition is given in Appendix A.2.

$$
\begin{array}{rlr}
[\![\mathsf{Type}_1]\!]_\varepsilon &= \mathsf{SN} & \\
[\![\mathsf{Type}_0]\!]_\varepsilon &= \lambda X \in \mathsf{SAT}.\mathsf{SN} & \\
[\![x]\!]_\varepsilon &= \varepsilon\, x & \text{if } x \in \mathsf{Con} \\
[\![\Box_s A]\!]_\varepsilon &= [\![A]\!]_\varepsilon & \\
[\![\lambda x : A.B]\!]_\varepsilon &= \lambda X \in \mathcal{K}[\![A]\!].[\![B]\!]_{\varepsilon[x \mapsto X]} & \text{if } A \in \mathsf{Kind}, B \in \mathsf{Con} \\
[\![A\, B]\!]_\varepsilon &= [\![A]\!]_\varepsilon([\![B]\!]_\varepsilon) & \text{if } B \in \mathsf{Con} \\
[\![(x :_{(s,r)} A) \to B]\!]_\varepsilon &= \lambda X \in \mathcal{K}[\![A]\!] \to \mathcal{K}[\![B]\!].\bigcap_{Y \in \mathcal{K}[\![A]\!]}([\![A]\!]_\varepsilon\, Y \to [\![B]\!]_{\varepsilon[x \mapsto Y]}\,(X\,(Y))) & \\
 & & \text{if } A, B \in \mathsf{Kind}
\end{array}
$$

Grades play no role in the reduction relation for GRTT, and hence, our interpretation erases graded modalities and their introductory and elimination forms (translated into substitutions). In fact, the above interpretation can be seen as a translation of $\text{GRTT}^{\{0,1\}}$ into non-substructural set theory; there is no data-usage tracking in the image of the interpretation. Tensors are translated into Cartesian products whose eliminators are translated into substitutions similarly to graded modalities. All terms however remain well-typed through the interpretation.

The interpretation of terms corresponds to term valuations that are used to close the term before interpreting it into the interpretation of its type.

Definition 8. *Valid term valuations*, $\Delta \odot \Gamma \models_\varepsilon \rho$, are defined as follows:

$$\frac{}{\emptyset \odot \emptyset \models_\emptyset \emptyset}\textsc{E} \quad \frac{\begin{array}{l} t \in ([\![A]\!]_\varepsilon)\,(\varepsilon\, x) \\ \Delta \odot \Gamma \models_\varepsilon \rho \\ (\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_1 \end{array}}{(\Delta, \sigma) \odot \Gamma, x : A \models_\varepsilon \rho[x \mapsto t]}\textsc{Ty} \quad \frac{\begin{array}{l} t \in [\![A]\!]_\varepsilon \\ \Delta \odot \Gamma \models_\varepsilon \rho \\ (\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_0 \end{array}}{(\Delta, \sigma) \odot \Gamma, x : A \models_\varepsilon \rho[x \mapsto t]}\textsc{Tm}$$

We interpret terms as substitutions, but graded modalities must be erased and their elimination forms converted into substitutions (and similarly for the eliminator for tensor products).

Definition 9. Suppose $\Delta \odot \Gamma \models_\varepsilon \rho$. Then the *interpretation of a term t* typable in $\Delta \odot \Gamma$ is $(\!|t|\!)_\rho = \rho\, t$, but where all let-expressions are translated into substitutions, and all graded modalities are erased.

Finally, we prove our main result using semantic typing which will imply strong normalization. Suppose $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$, then:

Definition 10. *Semantic typing*, $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \models t : A$, is defined as follows:

1. If $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_1$, then for every $\Delta \odot \Gamma \models_\varepsilon \rho$, $(\!|t|\!)_\rho \in [\![A]\!]_\varepsilon\,([\![t]\!]_\varepsilon)$.
2. If $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_0$, then for every $\Delta \odot \Gamma \models_\varepsilon \rho$, $(\!|t|\!)_\rho \in [\![A]\!]_\varepsilon$.

Theorem 3 (Soundness for Semantic Typing). $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \models t : A$.

Corollary 1 (Strong Normalization). *We have* $t \in \mathsf{SN}$.

5 Implementation

Our implementation **Gerty** is based on a bidirectionalised version of the typing rules here, somewhat following traditional schemes of bidirectional typing [19,20] but with grading (similar to Granule [45] but adapted considerably for the dependent setting). We briefly outline the implementation scheme and highlight a few key points, rules, and examples. We use this implementation to explore further applications of GRTT, namely optimising type checking algorithms.

Bidirectional typing splits declarative typing rules into *check* and *infer* modes. Furthermore, bidirectional GRTT rules split the grading context (left of $\odot$) into *input* and *output* contexts where $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$ is implemented via:

$$(\textit{check})\ \ \Delta; \Gamma \vdash t \Leftarrow A; \sigma_1; \sigma_2 \quad \textit{or} \quad (\textit{infer})\ \ \Delta; \Gamma \vdash t \Rightarrow A; \sigma_1; \sigma_2$$

where $\Leftarrow$ rules *check* that t has type A and $\Rightarrow$ rules *infer* (calculate) that t has type A. In both judgments, the context grading Δ and context Γ left of

$\vdash$ are inputs whereas the grade vectors σ_1 and σ_2 to the right of A are outputs. This input-output context approach resembles that employed in linear type checking [5,32,61]. Rather than following a “left over” scheme as in these works (where the output context explains what resources are left), the output grades here explain what has been used according to the analysis of grading (‘adding up’ rather than ‘taking away’).

For example, the following is the *infer* rule for function elimination:

$$\frac{\begin{array}{l}\Delta; \Gamma \vdash t_1 \Rightarrow (x :_{(s,r)} A) \to B; \sigma_2; \sigma_{13} \\ \Delta; \Gamma \vdash t_2 \Leftarrow A; \sigma_4; \sigma_1 \\ \Delta, \sigma_1; \Gamma, x : A \vdash B \Rightarrow \mathsf{Type}_l; \sigma_3, r; \mathbf{0} \qquad \sigma_{13} = \sigma_1 + \sigma_3\end{array}}{\Delta; \Gamma \vdash t_1\, t_2 \Rightarrow [t_2/x]B; \sigma_2 + s * \sigma_4; \sigma_3 + r * \sigma_4} \Rightarrow \lambda_e$$

The rule can be read by starting at the input of the conclusion (left of $\vdash$), then reading top down through each premise, to calculate the output grades in the rule’s conclusion. Any concrete value or already-bound variable appearing in the output grades of a premise can be read as causing an equality check in the type checker. The last premise checks that the output subject-type grade σ_{13} from the first premise matches $\sigma_1 + \sigma_3$ (which were calculated by later premises).

In contrast, function introduction is a *check* rule:

$$\frac{\Delta; \Gamma \vdash A \Rightarrow \mathsf{Type}_l; \sigma_1; \mathbf{0} \qquad \Delta, \sigma_1; \Gamma, x : A \vdash t \Leftarrow B; \sigma_2, s; \sigma_3, r}{\Delta; \Gamma \vdash \lambda x.t \Leftarrow (x :_{(s,r)} A) \to B; \sigma_2; \sigma_1 + \sigma_3} \Leftarrow \lambda_i$$

Thus, dependent functions can be checked against type $(x :_{(s,r)} A) \to B$ given input $\Delta; \Gamma$ by first inferring the type of A and checking that its output subject-type grade comprises all zeros $\mathbf{0}$. Then the body of the function t is checked against B under the context $\Delta, \sigma_1; \Gamma, x : A$ producing grade vectors σ_2, s' and σ_1, r' where it is checked that $s = s'$ and $r = r'$ (described implicitly in the rule), i.e., the calculated grades match those of the binder.

The implementation anticipates some further work for GRTT: the potential for grades which are first-class terms, for which we anticipate complex equations on grades. For grade equality, **Gerty** has two modes: one which normalises terms and then compares for syntactic equality, and the other which discharges constraints via an off-the-shelf SMT solver (we use Z3 [17]). We discuss briefly some performance implications in the next section.

Using Grades to Optimise Type Checking Abel posited that a dependent theory with quantitative resource tracking at the type level could leverage linearity-like optimisations in type checking [2]. Our implementation provides a research vehicle for exploring this idea; we consider one possible optimisation here.

Key to dependent type checking is the substitution of terms into types in elimination forms (i.e., application, tensor elimination). However, in a quantitative semiring setting, if a variable has 0 subject-type grade, then we know it is irrelevant to type formation (it is not semantically depended upon, i.e., during normalisation). Subsequently, substitutions into a 0-graded variable can be elided (or allocations to a closure environment can be avoided). We implemented this optimisation in **Gerty** when inferring the type of an application for $t_1\, t_2$ (rule $\Rightarrow \lambda_e$ above), where the type of t_1 is inferred as $(x :_{(s,0)} A) \to B$. For a quantitative semiring we know that x irrelevant in B, thus we need not perform the substitution $[t_2/x]B$ when type checking the application.

We evaluate this on simple **Gerty** programs of an n-ary "fanout" combinator implemented via an n-ary application combinator, e.g., for arity 3:

```
app3 : (a : (0, 6) Type 0) -> (b : (0, 2) Type 0)
-> (x0 : (1, 0) a) -> (x1 : (1, 0) a) -> (x2 : (1, 0) a)
-> (f:(1, 0) ((y0:(1,0) a) -> (y1:(1,0) a) -> (y2:(1,0) a) -> b)) -> b
app3 = \a -> \b -> \x0 -> \x1 -> \x2 -> \f -> f x0 x1 x2

fan3 : (a : (0, 4) Type 0) -> (b : (0, 2) Type 0)
-> (f : (1,0) ((z0 : (1,0) a) -> (z1 : (1,0) a) -> (z2 : (1,0) a) -> b))
-> (x : (3, 0) a) -> b
fan3 = \a -> \b -> \f -> \x -> app3 a b x x x f
```

Note that `fan3` uses its parameter `x` three times (hence the grade 3) which then incurs substitutions into the type of `app3` during type checking, but each such substitution is redundant since the type does not depend on these parameters, as reflected by the 0 subject-type grades.

To evaluate the optimisation and SMT solving vs. normalisation-based equality, we ran **Gerty** on the fan out program for arities from 3 to 8, with and without the optimisation and under the two equality approaches.
Table 1 gives the results. For grade equality by normalisation, the optimisation has a positive effect on speedup, getting increasingly significant (up to 38%) as the overall cost increases. For SMT-based grade equality, the optimisation causes some slow down for arity 4 and 5 (and just breaking even for arity 3). This is because working out whether the optimisation can be applied requires checking whether grades are equal to 0, which incurs extra SMT solver calls. Eventually, this cost is outweighed by the time saved by reducing substitutions. Since the grades here are all relatively simple, it is usually more efficient for the type checker to normalise and compare terms rather than compiling to SMT and starting up the external solver, as seen by longer times for the SMT approach.

The baseline performance here is poor (the implementation is not highly optimised) partly due to the overhead of computing type formation judgments often to accurately account for grading. However, such checks are often recomputed and could be optimised away by memoisation. Nevertheless this experiment gives the evidence that grades can indeed be used to optimise type checking. A thorough investigation of grade-directed optimisations is future work.

6 Discussion

Grading, Coeffects, and Quantitative Types The notion of *coeffects*, describing how a program depends on its context, arose in the literature from two directions:

	Normalisation			**SMT**		
n	Base ms	Optimised ms	Speedup	Base ms	Optimised ms	Speedup
3	45.71 (1.72)	44.08 (1.28)	1.04	77.12 (2.65)	76.91 (2.36)	1.00
4	108.75 (4.09)	89.73 (4.73)	1.21	136.18 (5.23)	162.95 (3.62)	0.84
5	190.57 (8.31)	191.25 (8.13)	1.00	279.49 (15.73)	289.73 (23.30)	0.96
6	552.11 (29.00)	445.26 (23.50)	1.24	680.11 (16.28)	557.08 (13.87)	1.22
7	1821.49 (49.44)	1348.85 (26.37)	1.35	1797.09 (43.53)	1368.45 (20.16)	1.31
8	6059.30 (132.01)	4403.10 (86.57)	1.38	5913.06 (118.83)	4396.90 (59.82)	1.34

Table 1. Performance analysis of grade-based optimisations to type checking. Times in milliseconds to 2 d.p. with the standard error given in brackets. Measurements are the mean of 10 trials (run on a 2.7 Ghz Intel Core, 8Gb of RAM, Z3 4.8.8).

as a dualisation of effect types [47,48] and a generalisation of Bounded Linear Logic to general resource semirings [25,10]. Coeffect systems can capture reuse bounds, information flow security [23], hardware scheduling constraints [25], and sensitivity for differential privacy [16,22]. A coeffect-style approach also enables linear types to be retrofitted to Haskell [8]. A common thread is the annotation of variables in the context with usage information, drawn from a semiring. Our approach generalises this idea to capture type, context, and computational usage.

McBride [43] reconciles linear and dependent types, allowing types to depend on linear values, refined by Atkey [6] as Quantitative Type Theory. QTT employs coeffect-style annotation of each assumption in a context with an element of a resource accounting algebra, with judgments of the form:

$$x_1 \overset{\rho_1}{:} A_1, \dots, x_n \overset{\rho_n}{:} A_n \vdash M \overset{\rho}{:} B$$

where ρ_i, ρ are elements of a semiring, and $\rho = 0$ or $\rho = 1$, respectively denoting a term which can be used in type formation (erased at runtime) or at runtime. Dependent function arrows are of the form $(x \overset{\rho}{:} A) \to B$, where ρ is a semiring element that denotes the computational usage of the parameter.

Variables used for type formation but not computation are annotated by 0. Subsequently, type formation rules are all of the form $0\Gamma \vdash T$, meaning every variable assumption has a 0 annotation. GRTT is similar to QTT, but differs in its more extensive grading to track usage in types, rather than blanketing all type usage with 0. In Atkey's formulation, a term can be promoted to a type if its result and dependency quantities are all 0. A set of rules provide formation of computational type terms, but these are also graded at 0. Subsequently, it is not possible to construct an inhabitant of Type that can be used at runtime. We avoid this shortcoming allowing matching on types. For example, a computation t that inspects a type variable a would be typed as: $(\Delta, \mathbf{0}, \Delta' \mid \sigma_1, 1, \sigma_1' \mid \sigma_2, r, \sigma_2') \odot \Gamma, a : \mathsf{Type}, \Gamma' \vdash t : B$ denoting 1 computational use and r type uses in B.

At first glance, it seems QTT could be encoded into GRTT taking the semiring $\mathcal{R}$ of QTT and parameterising GRTT by the semiring $\mathcal{R} \cup \{\hat{0}\}$ where $\hat{0}$ denotes arbitrary usage in type formation. However, there is impedance between the two systems as QTT always annotates type use with 0. It is not clear how to make this happen in GRTT whilst still having non-0 tracking at the computational level, since we use one semiring for both. Exploring an encoding is future work.

Choudhury et al. [13] give a system closely related (but arguably simpler) to QTT called GRAD. One key difference is that rather than annotating type usage with 0, grades are simply ignored in types. This makes for a surprisingly flexible system. In addition, they show that irrelevance is captured by the 0 grade using a heap-based semantics (a result leveraged in Section 3). GRAD however does not have the power of type-grades presented here.

Dependent Types and Modalities Dal Lago and Gaboardi extend PCF with linear and lightweight dependent types [15] (then adapted for differential privacy analysis [22]). They add a natural number type indexed by upper and lower bound terms which index a modality. Combined with linear arrows of the form $[a < I].\sigma \multimap \tau$ these describe functions using the parameter at most I times (where the modality acts as a binder for index variable a which denotes instantiations). Their system is leveraged to give fine-grained cost analyses in the context of Implicit Computational Complexity. Whilst a powerful system, their approach is restricted in terms of dependency, where only a specialised type can depend on specialised natural-number indexed terms (which are non-linear).

Gratzer et al. define a dependently-typed language with a Fitch-style modality [30]. It seems that such an approach could also be generalised to a graded modality, although we have used the natural-deduction style for our graded modality rather than the Fitch-style.

As discussed in Section 1, our approach closely resembles Abel's *resourceful dependent types* [2]. Our work expands on the idea, including tensors and the graded modalities. We considerably developed the associated metatheory, provide an implementation, and study applications.

Further Work One expressive extension is to capture analyses which have an ordering, e.g., grading by a *pre-ordered* semiring, allowing a notion of *approximation*. This would enable analyses such as bounded reuse from Bounded Linear Logic [28], intervals with least- and upper-bounds on use [45], and top-completed semirings, with an ∞-element denoting arbitrary usage as a fall-back. We have made progress into exploring the interaction between approximation and dependent types, and the remainder of this is left as future work.

A powerful extension of GRTT for future work is to allow grades to be first-class terms. Typing rules in GRTT involving grades could be adapted to internalise the elements as first-class terms. We could then, e.g., define the map function over sized vectors, which requires that the parameter function is used exactly the same number of times as the length of the vector:

$$\begin{aligned}&\mathit{map} : (n :_{(0,5)} \mathsf{nat}) \to (a :_{(0,n+1)} \mathsf{Type}) \to (b :_{(0,n+1)} \mathsf{Type}) \to\\ &\qquad (f :_{(n,0)} (x :_{(1,0)} a) \to b) \to (xs :_{(1,0)} \mathsf{Vec}\, n\, a) \to \mathsf{Vec}\, n\, b\end{aligned}$$

This type provides strong guarantees: the only well-typed implementations do the correct thing, up to permutations of the result vector. Without the grading, an implementation could apply f fewer than n times, replicating some of the transformed elements; here we know that f must be applied exactly n-times.

A further appealing possibility for GRTT is to allow the semiring to be defined internally, rather than as a meta-level parameter, leveraging dependent types for proofs of key properties. An implementation could specify what is required for a semiring instance, e.g., a record type capturing the operations and properties of a semiring. The rules of GRTT could then be extended, similarly to the extension to first-class grades, with the provision of the semiring(s) coming from GRTT terms. Thus, anywhere with a grading premise $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash r : \mathcal{R}$ would also require a premise $(\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash \mathcal{R} : \mathsf{Semiring}$. This opens up the ability for programmers and library developers to provide custom modes of resource tracking with their libraries, allowing domain-specific program verification.

Conclusions The paradigm of 'grading' exposes the inherent structure of a type theory, proof theory, or semantics by matching the underlying structure with some algebraic structure augmenting the types. This idea has been employed for reasoning about side effects via graded monads [35], and reasoning about data flow as discussed here by semiring grading. Richer algebras could be employed to capture other aspects, such as *ordered logics* in which the exchange rule can be controlled via grading (existing work has done this via modalities [34]).

We developed the core of grading in the context of dependent-types, treating types and terms equally (as one comes to expect in dependent-type theories). The tracking of data flow in types appears complex since we must account for how variables are used to form types in both the context and in the subject type, making sure not to repeat context formation use. The result however is a powerful system for studying dependencies in type theories, as shown by our ability to study different theories just be specialising grades. Whilst not yet a fully fledged implementation, **Gerty** is a useful test bed for further exploration.

Acknowledgments Orchard is supported by EPSRC grant EP/T013516/1.

References

1. Abadi, M., Banerjee, A., Heintze, N., Riecke, J.: A Core Calculus of Dependency. In: POPL. ACM (1999). https://doi.org/10.1145/292540.292555
2. Abel, A.: Resourceful Dependent Types. In: 24th International Conference on Types for Proofs and Programs, Abstracts (2018)
3. Abel, A., Bernardy, J.: A unified view of modalities in type systems. Proc. ACM Program. Lang. **4**(ICFP), 90:1–90:28 (2020). https://doi.org/10.1145/3408972
4. Abel, A., Scherer, G.: On irrelevance and algorithmic equality in predicative type theory. Log. Methods Comput. Sci. **8**(1) (2012). https://doi.org/10.2168/LMCS-8(1:29)2012
5. Allais, G.: Typing with Leftovers - A mechanization of Intuitionistic Multiplicative-Additive Linear Logic. In: Abel, A., Forsberg, F.N., Kaposi, A. (eds.) 23rd International Conference on Types for Proofs and Programs (TYPES 2017). Leibniz International Proceedings in Informatics (LIPIcs), vol. 104, pp. 1:1–1:22. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2018). https://doi.org/10.4230/LIPIcs.TYPES.2017.1

6. Atkey, R.: Syntax and Semantics of Quantitative Type Theory. In: Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018. pp. 56–65 (2018). https://doi.org/10.1145/3209108.3209189
7. Atkey, R., Ghani, N., Johann, P.: A relationally parametric model of dependent type theory. In: Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. pp. 503–515 (2014). https://doi.org/10.1145/2535838.2535852
8. Bernardy, J.P., Boespflug, M., Newton, R.R., Peyton Jones, S., Spiwack, A.: Linear Haskell: practical linearity in a higher-order polymorphic language. Proceedings of the ACM on Programming Languages **2**(POPL), 5 (2017). https://doi.org/10.1145/3158093
9. Brady, E., McBride, C., McKinna, J.: Inductive families need not store their indices. In: International Workshop on Types for Proofs and Programs. pp. 115–129. Springer (2003). https://doi.org/10.1007/978-3-540-24849-1_8
10. Brunel, A., Gaboardi, M., Mazza, D., Zdancewic, S.: A Core Quantitative Coeffect Calculus. In: Shao, Z. (ed.) Programming Languages and Systems - 23rd European Symposium on Programming, ESOP 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014, Proceedings. Lecture Notes in Computer Science, vol. 8410, pp. 351–370. Springer (2014). https://doi.org/10.1007/978-3-642-54833-8_19
11. Cardelli, L., Wegner, P.: On understanding types, data abstraction, and polymorphism. ACM Computing Surveys **17**(4), 471–523 (Dec 1985). https://doi.org/10.1145/6041.6042
12. Cervesato, I., Pfenning, F.: A linear logical framework. Information and Computation **179**(1), 19–75 (2002). https://doi.org/10.1109/LICS.1996.561339
13. Choudhury, P., Eades III, H., Eisenberg, R.A., Weirich, S.: A Graded Dependent Type System with a Usage-Aware Semantics. Proc. ACM Program. Lang. **5**(POPL) (Jan 2021). https://doi.org/10.1145/3434331
14. Coquand, T., Huet, G.: The Calculus of Constructions. Ph.D. thesis, INRIA (1986)
15. Dal Lago, U., Gaboardi, M.: Linear dependent types and relative completeness. In: Logic in Computer Science (LICS), 2011 26th Annual IEEE Symposium on. pp. 133–142. IEEE (2011). https://doi.org/10.1109/LICS.2011.22
16. De Amorim, A.A., Gaboardi, M., Gallego Arias, E.J., Hsu, J.: Really Natural Linear Indexed Type Checking. In: Proceedings of the 26nd 2014 International Symposium on Implementation and Application of Functional Languages. p. 5. ACM (2014). https://doi.org/10.1145/2746325.2746335
17. De Moura, L., Bjørner, N.: Z3: An efficient SMT solver. In: International conference on Tools and Algorithms for the Construction and Analysis of Systems. pp. 337–340. Springer (2008). https://doi.org/10.1007/978-3-540-78800-3_24
18. De Vries, E., Plasmeijer, R., Abrahamson, D.M.: Uniqueness typing simplified. In: Symposium on Implementation and Application of Functional Languages. pp. 201–218. Springer (2007). https://doi.org/10.1007/978-3-540-85373-2_12
19. Dunfield, J., Krishnaswami, N.R.: Sound and complete bidirectional typechecking for higher-rank polymorphism with existentials and indexed types. PACMPL **3**(POPL), 9:1–9:28 (2019). https://doi.org/10.1145/3290322
20. Dunfield, J., Pfenning, F.: Tridirectional typechecking. In: Proceedings of the 31st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. pp. 281–292. POPL '04, ACM, New York, NY, USA (2004). https://doi.org/10.1145/964001.964025

21. Eades, H., Stump, A.: Hereditary substitution for Stratified System F. In: International Workshop on Proof-Search in Type Theories, PSTT. vol. 10 (2010)
22. Gaboardi, M., Haeberlen, A., Hsu, J., Narayan, A., Pierce, B.C.: Linear dependent types for differential privacy. In: POPL. pp. 357–370 (2013). https://doi.org/10.1145/2429069.2429113
23. Gaboardi, M., Katsumata, S.y., Orchard, D., Breuvart, F., Uustalu, T.: Combining Effects and Coeffects via Grading. In: Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming. p. 476–489. ICFP 2016, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2951913.2951939
24. Geuvers, H.: A short and flexible proof of strong normalization for the calculus of constructions. In: Dybjer, P., Nordström, B., Smith, J. (eds.) Types for Proofs and Programs. pp. 14–38. Springer Berlin Heidelberg, Berlin, Heidelberg (1995). https://doi.org/10.1007/3-540-60579-7_2
25. Ghica, D.R., Smith, A.I.: Bounded linear types in a resource semiring. In: ESOP. pp. 331–350 (2014). https://doi.org/10.1007/978-3-642-54833-8_18
26. Girard, J.Y.: Une extension de l'interpretation de gödel a l'analyse, et son application a l'elimination des coupures dans l'analyse et la theorie des types. In: Studies in Logic and the Foundations of Mathematics, vol. 63, pp. 63–92. Elsevier (1971)
27. Girard, J.Y.: Linear logic. Theoretical Computer Science **50**, 1–102 (1987)
28. Girard, J.Y., Scedrov, A., Scott, P.J.: Bounded linear logic: a modular approach to polynomial-time computability. Theoretical computer science **97**(1), 1–66 (1992). https://doi.org/10.1016/0304-3975(92)90386-T
29. Girard, J., Taylor, P., Lafont, Y.: Proofs and types, vol. 7. Cambridge University Press Cambridge (1989)
30. Gratzer, D., Sterling, J., Birkedal, L.: Implementing a modal dependent type theory. Proc. ACM Program. Lang. **3**(ICFP), 107:1–107:29 (2019). https://doi.org/10.1145/3341711
31. Henglein, F., Makholm, H., Niss, H.: Effect types and region-based memory management. Advanced Topics in Types and Programming Languages pp. 87–135 (2005)
32. Hodas, J.S.: Logic programming in intutionistic linear logic: Theory, design and implementation. PhD Thesis, University of Pennsylvania, Department of Computer and Information Science (1994)
33. Hofmann, M.: Syntax and semantics of dependent types. Semantics and logics of computation **14**, 79 (1997)
34. Jiang, J., Eades III, H., de Paiva, V.: On the lambek calculus with an exchange modality. In: Ehrhard, T., Fernández, M., de Paiva, V., de Falco, L.T. (eds.) Proceedings Joint International Workshop on Linearity & Trends in Linear Logic and Applications, Linearity-TLLA@FLoC 2018, Oxford, UK, 7-8 July 2018. EPTCS, vol. 292, pp. 43–89 (2018). https://doi.org/10.4204/EPTCS.292.4
35. Katsumata, S.: Parametric effect monads and semantics of effect systems. In: Proceedings of POPL. pp. 633–646. ACM (2014). https://doi.org/10.1145/2535838.2535846
36. Krishnaswami, N.R., Dreyer, D.: Internalizing relational parametricity in the extensional calculus of constructions. In: Computer Science Logic 2013 (CSL 2013). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2013). https://doi.org/10.4230/LIPIcs.CSL.2013.432
37. Krishnaswami, N.R., Pradic, P., Benton, N.: Integrating linear and dependent types. In: ACM SIGPLAN Notices. vol. 50, pp. 17–30. ACM (2015)

38. Leivant, D.: Finitely stratified polymorphism. Information and Computation **93**(1), 93–113 (1991). https://doi.org/10.1016/0890-5401(91)90053-5
39. Luo, Z., Zhang, Y.: A linear dependent type theory. Types for Proofs and Programs (TYPES 2016), Novi Sad (2016)
40. Martin-Löf, P.: An Intuitionistic Theory of Types: Predicative Part. In: Rose, H.E., Shepherdson, J.C. (eds.) Studies in Logic and the Foundations of Mathematics, Logic Colloquium '73, vol. 80, pp. 73–118. Elsevier (Jan 1975). https://doi.org/10.1016/S0049-237X(08)71945-1
41. Martin-Löf, P.: Intuitionistic Type Theory (Jun 1980)
42. Martin-Löf, P.: Constructive Mathematics and Computer Programming. In: Cohen, L.J., Łoś, J., Pfeiffer, H., Podewski, K.P. (eds.) Studies in Logic and the Foundations of Mathematics, Logic, Methodology and Philosophy of Science VI, vol. 104, pp. 153–175. Elsevier (Jan 1982). https://doi.org/10.1016/S0049-237X(09)70189-2
43. McBride, C.: I Got Plenty o' Nuttin', pp. 207–233. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-30936-1_12
44. Nuyts, A., Vezzosi, A., Devriese, D.: Parametric quantifiers for dependent type theory. Proceedings of the ACM on Programming Languages **1**(ICFP), 32:1–32:29 (Aug 2017). https://doi.org/10.1145/3110276
45. Orchard, D., Liepelt, V.B., Eades III, H.: Quantitative Program Reasoning with Graded Modal Types. Proc. ACM Program. Lang. **3**(ICFP), 110:1–110:30 (Jul 2019). https://doi.org/10.1145/3341714
46. Palmgren, E.: On universes in type theory. Twenty-five years of constructive type theory **36**, 191–204 (1998)
47. Petricek, T., Orchard, D., Mycroft, A.: Coeffects: Unified Static Analysis of Context-Dependence. In: ICALP (2). pp. 385–397 (2013). https://doi.org/10.1007/978-3-642-39212-2_35
48. Petricek, T., Orchard, D., Mycroft, A.: Coeffects: A calculus of context-dependent computation. In: Proceedings of the 19th ACM SIGPLAN International Conference on Functional Programming. pp. 123–135. ICFP '14, ACM (2014). https://doi.org/10.1145/2628136.2628160
49. Pfenning, F.: Intensionality, extensionality, and proof irrelevance in modal type theory. In: Proceedings 16th Annual IEEE Symposium on Logic in Computer Science. pp. 221–230. IEEE (2001). https://doi.org/10.1109/LICS.2001.932499
50. Reed, J.: Extending higher-order unification to support proof irrelevance. In: International Conference on Theorem Proving in Higher Order Logics. pp. 238–252. Springer (2003). https://doi.org/10.1007/10930755_16
51. Reynolds, J.C.: Towards a theory of type structure. In: Programming Symposium. pp. 408–425. Springer (1974). https://doi.org/10.1007/3-540-06859-7_148
52. Reynolds, J.C.: Types, abstraction and parametric polymorphism. In: Information Processing 83, Proceedings of the IFIP 9th World Computer Congres. pp. 513–523 (1983)
53. Shi, R., Xi, H.: A linear type system for multicore programming in ATS. Science of Computer Programming **78**(8), 1176–1192 (2013). https://doi.org/10.1016/j.scico.2012.09.005
54. Tofte, M., Talpin, J.P.: Region-based memory management. Information and computation **132**(2), 109–176 (1997). https://doi.org/10.1006/inco.1996.2613
55. Tov, J.A., Pucella, R.: Practical affine types. In: Proceedings of the 38th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2011, Austin, TX, USA, January 26-28, 2011. pp. 447–458 (2011). https://doi.org/10.1145/1926385.1926436

56. Wadler, P.: Theorems for free! In: Proceedings of the fourth international conference on Functional programming languages and computer architecture. pp. 347–359 (1989). https://doi.org/10.1145/99370.99404
57. Wadler, P.: Linear Types Can Change the World! In: Programming Concepts and Methods. North (1990)
58. Wadler, P.: Propositions as types. Communications of the ACM **58**(12), 75–84 (2015). https://doi.org/10.1145/2699407
59. Walker, D.: Substructural type systems. Advanced Topics in Types and Programming Languages pp. 3–44 (2005)
60. Wood, J., Atkey, R.: A Linear Algebra Approach to Linear Metatheory. CoRR **abs/2005.02247** (2020), https://arxiv.org/abs/2005.02247
61. Zalakain, U., Dardha, O.: Pi with leftovers: a mechanisation in Agda. arXiv preprint arXiv:2005.05902 (2020)

A Appendix

A.1 Full Typing Rules for Grtt

$$\frac{}{\emptyset \odot \emptyset \vdash} \text{WF}\emptyset \qquad \frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l}{\Delta, \sigma \odot \Gamma, x : A \vdash} \text{WFEXT}$$

Fig. 1. Well-formed contexts for GRTT

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A = B : \mathsf{Type}_l}{(\Delta \mid \sigma) \odot \Gamma \vdash A \leq B} \leq_{\text{EQ}}$$

$$\frac{(\Delta \mid \sigma) \odot \Gamma \vdash A \leq B \quad (\Delta \mid \sigma) \odot \Gamma \vdash B \leq C}{(\Delta \mid \sigma) \odot \Gamma \vdash A \leq C} \leq_{\text{TRANS}}$$

$$\frac{\Delta \odot \Gamma \vdash \quad l \leq l'}{(\Delta \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_l \leq \mathsf{Type}_{l'}} \leq_{\text{TYPE}}$$

$$\frac{\begin{array}{l}(\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_1) \odot \Gamma \vdash A' \leq A \quad (\Delta, \sigma_1 \mid \sigma_2, r) \odot \Gamma, x : A' \vdash B \leq B'\end{array}}{(\Delta \mid \sigma_1 + \sigma_2) \odot \Gamma \vdash (x :_{(s,r)} A) \to B \leq (x :_{(s,r)} A') \to B'} \leq_{\to}$$

$$\frac{(\Delta, \sigma_1 \mid \sigma_2, r) \odot \Gamma, x : A \vdash B \leq B'}{(\Delta \mid \sigma_1 + \sigma_2) \odot \Gamma \vdash (x :_r A) \otimes B \leq (x :_r A) \otimes B'} \leq_{\otimes}$$

$$\frac{(\Delta \mid \sigma) \odot \Gamma \vdash A \leq A'}{(\Delta \mid \sigma) \odot \Gamma \vdash \Box_s A \leq \Box_s A'} \leq_{\Box}$$

Fig. 2. Subtyping for GRTT

$$\frac{\Delta \odot \Gamma \vdash}{(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}}\ \mathsf{Type} \qquad \frac{(\Delta_1, \sigma, \Delta_2) \odot \Gamma_1, x : A, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \sigma, \Delta_2 \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \sigma, 0, \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash x : A}\ \textsc{Var}$$

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : \mathsf{Type}_{l_1 \sqcup l_2}}\ \to$$

$$\frac{(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \quad (\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t : B}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t : (x :_{(s,r)} A) \to B}\ \lambda_i$$

$$\frac{\begin{array}{l}(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t_1 : (x :_{(s,r)} A) \to B \quad (\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : A\end{array}}{(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash t_1\, t_2 : [t_2/x]B}\ \lambda_e$$

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_r A) \otimes B : \mathsf{Type}_l}\ \otimes$$

$$\frac{\begin{array}{l}(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma \vdash t_1 : A \qquad (\Delta \mid \sigma_4 \mid \sigma_3 + r * \sigma_2) \odot \Gamma \vdash t_2 : [t_1/x]B\end{array}}{(\Delta \mid \sigma_2 + \sigma_4 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash (t_1, t_2) : (x :_r A) \otimes B}\ \otimes_i$$

$$\frac{\begin{array}{l}(\Delta \mid \sigma_3 \mid \sigma_1 + \sigma_2) \odot \Gamma \vdash t_1 : (x :_r A) \otimes B \\ (\Delta, (\sigma_1 + \sigma_2) \mid \sigma_5, r' \mid \mathbf{0}) \odot \Gamma, z : (x :_r A) \otimes B \vdash C : \mathsf{Type}_l \\ (\Delta, \sigma_1, (\sigma_2, r) \mid \sigma_4, s, s \mid \sigma_5, r', r') \odot \Gamma, x : A, y : B \vdash t_2 : [(x, y)/z]C\end{array}}{(\Delta \mid \sigma_4 + s * \sigma_3 \mid \sigma_5 + r' * \sigma_3) \odot \Gamma \vdash \mathsf{let}\,(x, y) = t_1 \,\mathsf{in}\, t_2 : [t_1/z]C}\ \otimes_e$$

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A : \mathsf{Type}_l}\ \Box \qquad \frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A}{(\Delta \mid s * \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box t : \Box_s A}\ \Box_i$$

$$\frac{\begin{array}{l}(\Delta, \sigma_2 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, z : \Box_s A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : \Box_s A \quad (\Delta, \sigma_2 \mid \sigma_3, s \mid \sigma_4, (s * r)) \odot \Gamma, x : A \vdash t_2 : [\Box x/z]B\end{array}}{(\Delta \mid \sigma_1 + \sigma_3 \mid \sigma_4 + r * \sigma_1) \odot \Gamma \vdash \mathsf{let}\,\Box x = t_1 \,\mathsf{in}\, t_2 : [t_1/z]B}\ \Box_e$$

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash A \leq B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : B}\ \textsc{Conv}$$

Fig. 3. Typing for GRTT

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t = t : A}\ \textsc{Refl} \qquad \frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A \quad (\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_2 = t_3 : A}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_3 : A}\ \textsc{Trans}$$

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_2 = t_1 : A}\ \textsc{Sym} \qquad \frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash A \leq B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : B}\ \textsc{ConvTy}$$

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A = C : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B = D : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B = (x :_{(s,r)} C) \to D : \mathsf{Type}_{l_1 \sqcup l_2}}\ \textsc{Eq}_{\to}$$

$$\frac{(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t_1 : B \quad (\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : A}{(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash (\lambda x.t_1)\, t_2 = [t_2/x]t_1 : [t_2/x]B}\ \textsc{Eq}_{\to_c}$$

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : (x :_{(s,r)} A) \to B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t = \lambda x.(t\, x) : (x :_{(s,r)} A) \to B}\ \textsc{Eq}_{\to_u}$$

$$\frac{(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t_1 = t_2 : B}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t_1 = \lambda x.t_2 : (x :_{(s,r)} A) \to B}\ \textsc{Eq}_{\to_i}$$

$$\frac{\begin{array}{l}(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t_1 = t_2 : (x :_{(s,r)} A) \to B \quad (\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_3 = t_4 : A\end{array}}{(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash t_1\, t_3 = t_2\, t_4 : [t_3/x]B}\ \textsc{Eq}_{\to_e}$$

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A = C : \mathsf{Type}_l \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B = D : \mathsf{Type}_l}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_r A) \otimes B = (x :_s C) \otimes D : \mathsf{Type}_l}\ \textsc{Eq}_{\otimes}$$

$$\frac{\begin{array}{l}(\Delta, (\sigma_1 + \sigma_2) \mid \sigma_5, r' \mid \mathbf{0}) \odot \Gamma, z : (x :_r A) \otimes B \vdash C : \mathsf{Type}_l \\ (\Delta \mid \sigma_3 \mid \sigma_1) \odot \Gamma \vdash t_1 : A \quad (\Delta \mid \sigma_6 \mid \sigma_2 + r * \sigma_3) \odot \Gamma \vdash t_2 : [t_1/x]B \\ (\Delta, \sigma_1, (\sigma_2, r) \mid \sigma_4, s, s \mid \sigma_5, r', r') \odot \Gamma, x : A, y : B \vdash t_3 : [(x,y)/z]C\end{array}}{(\Delta \mid \sigma_4 + s * (\sigma_3 + \sigma_6) \mid \sigma_5 + r' * (\sigma_3 + \sigma_6)) \odot \Gamma \vdash \mathsf{let}\,(x,y) = (t_1,t_2)\,\mathsf{in}\,t_3 = [t_1,t_2/x,y]t_3 : [(t_1,t_2)/z]C}\ \textsc{Eq}_{\otimes_c}$$

$$\frac{\begin{array}{l}(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma \vdash t_1 = t_2 : A \quad (\Delta \mid \sigma_4 \mid \sigma_3 + r * \sigma_2) \odot \Gamma \vdash t_3 = t_4 : [t_1/x]B\end{array}}{(\Delta \mid \sigma_2 + \sigma_4 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash (t_1,t_2) = (t_3,t_4) : (x :_r A) \otimes B}\ \textsc{Eq}_{\otimes_i}$$

$$\frac{\begin{array}{l}(\Delta \mid \sigma_3 \mid \sigma_1 + \sigma_2) \odot \Gamma \vdash t_1 = t_2 : (x :_r A) \otimes B \\ (\Delta, (\sigma_1 + \sigma_2) \mid \sigma_5, r' \mid \mathbf{0}) \odot \Gamma, z : (x :_r A) \otimes B \vdash C : \mathsf{Type}_l \\ (\Delta, \sigma_1, (\sigma_2, r) \mid \sigma_4, s, s \mid \sigma_5, r', r') \odot \Gamma, x : A, y : B \vdash t_3 = t_4 : [(x,y)/z]C\end{array}}{(\Delta \mid \sigma_4 + s * \sigma_3 \mid \sigma_5 + r' * \sigma_3) \odot \Gamma \vdash (\mathsf{let}\,(x,y) = t_1\,\mathsf{in}\,t_3) = (\mathsf{let}\,(x,y) = t_2\,\mathsf{in}\,t_4) : [t_1/z]C}\ \textsc{Eq}_{\otimes_e}$$

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : (x :_r A) \otimes B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t = (\mathsf{let}\,(x,y) = t\,\mathsf{in}\,(x,y)) : (x :_r A) \otimes B}\ \textsc{Eq}_{\otimes_u}$$

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A = B : \mathsf{Type}_l}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A = \Box_s B : \mathsf{Type}_l}\ \textsc{Eq}_{\Box} \qquad \frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A}{(\Delta \mid s * \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box t_1 = \Box t_2 : \Box_s A}\ \textsc{Eq}_{\Box_i}$$

$$\frac{\begin{array}{l}(\Delta, \sigma_2 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, z : \Box_s A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : A \qquad (\Delta, \sigma_2 \mid \sigma_3, s \mid \sigma_4, (s * r)) \odot \Gamma, x : A \vdash t_2 : [\Box x/z]B\end{array}}{(\Delta \mid \sigma_3 + s * \sigma_1 \mid \sigma_4 + s * r * \sigma_1) \odot \Gamma \vdash (\mathsf{let}\,\Box x = \Box t_1\,\mathsf{in}\,t_2) = [t_1/x]t_2 : [\Box t_1/z]B}\ \textsc{Eq}_{\Box_c}$$

$$\frac{\begin{array}{l}(\Delta, \sigma_2 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, z : \Box_s A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : \Box_s A \\ (\Delta, \sigma_2 \mid \sigma_3, s \mid \sigma_4, (s * r)) \odot \Gamma, x : A \vdash t_3 = t_4 : [\Box x/z]B\end{array}}{(\Delta \mid \sigma_1 + \sigma_3 \mid \sigma_4 + r * \sigma_1) \odot \Gamma \vdash (\mathsf{let}\,\Box x = t_1\,\mathsf{in}\,t_3) = (\mathsf{let}\,\Box x = t_2\,\mathsf{in}\,t_4) : [t_1/z]B}\ \textsc{Eq}_{\Box_e}$$

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : \Box_s A}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t = (\mathsf{let}\,\Box x = t\,\mathsf{in}\,\Box x) : \Box_s A}\ \textsc{Eq}_{\Box_u}$$

Fig. 4. Term equality for Grtt

A.2 Supplement

Specification and Analysis of Graded Modal Dependent Type Theory (GRTT)

Contents

1 Typing judgment overview

Typing judgments in GRTT have the form $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$, which may be written as $\begin{pmatrix}\Delta\\\sigma_1\\\sigma_2\end{pmatrix} \odot \Gamma \vdash t : A$, should it aid readability.

σ_1 and σ_2 are *grade vectors*, which are vectors of grades which describe subject (t) and subject type (A) use, respectively. Δ is a *context grade vector*, which is a vector of grade vectors, accounting for usage in the typing of assumptions in the context (Γ).

Throughout the theory, we implicitly assume that for any judgment $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \mathcal{J}$, we have $|\Delta| = |\sigma_1| = |\sigma_2| = |\Gamma|$ (i.e., that sizes align), and that for every vector σ in Δ, the size of σ is the same as its index in Δ. We assume these sizing requirements for all judgments (even those without grade vectors), not just typing ones.

The notation $\mathbf{0}$ is used to denote a vector consisting entirely of 0 grades, to whichever size would be necessary to satisfy the above sizing conditions. We may write $\mathbf{0}^{\pi}$ to specify a size (π), if we feel that it aids understanding.

Proofs over multiple judgments When lemmas should hold for multiple forms of judgment, we may use the syntax $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \mathcal{J}$ (where $\mathcal{J}$ ranges over judgments), optionally restricting $\mathcal{J}$ to a subset of judgments, with $\mathcal{J}$ by default ranging over the following forms of judgment:

- $\Delta \odot \Gamma \vdash$;
- $(\Delta \mid \sigma) \odot \Gamma \vdash A \leq B$;
- $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : A$;
- $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A$

Thus, for example, a lemma stating "if $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \mathcal{J}$ then $\Delta \odot \Gamma \vdash$", this would mean all of the following hold:

- if $\Delta \odot \Gamma \vdash$ then $\Delta \odot \Gamma \vdash$;
- if $(\Delta \mid \sigma) \odot \Gamma \vdash A \leq B$ then $\Delta \odot \Gamma \vdash$;
- if $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : A$ then $\Delta \odot \Gamma \vdash$;
- if $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A$ then $\Delta \odot \Gamma \vdash$

Note that repeated uses of $\mathcal{J}$ restrict the form of resulting judgments accordingly, and term operations map over the judgment, thus a lemma "if $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \mathcal{J}$ then $(\Delta' \mid \sigma_1' \mid \sigma_2') \odot \Gamma' \vdash [t/x]\mathcal{J}$" would mean all of the following hold:

- if $\Delta \odot \Gamma \vdash$ then $\Delta' \odot \Gamma' \vdash$;
- if $(\Delta \mid \sigma) \odot \Gamma \vdash A \leq B$ then $(\Delta' \mid \sigma') \odot \Gamma' \vdash [t/x]A \leq [t/x]B$;
- if $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : A$ then $(\Delta' \mid \sigma_1' \mid \sigma_2') \odot \Gamma' \vdash [t/x]t_1 : [t/x]A$;
- if $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A$ then $(\Delta' \mid \sigma_1' \mid \sigma_2') \odot \Gamma' \vdash [t/x]t_1 = [t/x]t_2 : [t/x]A$

Operations on terms lift naturally to operations on judgments. For example:

Definition 1.1 (Subsntitution for multiple judgments)**.** *We write $[t/x]\mathcal{J}$ to mean the substitution of t for x in each component of $\mathcal{J}$. For example, if $\mathcal{J}$ is $t_1 = t_2 : A$, then $[t/x]\mathcal{J}$ is $[t/x]t_1 = [t/x]t_2 : [t/x]A$.*

Definition 1.2 (Free-variable non-membership over multiple judgments)**.** *We write $x \notin \mathsf{FV}(\mathcal{J})$ to mean x is not in* any *of the components of $\mathcal{J}$. E.g., if $\mathcal{J}$ is $t_1 = t_2 : A$, then $x \notin \mathsf{FV}(\mathcal{J})$ is $x \notin (\mathsf{FV}(t_1) \cup \mathsf{FV}(t_2) \cup \mathsf{FV}(A))$.*

2 Definitions

Definition 2.1 (Well-formed contexts). *We say that a context* Γ *is* well-formed *under a context grade vector* Δ $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma$ $\Delta \odot \Gamma \vdash$*), if and only if for every* $\Delta_1, \sigma, \Delta_2, \Gamma_1, x, B,$ *and* Γ_2 *satisfying the equations* $\Delta = (\Delta_1, \sigma, \Delta_2)$*;* $\Gamma = (\Gamma_1, x : B, \Gamma_2)$*; and* $|\Delta_1| = |\Gamma_1|$*; we have* $(\Delta_1 \mid \sigma \mid \boldsymbol{0}) \odot \Gamma_1 \vdash B : \mathsf{Type}_l$ *for some level* l*.*

Definition 2.2 (Move on grade vectors). $\mathsf{mv}(\pi_1; \pi_2; \sigma)$ *moves the element at index* π_1 *to index* π_2 *in* σ *(pushing back elements as necessary). As a special case of this, we define* $\mathsf{exch}(\pi; \sigma) = \mathsf{mv}((\pi + 1); \pi; \sigma)$*.*

Definition 2.3 (Move on context grade vectors). *The operation* $\mathsf{mv}(\pi_1; \pi_2; \Delta)$ *is* $\mathsf{mv}(\pi_1; \pi_2; \sigma)$ *for each* σ *in* Δ*. As a special case of this, we define* $\mathsf{exch}(\pi; \Delta) = \mathsf{mv}((\pi + 1); \pi; \Delta)$*.*

Definition 2.4 (Contraction on context grade vectors). $\mathsf{contr}(\pi; \Delta)$ *combines the elements at index* π *and* $\pi{+}1$ *for each grade vector in* Δ*, via addition. This is defined as* $\mathsf{contr}(\pi; \Delta) = \Delta \backslash (\pi{+}1) + (\Delta/(\pi{+}1)) * (\boldsymbol{0}^{\pi}, 1)$*.*

Definition 2.5 (Insert on grade vectors). *The operation* $\mathsf{ins}(\pi; s; \sigma)$ *inserts the element* s *at index* π *in* σ*, such that all elements preceeding index* π *keep their positions, and every element at index* π *or greater in* σ *will be shifted one index later in the new vector.*

Definition 2.6 (Insert on context grade vectors). *The operation* $\mathsf{ins}(\pi; s; \Delta)$ *is* $\mathsf{ins}(\pi; s; \sigma)$ *for each* σ *in* Δ*.*

Definition 2.7 (Choose on grade vectors). *The operation* σ/π *selects the element at index* π *of* σ*.*

Definition 2.8 (Choose on context grade vectors). *The operation* Δ/π *is* σ/π *on each* σ *in* Δ*, producing a new grade vector of size* $|\Delta|$*.*

Definition 2.9 (Discard on grade vectors). *The operation* $\sigma\backslash\pi$ *removes the element at index* π *from* σ*.*

Definition 2.10 (Discard on context grade vectors). *The operation* $\Delta\backslash\pi$ *is* $\sigma\backslash\pi$ *for each* σ *in* Δ*.*

Definition 2.11 (Splash multiplication of grade vectors). *The operation* $\sigma_1 * \sigma_2$ *scales* σ_2 *by each element of* σ_1 *to produce a context grade vector.*

Definition 2.12 (Addition on grade vectors). *The operation* $\sigma_1 + \sigma_2$ *combines the two vectors element-wise using the semiring addition, right-padding the shorter vector with* 0*, to ensure correct sizing.*

Definition 2.13 (Addition on context grade vectors). *The operation* $\Delta_1 + \Delta_2$ *is pointwise addition of the elements of* Δ_1 *and* Δ_2 *in 'as much as possible.' I.e., if the corresponding vectors at a given index are of different sizes, then the shorter vector is treated as if it were right-padded with zeroes in the addition.*

3 List of results

Lemma 3.1 (Strengthening). *If* $(\Delta, \sigma_1, \Delta' \mid \sigma_2, s, \sigma_2' \mid \sigma_3, r, \sigma_3') \odot \Gamma, x : A, \Gamma' \vdash \mathcal{J}$*,* $x \notin \mathsf{FV}(\mathcal{J})$*, and* $x \notin \mathsf{FV}(\Gamma')$*, with* $|\Delta| = |\sigma_2| = |\sigma_3| = |\Gamma|$*, then* $(\Delta, (\Delta'\backslash\pi) \mid \sigma_2, \sigma_2' \mid \sigma_3, \sigma_3') \odot \Gamma, \Gamma' \vdash \mathcal{J}$ *where* $\pi = |\Gamma|$*.*

Lemma 3.2 (Judgments determine vector sizing). *If* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \mathcal{J}$*, then* $|\sigma_1| = |\sigma_2| = |\Delta| = |\Gamma|$ *and for each element* σ *of* Δ*, the size of* σ *is the same as its index.*

Lemma 3.3 (Judgmental contexts are well-formed). *If* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \mathcal{J}$*, then* $\Delta \odot \Gamma \vdash$*.*

Lemma 3.4 (Subcontext well-formedness). *If* $|\Delta_1| = |\Gamma_1|$ *and* $\Delta_1, \Delta_2 \odot \Gamma_1, \Gamma_2 \vdash$*, then* $\Delta_1 \odot \Gamma_1 \vdash$*.*

Lemma 3.5 (Typing an assumption in a judgmental context). *If* $(\Delta_1, \sigma, \Delta_2 \mid \sigma_1 \mid \sigma_2) \odot \Gamma_1, x : A, \Gamma_2 \vdash \mathcal{J}$ *with* $|\Delta_1| = |\Gamma_1|$*, then* $(\Delta_1 \mid \sigma \mid \boldsymbol{0}) \odot \Gamma_1 \vdash A : \mathsf{Type}_l$*, for some level* l*.*

Lemma 3.6 (Typing the type of a term). *Given* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$*, then there exists a level* l *such that* $(\Delta \mid \sigma_2 \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_l$*.*

Lemma 3.7 (Subtyping in context). *If* $(\Delta_1, \sigma_1, \Delta_2 \mid \sigma_2, s, \sigma_3 \mid \sigma_4, r, \sigma_5) \odot \Gamma_1, x : A, \Gamma_2 \vdash \mathcal{J}$ *and* $(\Delta_1 \mid \sigma_1) \odot \Gamma \vdash A' \leq A$*, then* $(\Delta_1, \sigma_1, \Delta_2 \mid \sigma_2, s, \sigma_3 \mid \sigma_4, r, \sigma_5) \odot \Gamma_1, x : A', \Gamma_2 \vdash \mathcal{J}$*.*

3.1 Inversions

Lemma 3.8 (Inversion on arrow typing). *If* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : C$ *then there exist grade vectors* $\hat{\sigma_1}$*,* $\hat{\sigma_1}'$*, and levels* l *and* l'*, such that* $(\Delta \mid \hat{\sigma_1} \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l'}$ *and* $(\Delta, \hat{\sigma_1} \mid \hat{\sigma_1}', r \mid \boldsymbol{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l''}$*, and* $\hat{\sigma_1} + \hat{\sigma_1}' = \sigma_1$*.*

Lemma 3.9 (Inversion on tensor typing). *If* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash (x :_r A) \otimes B : C$ *then there exist grade vectors* $\hat{\sigma_1}$*,* $\hat{\sigma_1}'$*, and levels* l *and* l'*, such that* $(\Delta \mid \hat{\sigma_1} \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l'}$ *and* $(\Delta, \hat{\sigma_1} \mid \hat{\sigma_1}', r \mid \boldsymbol{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l''}$*, and* $\hat{\sigma_1} + \hat{\sigma_1}' = \sigma_1$*.*

Lemma 3.10 (Inversion on box typing). *If* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box_s A : B$*, then* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash A : \mathsf{Type}_l$*, for some level* l*.*

Lemma 3.11 (Function inversion). *If* $(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t : C$*,* $(\Delta \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash C \leq (x :_{(s,r)} A) \to B$*, and* $(\Delta, \sigma_1 \mid \sigma_3, r \mid \boldsymbol{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l$*, then* $(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t : B$*.*

Lemma 3.12 (Pair inversion). *If* $(\Delta \mid \sigma_1 \mid \sigma_2 + \sigma_3) \odot \Gamma \vdash (t_1, t_2) : (x :_r A) \otimes B$ *and* $(\Delta, \sigma_2 \mid \sigma_3, r \mid \boldsymbol{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l$*, then there exist grade vectors* $\hat{\sigma_1}$ *and* $\hat{\sigma_1}'$ *such that* $(\Delta \mid \hat{\sigma_1} \mid \sigma_2) \odot \Gamma \vdash t_1 : A$*, and* $(\Delta \mid \hat{\sigma_1}' \mid \sigma_3 + r * \hat{\sigma_1}) \odot \Gamma \vdash t_2 : [t_1/x]B$*, with* $\hat{\sigma_1} + \hat{\sigma_1}' = \sigma_1$*.*

Lemma 3.13 (Box inversion). *If* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box t : B$ *and* $(\Delta \mid \sigma_2) \odot \Gamma \vdash B \leq \Box_s A$*, then* $(\Delta \mid \hat{\sigma_1} \mid \sigma_2) \odot \Gamma \vdash t : A$ *for some* $\hat{\sigma_1}$ *such that* $s * \hat{\sigma_1} = \sigma_1$*.*

Lemma 3.14 (Arrow subtyping inversion). *If* $(\Delta \mid \sigma_1 + \sigma_2 \mid \boldsymbol{0}) \odot \Gamma \vdash \mathcal{J}_1$ *where* $\mathcal{J}_1$ *is* $(x :_{(s,r)} A) \to B \leq C$ *or* $(x :_{(s,r)} A) \to B = C : D$ *and* $(\Delta \mid \sigma_1 + \sigma_2 \mid \boldsymbol{0}) \odot \Gamma \vdash \mathcal{J}_2$*, where* $\mathcal{J}_2$ *is respectively* $C \leq (y :_{(s',r')} A') \to B'$ *or* $C = (y :_{(s',r')} A') \to B' : E$ *(with* $(\Delta \mid \boldsymbol{0}) \odot \Gamma \vdash D \leq \mathsf{Type}_l$ *and* $(\Delta \mid \boldsymbol{0}) \odot \Gamma \vdash E \leq \mathsf{Type}_l$*), with* $(\Delta, \sigma_1 \mid \sigma_3, r \mid \boldsymbol{0}) \odot \Gamma, y : A' \vdash B' : \mathsf{Type}_{l'}$*, then* $x = y$*,* $s = s'$*,* $r = r'$*, and respectively, based on* $\mathcal{J}_1$*,* $(\Delta \mid \sigma_1) \odot \Gamma \vdash A' \leq A$*, and* $(\Delta, \sigma_1 \mid \sigma_3, r) \odot \Gamma, x : A' \vdash B \leq B'$*, or* $(\Delta \mid \sigma_1 \mid \boldsymbol{0}) \odot \Gamma \vdash A' = A : \mathsf{Type}_{l''}$*, and* $(\Delta, \sigma_1 \mid \sigma_3, r \mid \boldsymbol{0}) \odot \Gamma, x : A' \vdash B = B' : \mathsf{Type}_{l'''}$*.*

Lemma 3.15 (Tensor subtyping inversion). *If* $(\Delta \mid \sigma_1 + \sigma_2 \mid \boldsymbol{0}) \odot \Gamma \vdash \mathcal{J}_1$ *where* $\mathcal{J}_1$ *is* $(x :_r A) \otimes B \leq C$ *or* $(x :_r A) \otimes B = C : D$ *and* $(\Delta \mid \sigma_1 + \sigma_2 \mid \boldsymbol{0}) \odot \Gamma \vdash \mathcal{J}_2$*, where* $\mathcal{J}_2$ *is respectively* $C \leq (y :_{r'} A') \otimes B'$ *or* $C = (y :_{r'} A') \otimes B' : E$ *(with* $(\Delta \mid \boldsymbol{0}) \odot \Gamma \vdash D \leq \mathsf{Type}_l$ *and* $(\Delta \mid \boldsymbol{0}) \odot \Gamma \vdash E \leq \mathsf{Type}_l$*), with* $(\Delta, \sigma_1 \mid \sigma_3, r \mid \boldsymbol{0}) \odot \Gamma, y : A' \vdash B' : \mathsf{Type}_{l'}$*, then* $x = y$*,* $s = s'$*,* $r = r'$*,* $(\Delta \mid \sigma_1 \mid \boldsymbol{0}) \odot \Gamma \vdash A = A' : \mathsf{Type}_{l''''}$*, and respectively, based on* $\mathcal{J}_1$*,* $(\Delta, \sigma_1 \mid \sigma_3, r) \odot \Gamma, x : A' \vdash B \leq B'$*, or* $(\Delta, \sigma_1 \mid \sigma_3, r \mid \boldsymbol{0}) \odot \Gamma, x : A' \vdash B = B' : \mathsf{Type}_{l'''}$*.*

Lemma 3.16 (Box subtyping inversion). *If* $(\Delta \mid \sigma \mid \boldsymbol{0}) \odot \Gamma \vdash \mathcal{J}_1$ *where* $\mathcal{J}_1$ *is* $\Box_s A \leq B$ *or* $\Box_s A = B : D$*, and* $(\Delta \mid \sigma \mid \boldsymbol{0}) \odot \Gamma \vdash \mathcal{J}_2$*, where* $\mathcal{J}_2$ *is respectively* $B \leq \Box_{s'} A'$ *or* $\Box_{s'} A' = B : E$ *(where* $(\Delta \mid \boldsymbol{0}) \odot \Gamma \vdash D \leq \mathsf{Type}_l$ *and* $(\Delta \mid \boldsymbol{0}) \odot \Gamma \vdash E \leq \mathsf{Type}_l$*), then* $s = s'$*, and respectively, based on* $\mathcal{J}_1$*,* $(\Delta \mid \sigma) \odot \Gamma \vdash A \leq A'$*, or* $(\Delta \mid \sigma \mid \boldsymbol{0}) \odot \Gamma \vdash A = A' : \mathsf{Type}_{l'}$*.*

3.2 Vector manipulation

Lemma 3.17 (Factoring choose and discard)**.** *If* $|\sigma_1| = |\sigma_3| = \pi$*, then* $(\Delta_1, (\sigma_1, r, \sigma_2), \Delta_2)\backslash\pi + ((\Delta_1, (\sigma_1, r, \sigma_2), \Delta_2)/\pi) * \sigma_3 = (\Delta_1\backslash\pi + \Delta_1/\pi * \sigma_3), ((\sigma_1 + r * \sigma_3), \sigma_2), (\Delta_2\backslash\pi + \Delta_2/\pi * \sigma_3)$.

Lemma 3.18 (Factoring vector addition)**.** *If* $\hat{\sigma_1} + \hat{\sigma_1}' = \sigma_1$*;* $\hat{s} + \hat{s'} = s$*; and* $\hat{\sigma_3} + \hat{\sigma_3}' = \sigma_3$*, then* $((\hat{\sigma_1} + \hat{s} * \sigma_2), \hat{\sigma_3}) + ((\hat{\sigma_1}' + \hat{s}' * \sigma_2), \hat{\sigma_3}') = (\sigma_1 + s * \sigma_2), \sigma_3$ *for all* σ_2*.*

Lemma 3.19 (Addition across same-sized components)**.** *If* $|\sigma_1| = |\sigma_3|$ *then* $\sigma_1, \sigma_2 + \sigma_3 = (\sigma_1 + \sigma_3), \sigma_2$*.*

Lemma 3.20 (Vector addition across components)**.** *If* $|\sigma_1| = |\sigma_4|$ *and* $|\sigma_2| = |\sigma_5|$*, then* $\sigma_1, \sigma_2, \sigma_3 + \sigma_4, \sigma_5, \sigma_6 = (\sigma_1 + \sigma_4), (\sigma_2 + \sigma_5), (\sigma_3 + \sigma_6)$*.*

Lemma 3.21 (Moving an exchange)**.** *We have* $\mathsf{mv}((\pi_1 + \pi_2); \pi_3; (\mathsf{exch}((\pi_1 + \pi_2); \Delta))) = \mathsf{mv}((\pi_1 + \pi_2 + 1); \pi_3; \Delta)$*.*

3.3 Meta properties

3.3.1 Contraction

Lemma 3.22 (Contraction)**.** *The following rule is admissible:*

$$\frac{\begin{pmatrix} \Delta_1, \sigma_1, (\sigma_1, 0), \Delta_2 \\ \sigma_2, s_1, s_2, \sigma_3 \\ \sigma_4, r_1, r_2, \sigma_5 \end{pmatrix} \odot \Gamma_1, x : A, y : A, \Gamma_2 \vdash \mathcal{J} \quad |\Delta_1| = |\sigma_2| = |\sigma_4| = |\Gamma_1|}{\begin{pmatrix} \Delta_1, \sigma_1, \mathsf{contr}(|\Delta_1|; \Delta_2) \\ \sigma_2, (s_1 + s_2), \sigma_3 \\ \sigma_4, (r_1 + r_2), \sigma_5 \end{pmatrix} \odot \Gamma_1, z : A, [z, z/x, y]\Gamma_2 \vdash [z, z/x, y]\mathcal{J}} \ \textsc{Cont}$$

3.3.2 Exchange

Lemma 3.23 (Exchange)**.** *The following rule is admissible:*

$$\frac{\begin{pmatrix} \Delta_1, \sigma_1, (\sigma_2, 0), \Delta_2 \\ \sigma_3, s_1, s_2, \sigma_4 \\ \sigma_5, r_1, r_2, \sigma_6 \end{pmatrix} \odot \Gamma_1, x : A, y : B, \Gamma_2 \vdash \mathcal{J} \quad x \notin \mathsf{FV}\,(B) \quad |\Delta_1| = |\sigma_3| = |\sigma_5| = |\Gamma_1|}{\begin{pmatrix} \Delta_1, \sigma_2, (\sigma_1, 0), \mathsf{exch}(|\Delta_1|; \Delta_2) \\ \sigma_3, s_2, s_1, \sigma_4 \\ \sigma_5, r_2, r_1, \sigma_6 \end{pmatrix} \odot \Gamma_1, y : B, x : A, \Gamma_2 \vdash \mathcal{J}} \ \textsc{Exchange}$$

Corollary 3.23.1 (Exchange (general))**.** *As a corollary to Lemma 3.23, the following rule is admissible:*

$$\frac{\begin{matrix} \begin{pmatrix} \Delta_1, \Delta_2, (\sigma_1, \mathbf{0}^{|\Delta_2|}), \Delta_3 \\ \sigma_2, \sigma_3, s, \sigma_4 \\ \sigma_5, \sigma_6, r, \sigma_7 \end{pmatrix} \odot \Gamma_1, \Gamma_2, x : A, \Gamma_3 \vdash \mathcal{J} \\ \mathsf{Dom}(\Gamma_2) \cap \mathsf{FV}\,(A) = \emptyset \quad |\sigma_1| = |\Delta_1| = |\sigma_2| = |\sigma_5| = |\Gamma_1| \end{matrix}}{\begin{pmatrix} \Delta_1, \sigma_1, \mathsf{ins}(|\Delta_1|; 0; \Delta_2), \mathsf{mv}(|\Delta_1, \Delta_2|; |\Delta_1|; \Delta_3) \\ \sigma_2, s, \sigma_3, \sigma_4 \\ \sigma_5, r, \sigma_6, \sigma_7 \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2, \Gamma_3 \vdash \mathcal{J}} \ \textsc{ExchangeGen}$$

Where $\mathsf{Dom}(\Gamma)$ *is the domain of* Γ*.*

Corollary 3.23.2 (Exchange from end)**.** *As a corollary to Corollary 3.23.1, the following rule is admissible:*

$$\frac{\begin{pmatrix} \Delta_1, \Delta_2, (\sigma_1, \mathbf{0}^{|\Delta_2|}) \\ \sigma_2, \sigma_3, s \\ \sigma_4, \sigma_5, r \end{pmatrix} \odot \Gamma_1, \Gamma_2, x : A \vdash \mathcal{J} \quad \mathsf{Dom}(\Gamma_2) \cap \mathsf{FV}\,(A) = \emptyset \quad |\Delta_1| = |\sigma_2| = |\sigma_4| = |\Gamma_1|}{\begin{pmatrix} \Delta_1, \sigma_1, \mathsf{ins}(|\Delta_1|; 0; \Delta_2) \\ \sigma_2, s, \sigma_3 \\ \sigma_4, r, \sigma_5 \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2 \vdash \mathcal{J}} \ \textsc{ExchangeEnd}$$

3.3.3 Weakening

Lemma 3.24 (Weakening)**.** *The following rule is admissible:*

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \mathcal{J} \quad (\Delta \mid \sigma_3 \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_l}{(\Delta, \sigma_3 \mid \sigma_1, 0 \mid \sigma_2, 0) \odot \Gamma, x : A \vdash \mathcal{J}}\ \textsc{Weak}$$

Where $\mathcal{J}$ is typing, equality, or subtyping.

Lemma 3.25 (Weakening for well-formed contexts)**.** *The following rule is admissible:*

$$\frac{\Delta_1, \Delta_2 \odot \Gamma_1, \Gamma_2 \vdash \quad (\Delta_1 \mid \sigma_1 \mid \sigma_2) \odot \Gamma_1 \vdash t : A}{(\Delta_1, \Delta_2 \mid \sigma_1, \boldsymbol{0} \mid \sigma_2, \boldsymbol{0}) \odot \Gamma_1, \Gamma_2 \vdash t : A}\ \textsc{WeakWF}$$

Lemma 3.26 (Weakening (general))**.** *The following rule is admissible (for $\mathcal{J}$ is typing, equality, or subtyping):*

$$\frac{(\Delta_1, \Delta_2 \mid \sigma_1, \sigma_1' \mid \sigma_2, \sigma_2') \odot \Gamma_1, \Gamma_2 \vdash \mathcal{J} \quad (\Delta_1 \mid \sigma_3 \mid \boldsymbol{0}) \odot \Gamma_1 \vdash A : \mathsf{Type}_l \quad |\sigma_1| = |\sigma_2| = |\Gamma_1|}{(\Delta_1, \sigma_3, \mathsf{ins}(|\Delta_1|; 0; \Delta_2) \mid \sigma_1, 0, \sigma_1' \mid \sigma_2, 0, \sigma_2') \odot \Gamma_1, x : A, \Gamma_2 \vdash \mathcal{J}}\ \textsc{WeakGen}$$

3.3.4 Substitution

Lemma 3.27 (Substitution for judgments)**.** *If the following premises hold:*

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$
2. $(\Delta, \sigma_1, \Delta' \mid \sigma_3, s, \sigma_4 \mid \sigma_5, r, \sigma_6) \odot \Gamma_1, x : A, \Gamma_2 \vdash \mathcal{J}$
3. $|\sigma_3| = |\sigma_5| = |\Gamma_1|$

Then $\begin{pmatrix} \Delta, (\Delta' \backslash \pi + (\Delta'/\pi) * \sigma_2) \\ (\sigma_3 + s * \sigma_2), \sigma_4 \\ (\sigma_5 + r * \sigma_2), \sigma_6 \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]\mathcal{J}$, *where* $\pi = |\Gamma_1|$.

Lemma 3.28 (Equality through substitution)**.** *If the following premises hold:*

1. $(\Delta_1 \mid \sigma_1 \mid \sigma_2) \odot \Gamma_1 \vdash t_1 = t_2 : A$
2. $(\Delta_1, \sigma_2, \Delta_2 \mid \sigma_3, s, \sigma_4 \mid \sigma_5, r, \sigma_6) \odot \Gamma_1, x : A, \Gamma_2 \vdash t_3 : B$
3. $|\sigma_3| = |\sigma_5| = |\Gamma_1|$

Then $\begin{pmatrix} \Delta_1, (\Delta_2 \backslash \pi + (\Delta_2/\pi) * \sigma_1) \\ (\sigma_3 + s * \sigma_1), \sigma_4 \\ (\sigma_5 + r * \sigma_1), \sigma_6 \end{pmatrix} \odot \Gamma_1, [t_1/x]\Gamma_2 \vdash [t_1/x]t_3 = [t_2/x]t_3 : [t_1/x]B$, *where* $\pi = |\Gamma_1|$.

3.4 Properties of operations

Lemma 3.29 (Properties of insertion)**.** *The following properties hold for any valid insertion (i.e., where for all $i < |\Delta|$, $\pi \leq |\Delta[i]|$, for context grade vectors; and $\pi \leq |\sigma|$, for grade vectors):*

1. *(insPreservesSize)* $|\mathsf{ins}(\pi; R; \Delta)| = |\Delta|$*;*
2. *(insIncSizes) if $\pi \leq |\Delta[i]|$, then* $|(\mathsf{ins}(\pi; R; \Delta))[i]| = |\Delta[i]| + 1$*;*
3. *(insIncSizeGV)* $|\mathsf{ins}(\pi; R; \sigma)| = |\sigma| + 1$*;*
4. *(insCVthenGV)* $\mathsf{ins}(\pi; R; \Delta), \mathsf{ins}(\pi; R; \sigma) = \mathsf{ins}(\pi; R; (\Delta, \sigma))$

3.5 Properties of equality

Lemma 3.30 (Equality is an equivalence relation)**.** *For all, we have:*

- *(reflexivity) if* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$ *then* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t = t : A$*;*
- *(transitivity) if* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A$ *and* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_2 = t_3 : A$*, then* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_3 : A$*;*
- *(symmetry) if* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A$*, then* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_2 = t_1 : A$

Lemma 3.31 (Reduction implies equality)**.** *If* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : A$ *and* $t_1 \rightsquigarrow t_2$*, then* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A$*.*

Lemma 3.32 (Equality inversion)**.** *If* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A$*, then* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : A$ *and* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_2 : A$*.*

Lemma 3.33 (Deriving judgments under equal contexts)**.** *If* $(\Delta_1, \sigma_1, \Delta_2 \mid \sigma_2, s, \sigma_3 \mid \sigma_4, r, \sigma_5) \odot \Gamma_1, x : A, \Gamma_2 \vdash \mathcal{J}$ *and* $(\Delta_1 \mid \sigma_1 \mid \boldsymbol{0}) \odot \Gamma \vdash A = A' : \mathsf{Type}_l$*, then* $(\Delta_1, \sigma_1, \Delta_2 \mid \sigma_2, s, \sigma_3 \mid \sigma_4, r, \sigma_5) \odot \Gamma_1, x : A', \Gamma_2 \vdash \mathcal{J}$*.*

3.6 Properties of subtyping

Lemma 3.34 (Subtyping inversion to typing)**.** *If* $(\Delta \mid \sigma) \odot \Gamma \vdash A \leq B$*, then* $(\Delta \mid \sigma \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_l$ *and* $(\Delta \mid \sigma \mid \boldsymbol{0}) \odot \Gamma \vdash B : \mathsf{Type}_l$ *for some level* l*.*

3.7 Type preservation

Lemma 3.35 (Type preservation)**.** *If* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$ *and* $t \rightsquigarrow t'$*, then* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t' : A$*.*

4 Strong Normalization

Definition 4.1. *Typing can be broken up into the following stages:*

$$
\begin{aligned}
\mathsf{Kind} &:= \{A \mid \exists \Delta, \sigma_1, \Gamma.(\Delta \mid \sigma_1 \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_1\} \\
\mathsf{Type} &:= \{A \mid \exists \Delta, \sigma_1, \Gamma.(\Delta \mid \sigma_1 \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_0\} \\
\mathsf{Const} &:= \{t \mid \exists \Delta, \sigma_1, \sigma_2, \Gamma, A.(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A \wedge \\
&\qquad (\Delta \mid \sigma_2 \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_1\} \\
\mathsf{Term} &:= \{t \mid \exists \Delta, \sigma_1, \sigma_2, \Gamma, A.(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A \wedge \\
&\qquad (\Delta \mid \sigma_2 \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_0\}
\end{aligned}
$$

Lemma 4.2 (Classification)**.** *It is the case that* $\mathsf{Kind} \cap \mathsf{Type} = \emptyset$ *and* $\mathsf{Const} \cap \mathsf{Term} = \emptyset$*.*

Definition 4.3. *The set of base terms* $\mathcal{B}$ *is defined by:*

1. *For* x *a variable,* $x \in \mathcal{B}$*,*
2. $\mathsf{Type}_0, \mathsf{Type}_1 \in \mathcal{B}$*,*
3. *If* $t_1 \in \mathcal{B}$ *and* $t_2 \in \mathsf{SN}$*, then* $(t_1\, t_2) \in \mathcal{B}$*,*
4. *If* $t_2 \in \mathcal{B}$ *and* $t_1 \in \mathsf{SN}$*, then* $(\mathsf{let}\,(x, y) = t_1\,\mathsf{in}\,t_2) \in \mathcal{B}$*,*
5. *If* $t_2 \in \mathcal{B}$ *and* $t_1 \in \mathsf{SN}$*, then* $(\mathsf{let}\,\Box x = t_1\,\mathsf{in}\,t_2) \in \mathcal{B}$*,*

6. *If* $A, B \in \mathsf{SN}$*, then* $((x :_{(r,s)} A) \to B) \in \mathcal{B}$ *for any* $r, s \in R$*,*

7. *If* $A, B \in \mathsf{SN}$*, then* $((x :_{r} A) \otimes B) \in \mathcal{B}$ *for any* $r \in R$*,*

8. *If* $A \in \mathsf{SN}$*, then* $(\Box_r A) \in \mathcal{B}$ *for any* $r \in R$*.*

Definition 4.4. *The* key redex *of a term is defined by:*

1. *If* t *is a redex, then* t *is its own key redex,*

2. *If* t_1 *has key redex* t*, then* $(t_1\, t_2)$ *has key redex* t*,*

3. *If* t_1 *has key redex* t*, then* $(\mathsf{let}\,(x, y) = t_1 \,\mathsf{in}\, t_2)$ *has key redex* t*,*

4. *If* t_1 *has key redex* t*, then* $(\mathsf{let}\,\Box x = t_1 \,\mathsf{in}\, t_2)$ *has key redex* t*.*

The term obtained from t *by contracting its key redex is denoted by* $\mathsf{red_k}\, t$*.*

Lemma 4.5. *The following are both true:*

1. $\mathcal{B} \subseteq \mathsf{SN}$

2. *The key redex of a term is unique and a head redex.*

Definition 4.6. *A set of terms* X *is* saturated *if:*

1. $X \subset \mathsf{SN}$*,*

2. $\mathcal{B} \subset X$*,*

3. *If* $\mathsf{red_k}\, t \in X$ *and* $t \in \mathsf{SN}$*, then* $t \in X$*.*

The collection of saturated sets is denoted by SAT*.*

Lemma 4.7 (SN is saturated)**.** *Every saturated set is non-empty and* SN *is saturated.*

Proof. By definition. ∎

Definition 4.8. *For* $T \in \mathsf{Kind}$*, the kind interpretation,* $\mathcal{K}[\![T]\!]$*, is defined inductively as follows:*

$$
\begin{array}{l}
\mathcal{K}[\![\mathsf{Type}_0]\!] = \mathsf{SAT} \\
\mathcal{K}[\![(x :_{(r,s)} A) \to B]\!] = \{ f \mid f : \mathcal{K}[\![A]\!] \to \mathcal{K}[\![B]\!] \},\ \textit{if } A, B \in \mathsf{Kind} \\
\mathcal{K}[\![(x :_{(r,s)} A) \to B]\!] = \mathcal{K}[\![A]\!],\ \textit{if } A \in \mathsf{Kind}, B \in \mathsf{Type} \\
\mathcal{K}[\![(x :_{(r,s)} A) \to B]\!] = \mathcal{K}[\![B]\!],\ \textit{if } A \in \mathsf{Type}, B \in \mathsf{Kind} \\
\mathcal{K}[\![(x :_{s} A) \otimes B]\!] = \mathcal{K}[\![A]\!] \times \mathcal{K}[\![B]\!],\ \textit{if } A, B \in \mathsf{Kind} \\
\mathcal{K}[\![(x :_{s} A) \otimes B]\!] = \mathcal{K}[\![A]\!],\ \textit{if } A \in \mathsf{Kind}, B \in \mathsf{Type} \\
\mathcal{K}[\![(x :_{s} A) \otimes B]\!] = \mathcal{K}[\![B]\!],\ \textit{if } A \in \mathsf{Type}, B \in \mathsf{Kind} \\
\mathcal{K}[\![\Box_s A]\!] = \mathcal{K}[\![A]\!]
\end{array}
$$

Definition 4.9. Type valuations, $\Delta \odot \Gamma \models \varepsilon$*, are defined as follows:*

$$
\frac{}{\emptyset \odot \emptyset \models \emptyset}\ \textsc{Ep\_Empty}
\qquad
\frac{\begin{array}{l} X \in \mathcal{K}[\![A]\!] \\ \Delta \odot \Gamma \models \varepsilon \quad (\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_1 \end{array}}{(\Delta, \sigma_2) \odot (\Gamma, x : A) \models \varepsilon[x \mapsto X]}\ \textsc{Ep\_ExtTy}
$$

$$
\frac{\Delta \odot \Gamma \models \varepsilon \quad (\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_0}{(\Delta, \sigma_2) \odot (\Gamma, x : A) \models \varepsilon}\ \textsc{Ep\_ExtTm}
$$

Definition 4.10. *Given a type valuation $\Delta \odot \Gamma \models \varepsilon$ and a type $T \in (\mathsf{Kind} \cup \mathsf{Type} \cup \mathsf{Con})$ with T typable in $\Delta \odot \Gamma$, we define the interpretation of types, $[\![T]\!]_\varepsilon$ inductively as follows:*

$$
\begin{array}{rcl}
[\![\mathsf{Type}_1]\!]_\varepsilon & = & \mathsf{SN} \\
[\![\mathsf{Type}_0]\!]_\varepsilon & = & \lambda X \in \mathsf{SAT}.\mathsf{SN} \\
[\![x]\!]_\varepsilon & = & \varepsilon\, x,\ \textit{if } x \in \mathsf{Con} \\
[\![(x :_{(r,s)} A) \to B]\!]_\varepsilon & = & \lambda X \in \mathcal{K}[\![A]\!] \to \mathcal{K}[\![B]\!]. \bigcap_{Y \in \mathcal{K}[\![A]\!]}(([\![A]\!]_\varepsilon)\,(Y) \to ([\![B]\!]_{\varepsilon[x \mapsto Y]})\,(X\,(Y))), \\
 & & \textit{if } A, B \in \mathsf{Kind} \\
[\![(x :_{(r,s)} A) \to B]\!]_\varepsilon & = & \bigcap_{Y \in \mathcal{K}[\![A]\!]}(([\![A]\!]_\varepsilon)\,(Y) \to ([\![B]\!]_{\varepsilon[x \mapsto Y]})), \\
 & & \textit{if } A \in \mathsf{Kind}, B \in \mathsf{Type} \\
[\![(x :_{(r,s)} A) \to B]\!]_\varepsilon & = & \lambda X \in \mathcal{K}[\![B]\!].[\![A]\!]_\varepsilon \to ([\![B]\!]_\varepsilon)\,(X),\ \textit{if } A \in \mathsf{Type}, B \in \mathsf{Kind} \\
[\![(x :_{(r,s)} A) \to B]\!]_\varepsilon & = & [\![A]\!]_\varepsilon \to [\![B]\!]_\varepsilon,\ \textit{if } A, B \in \mathsf{Type} \\
[\![\lambda x : A.B]\!]_\varepsilon & = & \lambda X \in \mathcal{K}[\![A]\!].[\![B]\!]_{\varepsilon[x \mapsto X]},\ \textit{if } A \in \mathsf{Kind}, B \in \mathsf{Con} \\
[\![\lambda x : A.B]\!]_\varepsilon & = & [\![A]\!]_\varepsilon,\ \textit{for any } A \in \mathsf{Kind}, B \in \mathsf{Term} \\
[\![\lambda x : A.B]\!]_\varepsilon & = & [\![B]\!]_\varepsilon,\ \textit{for any } A \in \mathsf{Type}, B \in \mathsf{Con} \\
[\![A\,B]\!]_\varepsilon & = & [\![A]\!]_\varepsilon([\![B]\!]_\varepsilon),\ \textit{if } B \in \mathsf{Con} \\
[\![A\,t]\!]_\varepsilon & = & [\![A]\!]_\varepsilon,\ \textit{if } t \in \mathsf{Term} \\
[\![(x :_s A) \otimes B]\!]_\varepsilon & = & \lambda X \in \mathcal{K}[\![A]\!] \times \mathcal{K}[\![B]\!].([\![A]\!]_\varepsilon)\,(\pi_1\,X) \times ([\![B]\!]_{\varepsilon[x \mapsto \pi_1\,X]})\,(\pi_2\,X), \\
 & & \textit{if } A, B \in \mathsf{Kind} \\
[\![(x :_s A) \otimes B]\!]_\varepsilon & = & \lambda X \in \mathcal{K}[\![B]\!].[\![A]\!]_\varepsilon \times ([\![B]\!]_\varepsilon)\,(X),\ \textit{if } A \in \mathsf{Type}, B \in \mathsf{Kind} \\
[\![(x :_s A) \otimes B]\!]_\varepsilon & = & \lambda X \in \mathcal{K}[\![A]\!].([\![A]\!]_\varepsilon)\,(X) \times [\![B]\!]_{\varepsilon[x \mapsto X]},\ \textit{if } A \in \mathsf{Kind}, B \in \mathsf{Type} \\
[\![(x :_s A) \otimes B]\!]_\varepsilon & = & [\![A]\!]_\varepsilon \times [\![B]\!]_\varepsilon,\ \textit{if } A, B \in \mathsf{Type} \\
[\![(A, B)]\!]_\varepsilon & = & ([\![A]\!]_\varepsilon, [\![B]\!]_\varepsilon) \\
[\![(A, t)]\!]_\varepsilon & = & [\![A]\!]_\varepsilon,\ \textit{if } t \in \mathsf{Term} \\
[\![(t, B)]\!]_\varepsilon & = & [\![B]\!]_\varepsilon,\ \textit{if } t \in \mathsf{Term} \\
[\![\mathsf{let}\,(x : A, y : B) = t\,\mathsf{in}\,C]\!]_\varepsilon & = & [\![C]\!]_{\varepsilon[x \mapsto \pi_1\,[\![t]\!]_\varepsilon, y \mapsto \pi_2\,[\![t]\!]_\varepsilon]},\ \textit{if } A, B \in \mathsf{Kind} \\
[\![\mathsf{let}\,(x : A, y : B) = t\,\mathsf{in}\,C]\!]_\varepsilon & = & [\![C]\!]_{\varepsilon[x \mapsto [\![t]\!]_\varepsilon]},\ \textit{if } A \in \mathsf{Kind}, B \in \mathsf{Type} \\
[\![\mathsf{let}\,(x : A, y : B) = t\,\mathsf{in}\,C]\!]_\varepsilon & = & [\![C]\!]_{\varepsilon[y \mapsto [\![t]\!]_\varepsilon]},\ \textit{if } A \in \mathsf{Type}, B \in \mathsf{Kind} \\
[\![\mathsf{let}\,(x : A, y : B) = t\,\mathsf{in}\,C]\!]_\varepsilon & = & [\![C]\!]_\varepsilon,\ \textit{if } A, B \in \mathsf{Type} \\
[\![\Box_{(r,s)} A]\!]_\varepsilon & = & [\![A]\!]_\varepsilon \\
[\![\Box A]\!]_\varepsilon & = & [\![A]\!]_\varepsilon \\
[\![\mathsf{let}\,\Box(x : A) = t\,\mathsf{in}\,B]\!]_\varepsilon & = & [\![B]\!]_{\varepsilon[x \mapsto [\![t]\!]_\varepsilon]},\ \textit{if } A \in \mathsf{Kind} \\
[\![\mathsf{let}\,\Box(x : A) = t\,\mathsf{in}\,B]\!]_\varepsilon & = & [\![B]\!]_\varepsilon,\ \textit{if } A \in \mathsf{Type}
\end{array}
$$

Definition 4.11. *Suppose $\Delta \odot \Gamma \models \varepsilon$. Then* Valid term valuations, $\Delta \odot \Gamma \models_\varepsilon \rho$, *are defined as follows:*

$$
\frac{}{\emptyset \odot \emptyset \models_\emptyset \emptyset}\ \textsc{Rho\_Empty}
\qquad
\frac{\begin{array}{l} t \in ([\![A]\!]_\varepsilon)\,(\varepsilon\,x) \\ \Delta \odot \Gamma \models_\varepsilon \rho \quad (\Delta \mid \sigma_2 \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_1 \end{array}}{(\Delta, \sigma_2) \odot (\Gamma, x : A) \models_\varepsilon \rho[x \mapsto t]}\ \textsc{Rho\_ExtTy}
$$

$$
\frac{\begin{array}{l} t \in [\![A]\!]_\varepsilon \\ \Delta \odot \Gamma \models_\varepsilon \rho \quad (\Delta \mid \sigma_2 \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_0 \end{array}}{(\Delta, \sigma_2) \odot (\Gamma, x : A) \models_\varepsilon \rho[x \mapsto t]}\ \textsc{Rho\_ExtTm}
$$

Definition 4.12. *Suppose $\Delta \odot \Gamma \models_\varepsilon \rho$. Then the* interpretation of a term t *typable in $\Delta \odot \Gamma$ is $(\!|t|\!)_\rho = \rho\, t$, but where all let-expressions are translated into substitutions, and all graded modalities are erased.*

Definition 4.13. *Suppose $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$. Then* semantic typing, $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \models t : A$, *is defined as follows:*

1. *If $(\Delta \mid \sigma \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_1$, then for every $\Delta \odot \Gamma \models_\varepsilon \rho$, $(\!|t|\!)_\rho \in [\![A]\!]_\varepsilon\,([\![t]\!]_\varepsilon)$.*
2. *If $(\Delta \mid \sigma \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_0$, then for every $\Delta \odot \Gamma \models_\varepsilon \rho$, $(\!|t|\!)_\rho \in [\![A]\!]_\varepsilon$.*

Lemma 4.14 (Substitution for Typing Interpretation)**.** *Suppose $\Delta \odot \Gamma \models \varepsilon$ and we have types $T_2 \in (\mathsf{Kind} \cup \mathsf{Con})$ and $T_1 \in \mathsf{Con}$ with T_1 and T_2 typable in $\Delta \odot \Gamma$, and a term $t \in \mathsf{Term}$ typable in $\Delta \odot \Gamma$. Then:*

1. $[\![T_2]\!]_{\varepsilon[x \mapsto [\![T_1]\!]_\varepsilon]} = [\![[T_1/x]T_2]\!]_\varepsilon$.
2. $[\![T_2]\!]_\varepsilon = [\![[t/x]T_2]\!]_\varepsilon$.

Proof. By straightforward induction on T_2 with the fact that substitutions disappear in the kind interpretation. $\square$

Lemma 4.15 (Equality of Interpretations). *Suppose $\Delta \odot \Gamma \models \varepsilon$. Then:*

1. *if $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash T_1 = T_2 : \mathsf{Type}_1$, then $\mathcal{K}[\![T_1]\!] = \mathcal{K}[\![T_2]\!]$.*
2. *if $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash T_1 = T_2 : \mathsf{Type}_l$, then $[\![T_1]\!]_\varepsilon = [\![T_2]\!]_\varepsilon$.*

Proof. Part one follows by induction on $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash T_1 = T_2 : \mathsf{Type}_l$, and so does part two, but it also depends on part one and the previous lemma. $\square$

Lemma 4.16 (Interpretation Soundness). *Suppose $\Delta \odot \Gamma \models \varepsilon$ and $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_1$. Then:*

1. *If $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$, then $[\![t]\!]_\varepsilon \in \mathcal{K}[\![A]\!]$*
2. *$[\![A]\!]_\varepsilon \in \mathcal{K}[\![A]\!] \to \mathsf{SAT}$*

Proof. This is a proof by simultaneous induction over $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$ and $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_1$. We consider part 1 assuming 2, and vice versa.

Proof of part 1:

Case 1:

$$\frac{\Delta \odot \Gamma \vdash}{(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_0 : \mathsf{Type}_1}\ \textsc{T\_Type}$$

This case holds trivially, because Type_1 cannot be of type Type_1.

Case 2:

$$\frac{\Delta_1, \sigma, \Delta_2 \odot \Gamma_1, x : A, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \sigma, \Delta_2 \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \sigma, 0, \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash x : A}\ \textsc{T\_Var}$$

In this case we have:

$$\begin{aligned}
&\Delta = (\Delta_1, \sigma, \Delta_2)\\
&\sigma_1 = (\mathbf{0}^{|\Delta_1|}, 1, \mathbf{0})\\
&\sigma_2 = (\sigma, 0, \mathbf{0})\\
&t = x
\end{aligned}$$

Thus, we must show that:

$$[\![x]\!]_\varepsilon \in \mathcal{K}[\![A]\!]$$

We know by Definition C.10 and Definition C.9 that $[\![x]\!]_\varepsilon = \varepsilon\, x \in \mathcal{K}[\![A]\!]$; thus, we obtain our result.

Case 3:

$$\frac{(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, x : B_1 \vdash B_2 : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_3 + \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} B_1) \to B_2 : \mathsf{Type}_{l_1 \sqcup l_2}}\ \textsc{T\_Arrow}$$

In this case we know:

$$\begin{aligned}
&\sigma_1 = (\sigma_3 + \sigma_4)\\
&\sigma_2 = \mathbf{0}\\
&t = (x :_{(s,r)} B_1) \to B_2\\
&A = \mathsf{Type}_{l_1 \sqcup l_2}
\end{aligned}$$

However, by assumption we know that $(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_{l_1 \sqcup l_2} : \mathsf{Type}_1$, and hence, $\mathsf{Type}_{l_1 \sqcup l_2} = \mathsf{Type}_0$ which implies that $l_1 = l_2 = 0$. This all implies that $B_1, B_2 \in \mathsf{Type}$. Furthermore, we know that $(\Delta, \sigma_3) \odot (\Gamma, x : B_1) \models \varepsilon$, because $B_1 \in \mathsf{Type}$.

Thus, by Definition C.8 we must show that:

$$[\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon \in \mathcal{K}[\![\mathsf{Type}_0]\!] = \mathsf{SAT}$$

By Definition C.10 we know that:

$$[\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon = [\![B_1]\!]_\varepsilon \to [\![B_2]\!]_\varepsilon$$

By the IH:

IH(1): $[\![B_1]\!]_\varepsilon \in \mathcal{K}[\![\mathsf{Type}_0]\!] = \mathsf{SAT}$

IH(2): $[\![B_2]\!]_\varepsilon \in \mathcal{K}[\![\mathsf{Type}_0]\!] = \mathsf{SAT}$

Therefore, $[\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon = ([\![B_1]\!]_\varepsilon \to [\![B_2]\!]_\varepsilon) = (\mathsf{SAT} \to \mathsf{SAT}) \in \mathsf{SAT}$ which holds by the closer of SAT under function spaces.

Case 4:

$$\frac{(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, x : B_1 \vdash B_2 : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_3 + \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash (x :_r B_1) \otimes B_2 : \mathsf{Type}_{l_1 \sqcup l_2}} \text{ T\_TEN}$$

This case follows nearly exactly as the previous case, but ending with a cartesian product, $\mathsf{SAT} \times \mathsf{SAT}$, rather than the function space.

Case 5:

$$\frac{(\Delta, \sigma_3 \mid \sigma_4, s \mid \sigma_5, r) \odot \Gamma, x : B_1 \vdash t' : B_2}{(\Delta \mid \sigma_4 \mid \sigma_3 + \sigma_5) \odot \Gamma \vdash \lambda x : B_1.t' : (x :_{(s,r)} B_1) \to B_2} \text{ T\_FUN}$$

In this case we know:

$$\begin{aligned} &\sigma_1 = \sigma_4 \\ &\sigma_2 = \sigma_3 + \sigma_5 \\ &t = \lambda x : B_1.t' \\ &A = (x :_{(s,r)} B_1) \to B_2 \end{aligned}$$

We also know that $(\Delta \mid \sigma_3 + \sigma_5 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} B_1) \to B_2 : \mathsf{Type}_1$ by assumption. This implies by inversion that $B_1, B_2 \in \mathsf{Kind}$, $B_1 \in \mathsf{Kind}$ and $B_2 \in \mathsf{Type}$, or $B_1 \in \mathsf{Type}$ and $B_2 \in \mathsf{Kind}$. We consider each case in turn:

Subcase 1: Suppose $B_1, B_2 \in \mathsf{Kind}$. Then we must show that:

$$[\![\lambda x : B_1.t']\!]_\varepsilon \in \mathcal{K}[\![(x :_{(s,r)} B_1) \to B_2]\!] = \{f \mid f : \mathcal{K}[\![B_1]\!] \to \mathcal{K}[\![B_2]\!]\}$$

By Definition C.10 we know that:

$$[\![\lambda x : B_1.t']\!]_\varepsilon = \lambda X \in \mathcal{K}[\![B_1]\!].[\![t']\!]_{\varepsilon[x \mapsto X]}$$

Suppose $X \in \mathcal{K}[\![B_1]\!]$. Then we will show that $[\![t']\!]_{\varepsilon[x \mapsto X]} \in \mathcal{K}[\![B_2]\!]$. We know by assumption that $\Delta \odot \Gamma \models \varepsilon$ and $(\Delta, \sigma_3 \mid \sigma_4, s \mid \sigma_5, r) \odot \Gamma, x : B_1 \vdash t' : B_2$. Hence, by Lemma 3.5, we know that $(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_1$. Thus, we know that $(\Delta, \sigma_3) \odot (\Gamma, x : B_1) \models \varepsilon[x \mapsto X]$.

Therefore, by the IH:

IH(1): $[\![t']\!]_{\varepsilon[x \mapsto X]} \in \mathcal{K}[\![B_2]\!]$

which is what was to be shown.

Subcase 2: Suppose $B_1 \in \mathsf{Kind}$ and $B_2 \in \mathsf{Type}$. Then we must show that:

$$[\![\lambda x : B_1.t']\!]_\varepsilon \in \mathcal{K}[\![(x :_{(s,r)} B_1) \to B_2]\!] = \mathcal{K}[\![B_1]\!]$$

By Definition C.10 we know that:

$$[\![\lambda x : B_1.t']\!]_\varepsilon = \mathcal{K}[\![B_1]\!]$$

which is what was to be shown.

Subcase 3: Suppose $B_1 \in \mathsf{Type}$ and $B_2 \in \mathsf{Kind}$. Then we must show that:

$$[\![\lambda x : B_1.t']\!]_\varepsilon \in \mathcal{K}[\![(x :_{(s,r)} B_1) \to B_2]\!] = \mathcal{K}[\![B_2]\!]$$

By Definition C.10 we know that:

$$[\![\lambda x : B_1.t']\!]_\varepsilon = \mathcal{K}[\![B_2]\!]$$

which is what was to be shown.

Case 6:

$$\frac{(\Delta \mid \sigma_4 \mid \sigma_3 + \sigma_5) \odot \Gamma \vdash t_1 : (x :_{(s,r)} B_1) \to B_2 \quad (\Delta \mid \sigma_6 \mid \sigma_3) \odot \Gamma \vdash t_2 : B_1}{(\Delta \mid \sigma_4 + s * \sigma_6 \mid \sigma_5 + r * \sigma_6) \odot \Gamma \vdash t_1\, t_2 : [t_2/x]B_2} \text{ T\_APP}$$

In this case we know that:

$$\begin{array}{l} \sigma_1 = \sigma_4 + s * \sigma_6 \\ \sigma_2 = \sigma_5 + r * \sigma_6 \\ t = t_1\, t_2 \\ A = [t_2/x]B_2 \end{array}$$

It suffices to show that:

$$[\![t_1\, t_2]\!]_\varepsilon \in \mathcal{K}[\![[t_2/x]B_2]\!] = \mathcal{K}[\![B_2]\!]$$

In this case we know by assumption that $[t_2/x]B_2 \in \mathsf{Kind}$ which implies that $B_2 \in \mathsf{Kind}$, and this further implies that $((x :_{(s,r)} B_1) \to B_2) \in \mathsf{Kind}$. However, there are two cases for B_1, either $B_1 \in \mathsf{Kind}$ or $B_1 \in \mathsf{Type}$. We consider each case in turn.

Subcase 1: Suppose $B_1 \in \mathsf{Kind}$. Then we know by assumption that $\Delta \odot \Gamma \models \varepsilon$. Thus, we apply the IH to obtain:

IH(1): $[\![t_1]\!]_\varepsilon \in \mathcal{K}[\![(x :_{(s,r)} B_1) \to B_2]\!] = \{f \mid f : \mathcal{K}[\![B_1]\!] \to \mathcal{K}[\![B_2]\!]\}$
IH(2): $[\![t_2]\!]_\varepsilon \in \mathcal{K}[\![B_1]\!]$

Using the IH's and Definition C.10 we know:

$$[\![t_1\, t_2]\!]_\varepsilon = [\![t_1]\!]_\varepsilon\, ([\![t_2]\!]_\varepsilon) \in \mathcal{K}[\![B_2]\!]$$

which was what was to be shown.

Subcase 2: Suppose $B_1 \in \mathsf{Type}$. Then we know by assumption that $\Delta \odot \Gamma \models \varepsilon$. Thus, we apply the IH to obtain:

IH(1): $[\![t_1]\!]_\varepsilon \in \mathcal{K}[\![(x :_{(s,r)} B_1) \to B_2]\!] = \mathcal{K}[\![B_2]\!]$

Using the IH's and Definition C.10 we know:

$$[\![t_1\, t_2]\!]_\varepsilon = [\![t_1]\!]_\varepsilon \in \mathcal{K}[\![B_2]\!]$$

which was what was to be shown.

Case 7:

$$\frac{\begin{array}{l}(\Delta, \sigma_3 \mid \sigma_5, r \mid \mathbf{0}) \odot \Gamma, x : B_1 \vdash B_2 : \mathsf{Type}_l \\ (\Delta \mid \sigma_4 \mid \sigma_3) \odot \Gamma \vdash t_1 : B_1 \\ (\Delta \mid \sigma_6 \mid \sigma_5 + r * \sigma_4) \odot \Gamma \vdash t_2 : [t_1/x]B_2\end{array}}{(\Delta \mid \sigma_4 + \sigma_6 \mid \sigma_3 + \sigma_5) \odot \Gamma \vdash (t_1, t_2) : (x :_r B_1) \otimes B_2} \text{ T\_PAIR}$$

In this case we know that:

$$\begin{array}{l}\sigma_1 = \sigma_4 + \sigma_6 \\ \sigma_2 = \sigma_3 + \sigma_5 \\ t = (t_1, t_2) \\ A = (x :_r B_1) \otimes B_2\end{array}$$

It suffices to show that:

$$[\![(t_1, t_2)]\!]_\varepsilon \in \mathcal{K}[\![(x :_r B_1) \otimes B_2]\!]$$

By assumption we know that $((x :_r B_1) \otimes B_2) \in \mathsf{Kind}$. Thus, it must be the case that either $B_1, B_2 \in \mathsf{Kind}$, $B_1 \in \mathsf{Kind}$ and $B_2 \in \mathsf{Type}$, or $B_1 \in \mathsf{Type}$ and $B_2 \in \mathsf{Kind}$. We consider each of these cases in turn.

Subcase 1: Suppose $B_1, B_2 \in \mathsf{Kind}$ so $l = 1$. Then by Definition C.8 and Definition C.10 it suffices to show that:

$$\begin{aligned}[\![(t_1, t_2)]\!]_\varepsilon &= ([\![t_1]\!]_\varepsilon, [\![t_2]\!]_\varepsilon) \\ &\in \mathcal{K}[\![(x :_r B_1) \otimes B_2]\!] \\ &= \mathcal{K}[\![B_1]\!] \times \mathcal{K}[\![B_2]\!]\end{aligned}$$

We know by assumption that $\Delta \odot \Gamma \models \varepsilon$ and $(\Delta \mid \sigma_4 \mid \sigma_3) \odot \Gamma \vdash t_1 : B_1$. By kinding for typing (Lemma 3.6) we know that $(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_1$, and by assumption we know that $(\Delta, \sigma_3 \mid \sigma_5, r \mid \mathbf{0}) \odot (\Gamma, x : B_1) \vdash B_2 : \mathsf{Type}_1$. By substitution for typing (Lemma 3.27), we know that $(\Delta \mid \sigma_5 + r * \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash [t_1/x]B_2 : \mathsf{Type}_1$. We can now apply the IH to the premises for t_1 and t_2 using the fact that we know $(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_1$ and $(\Delta \mid \sigma_5 + r * \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash [t_1/x]B_2 : \mathsf{Type}_1$.

Thus, by the IH:

IH(1): $[\![t_1]\!]_\varepsilon \in \mathcal{K}[\![B_1]\!]$

IH(2): $[\![t_2]\!]_\varepsilon \in \mathcal{K}[\![[t_1/x]B_2]\!] = \mathcal{K}[\![B_2]\!]$

Therefore, $([\![t_1]\!]_\varepsilon, [\![t_2]\!]_\varepsilon) \in \mathcal{K}[\![B_1]\!] \times \mathcal{K}[\![B_2]\!]$.

Subcase 2: Suppose $B_1 \in \mathsf{Kind}$ and $B_2 \in \mathsf{Type}$. Then by Definition C.8 and Definition C.10 it suffices to show that:

$$\begin{aligned}[\![(t_1, t_2)]\!]_\varepsilon &= [\![t_1]\!]_\varepsilon \\ &\in \mathcal{K}[\![(x :_r B_1) \otimes B_2]\!] \\ &= \mathcal{K}[\![B_1]\!]\end{aligned}$$

We now by assumption that $\Delta \odot \Gamma \models \varepsilon$ and $(\Delta \mid \sigma_4 \mid \sigma_3) \odot \Gamma \vdash t_1 : B_1$. By kinding for typing (Lemma 3.6) we know that $(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_1$. Thus, by the IH:

IH(1): $[\![t_1]\!]_\varepsilon \in \mathcal{K}[\![B_1]\!]$

which is what was to be shown.

Subcase 3: Suppose $B_1 \in \mathsf{Type}$ and $B_2 \in \mathsf{Kind}$. Then by Definition C.8 and Definition C.10 it suffices to show that:

$$\begin{aligned}[\![(t_1, t_2)]\!]_\varepsilon &= [\![t_2]\!]_\varepsilon \\ &\in \mathcal{K}[\![(x :_r B_1) \otimes B_2]\!] \\ &= \mathcal{K}[\![B_2]\!]\end{aligned}$$

We know by assumption that $\Delta \odot \Gamma \models \varepsilon$ $(\Delta \mid \sigma_4 \mid \sigma_3) \odot \Gamma \vdash t_1 : B_1$, and $(\Delta, \sigma_3 \mid \sigma_5, r \mid \mathbf{0}) \odot (\Gamma, x : B_1) \vdash B_2 : \mathsf{Type}_1$. By substitution for typing (Lemma 3.27), we know that

$(\Delta \mid \sigma_5 + r * \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash [t_1/x]B_2 : \mathsf{Type}_1$. We can now apply the IH to the premises for t_2 using the fact that we know $(\Delta \mid \sigma_5 + r * \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash [t_1/x]B_2 : \mathsf{Type}_1$.

Thus, by the IH:

IH(1): $[\![t_2]\!]_\varepsilon \in \mathcal{K}[\![[t_1/x]B_2]\!] = \mathcal{K}[\![B_2]\!]$

which is what was to be shown.

Case 8:

$$\frac{\begin{array}{c}(\Delta \mid \sigma_5 \mid \sigma_3 + \sigma_4) \odot \Gamma \vdash t_1 : (x :_r B_1) \otimes B_2 \\ (\Delta, (\sigma_3 + \sigma_4) \mid \sigma_7, r' \mid \mathbf{0}) \odot \Gamma, z : (x :_r B_1) \otimes B_2 \vdash C : \mathsf{Type}_1 \\ (\Delta, \sigma_3, (\sigma_4, r) \mid \sigma_6, s, s \mid \sigma_7, r', r') \odot \Gamma, x : B_1, y : B_2 \vdash t_2 : [(x,y)/z]C\end{array}}{(\Delta \mid \sigma_6 + s * \sigma_5 \mid \sigma_7 + r' * \sigma_5) \odot \Gamma \vdash \mathsf{let}\,(x : B_1, y : B_2) = t_1 \,\mathsf{in}\, t_2 : [t_1/z]C}\ \text{T\_TenCut}$$

In this case we know that:

$$\begin{array}{l}\sigma_1 = \sigma_6 + s * \sigma_5 \\ \sigma_2 = \sigma_7 + r' * \sigma_5 \\ t = (\mathsf{let}\,(x : B_1, y : B_2) = t_1 \,\mathsf{in}\, t_2) \\ A = [t_1/z]C\end{array}$$

It suffices to show that:

$$[\![\mathsf{let}\,(x : B_1, y : B_2) = t_1 \,\mathsf{in}\, t_2]\!]_\varepsilon \in \mathcal{K}[\![[t_1/z]C]\!] = \mathcal{K}[\![C]\!]$$

Based on these assumptions, it must be the case that $C \in \mathsf{Kind}$ and $t_2 \in \mathsf{Type}$. First, we can conclude by kinding for typing that:

$$(\Delta, \sigma_3, (\sigma_4, r) \mid \sigma_{11} \mid \mathbf{0}) \odot (\Gamma, x : B_1, y : B_2) \vdash [(x,y)/z]C : \mathsf{Type}_1$$

for some vector σ_1. Then by well-formed contexts for typing we know that:

$$(\Delta, \sigma_3, (\sigma_4, r)) \odot (\Gamma, x : B_1, y : B_2) \vdash$$

This then implies that:

$$(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_{l_1} (\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot (\Gamma, x : B_1) \vdash B_2 : \mathsf{Type}_{l_2}$$

We will use these to apply the IH and define the typing evaluation below.

We must now consider cases for when $B_1, B_2 \in \mathsf{Kind}$, $B_1 \in \mathsf{Kind}$ and $B_2 \in \mathsf{Type}$, $B_1 \in \mathsf{Type}$ and $B_2 \in \mathsf{Kind}$, and $B_1, B_2 \in \mathsf{Type}$. We consider each case in turn.

Subcase 1: Suppose $B_1, B_2 \in \mathsf{Kind}$. Then we know:

$$\begin{array}{l}(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_1 \\ (\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot (\Gamma, x : B_1) \vdash B_2 : \mathsf{Type}_1 \\ (\Delta \mid \sigma_3 + \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash (x :_r B_1) \otimes B_2 : \mathsf{Type}_1\end{array}$$

By Definition C.8 and Definition C.10 it suffices to show:

$$\begin{array}{l}[\![\mathsf{let}\,(x : B_1, y : B_2) = t_1 \,\mathsf{in}\, t_2]\!]_\varepsilon \\ \quad = [\![t_2]\!]_{\varepsilon[x \mapsto \pi_1 [\![t_1]\!]_\varepsilon, y \mapsto \pi_2 [\![t_1]\!]_\varepsilon]} \\ \quad \in \mathcal{K}[\![[t_1/z]C]\!] \\ \quad = \mathcal{K}[\![C]\!]\end{array}$$

By the applying the IH to the premise for t_1 and Definition C.8:

IH(1): $[\![t_1]\!]_\varepsilon \in \mathcal{K}[\![(x :_r B_1) \otimes B_2]\!] = \mathcal{K}[\![B_1]\!] \times \mathcal{K}[\![B_2]\!]$

Hence, $\pi_1 \llbracket t_1 \rrbracket_\varepsilon \in \mathcal{K}\llbracket B_1 \rrbracket$ and $\pi_2 \llbracket t_1 \rrbracket_\varepsilon \in \mathcal{K}\llbracket B_2 \rrbracket$. This along with the kinding judgments given above imply that

$$(\Delta, \sigma_3, (\sigma_4, r)) \odot (\Gamma, x : B_1, y : B_2) \models \varepsilon[x \mapsto \pi_1 \llbracket t_1 \rrbracket_\varepsilon, y \mapsto \pi_2 \llbracket t_1 \rrbracket_\varepsilon]$$

given the assumption $\Delta \odot \Gamma \models \varepsilon$. We also know:

$$(\Delta, \sigma_3, (\sigma_4, r) \mid \sigma_{12} \mid \mathbf{0}) \odot (\Gamma, x : B_1, y : B_2) \vdash [(x, y)/z]C : \mathsf{Type}_1$$

by applying kinding for typing to the premise for t_2. We now have everything we need to apply the IH to the premise for t_2:

IH(2): $\llbracket t_2 \rrbracket_{\varepsilon[x \mapsto \pi_1 \llbracket t_1 \rrbracket_\varepsilon, y \mapsto \pi_2 \llbracket t_1 \rrbracket_\varepsilon]} \in \mathcal{K}\llbracket [(x, y)/z]C \rrbracket = \mathcal{K}\llbracket C \rrbracket$

Now by Definition C.10 and the previous results:

$$\begin{array}{l} \llbracket \mathsf{let}\,(x : B_1, y : B_2) = t_1\,\mathsf{in}\,t_2 \rrbracket_\varepsilon \\ \quad = \llbracket t_2 \rrbracket_{\varepsilon[x \mapsto \pi_1 \llbracket t_1 \rrbracket_\varepsilon, y \mapsto \pi_2 \llbracket t_1 \rrbracket_\varepsilon]} \\ \quad \in K[|C|] \end{array}$$

which was what was to be shown.

Subcase 2: Suppose $B_1 \in \mathsf{Kind}$ and $B_2 \in \mathsf{Type}$. Then we know:

$$\begin{array}{l} (\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_1 \\ (\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot (\Gamma, x : B_1) \vdash B_2 : \mathsf{Type}_0 \\ (\Delta \mid \sigma_3 + \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash (x :_r B_1) \otimes B_2 : \mathsf{Type}_1 \end{array}$$

By Definition C.8 and Definition C.10 it suffices to show:

$$\begin{array}{l} \llbracket \mathsf{let}\,(x : B_1, y : B_2) = t_1\,\mathsf{in}\,t_2 \rrbracket_\varepsilon \\ \quad = \llbracket t_2 \rrbracket_{\varepsilon[x \mapsto \llbracket t_1 \rrbracket_\varepsilon]} \\ \quad \in \mathcal{K}\llbracket [t_1/z]C \rrbracket \\ \quad = \mathcal{K}\llbracket C \rrbracket \end{array}$$

By the applying the IH to the premise for t_1 and Definition C.8:

IH(1): $\llbracket t_1 \rrbracket_\varepsilon \in \mathcal{K}\llbracket (x :_r B_1) \otimes B_2 \rrbracket = \mathcal{K}\llbracket B_1 \rrbracket$

This along with the kinding judgments given above imply that

$$(\Delta, \sigma_3, (\sigma_4, r)) \odot (\Gamma, x : B_1, y : B_2) \models \varepsilon[x \mapsto \llbracket t_1 \rrbracket_\varepsilon]$$

given the assumption $\Delta \odot \Gamma \models \varepsilon$. We also know:

$$(\Delta, \sigma_3, (\sigma_4, r) \mid \sigma_{12} \mid \mathbf{0}) \odot (\Gamma, x : B_1, y : B_2) \vdash [(x, y)/z]C : \mathsf{Type}_1$$

by applying kinding for typing to the premise for t_2. We now have everything we need to apply the IH to the premise for t_2:

IH(2): $\llbracket t_2 \rrbracket_{\varepsilon[x \mapsto \llbracket t_1 \rrbracket_\varepsilon]} \in \mathcal{K}\llbracket [(x, y)/z]C \rrbracket = \mathcal{K}\llbracket C \rrbracket$

Now by Definition C.10 and the previous results:

$$\begin{array}{l} \llbracket \mathsf{let}\,(x : B_1, y : B_2) = t_1\,\mathsf{in}\,t_2 \rrbracket_\varepsilon \\ \quad = \llbracket t_2 \rrbracket_{\varepsilon[x \mapsto \llbracket t_1 \rrbracket_\varepsilon]} \\ \quad \in K[|C|] \end{array}$$

which was what was to be shown.

Subcase 3: Suppose $B_1 \in \mathsf{Type}$ and $B_2 \in \mathsf{Kind}$. This case is similar to the previous case using:

$$(\Delta, \sigma_3, (\sigma_4, r)) \odot (\Gamma, x : B_1, y : B_2) \models \varepsilon[x \mapsto [\![t_1]\!]_\varepsilon]$$

Subcase 4: Suppose $B_1, B_2 \in \mathsf{Type}$. This case is similar to the previous case using:

$$(\Delta, \sigma_3, (\sigma_4, r)) \odot (\Gamma, x : B_1, y : B_2) \models \varepsilon$$

Case 9:

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash B : \mathsf{Type}_0}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s B : \mathsf{Type}_0}\ \text{T\_Box}$$

This case follows directly from the induction hypothesis.

Case 10:

$$\frac{(\Delta \mid \sigma_3 \mid \sigma_4) \odot \Gamma \vdash t' : B}{(\Delta \mid s * \sigma_3 \mid \sigma_4) \odot \Gamma \vdash \Box t' : \Box_s B}\ \text{T\_BoxI}$$

This case follows directly from the induction hypothesis.

Case 11:

$$\frac{\begin{array}{l}(\Delta \mid \sigma_3 \mid \sigma_7) \odot \Gamma \vdash t_1 : \Box_s B_1 \\ (\Delta, \sigma_7 \mid \sigma_5, s \mid \sigma_6, (s * r)) \odot \Gamma, x : B_1 \vdash t_2 : [\Box x/z] B_2 \\ (\Delta, \sigma_7 \mid \sigma_6, r \mid \mathbf{0}) \odot \Gamma, z : \Box_s B_1 \vdash B_2 : \mathsf{Type}_l\end{array}}{(\Delta \mid \sigma_3 + \sigma_5 \mid \sigma_6 + r * \sigma_3) \odot \Gamma \vdash \mathsf{let}\,\Box(x : B_1) = t_1\,\mathsf{in}\,t_2 : [t_1/z] B_2}\ \text{T\_BoxE}$$

In this case we know that:

$$\begin{array}{l}\sigma_1 = \sigma_3 + \sigma_5 \\ \sigma_2 = \sigma_6 + r * \sigma_3 \\ t = (\mathsf{let}\,\Box(x : B_1) = t_1\,\mathsf{in}\,t_2) \\ A = [t_1/z] B_2\end{array}$$

It suffices to show that:

$$[\![\mathsf{let}\,\Box(x : B_1) = t_1\,\mathsf{in}\,t_2]\!]_\varepsilon \in \mathcal{K}[\![[t_1/z]B_2]\!] = \mathcal{K}[\![B_2]\!]$$

In this case we know that $([t_1/z]B_2) \in \mathsf{Kind}$, and thus, $B_2 \in \mathsf{Kind}$ and either $B_1 \in \mathsf{Kind}$ or $B_1 \in \mathsf{Type}$. We cover both of these cases in turn.

Subcase 1: Suppose $B_1 \in \mathsf{Kind}$. It suffices to show that:

$$\begin{array}{ll}[\![\mathsf{let}\,\Box(x : B_1) = t_1\,\mathsf{in}\,t_2]\!]_\varepsilon & = [\![t_2]\!]_{\varepsilon[x \mapsto [\![t_1]\!]_\varepsilon]} \\ & \in \mathcal{K}[\![[t_1/z]B_2]\!] \\ & = \mathcal{K}[\![B_2]\!]\end{array}$$

As we have seen in the previous cases we can apply well-formed contexts for typing to obtain that:

$$(\Delta \mid \sigma_7 \mid \mathbf{0}) \odot \Gamma \vdash \Box_s B_1 : \mathsf{Type}_1$$

We can now apply the IH to the premise for t_1 to obtain:

IH(1): $[\![t_1]\!]_\varepsilon \in \mathcal{K}[\![\Box_s B_1]\!] = \mathcal{K}[\![B_1]\!]$

Using the previous two facts along with the assumption that $\Delta \odot \Gamma \models \varepsilon$ we may obtain

$$(\Delta, \sigma_7) \odot (\Gamma, x : B_1) \models \varepsilon[x \mapsto [\![t_1]\!]_\varepsilon]$$

In addition, we know that $(\Delta, \sigma_7 \mid \sigma_6, r \mid \mathbf{0}) \odot \Gamma, z : \Box_s B_1 \vdash B_2 : \mathsf{Type}_1$. Thus, we can now apply the induction hypothesis a second time.

IH(2): $[\![t_2]\!]_{\varepsilon[x\mapsto[\![t_1]\!]_\varepsilon]} \in \mathcal{K}[\![B_2]\!]$

which was what was to be shown.

Subcase 2: Suppose $B_1 \in \mathsf{Type}$. It suffices to show that:

$$\begin{aligned}[\![\mathsf{let}\,\square(x : B_1) = t_1 \,\mathsf{in}\, t_2]\!]_\varepsilon &= [\![t_2]\!]_\varepsilon \\ &\in \mathcal{K}[\![[t_1/z]B_2]\!] \\ &= \mathcal{K}[\![B_2]\!]\end{aligned}$$

As we have seen in the previous cases we can apply well-formed contexts for typing to obtain that:

$$(\Delta \mid \sigma_7 \mid \mathbf{0}) \odot \Gamma \vdash \square_s B_1 : \mathsf{Type}_0$$

Using this along with the assumption that $\Delta \odot \Gamma \models \varepsilon$ we may obtain

$$(\Delta, \sigma_7) \odot (\Gamma, x : B_1) \models \varepsilon$$

In addition, we know that $(\Delta, \sigma_7 \mid \sigma_6, r \mid \mathbf{0}) \odot \Gamma, z : \square_s B_1 \vdash B_2 : \mathsf{Type}_1$. Thus, we can now apply the induction hypothesis a second time.

IH(2): $[\![t_2]\!]_\varepsilon \in \mathcal{K}[\![B_2]\!]$

which was what was to be shown.

Case 12:

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A' \quad (\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash A' = A : \mathsf{Type}_1}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A}\ \textsc{T\_TyConv}$$

This case follows by first applying the induction hypothesis to the typing premise, and then applying Lemma C.15 to obtain that $\mathcal{K}[\![A]\!] = \mathcal{K}[\![A']\!]$ obtaining our result.

We now move onto the second part of this result assuming the first. In this part we will show:

If $\Delta \odot \Gamma \models \varepsilon$ and $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_1$, then $[\![A]\!]_\varepsilon \in \mathcal{K}[\![A]\!] \to \mathsf{SAT}$.

Recall that this is a proof by mutual induction on $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$ and $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_1$.

Case 1:

$$\frac{\Delta \odot \Gamma \vdash}{(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_0 : \mathsf{Type}_1}\ \textsc{T\_Type}$$

In this case we know that:

$$\begin{aligned}\sigma &= \mathbf{0} \\ A &= \mathsf{Type}_0\end{aligned}$$

It suffices to show that:

$$[\![\mathsf{Type}_0]\!]_\varepsilon \in (\mathcal{K}[\![\mathsf{Type}_0]\!] \to \mathsf{SAT}) = (\mathsf{SAT} \to \mathsf{SAT})$$

But, $[\![\mathsf{Type}_0]\!]_\varepsilon = \lambda x \in \mathsf{SAT}.\mathsf{SN}$, and by Lemma C.7 $\mathsf{SN} \in \mathsf{SAT}$; hence, $(\lambda x \in \mathsf{SAT}.\mathsf{SN}) \in (\mathsf{SAT} \to \mathsf{SAT})$.

Case 2:

$$\frac{\Delta_1, \mathbf{0}, \Delta_2 \odot \Gamma_1, x : \mathsf{Type}_1, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \mathbf{0}, \Delta_2 \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \mathbf{0}) \odot (\Gamma_1, x : \mathsf{Type}_1, \Gamma_2) \vdash x : \mathsf{Type}_1}\ \textsc{T\_Var}$$

This case is impossible, because the well-formed context premise fails, because Type_1 has no type.

Case 3:

$$\frac{\begin{array}{c}(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_{l_1} \\ (\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, x : B_1 \vdash B_2 : \mathsf{Type}_{l_2}\end{array}}{(\Delta \mid \sigma_3 + \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} B_1) \to B_2 : \mathsf{Type}_1} \text{ T\_ARROW}$$

In this case we know that:

$$\begin{aligned}\sigma &= \sigma_3 + \sigma_4 \\ A &= (x :_{(s,r)} B_1) \to B_2\end{aligned}$$

It suffices to show that:

$$[\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon \in (\mathcal{K}[\![(x :_{(s,r)} B_1) \to B_2]\!] \to \mathsf{SAT})$$

In this case either $B_1, B_2 \in \mathsf{Kind}$, $B_1 \in \mathsf{Kind}$ and $B_2 \in \mathsf{Type}$, or $B_1 \in \mathsf{Type}$ and $B_2 \in \mathsf{Kind}$. We consider each case in turn.

Subcase 1: Suppose $B_1, B_2 \in \mathsf{Kind}$. It suffices to show:

$$\begin{aligned}&[\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon \\ &= \lambda X \in \mathcal{K}[\![B_1]\!] \to \mathcal{K}[\![B_2]\!]. \textstyle\bigcap_{Y \in \mathcal{K}[\![B_1]\!]}(([\![B_1]\!]_\varepsilon)\,(Y) \to ([\![B_2]\!]_{\varepsilon[x \mapsto Y]})\,(X\,(Y))) \\ &\in (\mathcal{K}[\![(x :_{(s,r)} B_1) \to B_2]\!] \to \mathsf{SAT}) \\ &= \{f \mid f : \mathcal{K}[\![B_1]\!] \to \mathcal{K}[\![B_2]\!]\} \to \mathsf{SAT}\end{aligned}$$

Now suppose $X \in (\mathcal{K}[\![B_1]\!] \to \mathcal{K}[\![B_2]\!])$ and $Y \in \mathcal{K}[\![B_1]\!]$. We know by assumption that $(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_{l_1}$ and $\Delta \odot \Gamma \models \varepsilon$, and so we can apply the induction hypothesis to the premise for B_1 to obtain:

IH(1): $[\![B_1]\!]_\varepsilon \in (\mathcal{K}[\![B_1]\!] \to \mathsf{SAT})$

The previous facts now allow us, by Definition C.9, to obtain:

$$(\Delta, \sigma_3) \odot (\Gamma, x : B_1) \models \varepsilon[x \mapsto Y]$$

Thus, we can now apply the induction hypothesis to the premise for B_2 to obtain:

IH(2): $[\![B_2]\!]_{\varepsilon[x \mapsto Y]} \in (\mathcal{K}[\![B_2]\!] \to \mathsf{SAT})$

Then we know by IH(1) that $[\![B_1]\!]_\varepsilon\,(Y) \in \mathsf{SAT}$ and by IH(2) $[\![B_2]\!]_{\varepsilon[x \mapsto Y]}\,(X\,(Y)) \in \mathsf{SAT}$, thus:

$$([\![B_1]\!]_\varepsilon\,(Y) \to [\![B_2]\!]_{\varepsilon[x \mapsto Y]}\,(X\,(Y))) \in (\mathsf{SAT} \to \mathsf{SAT}) \in \mathsf{SAT}$$

Then by Lemma C.8:

$$\bigcap_{Y \in \mathcal{K}[\![B_1]\!]} ([\![B_1]\!]_\varepsilon\,(Y) \to [\![B_2]\!]_{\varepsilon[x \mapsto Y]}\,(X\,(Y))) \in \mathsf{SAT}$$

Therefore, we obtain our result.

Subcase 2: Suppose $B_1 \in \mathsf{Kind}$ and $B_2 \in \mathsf{Type}$. Similar to the previous case.

Subcase 3: Suppose $B_1 \in \mathsf{Type}$ and $B_2 \in \mathsf{Kind}$. It suffices to show:

$$\begin{aligned}&[\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon \\ &= \lambda X \in \mathcal{K}[\![B_2]\!].([\![B_1]\!]_\varepsilon \to ([\![B_2]\!]_\varepsilon)\,(X)) \\ &\in (\mathcal{K}[\![(x :_{(s,r)} B_1) \to B_2]\!] \to \mathsf{SAT}) \\ &= (\mathcal{K}[\![B_2]\!] \to \mathsf{SAT})\end{aligned}$$

Now suppose $X \in \mathcal{K}[\![B_2]\!]$. We know by assumption that $(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_0$ and $\Delta \odot \Gamma \models \varepsilon$, and so we can apply the first part of the induction hypothesis to the premise for B_1 to obtain:

IH(1): $[\![B_1]\!]_\varepsilon \in \mathcal{K}[\![\mathsf{Type}_0]\!] = \mathsf{SAT}$

Now we know by assumption that $(\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, x : B_1 \vdash B_2 : \mathsf{Type}_0$, and we can now show by Definition C.9 that $(\Delta, \sigma_3) \odot (\Gamma, x : B_1) \models \varepsilon$ holds. So we can apply the IH to the former judgment to obtain:

IH(2): $[\![B_2]\!]_\varepsilon \in (\mathcal{K}[\![B_2]\!] \to \mathsf{SAT})$

At this point we can see that $(\lambda X \in \mathcal{K}[\![B_2]\!].([\![B_1]\!]_\varepsilon \to [\![B_2]\!]_\varepsilon (X))) \in \mathsf{SAT}$ by the previous facts, and the fact that SAT is closed under function spaces.

Case 4:

$$\frac{\begin{array}{c}(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_{l_1} \\ (\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, x : B_1 \vdash B_2 : \mathsf{Type}_{l_2}\end{array}}{(\Delta \mid \sigma_3 + \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash (x :_r B_1) \otimes B_2 : \mathsf{Type}_{l_1 \sqcup l_2}} \text{ T\_Ten}$$

This case is similar to the previous case.

Case 5:

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash B : \mathsf{Type}_1}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s B : \mathsf{Type}_1} \text{ T\_Box}$$

This case follows from the IH.

Case 6:

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash A : A' \quad (\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash A' = \mathsf{Type}_1 : B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash A : \mathsf{Type}_1} \text{ T\_TyConv}$$

This case is impossible, because Type_1 has no type B.

$\square$

Theorem 4.17 (Soundness for Semantic Typing)**.** *If* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$, *then* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \models t : A$.

Proof. This is a proof by induction on the assumed typing derivation.

Case 1:

$$\frac{\Delta \odot \Gamma \vdash}{(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_0 : \mathsf{Type}_1} \text{ T\_Type}$$

In this case we have:

$$\begin{array}{l}\sigma_1 = \mathbf{0} \\ \sigma_2 = \mathbf{0} \\ t = \mathsf{Type}_0 \\ A = \mathsf{Type}_1\end{array}$$

We can now see that this case holds trivially, because Type_1 has no type.

Case 2:

$$\frac{\Delta_1, \sigma, \Delta_2 \odot \Gamma_1, x : A, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \sigma, \Delta_2 \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \sigma, 0, \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash x : A} \text{ T\_Var}$$

In this case we have:

$$\begin{array}{l}\Delta = (\Delta_1, \sigma, \Delta_2) \\ \sigma_1 = \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \\ \sigma_2 = \sigma, 0, \mathbf{0} \\ t = x\end{array}$$

Now either $A \in \mathsf{Kind}$ or $A \in \mathsf{Type}$. We consider both cases in turn.

Subcase 1: Suppose $A \in \mathsf{Kind}$ and $\Delta \odot \Gamma \models_\varepsilon \rho$. It suffices to show:

$$(\!|x|\!)_\rho = \rho\, x \in [\![A]\!]_\varepsilon\,([\![x]\!]_\varepsilon) = [\![A]\!]_\varepsilon\,(\varepsilon\, x)$$

But, this holds by Definition C.12, because the well-formed context premise above implies the proper kinding of A.

Subcase 2: Suppose $A \in \mathsf{Type}$ and $\Delta \odot \Gamma \models_\varepsilon \rho$. It suffices to show:

$$(\!|x|\!)_\rho = \rho\, x \in [\![A]\!]_\varepsilon$$

But, this holds by Definition C.12, because the well-formed context premise above implies the proper kinding of A.

Case 3:

$$\frac{\begin{array}{c}(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_0 \\ (\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, x : B_1 \vdash B_2 : \mathsf{Type}_0\end{array}}{(\Delta \mid \sigma_3 + \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} B_1) \to B_2 : \mathsf{Type}_0}\ \textsc{T\_Arrow}$$

In this case we have:

$$\begin{array}{l}\sigma_1 = (\sigma_3 + \sigma_4) \\ \sigma_2 = \mathbf{0} \\ t = (x :_{(s,r)} B_1) \to B_2 \\ A = \mathsf{Type}_0\end{array}$$

We only need to consider the first case of this theorem, because Type_1 has no type. So suppose $\Delta \odot \Gamma \models_\varepsilon \rho$. It suffices to show:

$$\begin{array}{l}(\!|(x :_{(s,r)} B_1) \to B_2|\!)_\rho \\ = ((x :_{(s,r)} (\!|B_1|\!)_\rho) \to (\!|B_2|\!)_\rho) \\ \in [\![\mathsf{Type}_0]\!]_\varepsilon\,([\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon) \\ = \mathsf{SN}\end{array}$$

We know by assumption that $\Delta \odot \Gamma \models_\varepsilon \rho$ so we can apply the IH to conclude:

IH(1): $(\!|B_1|\!)_\rho \in [\![\mathsf{Type}_0]\!]_\varepsilon\,([\![B_1]\!]_\varepsilon) = \mathsf{SN}$

Now suppose $t \in [\![B_1]\!]_\varepsilon$. Then we know by Definition C.12 that $(\Delta, \sigma_3) \odot (\Gamma, x : B_1) \models_\varepsilon \rho[x \mapsto t]$. Thus, by applying the IH to the premise for B_2 we may conclude that

IH(2): $(\!|B_2|\!)_{\rho[x \mapsto t]} \in [\![\mathsf{Type}_0]\!]_\varepsilon\,([\![B_2]\!]_\varepsilon) = \mathsf{SN}$

holds for every t. Therefore, we may conclude out result.

Case 4:

$$\frac{\begin{array}{c}(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_0 \\ (\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, x : B_1 \vdash B_2 : \mathsf{Type}_0\end{array}}{(\Delta \mid \sigma_3 + \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash (x :_r B_1) \otimes B_2 : \mathsf{Type}_0}\ \textsc{T\_Ten}$$

Similar to the previous case.

Case 5:

$$\frac{(\Delta, \sigma_3 \mid \sigma_4, s \mid \sigma_5, r) \odot \Gamma, x : B_1 \vdash t' : B_2}{(\Delta \mid \sigma_4 \mid \sigma_3 + \sigma_5) \odot \Gamma \vdash \lambda x : B_1.t' : (x :_{(s,r)} B_1) \to B_2}\ \textsc{T\_Fun}$$

In this case we have:

$$\begin{array}{l}\sigma_1 = \sigma_4 \\ \sigma_2 = (\sigma_3 + \sigma_5) \\ t = \lambda x : B_1.t' \\ A = (x :_{(s,r)} B_1) \to B_2\end{array}$$

We have two cases to consider, either $((x :_{(s,r)} B_1) \to B_2) \in \mathsf{Kind}$ or $((x :_{(s,r)} B_1) \to B_2) \in \mathsf{Type}$. We cover both cases in turn.

Subcase 1: Suppose $((x :_{(s,r)} B_1) \to B_2) \in \mathsf{Kind}$. We now have three subcases depending on the typing for both B_1 and B_2.

Subcase 1: Suppose $B_1, B_2 \in \mathsf{Kind}$. It suffices to show:

$$\begin{array}{l} (\!|\lambda x : B_1.t'|\!)_\rho \\ = \lambda x : (\!|B_1|\!)_\rho.(\!|t'|\!)_\rho \\ \in [\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon\,([\![\lambda x : B_1.t']\!]_\varepsilon) \\ = \bigcap_{Y \in \mathcal{K}[\![B_1]\!]}(([\![B_1]\!]_\varepsilon)\,(Y) \to ([\![B_2]\!]_{\varepsilon[x \mapsto Y]})\,([\![\lambda x : B_1.t']\!]_\varepsilon\,(Y))) \\ = \bigcap_{Y \in \mathcal{K}[\![B_1]\!]}(([\![B_1]\!]_\varepsilon)\,(Y) \to ([\![B_2]\!]_{\varepsilon[x \mapsto Y]})\,([\![t']\!]_{\varepsilon[x \mapsto Y]})) \end{array}$$

So suppose we have a $Y \in \mathcal{K}[\![B_1]\!]$ and a $t \in [\![B_1]\!]_\varepsilon\,(Y) = [\![B_1]\!]_{(\varepsilon[x \mapsto Y])}\,(\varepsilon\, x)$ (since $x \notin \mathsf{FV}\,(B_2)$). Then we know that $(\Delta, \sigma_3) \odot (\Gamma, x : B_1) \models_{\varepsilon[x \mapsto Y]} \rho[x \mapsto t]$. Then by the IH:

IH(1): $(\!|t'|\!)_{\rho[x \mapsto t]} \in [\![B_2]\!]_{\varepsilon[x \mapsto Y]}\,([\![t']\!]_{\varepsilon[x \mapsto Y]})$

Thus, we obtain our result.

Subcase 2: Suppose $B_1 \in \mathsf{Kind}$ and $B_2 \in \mathsf{Type}$. This case is similar to the previous case, except we will use part two of the IH.

Subcase 3: Suppose $B_1 \in \mathsf{Type}$ and $B_2 \in \mathsf{Kind}$. Similar to the previous case.

Subcase 2: Suppose $((x :_{(s,r)} B_1) \to B_2) \in \mathsf{Type}$. This case is similar to the above, but we will use the second part of the IH.

Case 6:

$$\frac{\begin{array}{c}(\Delta \mid \sigma_4 \mid \sigma_3 + \sigma_5) \odot \Gamma \vdash t_1 : (x :_{(s,r)} B_1) \to B_2 \\ (\Delta \mid \sigma_6 \mid \sigma_3) \odot \Gamma \vdash t_2 : B_1\end{array}}{(\Delta \mid \sigma_4 + s * \sigma_6 \mid \sigma_5 + r * \sigma_6) \odot \Gamma \vdash t_1\, t_2 : [t_2/x]B_2}\ \textsc{T\_App}$$

In this case we have:

$$\begin{array}{l} \sigma_1 = (\sigma_4 + s * \sigma_6) \\ \sigma_2 = (\sigma_5 + r * \sigma_6) \\ t = (t_1\, t_2) \\ A = [t_2/x]B_2 \end{array}$$

We have several cases to consider.

Subcase 1: Suppose $B_1, [t_2/x]B_2 \in \mathsf{Kind}$. It suffices to show:

$$\begin{array}{l} (\!|t_1\, t_2|\!)_\rho \\ = (\!|t_1|\!)_\rho\, (\!|t_2|\!)_\rho \\ \in [\![[t_2/x]B_2]\!]_\varepsilon\,([\![t_1\, t_2]\!]_\varepsilon) \\ = [\![B_2]\!]_{\varepsilon[x \mapsto [\![t_2]\!]_\varepsilon]}\,([\![t_1]\!]_\varepsilon\,([\![t_2]\!]_\varepsilon)) \end{array}$$

By the IH we have:

IH(1): $(\!|t_1|\!)_\rho \in [\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon\,([\![t_1]\!]_\varepsilon)$

IH(2): $(\!|t_2|\!)_\rho \in [\![B_1]\!]_\varepsilon\,([\![t_2]\!]_\varepsilon)$

Notice that by Lemma C.16 we know that $[\![t_2]\!]_\varepsilon \in \mathcal{K}[\![B_1]\!]$. So using C.10 we know:

$$\begin{array}{l} (\!|t_1|\!)_\rho \in [\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon\,([\![t_1]\!]_\varepsilon) \\ = \bigcap_{Y \in \mathcal{K}[\![A]\!]}(([\![A]\!]_\varepsilon)\,(Y) \to ([\![B]\!]_{\varepsilon[x \mapsto Y]})\,([\![t_1]\!]_\varepsilon\,(Y))) \end{array}$$

Thus, $(\!|t_1|\!)_\rho\, (\!|t_2|\!)_\rho \in ([\![B]\!]_{\varepsilon[x \mapsto [\![t_2]\!]_\varepsilon]})\,([\![t_1]\!]_\varepsilon\,([\![t_2]\!]_\varepsilon))$.

Subcase 2: Suppose $B_1 \in \mathsf{Kind}$ and $[t_2/x]B_2 \in \mathsf{Type}$. It suffices to show:

$$\begin{array}{l} (\!|t_1\, t_2|\!)_\rho \\ = (\!|t_1|\!)_\rho\, (\!|t_2|\!)_\rho \\ \in [\![[t_2/x]B_2]\!]_\varepsilon \\ = [\![B_2]\!]_{\varepsilon[x \mapsto [\![t_2]\!]_\varepsilon]} \end{array}$$

By the IH we have:

IH(1): $(\!|t_1|\!)_\rho \in [\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon\, ([\![t_1]\!]_\varepsilon)$

IH(2): $(\!|t_2|\!)_\rho \in [\![B_1]\!]_\varepsilon\, ([\![t_2]\!]_\varepsilon)$

Notice that by Lemma C.16 we know that $[\![t_2]\!]_\varepsilon \in \mathcal{K}[\![B_1]\!]$. So using C.10 we know:

$$\begin{array}{rl} & (\!|t_1|\!)_\rho \in [\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon\, ([\![t_1]\!]_\varepsilon) \\ = & \bigcap_{Y \in \mathcal{K}[\![B_1]\!]} (([\![B_1]\!]_\varepsilon)\,(Y) \to ([\![B_2]\!]_{\varepsilon[x \mapsto Y]})) \end{array}$$

Thus, $(\!|t_1|\!)_\rho\, (\!|t_2|\!)_\rho \in ([\![B_2]\!]_{\varepsilon[x \mapsto [\![t_2]\!]_\varepsilon]})$.

Subcase 3: Suppose $B_1 \in \mathsf{Type}$ and $[t_2/x]B_2 \in \mathsf{Type}$. It suffices to show:

$$\begin{array}{l} (\!|t_1\, t_2|\!)_\rho \\ = (\!|t_1|\!)_\rho\, (\!|t_2|\!)_\rho \\ \in [\![[t_2/x]B_2]\!]_\varepsilon \\ = [\![B_2]\!]_{\varepsilon[x \mapsto [\![t_2]\!]_\varepsilon]} \end{array}$$

By the IH we have:

IH(1): $(\!|t_1|\!)_\rho \in [\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon$

IH(2): $(\!|t_2|\!)_\rho \in [\![B_1]\!]_\varepsilon$

Using C.10 we know:

$$\begin{array}{rl} & (\!|t_1|\!)_\rho \in [\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon\, ([\![t_1]\!]_\varepsilon) \\ = & [\![B_1]\!]_\varepsilon \to [\![B_2]\!]_\varepsilon \end{array}$$

Thus, $(\!|t_1|\!)_\rho\, (\!|t_2|\!)_\rho \in [\![B_2]\!]_\varepsilon$.

Case 7:

$$\frac{\begin{array}{l}(\Delta, \sigma_3 \mid \sigma_5, r \mid \mathbf{0}) \odot \Gamma, x : B_1 \vdash B_2 : \mathsf{Type}_l \\ (\Delta \mid \sigma_4 \mid \sigma_3) \odot \Gamma \vdash t_1 : B_1 \\ (\Delta \mid \sigma_6 \mid \sigma_5 + r * \sigma_4) \odot \Gamma \vdash t_2 : [t_1/x]B_2\end{array}}{(\Delta \mid \sigma_4 + \sigma_6 \mid \sigma_3 + \sigma_5) \odot \Gamma \vdash (t_1, t_2) : (x :_r B_1) \otimes B_2}\ \textsc{T\_Pair}$$

This case is similar to the case for λ-abstraction above.

Case 8:

$$\frac{\begin{array}{l}(\Delta \mid \sigma_5 \mid \sigma_3 + \sigma_4) \odot \Gamma \vdash t_1 : (x :_r B_1) \otimes B_2 \\ (\Delta, (\sigma_3 + \sigma_4) \mid \sigma_7, r' \mid \mathbf{0}) \odot \Gamma, z : (x :_r B_1) \otimes B_2 \vdash C : \mathsf{Type}_1 \\ (\Delta, \sigma_3, (\sigma_4, r) \mid \sigma_6, s, s \mid \sigma_7, r', r') \odot \Gamma, x : B_1, y : B_2 \vdash t_2 : [(x,y)/z]C\end{array}}{(\Delta \mid \sigma_6 + s * \sigma_5 \mid \sigma_7 + r' * \sigma_5) \odot \Gamma \vdash \mathsf{let}\,(x : B_1, y : B_2) = t_1\,\mathsf{in}\,t_2 : [t_1/z]C}\ \textsc{T\_TenCut}$$

Similar to the application case above.

Case 9:

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash B : \mathsf{Type}_0}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s B : \mathsf{Type}_0}\ \textsc{T\_Box}$$

This case follows from the IH.

Case 10:

$$\frac{(\Delta \mid \sigma_3 \mid \sigma_2) \odot \Gamma \vdash t' : B}{(\Delta \mid s * \sigma_3 \mid \sigma_2) \odot \Gamma \vdash \Box t' : \Box_s B} \text{ T\_BoxI}$$

In this case we have:

$$\begin{array}{l} \sigma_1 = (s * \sigma_3) \\ t = (\Box t') \\ A = \Box_s B \end{array}$$

We have two cases to consider.

Subcase 1: Suppose $B \in \mathsf{Kind}$. It suffices to show:

$$\begin{array}{ll} & (\!|\Box t'|\!)_\rho \\ = & (\!|t'|\!)_\rho \\ \in & [\![\Box_s B]\!]_\varepsilon \, ([\![\Box t']\!]_\varepsilon) \\ = & [\![B]\!]_\varepsilon \, ([\![t']\!]_\varepsilon) \end{array}$$

At this point, this case holds by the IH.

Subcase 2: Suppose $B \in \mathsf{Type}$. Similar to the previous case.

Case 11:

$$\frac{\begin{array}{l} (\Delta \mid \sigma_3 \mid \sigma_7) \odot \Gamma \vdash t_1 : \Box_s B_1 \\ (\Delta, \sigma_7 \mid \sigma_6, r \mid \mathbf{0}) \odot \Gamma, z : \Box_s B_1 \vdash B_2 : \mathsf{Type}_0 \\ (\Delta, \sigma_7 \mid \sigma_5, s \mid \sigma_6, (s * r)) \odot \Gamma, x : B_1 \vdash t_2 : [\Box x/z]B_2 \end{array}}{(\Delta \mid \sigma_3 + \sigma_5 \mid \sigma_6 + r * \sigma_3) \odot \Gamma \vdash \mathsf{let}\, \Box(x : B_1) = t_1 \,\mathsf{in}\, t_2 : [t_1/z]B_2} \text{ T\_BoxE}$$

In this case we have:

$$\begin{array}{l} \sigma_1 = (\sigma_3 + \sigma_5) \\ \sigma_2 = (\sigma_6 + r * \sigma_3) \\ t = (\mathsf{let}\, \Box(x : B_1) = t_1 \,\mathsf{in}\, t_2) \\ A = [t_1/z]B_2 \end{array}$$

We have several cases to consider:

Subcase 1: Suppose $B_1, B_2 \in \mathsf{Kind}$. It suffices to show:

$$\begin{array}{ll} & (\!|\mathsf{let}\, \Box(x : B_1) = t_1 \,\mathsf{in}\, t_2|\!)_\rho \\ = & (\!|t_2|\!)_{\rho[x \mapsto (\!|t_1|\!)_\rho]} \\ \in & [\![[t_1/z]B_2]\!]_\varepsilon \, ([\![\mathsf{let}\, \Box(x : B_1) = t_1 \,\mathsf{in}\, t_2]\!]_\varepsilon) \\ = & [\![B_2]\!]_{\varepsilon[x \mapsto [\![t_1]\!]_\varepsilon]} \, ([\![t_2]\!]_{\varepsilon[x \mapsto [\![t_1]\!]_\varepsilon]}) \end{array}$$

By the IH:

IH(1): $(\!|t_1|\!)_\rho \in [\![\Box_s B_1]\!]_\varepsilon \, ([\![t_1]\!]_\varepsilon) = [\![B_1]\!]_\varepsilon \, ([\![t_1]\!]_\varepsilon)$

At this point we need:

$$(\Delta, \sigma_7) \odot (\Gamma, x : B_1) \models_{\varepsilon[x \mapsto [\![t_1]\!]_\varepsilon]} \rho[x \mapsto (\!|t_1|\!)_\rho]$$

But, this follows by definition and Lemma C.16.

By the IH:

IH(2): $(\!|t_2|\!)_{\rho[x \mapsto (\!|t_1|\!)_\rho]} \in [\![B_2]\!]_{\varepsilon[x \mapsto [\![t_1]\!]_\varepsilon]} \, ([\![t_2]\!]_{\varepsilon[x \mapsto [\![t_1]\!]_\varepsilon]})$

Subcase 2: Suppose $B_1 \in \mathsf{Kind}$ and $B_2 \in \mathsf{Type}$. Similar to the previous case.

Subcase 3: Suppose $B_1 \in \mathsf{Type}$ and $B_2 \in \mathsf{Kind}$. It suffices to show:

$$\begin{array}{rl} & (\!|\mathsf{let}\,\square(x : B_1) = t_1 \,\mathsf{in}\, t_2|\!)_\rho \\ = & (\!|t_2|\!)_{\rho[x \mapsto (\!|t_1|\!)_\rho]} \\ \in & [\![[t_1/z]B_2]\!]_\varepsilon\,([\![\mathsf{let}\,\square(x : B_1) = t_1 \,\mathsf{in}\, t_2]\!]_\varepsilon) \\ = & [\![B_2]\!]_\varepsilon\,([\![t_2]\!]_\varepsilon) \end{array}$$

By the IH:

IH(1): $(\!|t_1|\!)_\rho \in [\![\square_s B_1]\!]_\varepsilon = [\![B_1]\!]_\varepsilon$

At this point we need:

$$(\Delta, \sigma_7) \odot (\Gamma, x : B_1) \models_\varepsilon \rho[x \mapsto (\!|t_1|\!)_\rho]$$

But, this follows by definition.

By the IH:

IH(2): $(\!|t_2|\!)_{\rho[x \mapsto (\!|t_1|\!)_\rho]} \in [\![B_2]\!]_\varepsilon\,([\![t_2]\!]_\varepsilon)$

Subcase 4: Suppose $B_1, B_2 \in \mathsf{Type}$. Similar to the previous case.

Case 12:

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A' \quad (\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash A' = A : \mathsf{Type}_l}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A}\ \textsc{T\_TyConv}$$

By the induction hypothesis we know $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \models t : A'$ holds, but this implies $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \models t : A'$ holds by Lemma C.15.

□

Corollary 4.17.1 (Strong Normalization)**.** *For every* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$*,* $t \in \mathsf{SN}$*.*

Proof. Similarly to CC, we can define a notion of canonical element in $\mathcal{K}[\![A]\!]$, and define a term valuation $\Delta \odot \Gamma \models_\varepsilon \rho$, and then conclude SN by the previous theorem. □

A Proofs for Graded Modal Dependent Type Theory

Lemma 3.1 (Strengthening)**.** *If* $(\Delta, \sigma_1, \Delta' \mid \sigma_2, s, \sigma_2' \mid \sigma_3, r, \sigma_3') \odot \Gamma, x : A, \Gamma' \vdash \mathcal{J}$*,* $x \notin \mathsf{FV}(\mathcal{J})$*, and* $x \notin \mathsf{FV}(\Gamma')$*, with* $|\Delta| = |\sigma_2| = |\sigma_3| = |\Gamma|$*, then* $(\Delta, (\Delta' \backslash \pi) \mid \sigma_2, \sigma_2' \mid \sigma_3, \sigma_3') \odot \Gamma, \Gamma' \vdash \mathcal{J}$ *where* $\pi = |\Gamma|$*.*

Proof. For well-formedness:

By induction on the definition of well-formed contexts.

- Case Wf_Empty. Trivial since it does not match the form of the lemma.
- Case Wf_Ext We consider two cases depending on the syntactic structure of Γ_2 and Δ_2 (simultaneously, since they must have the same size by the lemma statement).
  - $\Gamma_2 = \emptyset$ and $\Delta_2 = \emptyset$ then we have:

$$\frac{(\Delta_1 \mid \sigma \mid \mathbf{0}) \odot \Gamma_1 \vdash A : \mathsf{Type}_l}{\Delta_1, \sigma \odot \Gamma_1, x : A \vdash}\ \textsc{Wf\_Ext}$$

By Lemma 3.3 (on the first premise) then we have $\Delta_1 \odot \Gamma_1 \vdash$ which is the goal here.

– $\Gamma_2 = \Gamma_2', y : A'$ and $\Delta_2 = \Delta_2', \sigma'$ then we have:

$$\frac{(\Delta_1, \sigma, \Delta_2' \mid \sigma' \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2' \vdash A' : \mathsf{Type}_l}{\Delta_1, \sigma, \Delta_2', \sigma' \odot \Gamma_1, x : A, \Gamma_2', y : A' \vdash} \text{ W}_{\text{F}}\text{\_Ext}$$

By Lemma 3.1 on the premise we get $(\Delta_1, (\Delta_2' \backslash \pi) \mid \sigma' \backslash \pi \mid \mathbf{0}) \odot \Gamma_1, \Gamma_2' \vdash A' : \mathsf{Type}_l$
We then construct the derivation:

$$\frac{(\Delta_1, (\Delta_2' \backslash \pi) \mid \sigma' \backslash \pi \mid \mathbf{0}) \odot \Gamma_1, \Gamma_2' \vdash A' : \mathsf{Type}_l}{\Delta_1, (\Delta_2' \backslash \pi), \sigma' \backslash \pi \odot \Gamma_1, \Gamma_2', y : A' \vdash} \text{ W}_{\text{F}}\text{\_Ext}$$

Since $(\Delta_2' \backslash \pi), \sigma' \backslash \pi = (\Delta_2', \sigma') \backslash \pi$ then the above derivation satisfies the goal.

For typing: By induction on $(\Delta, \sigma_1, \Delta' \mid \sigma_2, s, \sigma_2' \mid \sigma_3, r, \sigma_3') \odot \Gamma, x : A, \Gamma' \vdash t : B$

Case T_Type

$$\frac{\Delta, \sigma_1, \Delta' \odot \Gamma, x : A, \Gamma' \vdash}{(\Delta, \sigma_1, \Delta' \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma, x : A, \Gamma' \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}} \text{ T\_Type}$$

By induction on $\Delta, \sigma_1, \Delta' \odot \Gamma, x : A, \Gamma' \vdash$ we have that $\Delta, (\Delta' \backslash \pi) \odot \Gamma, \Gamma' \vdash$ we can thus form the derivation:

$$\frac{\Delta, (\Delta' \backslash \pi) \odot \Gamma, \Gamma' \vdash}{(\Delta, (\Delta' \backslash \pi) \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}} \text{ T\_Type}$$

thus satisfying the goal here.

Case T_Var

$$\frac{\Delta_1, \sigma_1, \Delta_2, \sigma_1, \Delta_2' \odot \Gamma_1, y : A', \Gamma_2, x : A, \Gamma_2' \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \sigma_1, \Delta_2, \sigma_1, \Delta_2' \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \sigma_1, 0, \mathbf{0}) \odot \Gamma_1, y : A', \Gamma_2, x : A, \Gamma_2' \vdash y : A'} \text{ T\_Var}$$

By induction on $\Delta_1, \sigma_1', \Delta_2, \sigma_1, \Delta_2' \odot \Gamma_1, y : A', \Gamma_2, x : A, \Gamma_2' \vdash$ we have that $\Delta_1, \sigma_1', \Delta_2, (\Delta_2' \backslash \pi) \odot \Gamma_1, y : A', \Gamma_2, \Gamma_2'$.
Thus, we can build the var rule:

$$\frac{\Delta_1, \sigma_1, \Delta_2, (\Delta_2' \backslash \pi) \odot \Gamma_1, y : A', \Gamma_2, \Gamma_2' \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \sigma_1, \Delta_2, (\Delta_2' \backslash \pi) \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \sigma_1, \mathbf{0}) \odot \Gamma_1, y : A', \Gamma_2, \Gamma_2' \vdash y : A'} \text{ T\_Var}$$

which satisfies the goal here.

Case T_Arrow

$$\frac{\begin{array}{c}(\Delta, \sigma_1, \Delta' \mid \sigma_4 \mid \mathbf{0}) \odot \Gamma, x : A, \Gamma' \vdash A' : \mathsf{Type}_{l_1} \\ (\Delta, \sigma_1, \Delta', \sigma_4 \mid \sigma_5, r' \mid \mathbf{0}) \odot \Gamma, x : A, \Gamma', y : A' \vdash B : \mathsf{Type}_{l_2}\end{array}}{(\Delta, \sigma_1, \Delta' \mid \sigma_4 + \sigma_5 \mid \mathbf{0}) \odot \Gamma, x : A, \Gamma' \vdash (y :_{(s', r')} A') \to B : \mathsf{Type}_{l_1 \sqcup l_2}} \text{ T\_Arrow}$$

Thus, in the case we have that we have that $\sigma_3 = \mathbf{0}$, $r = 0$, and $\sigma_3' = \mathbf{0}$ and that $\sigma_4 + \sigma_5 = \sigma_2, s, \sigma_2'$ thus we must have $\sigma_4', s_1, \sigma_4'' = \sigma_4$ and $\sigma_5', s_2, \sigma_5'' = \sigma_5$ and $s_1 + s_2 = s$ such that $(\sigma_1'', s_1, \sigma_1''') + (\sigma_2'', s_2, \sigma_2''') = \sigma_2, s, \sigma_2'$.

By induction on the premises we get:

$$(\Delta, (\Delta' \backslash \pi) \mid \sigma'_4, \sigma''_4 \mid \mathbf{0}) \odot \Gamma, \Gamma' \vdash A' : \mathsf{Type}_{l_1} \quad \text{(ih1)}$$
$$(\Delta, (\Delta' \backslash \pi), \sigma'_4, \sigma''_4 \mid \sigma'_5, \sigma''_5, r' \mid \mathbf{0}) \odot \Gamma, \Gamma', y : A' \vdash B : \mathsf{Type}_{l_2} \quad \text{(ih2)}$$

Then we can derive:

$$\frac{\text{(ih1)} \qquad \text{(ih2)}}{(\Delta, (\Delta' \backslash \pi) \mid \sigma'_4, \sigma''_4 + \sigma'_5, \sigma''_5 \mid \mathbf{0}) \odot \Gamma, \Gamma' \vdash (y :_{(s',r')} A') \to B : \mathsf{Type}_{l_1 \sqcup l_2}} \text{T\_ARROW}$$

which satisfies this case.

Case T_TEN Same reasoning as above since the structure is exactly the same

Case T_FUN

$$\frac{\begin{array}{c}(\Delta, \sigma, \Delta', \sigma_1 \mid \sigma_4, r' \mid \mathbf{0}) \odot \Gamma, x : A, \Gamma', y : A' \vdash B : \mathsf{Type}_l \\ (\Delta, \sigma, \Delta', \sigma_1 \mid \sigma_2, s, \sigma'_2, s' \mid \sigma_4, r') \odot \Gamma, x : A, \Gamma', y : A' \vdash t : B\end{array}}{(\Delta, \sigma, \Delta' \mid \sigma_2, s, \sigma'_2 \mid \sigma_1 + \sigma_4) \odot \Gamma, x : A, \Gamma' \vdash \lambda y.t : (x :_{(s',r')} A') \to B} \text{T\_FUN}$$

with $\sigma_1 + \sigma_4 = \sigma_3, r, \sigma'_3$ thus let $\sigma_1 = \sigma'_1, r_1, \sigma''_1$ $\sigma_4 = \sigma'_4, r_2, \sigma''_4$.
By induction we have:

$$(\Delta, (\Delta', \sigma_1) \backslash \pi \mid \sigma'_4, \sigma''_4, r' \mid \mathbf{0}) \odot \Gamma, x : A, \Gamma', y : A' \vdash B : \mathsf{Type}_l$$
$$(\Delta, (\Delta', \sigma_1) \backslash \pi \mid \sigma_2, \sigma'_2, s' \mid \sigma'_4, \sigma''_4, r') \odot \Gamma, \Gamma', y : A' \vdash t : B$$

and thus $\sigma_1 \backslash \pi = \sigma'_1, \sigma''_1$.
From which we form the derivation:

$$\frac{\begin{array}{c}(\Delta, (\Delta', \sigma_1) \backslash \pi \mid \sigma'_4, \sigma''_4, r' \mid \mathbf{0}) \odot \Gamma, x : A, \Gamma', y : A' \vdash B : \mathsf{Type}_l \\ (\Delta, (\Delta', \sigma_1) \backslash \pi \mid \sigma_2, \sigma'_2, s' \mid \sigma'_4, \sigma''_4, r') \odot \Gamma, \Gamma', y : A' \vdash t : B\end{array}}{(\Delta, (\Delta' \backslash \pi) \mid \sigma_2, \sigma'_2 \mid \sigma'_1, \sigma''_1 + \sigma'_4, \sigma''_4) \odot \Gamma, \Gamma' \vdash \lambda y.t : (x :_{(s',r')} A') \to B} \text{T\_FUN}$$

satisfying the goal here.

- Case T_APP

$$\frac{\begin{array}{c}(\Delta, \sigma, \Delta', \sigma_1 \mid \sigma_6, r' \mid \mathbf{0}) \odot \Gamma, x : A, \Gamma', y : A' \vdash B : \mathsf{Type}_l \\ (\Delta, \sigma, \Delta' \mid \sigma_5 \mid \sigma_1 + \sigma_6) \odot \Gamma, x : A, \Gamma' \vdash t_1 : (y :_{(s',r')} A') \to B \\ (\Delta, \sigma, \Delta' \mid \sigma_4 \mid \sigma_1) \odot \Gamma, x : A, \Gamma' \vdash t_2 : A'\end{array}}{(\Delta, \sigma, \Delta' \mid \sigma_5 + s' * \sigma_4 \mid \sigma_6 + r' * \sigma_4) \odot \Gamma, x : A, \Gamma' \vdash t_1\, t_2 : [t_2/x]B} \text{T\_APP}$$

with $\sigma_5 + s' * \sigma_4 = \sigma_2, s, \sigma'_2$ and $\sigma_6 + r' * \sigma_4 = \sigma_3, r, \sigma'_3$ thus let $\sigma_5 = \sigma'_5, s_1, \sigma''_5$ and $\sigma_4 = \sigma'_4, s_2, \sigma''_4$ and $\sigma_6 = \sigma'_6, r_1, \sigma''_6$ and $\sigma_1 = \sigma'_1, r_3, \sigma''_1$.

therefore we have:

$$\begin{aligned}
\sigma_2 &= \sigma_5' + s' * \sigma_4'' \\
s &= s_1 + s' * s_2 \\
\sigma_2' &= \sigma_5'' + s' * \sigma_4'' \\
\sigma_3 &= \sigma_6' + r' * \sigma_4'' \\
r &= r_1 + r' * s_2 \\
\sigma_3' &= \sigma_6'' + r' * \sigma_4''
\end{aligned}$$

By induction we then have:

$$(\Delta, (\Delta' \backslash \pi), (\sigma_1', \sigma_1'') \mid \sigma_6', \sigma_6'', r' \mid \mathbf{0}) \odot \Gamma, \Gamma', y : A' \vdash B : \mathsf{Type}_l \tag{ih0}$$

$$(\Delta, (\Delta' \backslash \pi) \mid \sigma_5', \sigma_5'' \mid (\sigma_1' + \sigma_6'), (\sigma_1'' + \sigma_6'')) \odot \Gamma, \Gamma' \vdash t_1 : (y :_{(s', r')} A') \to B \tag{ih1}$$

$$(\Delta, (\Delta' \backslash \pi) \mid \sigma_4', \sigma_4'' \mid \sigma_1', \sigma_1'') \odot \Gamma, \Gamma' \vdash t_2 : A' \tag{ih2}$$

Then we can form the derivation:

$$\frac{(\text{ih0}) \qquad (\text{ih1}) \qquad (\text{ih2})}{(\Delta, (\Delta' \backslash \pi) \mid \sigma_5', \sigma_5'' + s' * \sigma_4', \sigma_4'' \mid \sigma_6', \sigma_6'' + r' * \sigma_4', \sigma_4'') \odot \Gamma, \Gamma' \vdash t_1\, t_2 : [t_2/x]B} \text{ T\_App}$$

whose concluding judgment is equal to :

$$(\Delta, (\Delta' \backslash \pi) \mid \sigma_5' + s' * \sigma_4', \sigma_5'' + s' * \sigma_4'' \mid \sigma_6' + r' * \sigma_4', \sigma_6'' + r' * \sigma_4'') \odot \Gamma, \Gamma' \vdash t_1\, t_2 : [t_2/x]B$$

thus matching the goal

- Case T_Pair

$$\frac{\begin{array}{c}(\Delta, \sigma, \Delta', \sigma_1 \mid \sigma_6, r' \mid \mathbf{0}) \odot \Gamma, x : A, \Gamma', y : A' \vdash B : \mathsf{Type}_l \\ (\Delta, \sigma, \Delta' \mid \sigma_5 \mid \sigma_1) \odot \Gamma, x : A, \Gamma' \vdash t_1 : A' \\ (\Delta, \sigma, \Delta' \mid \sigma_4 \mid \sigma_6 + r' * \sigma_5) \odot \Gamma, x : A, \Gamma' \vdash t_2 : [t_1/x]B\end{array}}{(\Delta, \sigma, \Delta' \mid \sigma_5 + \sigma_4 \mid \sigma_1 + \sigma_6) \odot \Gamma, x : A, \Gamma' \vdash (t_1, t_2) : (y :_{r'} A') \otimes B} \text{ T\_Pair}$$

with $\sigma_2, s, \sigma_2' = \sigma_5 + \sigma_4$ thus we have $\sigma_5 = \sigma_5', s_1, \sigma_5''$ and $\sigma_4 = \sigma_4', s_2, \sigma_4''$ and with $\sigma_3, r, \sigma_3' = \sigma_1 + \sigma_6$ thus we have $\sigma_1 = \sigma_1', r_1, \sigma_1''$ and $\sigma_6 = \sigma_6', r_2, \sigma_6''$ such that:

$$\begin{aligned}
\sigma_2 &= \sigma_5' + \sigma_4' \\
s &= s_1 + s_2 \\
\sigma_2' &= \sigma_5'' + \sigma_4'' \\
\sigma_3 &= \sigma_1' + \sigma_6' \\
r &= r_1 + r_2 \\
\sigma_3' &= \sigma_1'' + \sigma_6''
\end{aligned}$$

By induction we have that:

$$(\Delta, (\Delta' \backslash \pi), (\sigma_1', \sigma_1'') \mid \sigma_6', \sigma_6'', r' \mid \mathbf{0}) \odot \Gamma, \Gamma', y : A' \vdash B : \mathsf{Type}_l \tag{ih1}$$

$$(\Delta, (\Delta' \backslash \pi) \mid \sigma_5', \sigma_5'' \mid \sigma_1', \sigma_1'') \odot \Gamma, \Gamma' \vdash t_1 : A' \tag{ih2}$$

$$(\Delta, (\Delta' \backslash \pi) \mid \sigma_4', \sigma_4'' \mid \sigma_6' + r' * \sigma_5', \sigma_6'' + r' * \sigma_6'') \odot \Gamma, \Gamma' \vdash t_2 : [t_1/x]B \tag{ih3}$$

From which we construct the application:

$$\frac{\text{(ih1)} \qquad \text{(ih2)} \qquad \text{(ih3)}}{(\Delta, (\Delta' \backslash \pi) \mid \sigma'_5, \sigma''_5 + \sigma'_4, \sigma''_4 \mid \sigma'_1, \sigma''_1 + \sigma'_6, \sigma''_6) \odot \Gamma, \Gamma' \vdash (t_1, t_2) : (y :_{r'} A') \otimes B} \text{ T\_PAIR}$$

which is equal to the judgment:

$$(\Delta, (\Delta' \backslash \pi) \mid \sigma'_5 + \sigma'_4, \sigma''_4 + \sigma''_5 \mid \sigma'_1 + \sigma'_6, \sigma''_1 + \sigma''_6) \odot \Gamma, \Gamma' \vdash (t_1, t_2) : (y :_{r'} A') \otimes B$$

satisfying the goal of this case.

- Case T_TENCUT

$$\frac{\begin{array}{c}(\Delta, \sigma, \Delta' \mid \sigma_7 \mid \sigma_1 + \sigma_6) \odot \Gamma, x : A, \Gamma' \vdash t_1 : (w :_{r'} A') \otimes B \\ (\Delta, \sigma, \Delta', (\sigma_1 + \sigma_6) \mid \sigma_5, r'' \mid \mathbf{0}) \odot \Gamma, x : A, \Gamma', z : (w :_{r'} A') \otimes B \vdash C : \mathsf{Type}_l \\ (\Delta, \sigma, \Delta', \sigma_1, (\sigma_6, r') \mid \sigma_4, s', s' \mid \sigma_5, r'', r'') \odot \Gamma, x : A, \Gamma', w : A', y : B \vdash t_2 : [(w, y)/z]C\end{array}}{(\Delta, \sigma, \Delta' \mid \sigma_4 + s' * \sigma_7 \mid \sigma_5 + r'' * \sigma_7) \odot \Gamma, x : A, \Gamma' \vdash \mathsf{let}\,(w, y) = t_1 \,\mathsf{in}\, t_2 : [t_1/z]C} \text{ T\_TENCUT}$$

where $\sigma_2, s, \sigma'_2 = \sigma_4 + s' * \sigma_7$ and $\sigma_3, r, \sigma'_3 = \sigma_5 + r'' * \sigma_7$. Let $\sigma_4 = \sigma'_4, s_1, \sigma''_4$ and $\sigma_7 = \sigma'_7, s_2, \sigma''_7$ and $\sigma_5 = \sigma'_5, r_1, \sigma''_5$ and $\sigma_1 = \sigma'_1, r_2, \sigma''_1$ and $\sigma_6 = \sigma'_6, r_3, \sigma''_6$ then

$$\begin{aligned} \sigma_2 &= \sigma'_4 + s' * \sigma'_7 \\ s &= s_1 + s' * s_2 \\ \sigma'_2 &= \sigma''_4 + s' * \sigma''_7 \\ \sigma_3 &= \sigma'_5 + r'' * \sigma''_5 \\ r &= r_1 + r'' * s_2 \\ \sigma'_3 &= \sigma''_5 + r'' * \sigma''_7 \end{aligned}$$

Then by induction we have:

$$(\Delta, (\Delta' \backslash \pi) \mid \sigma'_7, \sigma''_7 \mid \sigma'_1 + \sigma'_6, \sigma''_1 + \sigma''_6) \odot \Gamma, \Gamma' \vdash t_1 : (w :_{r'} A') \otimes B \quad \text{(ih1)}$$

$$(\Delta, (\Delta' \backslash \pi), (\sigma'_1 + \sigma'_6, \sigma''_1 + \sigma''_6) \mid \sigma'_5, \sigma''_5, r'' \mid \mathbf{0}) \odot \Gamma, \Gamma', z : (w :_{r'} A') \otimes B \vdash C : \mathsf{Type}_l \quad \text{(ih2)}$$

$$(\Delta, (\Delta' \backslash \pi), (\sigma'_1, \sigma''_1), (\sigma'_6, \sigma''_6, r') \mid \sigma'_4, \sigma''_4, s', s' \mid \sigma'_5, \sigma''_5, r'', r'') \odot \Gamma, \Gamma', w : A', y : B \vdash t_2 : [(w, y)/z]C \quad \text{(ih3)}$$

Then we can form the derivation:

$$\frac{\text{(ih1)} \quad \text{(ih2)} \quad \text{(ih3)}}{(\Delta, (\Delta' \backslash \pi) \mid \sigma'_4, \sigma''_4 + s' * (\sigma'_7, \sigma''_7) \mid \sigma'_5, \sigma''_5 + r'' * \sigma'_7, \sigma''_7) \odot \Gamma, \Gamma' \vdash \mathsf{let}\,(w, y) = t_1 \,\mathsf{in}\, t_2 : [t_1/z]C} \text{ T\_TENCUT}$$

giving the equivalent judgment:

$$(\Delta, (\Delta' \backslash \pi) \mid \sigma'_4 + s' * \sigma'_7, \sigma''_4 + s' * \sigma''_7 \mid \sigma'_5 + r'' * \sigma'_7, \sigma''_5 + r'' * \sigma''_7) \odot \Gamma, \Gamma' \vdash \mathsf{let}\,(w, y) = t_1 \,\mathsf{in}\, t_2 : [t_1/z]C$$

satisfying the goal.

- Case T_Box

$$\frac{(\Delta, \sigma_1, \Delta' \mid \sigma \mid \mathbf{0}) \odot \Gamma, x : A, \Gamma' \vdash A' : \mathsf{Type}_l}{(\Delta, \sigma_1, \Delta' \mid \sigma \mid \mathbf{0}) \odot \Gamma, x : A, \Gamma' \vdash \Box_{s'} A' : \mathsf{Type}_l} \text{ T\_Box}$$

where $\sigma = \sigma_2, s, \sigma_2'$ and $\mathbf{0} = \sigma_3, r, \sigma_3'$ thus let $\sigma = \sigma', s_1, \sigma''$.
By induction on the first premise we have:

$$(\Delta, (\Delta' \backslash \pi) \mid \sigma', \sigma'' \mid \mathbf{0}) \odot \Gamma, \Gamma' \vdash A' : \mathsf{Type}_l \tag{ih}$$

Thus we can form the derivation:

$$\frac{\text{(ih)}}{(\Delta, (\Delta' \backslash \pi) \mid \sigma', \sigma'' \mid \mathbf{0}) \odot \Gamma, \Gamma' \vdash \Box_{s'} A' : \mathsf{Type}_l} \text{ T\_Box}$$

Satisfying the goal.

- Case T_BoxI

$$\frac{(\Delta, \sigma, \Delta' \mid \sigma_1 \mid \sigma_3, r, \sigma_3') \odot \Gamma, x : A, \Gamma' \vdash t : A'}{(\Delta, \sigma, \Delta' \mid s' * \sigma_1 \mid \sigma_3, r, \sigma_3') \odot \Gamma, x : A, \Gamma' \vdash \Box t : \Box_{s'} A'} \text{ T\_BoxI}$$

where $s' * \sigma_1 = \sigma_2, s, \sigma_2'$ thus let: $\sigma_1 = \sigma_1', s_1, \sigma_1''$ then:

$$\begin{aligned} \sigma_2 &= s' * \sigma_1' \\ s &= s' * s_1 \\ \sigma_2' &= s' * \sigma_1'' \end{aligned}$$

By induction on the first premise have that:

$$(\Delta, (\Delta' \backslash \pi) \mid \sigma_1', \sigma_1'' \mid \sigma_3, \sigma_3') \odot \Gamma, \Gamma' \vdash t : A' \tag{ih}$$

$$\frac{\text{(ih)}}{(\Delta, (\Delta' \backslash \pi) \mid s' * \sigma_1', \sigma_1'' \mid \sigma_3, \sigma_3') \odot \Gamma, \Gamma' \vdash \Box t : \Box_{s'} A'} \text{ T\_BoxI}$$

which yields the equivalent judgment:

$$(\Delta, (\Delta' \backslash \pi) \mid s' * \sigma_1', s' * \sigma_1'' \mid \sigma_3, \sigma_3') \odot \Gamma, \Gamma' \vdash \Box t : \Box_{s'} A'$$

satisfying the goal

- Case T_BoxE

$$\frac{\begin{array}{c}(\Delta, \sigma, \Delta', \sigma_5 \mid \sigma_4, r' \mid \mathbf{0}) \odot \Gamma, x : A, \Gamma', z : \Box_{s'} A' \vdash B : \mathsf{Type}_l \\ (\Delta, \sigma, \Delta' \mid \sigma_1 \mid \sigma_5) \odot \Gamma, x : A, \Gamma' \vdash t_1 : \Box_{s'} A' \\ (\Delta, \sigma, \Delta', \sigma_5 \mid \sigma_7, s' \mid \sigma_4, (s' * r')) \odot \Gamma, x : A, \Gamma', y : A' \vdash t_2 : [\Box y / z] B\end{array}}{(\Delta, \sigma, \Delta' \mid \sigma_1 + \sigma_7 \mid \sigma_4 + r' * \sigma_1) \odot \Gamma, x : A, \Gamma' \vdash \mathsf{let}\, \Box x = t_1 \,\mathsf{in}\, t_2 : [t_1 / z] B} \text{ T\_BoxE}$$

where $\sigma_2, s, \sigma'_2 = \sigma_1 + \sigma_7$ and $\sigma_3, r, \sigma'_3 = \sigma_4 + r' * \sigma_1$. Let $\sigma_1 = \sigma'_1, s_1, \sigma''_1$ and $\sigma_7 = \sigma'_7, s_2, \sigma''_7$ and $\sigma_6 = \sigma'_6, r_1, \sigma''_6$ and $\sigma_4 = \sigma'_4, r_2, \sigma''_4$ and $\sigma_5 = \sigma'_5, r_3, \sigma''_5$ then we have that:

$$\begin{aligned}
\sigma_2 &= \sigma'_1 + \sigma'_7 \\
s &= s_1 + s_2 \\
\sigma'_2 &= \sigma''_1 + \sigma''_7 \\
\sigma_3 &= \sigma'_4 + r' * \sigma'_6 \\
r &= r_1 + r_2 \\
\sigma'_3 &= \sigma''_4 + r' * \sigma''_6
\end{aligned}$$

Thus by induction we get that:

$$(\Delta, (\Delta' \backslash \pi), (\sigma'_5, \sigma''_5) \mid \sigma'_4, \sigma''_4, r' \mid \mathbf{0}) \odot \Gamma, \Gamma', z : \Box_{s'} A' \vdash B : \mathsf{Type}_l \qquad \text{(ih0)}$$
$$(\Delta, (\Delta' \backslash \pi) \mid \sigma'_1, \sigma''_1 \mid \sigma'_5, \sigma''_5) \odot \Gamma, \Gamma' \vdash t_1 : \Box_{s'} A' \qquad \text{(ih1)}$$
$$(\Delta, (\Delta' \backslash \pi), (\sigma'_5, \sigma''_5) \mid \sigma'_7, \sigma''_7, s' \mid \sigma'_4, \sigma''_4, (s' * r')) \odot \Gamma, \Gamma', y : A' \vdash t_2 : [\Box y/z]B \qquad \text{(ih2)}$$

then we can form the derivation:

$$\frac{\text{(ih0)} \qquad \text{(ih1)} \qquad \text{(ih2)}}{(\Delta, (\Delta' \backslash \pi) \mid \sigma'_1, \sigma''_1 + \sigma'_7, \sigma''_7 \mid \sigma'_4, \sigma''_4 + r' * \sigma'_6, \sigma''_6) \odot \Gamma, \Gamma' \vdash \mathsf{let}\,\Box x = t_1\,\mathsf{in}\,t_2 : [t_1/z]B}\ \textsc{T\_BoxE}$$

which is equal to the judgment:

$$(\Delta, (\Delta' \backslash \pi) \mid \sigma'_1 + \sigma'_7, \sigma''_1 + \sigma''_7 \mid (\sigma'_4 + r' * \sigma'_6), (\sigma''_4 + r' * \sigma''_6)) \odot \Gamma, \Gamma' \vdash \mathsf{let}\,\Box x = t_1\,\mathsf{in}\,t_2 : [t_1/z]B$$

satisfying the goal.

- Case T_TyConv

$$\frac{(\Delta, \sigma_1, \Delta' \mid \sigma_2, s, \sigma'_2 \mid \sigma_3, r, \sigma'_3) \odot \Gamma, x : A, \Gamma' \vdash t : C \quad (\Delta, \sigma_1, \Delta' \mid \sigma_3, r, \sigma'_3) \odot \Gamma, x : A, \Gamma' \vdash C \leq B}{(\Delta, \sigma_1, \Delta' \mid \sigma_2, s, \sigma'_2 \mid \sigma_3, r, \sigma'_3) \odot \Gamma, x : A, \Gamma' \vdash t : B}\ \textsc{T\_TyConv}$$

By induction we have that:

$$(\Delta, (\Delta' \backslash \pi) \mid \sigma_2, \sigma'_2 \mid \sigma_3, \sigma'_3) \odot \Gamma, \Gamma' \vdash t : C \qquad \text{(ih1)}$$
$$(\Delta, (\Delta' \backslash \pi) \mid \sigma_3, \sigma'_3) \odot \Gamma, \Gamma' \vdash C \leq B \qquad \text{(ih2)}$$

Then we can build the judgment:

$$\frac{\text{(ih1)} \qquad \text{(ih2)}}{(\Delta, (\Delta' \backslash \pi) \mid \sigma_2, \sigma'_2 \mid \sigma_3, \sigma'_3) \odot \Gamma, \Gamma' \vdash t : B}\ \textsc{T\_TyConv}$$

As required.

For subtyping, by standard induction and re-application (see Section A.8).
For equality, by standard induction and re-application (see Section A.8). $\square$

Lemma 3.3 (Judgmental contexts are well-formed)**.** *If* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \mathcal{J}$*, then* $\Delta \odot \Gamma \vdash$*.*

Proof. For well-formedness: our goal is the premise.
For typing: By induction on the form of $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$, as follows:

Case.

$$\frac{\Delta \odot \Gamma \vdash}{(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}}\ \textsc{T\_Type}$$

We need to show $\Delta \odot \Gamma \vdash$, which holds by premise.

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : \mathsf{Type}_{l_1 \sqcup l_2}}\ \textsc{T\_Arrow}$$

This holds by induction on the typing premise for A. Each of the cases for T_Ten, T_App, T_Pair, T_Box, T_BoxI, T_BoxE, and T_TenCut proceed similarly.

Case.

$$\frac{\Delta_1, \sigma, \Delta_2 \odot \Gamma_1, x : A, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \sigma, \Delta_2 \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \sigma, 0, \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash x : A}\ \textsc{T\_Var}$$

We need to show $\Delta_1, \sigma_1, \Delta_2 \odot \Gamma_1, x : A, \Gamma_2 \vdash$, which holds by premise.

Case.

$$\frac{(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \quad (\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t : B}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t : (x :_{(s,r)} A) \to B}\ \textsc{T\_Fun}$$

Our goal is $\Delta \odot \Gamma \vdash$. By induction on the typing premise for t, we know that $\Delta, \sigma_1 \odot \Gamma, x : A \vdash$, therefore, we have $\Delta \odot \Gamma \vdash$ by Lemma 3.4, as required.

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash A \leq B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : B}\ \textsc{T\_TyConv}$$

Our goal is $\Delta \odot \Gamma \vdash$. This holds by induction on the typing premise for $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$.

For subtyping: all cases proceed trivially by induction.

For equality: all cases proceed trivially by induction, with the exception of TEQ_FUN, which proceeds similarly to T_FUN. ☐

Lemma 3.2 (Judgments determine vector sizing). *If $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \mathcal{J}$, then $|\sigma_1| = |\sigma_2| = |\Delta| = |\Gamma|$ and for each element σ of Δ, the size of σ is the same as its index.*

Proof. This holds by assumption over judgments. This may also be proven inductively, by adding size annotations to occurrences of $\mathbf{0}$. ☐

Lemma 3.4 (Subcontext well-formedness). *If $|\Delta_1| = |\Gamma_1|$ and $\Delta_1, \Delta_2 \odot \Gamma_1, \Gamma_2 \vdash$, then $\Delta_1 \odot \Gamma_1 \vdash$.*

Proof. By induction on the definition of well-formed contexts.

- Case WF_EMPTY. Trivial since it does not match the form of the lemma.
- Case WF_EXT We consider two cases depending on the syntactic structure of Γ_2 and Δ_2 (simultaneously, since they must have the same size by the lemma statement).
  - $\Gamma_2 = \emptyset$ and $\Delta_2 = \emptyset$ then this lemma is trivial since we have $\Delta_1 \odot \Gamma_1 \vdash$ already.
  - $\Gamma_2 = \Gamma_2', x : A$ and $\Delta_2 = \Delta_2', \sigma$ then we have:

$$\frac{(\Delta_1, \Delta_2' \mid \sigma \mid \mathbf{0}) \odot \Gamma_1, \Gamma_2 \vdash A : \mathsf{Type}_l}{\Delta_1, \Delta_2', \sigma \odot \Gamma_1, \Gamma_2, x : A \vdash} \text{WF\_EXT}$$

    By Lemma 3.3 then we have $\Delta_1, \Delta_2' \odot \Gamma_1, \Gamma_2 \vdash$
    By induction on this then we get that $\Delta_1' \odot \Gamma_1 \vdash$ which is the goal of this case.

☐

Lemma 3.5 (Typing an assumption in a judgmental context). *If $(\Delta_1, \sigma, \Delta_2 \mid \sigma_1 \mid \sigma_2) \odot \Gamma_1, x : A, \Gamma_2 \vdash \mathcal{J}$ with $|\Delta_1| = |\Gamma_1|$, then $(\Delta_1 \mid \sigma \mid \mathbf{0}) \odot \Gamma_1 \vdash A : \mathsf{Type}_l$, for some level l.*

Proof. By Lemma 3.3 we have $\Delta_1, \sigma, \Delta_2 \odot \Gamma_1, x : A, \Gamma_2 \vdash$, therefore by Lemma 3.4 we have $\Delta_1, \sigma \odot \Gamma_1, x : A \vdash$, which must come from the following derivation:

$$\frac{(\Delta_1 \mid \sigma \mid \mathbf{0}) \odot \Gamma_1 \vdash A : \mathsf{Type}_l}{\Delta_1, \sigma \odot \Gamma_1, x : A \vdash} \text{WF\_EXT}$$

Therefore we have $(\Delta_1 \mid \sigma \mid \mathbf{0}) \odot \Gamma_1 \vdash A : \mathsf{Type}_l$ by the premise, as required. ☐

Lemma 3.6 (Typing the type of a term). *Given $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$, then there exists a level l such that $(\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l$.*

Proof. By induction on the form of $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$, as follows:

Case.

$$\frac{\Delta \odot \Gamma \vdash}{(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}} \text{T\_TYPE}$$

Then we will show $(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_{\mathsf{suc}\ l} : \mathsf{Type}_{\mathsf{suc}\ \mathsf{suc}\ l}$ which holds by T_TYPE and the well-formedness premise for Γ.

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : \mathsf{Type}_{l_1 \sqcup l_2}} \text{ T\_ARROW}$$

Then we will show $(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_{l_1 \sqcup l_2} : \mathsf{Type}_{l_3}$ for some level l_3. Consider $l_2 \leq l_1$, then we have $l_1 \sqcup l_2 = l_1$, and thus need to show $(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_{l_1} : \mathsf{Type}_{l_3}$, which holds by induction on the typing premise for A. Consider $l_1 < l_2$, then we have $l_1 \sqcup l_2 = l_2$, and thus need to show $(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_{l_2} : \mathsf{Type}_{l_3}$. By induction on the typing premise for B, we have $(\Delta, \sigma_1 \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma, x : A \vdash \mathsf{Type}_{l_2} : \mathsf{Type}_{l_3}$, therefore, by Lemma 3.1, we have $(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_{l_2} : \mathsf{Type}_{l_3}$, as required.
The proof for T_TEN follows the same process.

Case.

$$\frac{\Delta_1, \sigma, \Delta_2 \odot \Gamma_1, x : A, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \sigma, \Delta_2 \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \sigma, 0, \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash x : A} \text{ T\_VAR}$$

Then we will show $(\Delta_1, \sigma, \Delta_2 \mid \sigma, 0, \mathbf{0} \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash A : \mathsf{Type}_l$ (for some level l). This holds by the following derivation:

$$\frac{\Delta_1, \sigma, \Delta_2 \odot \Gamma_1, x : A, \Gamma_2 \vdash \quad \dfrac{\Delta_1, \sigma, \Delta_2 \odot \Gamma_1, x : A, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1 \mid \sigma \mid \mathbf{0}) \odot \Gamma_1 \vdash A : \mathsf{Type}_l} \text{ LEMMA 3.5}}{(\Delta_1, \sigma, \Delta_2 \mid \sigma, 0, \mathbf{0} \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash A : \mathsf{Type}_l} \text{ LEMMA 3.25}$$

Case.

$$\frac{(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \quad (\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t : B}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t : (x :_{(s,r)} A) \to B} \text{ T\_FUN}$$

Then we will show $(\Delta \mid \sigma_1 + \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : \mathsf{Type}_{l'}$ for some level l'. Using Lemma 3.5 on the typing premise for t gives $(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l_1}$ (for some level l_1). We can then achieve our goal via an application to T_ARROW, as follows:

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l}{(\Delta \mid \sigma_1 + \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : \mathsf{Type}_{l_1 \sqcup l}} \text{ T\_ARROW}$$

Case.

$$\frac{\begin{array}{c}(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t_1 : (x :_{(s,r)} A) \to B \quad (\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : A\end{array}}{(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash t_1\, t_2 : [t_2/x]B} \text{T\_App}$$

Then we will show $(\Delta \mid \sigma_3 + r * \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash [t_2/x]B : \mathsf{Type}_l$. We have the following:

1. $(\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : A$
2. $(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l$
3. $|\sigma_3| = |\mathbf{0}| = |\Gamma|$ (trivially, and by Lemma 3.2)

Which we can pass to Lemma 3.27, to obtain $(\Delta \mid \sigma_3 + r * \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash [t_2/x]B : \mathsf{Type}_l$, as required.

Case.

$$\frac{\begin{array}{c}(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma \vdash t_1 : A \quad (\Delta \mid \sigma_4 \mid \sigma_3 + r * \sigma_2) \odot \Gamma \vdash t_2 : [t_1/x]B\end{array}}{(\Delta \mid \sigma_2 + \sigma_4 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash (t_1, t_2) : (x :_r A) \otimes B} \text{T\_Pair}$$

Then we will show $(\Delta \mid \sigma_1 + \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash (x :_r A) \otimes B : \mathsf{Type}_{l'}$ for some level l'. Applying Lemma 3.5 to the typing premise for B, we have $(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l''}$ (for some level l''). We can then apply this result, along with the typing premise for B to T_Ten, to obtain:

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l''} \quad (\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l}{(\Delta \mid \sigma_1 + \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash (x :_r A) \otimes B : \mathsf{Type}_{l'' \sqcup l}} \text{T\_Ten}$$

As required.

Case.

$$\frac{\begin{array}{c}(\Delta \mid \sigma_3 \mid \sigma_1 + \sigma_2) \odot \Gamma \vdash t_1 : (x :_r A) \otimes B \\ (\Delta, (\sigma_1 + \sigma_2) \mid \sigma_5, r' \mid \mathbf{0}) \odot \Gamma, z : (x :_r A) \otimes B \vdash C : \mathsf{Type}_l \\ (\Delta, \sigma_1, (\sigma_2, r) \mid \sigma_4, s, s \mid \sigma_5, r', r') \odot \Gamma, x : A, y : B \vdash t_2 : [(x, y)/z]C\end{array}}{(\Delta \mid \sigma_4 + s * \sigma_3 \mid \sigma_5 + r' * \sigma_3) \odot \Gamma \vdash \mathsf{let}\,(x, y) = t_1 \,\mathsf{in}\, t_2 : [t_1/z]C} \text{T\_TenCut}$$

Then we will show $(\Delta \mid \sigma_5 + r' * \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash [t_1/z]C : \mathsf{Type}_l$. We can form the following premises:

1. $(\Delta \mid \sigma_3 \mid \sigma_1 + \sigma_2) \odot \Gamma \vdash t_1 : (x :_r A) \otimes B$
2. $(\Delta, (\sigma_1 + \sigma_2) \mid \sigma_5, r' \mid \mathbf{0}) \odot \Gamma, z : (x :_r A) \otimes B \vdash C : \mathsf{Type}_l$
3. $|\sigma_5| = |\mathbf{0}| = |\Gamma|$ (trivially, and by Lemma 3.2)

Which we can pass to Lemma 3.27, to obtain $(\Delta \mid \sigma_5 + r' * \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash [t_1/z]C : \mathsf{Type}_l$, as required.

Case.

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A : \mathsf{Type}_l}\ \text{T\_Box}$$

Then we will show $(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}$ which holds by the following derivation:

$$\frac{\dfrac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l}{\Delta \odot \Gamma \vdash}\ \text{L. 3.3}}{(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}}\ \text{T\_Type}$$

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A}{(\Delta \mid s * \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box t : \Box_s A}\ \text{T\_BoxI}$$

Then we will show $(\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A : \mathsf{Type}_l$, for some level l. By induction on the typing premise for t, we have $(\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l$, therefore, we can perform the following application to achieve our goal:

$$\frac{(\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l}{(\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A : \mathsf{Type}_l}\ \text{T\_Box}$$

Case.

$$\frac{\begin{array}{l}(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : \Box_s A \\ (\Delta, \sigma_2 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, z : \Box_s A \vdash B : \mathsf{Type}_l \\ (\Delta, \sigma_2 \mid \sigma_3, s \mid \sigma_4, (s * r)) \odot \Gamma, x : A \vdash t_2 : [\Box x/z]B\end{array}}{(\Delta \mid \sigma_1 + \sigma_3 \mid \sigma_4 + r * \sigma_1) \odot \Gamma \vdash \mathsf{let}\,\Box x = t_1\,\mathsf{in}\,t_2 : [t_1/z]B}\ \text{T\_BoxE}$$

Then we will show $(\Delta \mid \sigma_4 + r * \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash [t_1/z]B : \mathsf{Type}_l$. We can form the following premises:

1. $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : \Box_s A$
2. $(\Delta, \sigma_2 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, z : \Box_s A \vdash B : \mathsf{Type}_l$
3. $|\sigma_4| = |\mathbf{0}| = |\Gamma|$ (trivially, and by Lemma 3.2)

Which we can pass to Lemma 3.27, to obtain $(\Delta \mid \sigma_4 + r * \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash [t_1/z]B : \mathsf{Type}_l$, as required.

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash A \leq B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : B}\ \text{T\_TyConv}$$

Then we will show $(\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash B : \mathsf{Type}_l$, which holds by Lemma 3.34.

$\square$

Lemma 3.7 (Subtyping in context)**.** *If* $(\Delta_1, \sigma_1, \Delta_2 \mid \sigma_2, s, \sigma_3 \mid \sigma_4, r, \sigma_5) \odot \Gamma_1, x : A, \Gamma_2 \vdash \mathcal{J}$ *and* $(\Delta_1 \mid \sigma_1) \odot \Gamma \vdash A' \leq A$*, then* $(\Delta_1, \sigma_1, \Delta_2 \mid \sigma_2, s, \sigma_3 \mid \sigma_4, r, \sigma_5) \odot \Gamma_1, x : A', \Gamma_2 \vdash \mathcal{J}$*.*

Proof. For well-formed contexts:

Case.

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l}{\Delta, \sigma \odot \Gamma, x : A \vdash} \text{ WF\_EXT}$$

With $(\Delta \mid \mathbf{0}) \odot \Gamma \vdash A' \leq A$. Then our goal holds by the following derivation:

$$\frac{\dfrac{(\Delta \mid \sigma) \odot \Gamma \vdash A' \leq A}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A' : \mathsf{Type}_l} \text{ L. 3.34}}{\Delta, \sigma \odot \Gamma, x : A' \vdash} \text{ WF\_EXT}$$

Case.

$$\frac{(\Delta_1, \sigma_1, \Delta_2 \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash B : \mathsf{Type}_{l'}}{\Delta_1, \sigma_1, \Delta_2, \sigma_2 \odot \Gamma_1, x : A, \Gamma_2, y : B \vdash} \text{ WF\_EXT}$$

Then our goal holds by the following derivation:

$$\frac{\dfrac{(\Delta_1, \sigma_1, \Delta_2 \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash B : \mathsf{Type}_{l'} \quad (\Delta_1 \mid \sigma_1) \odot \Gamma_1 \vdash A' \leq A}{(\Delta_1, \sigma_1, \Delta_2 \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma_1, x : A', \Gamma_2 \vdash B : \mathsf{Type}_{l'}} \text{ I.H.}}{\Delta_1, \sigma_1, \Delta_2, \sigma_2 \odot \Gamma_1, x : A', \Gamma_2, y : B \vdash} \text{ WF\_EXT}$$

For typing: most cases hold by induction and reapplication. Use $\dfrac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash A \leq B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : B} \text{ T\_TYCON}$ when necessary. For equality, all cases proceed by induction then re-application to respective rules. For subtyping, all cases proceed by induction then re-application to respective rules. $\square$

A.1 Proofs for inversions

Lemma 3.8 (Inversion on arrow typing)**.** *If* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : C$ *then there exist grade vectors* $\hat{\sigma_1}$, $\hat{\sigma_1}'$*, and levels* l *and* l'*, such that* $(\Delta \mid \hat{\sigma_1} \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l'}$ *and* $(\Delta, \hat{\sigma_1} \mid \hat{\sigma_1}', r \mid \boldsymbol{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l''}$*, and* $\hat{\sigma_1} + \hat{\sigma_1}' = \sigma_1$*.*

Proof. By induction on $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : C$, as follows:

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : \mathsf{Type}_{l_1 \sqcup l_2}} \text{ T\_ARROW}$$

Then we have $(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l_1}$, $(\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l_2}$, with $\sigma_1 + \sigma_2 = \sigma_1 + \sigma_2$, as required.

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : D \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash D \leq C}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : C} \text{ T\_TyConv}$$

Then our goal holds by induction.

$\square$

Lemma 3.9 (Inversion on tensor typing). *If $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash (x :_r A) \otimes B : C$ then there exist grade vectors $\hat{\sigma_1}$, $\hat{\sigma_1}'$, and levels l and l', such that $(\Delta \mid \hat{\sigma_1} \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l'}$ and $(\Delta, \hat{\sigma_1} \mid \hat{\sigma_1}', r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l''}$, and $\hat{\sigma_1} + \hat{\sigma_1}' = \sigma_1$.*

Proof. By induction on $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash (x :_r A) \otimes B : C$, as follows:

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_r A) \otimes B : \mathsf{Type}_{l_1 \sqcup l_2}} \text{ T\_Ten}$$

Then we have $(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l_1}$, $(\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l_2}$, with $\sigma_1 + \sigma_2 = \sigma_1 + \sigma_2$, as required.

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash (x :_r A) \otimes B : D \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash D \leq C}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash (x :_r A) \otimes B : C} \text{ T\_TyConv}$$

Then our goal holds by induction.

$\square$

Lemma 3.10 (Inversion on box typing). *If $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box_s A : B$, then $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash A : \mathsf{Type}_l$, for some level l.*

Proof. By induction on $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box_s A : B$, as follows:

Case.

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A : \mathsf{Type}_l} \text{ T\_Box}$$

Then we have $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l$, as required.

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box_{s} A : C \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash C \leq B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box_{s} A : B} \text{ T\_TyConv}$$

Then our goal holds by induction.

□

Lemma 3.11 (Function inversion)**.** *If* $(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t : C$*,* $(\Delta \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash C \leq (x :_{(s,r)} A) \to B$*, and* $(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l$*, then* $(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t : B$*.*

Proof. By induction on the form of $(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t : C$, as follows:

Case.

$$\frac{(\Delta, \sigma_1 \mid \sigma_3, r' \mid \mathbf{0}) \odot \Gamma, x' : A' \vdash B' : \mathsf{Type}_l \quad (\Delta, \sigma_1 \mid \sigma_2, s' \mid \sigma_3, r') \odot \Gamma, x' : A' \vdash t : B'}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t : (x' :_{(s',r')} A') \to B'} \text{ T\_Fun}$$

By Lemma 3.14 we have $x' = x$, $s' = s$, $r' = r$, $(\Delta \mid \sigma_1) \odot \Gamma \vdash A \leq A'$, and $(\Delta, \sigma_1 \mid \sigma_3, r) \odot \Gamma, x : A \vdash B' \leq B$. Therefore we have by Lemma 3.7 that $(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t : B'$, and applying this to T_TyConv, we have $(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t : B$, as required.

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \lambda x.t : C \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash C \leq D}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \lambda x.t : D} \text{ T\_TyConv}$$

Then our goal holds by the following derivation:

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \lambda x.t : C \quad \dfrac{(\Delta \mid \sigma_2) \odot \Gamma \vdash C \leq D \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash D \leq (x :_{(s,r)} A) \to B}{(\Delta \mid \sigma_2) \odot \Gamma \vdash C \leq (x :_{(s,r)} A) \to B} \text{ ST\_Trans}}{(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t : B} \text{ I.H.}$$

□

Lemma 3.12 (Pair inversion)**.** *If* $(\Delta \mid \sigma_1 \mid \sigma_2 + \sigma_3) \odot \Gamma \vdash (t_1, t_2) : (x :_{r} A) \otimes B$ *and* $(\Delta, \sigma_2 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l$*, then there exist grade vectors* $\hat{\sigma_1}$ *and* $\hat{\sigma_1}'$ *such that* $(\Delta \mid \hat{\sigma_1} \mid \sigma_2) \odot \Gamma \vdash t_1 : A$*, and* $(\Delta \mid \hat{\sigma_1}' \mid \sigma_3 + r * \hat{\sigma_1}) \odot \Gamma \vdash t_2 : [t_1/x]B$*, with* $\hat{\sigma_1} + \hat{\sigma_1}' = \sigma_1$*.*

Proof. Proof proceeds similarly to the proof for Lemma 3.11, but using Lemma 3.15. □

Lemma 3.13 (Box inversion)**.** *If* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box t : B$ *and* $(\Delta \mid \sigma_2) \odot \Gamma \vdash B \leq \Box_{s} A$*, then* $(\Delta \mid \hat{\sigma_1} \mid \sigma_2) \odot \Gamma \vdash t : A$ *for some* $\hat{\sigma_1}$ *such that* $s * \hat{\sigma_1} = \sigma_1$*.*

Proof. By induction on the form of $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box t : B$, as follows:

Case.

$$\frac{(\Delta \mid \hat{\sigma_1} \mid \sigma_2) \odot \Gamma \vdash t : A'}{(\Delta \mid s' * \hat{\sigma_1} \mid \sigma_2) \odot \Gamma \vdash \Box t : \Box_{s'} A'} \text{ T\_BOXI}$$

Then by Lemma 3.16 we have:

- $s' = s$ (therefore $s' * \hat{\sigma_1} = s * \hat{\sigma_1}$);
- $(\Delta \mid \sigma_2) \odot \Gamma \vdash A' \leq A$;

Then our goal holds by the following derivation:

$$\frac{(\Delta \mid \hat{\sigma_1} \mid \sigma_2) \odot \Gamma \vdash t : A' \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash A' \leq A}{(\Delta \mid \hat{\sigma_1} \mid \sigma_2) \odot \Gamma \vdash t : A} \text{ T\_TYCONV}$$

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box t : C \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash C \leq B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box t : B} \text{ T\_TYCONV}$$

Then our goal holds by the following derivation:

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box t : C \quad \dfrac{(\Delta \mid \sigma_2) \odot \Gamma \vdash C \leq B \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash B \leq \Box_s A}{(\Delta \mid \sigma_2) \odot \Gamma \vdash C \leq \Box_s A} \text{ ST\_TRANS}}{(\Delta \mid \hat{\sigma_1} \mid \sigma_2) \odot \Gamma \vdash t : A} \text{ I.H.}$$

∎

Lemma 3.14 (Arrow subtyping inversion)**.** *If $(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash \mathcal{J}_1$ where $\mathcal{J}_1$ is $(x :_{(s,r)} A) \to B \leq C$ or $(x :_{(s,r)} A) \to B = C : D$ and $(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash \mathcal{J}_2$, where $\mathcal{J}_2$ is respectively $C \leq (y :_{(s',r')} A') \to B'$ or $C = (y :_{(s',r')} A') \to B' : E$ (with $(\Delta \mid \mathbf{0}) \odot \Gamma \vdash D \leq \mathsf{Type}_l$ and $(\Delta \mid \mathbf{0}) \odot \Gamma \vdash E \leq \mathsf{Type}_l$), with $(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, y : A' \vdash B' : \mathsf{Type}_{l'}$, then $x = y$, $s = s'$, $r = r'$, and respectively, based on $\mathcal{J}_1$, $(\Delta \mid \sigma_1) \odot \Gamma \vdash A' \leq A$, and $(\Delta, \sigma_1 \mid \sigma_3, r) \odot \Gamma, x : A' \vdash B \leq B'$, or $(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A' = A : \mathsf{Type}_{l''}$, and $(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A' \vdash B = B' : \mathsf{Type}_{l'''}$.*

Proof. Proof proceeds similarly to the proof for Lemma 3.16. ∎

Lemma 3.15 (Tensor subtyping inversion)**.** *If $(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash \mathcal{J}_1$ where $\mathcal{J}_1$ is $(x :_r A) \otimes B \leq C$ or $(x :_r A) \otimes B = C : D$ and $(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash \mathcal{J}_2$, where $\mathcal{J}_2$ is respectively $C \leq (y :_{r'} A') \otimes B'$ or $C = (y :_{r'} A') \otimes B' : E$ (with $(\Delta \mid \mathbf{0}) \odot \Gamma \vdash D \leq \mathsf{Type}_l$ and $(\Delta \mid \mathbf{0}) \odot \Gamma \vdash E \leq \mathsf{Type}_l$), with $(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, y : A' \vdash B' : \mathsf{Type}_{l'}$, then $x = y$, $s = s'$, $r = r'$, $(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A = A' : \mathsf{Type}_{l''''}$, and respectively, based on $\mathcal{J}_1$, $(\Delta, \sigma_1 \mid \sigma_3, r) \odot \Gamma, x : A' \vdash B \leq B'$, or $(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A' \vdash B = B' : \mathsf{Type}_{l'''}$.*

Proof. Proof proceeds similarly to the proof for Lemma 3.14. ∎

Lemma 3.16 (Box subtyping inversion). *If* $(\Delta \mid \sigma \mid \boldsymbol{0}) \odot \Gamma \vdash \mathcal{J}_1$ *where* $\mathcal{J}_1$ *is* $\Box_s A \leq B$ *or* $\Box_s A = B : D$*, and* $(\Delta \mid \sigma \mid \boldsymbol{0}) \odot \Gamma \vdash \mathcal{J}_2$*, where* $\mathcal{J}_2$ *is respectively* $B \leq \Box_{s'} A'$ *or* $\Box_{s'} A' = B : E$ *(where* $(\Delta \mid \boldsymbol{0}) \odot \Gamma \vdash D \leq \mathsf{Type}_l$ *and* $(\Delta \mid \boldsymbol{0}) \odot \Gamma \vdash E \leq \mathsf{Type}_l$*), then* $s = s'$*, and respectively, based on* $\mathcal{J}_1$*,* $(\Delta \mid \sigma) \odot \Gamma \vdash A \leq A'$*, or* $(\Delta \mid \sigma \mid \boldsymbol{0}) \odot \Gamma \vdash A = A' : \mathsf{Type}_{l'}$*.*

Proof. For subtyping, by induction, as follows:

Case.

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A = B : \mathsf{Type}_l}{(\Delta \mid \sigma) \odot \Gamma \vdash \Box_s A \leq B}\ \text{ST\_EQ}$$

With:

Case.

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash B = \Box_{s'} A' : \mathsf{Type}_l}{(\Delta \mid \sigma) \odot \Gamma \vdash B \leq \Box_{s'} A'}\ \text{ST\_EQ}$$

Then by induction we have $s = s'$ and $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A = A' : \mathsf{Type}_l$, and therefore have $(\Delta \mid \sigma) \odot \Gamma \vdash A \leq A'$ by ST_EQ.

Case.

$$\frac{(\Delta \mid \sigma) \odot \Gamma \vdash B \leq C \quad (\Delta \mid \sigma) \odot \Gamma \vdash C \leq \Box_{s'} A'}{(\Delta \mid \sigma) \odot \Gamma \vdash B \leq \Box_{s'} A'}\ \text{ST\_TRANS}$$

Then by ST_TRANS we have $(\Delta \mid \sigma) \odot \Gamma \vdash \Box_s A \leq C$, and therefore our goal holds by induction.

Case.

$$\frac{(\Delta \mid \sigma) \odot \Gamma \vdash A'' \leq A'}{(\Delta \mid \sigma) \odot \Gamma \vdash \Box_{s'} A'' \leq \Box_{s'} A'}\ \text{ST\_BOX}$$

Then by induction with $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A = \Box_{s'} A'' : \mathsf{Type}_l$ and $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_{s'} A'' = \Box_{s'} A'' : \mathsf{Type}_l$, then we have $s = s'$, and $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A = A'' : \mathsf{Type}_l$, and therefore by ST_EQ we have $(\Delta \mid \sigma) \odot \Gamma \vdash A \leq A''$, and therefore by ST_TRANS, we have $(\Delta \mid \sigma) \odot \Gamma \vdash A \leq A'$, as required.

Case.

$$\frac{(\Delta \mid \sigma) \odot \Gamma \vdash \Box_s A \leq C \quad (\Delta \mid \sigma) \odot \Gamma \vdash C \leq B}{(\Delta \mid \sigma) \odot \Gamma \vdash \Box_s A \leq B} \text{ ST\_TRANS}$$

Then by ST_TRANS we have $(\Delta \mid \sigma) \odot \Gamma \vdash C \leq \Box_{s'} A'$, and our goal holds by induction.

Case.

$$\frac{(\Delta \mid \sigma) \odot \Gamma \vdash A \leq A''}{(\Delta \mid \sigma) \odot \Gamma \vdash \Box_s A \leq \Box_s A''} \text{ ST\_BOX}$$

With:

Case.

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A'' = \Box_{s'} A' : \mathsf{Type}_l}{(\Delta \mid \sigma) \odot \Gamma \vdash \Box_s A'' \leq \Box_{s'} A'} \text{ ST\_EQ}$$

Then by induction with $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A'' = \Box_{s'} A' : \mathsf{Type}_l$ and $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_{s'} A' = \Box_{s'} A' : \mathsf{Type}_l$, then we have $s = s'$, and $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A'' = A' : \mathsf{Type}_l$, and therefore by ST_EQ we have $(\Delta \mid \sigma) \odot \Gamma \vdash A'' \leq A'$, and therefore by ST_TRANS, we have $(\Delta \mid \sigma) \odot \Gamma \vdash A \leq A'$, as required.

Case.

$$\frac{(\Delta \mid \sigma) \odot \Gamma \vdash \Box_s A'' \leq C \quad (\Delta \mid \sigma) \odot \Gamma \vdash C \leq \Box_{s'} A'}{(\Delta \mid \sigma) \odot \Gamma \vdash \Box_s A'' \leq \Box_{s'} A'} \text{ ST\_TRANS}$$

Then by induction we have $(\Delta \mid \sigma) \odot \Gamma \vdash A'' \leq A'$, and therefore our goal holds by ST_TRANS.

Case.

$$\frac{(\Delta \mid \sigma) \odot \Gamma \vdash A'' \leq A'}{(\Delta \mid \sigma) \odot \Gamma \vdash \Box_s A'' \leq \Box_s A'} \text{ ST\_BOX}$$

Then our goal holds by ST_TRANS.

For equality, by induction, as follows:

Case.

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A = C : D \quad (\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash C = B : D}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A = B : D} \text{ TEQ\_TRANS}$$

Then by TEQ_TRANS we have $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash C = \Box_{s'} A' : D$, and therefore our goal holds by induction.

Case.

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A = B : C \quad (\Delta \mid \mathbf{0}) \odot \Gamma \vdash C \leq D}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A = B : D} \text{ TEQ\_CONVTY}$$

By ST_TRANS, we have $(\Delta \mid \mathbf{0}) \odot \Gamma \vdash C \leq \mathsf{Type}_l$. With:

Case.

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_{s'} A' : E}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_{s'} A' = \Box_{s'} A' : E} \text{ TEQ\_REFL}$$

Then our goal holds by induction.

Case.

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_{s'} A' = B : E}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash B = \Box_{s'} A' : E} \text{ TEQ\_SYM}$$

Then we have by induction that $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A' = A : \mathsf{Type}_l$ and thus by TEQ_SYM than $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A = A' : \mathsf{Type}_l$, as required.

Case.

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash B = C : E \quad (\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash C = \Box_{s'} A' : E}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash B = \Box_{s'} A' : E} \text{ TEQ\_TRANS}$$

Use TEQ_CONVTY to obtain $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A = B : \mathsf{Type}_l$ and $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash C = \Box_{s'} A' : \mathsf{Type}_l$, then use TEQ_TRANS to obtain $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A = C : \mathsf{Type}_l$, and therefore our goal holds by induction.

Case.

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash B = \Box_{s'} A' : C' \quad (\Delta \mid \sigma) \odot \Gamma \vdash C' \leq E}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash B = \Box_{s'} A' : E} \text{ TEQ\_ConvTy}$$

Then by ST_Trans we have $(\Delta \mid \sigma) \odot \Gamma \vdash C' \leq \mathsf{Type}_l$, and our goal holds by induction.

Case.

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A'' = A' : \mathsf{Type}_l}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_{s'} A'' = \Box_{s'} A' : \mathsf{Type}_l} \text{ TEQ\_Box}$$

Then by induction with $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A = \Box_{s'} A'' : D$ and $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_{s'} A'' = \Box_{s'} A'' : \mathsf{Type}_l$, then we have $s = s'$, and $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A = A'' : \mathsf{Type}_l$, and therefore by TEQ_Trans, we have $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A = A' : \mathsf{Type}_{l'}$, as required.

Remaining cases proceed similarly.

$\square$

A.2 Proofs of meta properties

A.2.1 Contraction

Lemma 3.22 (Contraction)**.** *The following rule is admissible:*

$$\frac{\begin{pmatrix} \Delta_1,\sigma_1,(\sigma_1,0),\Delta_2 \\ \sigma_2,s_1,s_2,\sigma_3 \\ \sigma_4,r_1,r_2,\sigma_5 \end{pmatrix} \odot \Gamma_1, x : A, y : A, \Gamma_2 \vdash \mathcal{J} \quad |\Delta_1| = |\sigma_2| = |\sigma_4| = |\Gamma_1|}{\begin{pmatrix} \Delta_1,\sigma_1,\mathsf{contr}(|\Delta_1|;\Delta_2) \\ \sigma_2,(s_1+s_2),\sigma_3 \\ \sigma_4,(r_1+r_2),\sigma_5 \end{pmatrix} \odot \Gamma_1, z : A, [z, z/x, y]\Gamma_2 \vdash [z, z/x, y]\mathcal{J}} \text{ Cont}$$

Proof. For this lemma, it is sufficient to consider that z and x are the same, in which case our goal is:

$$\begin{pmatrix} \Delta_1,\sigma_1,\mathsf{contr}(|\Delta_1|;\Delta_2) \\ \sigma_2,(s_1+s_2),\sigma_3 \\ \sigma_4,(r_1+r_2),\sigma_5 \end{pmatrix} \odot \Gamma_1, x : A, [x/y]\Gamma_2 \vdash [x/y]\mathcal{J}$$

We can form the following premises for Lemma 3.27:

$$\mathcal{C} \quad \cfrac{\cfrac{\cfrac{\begin{pmatrix} \Delta_1,\sigma_1,(\sigma_1,0),\Delta_2 \\ \sigma_2,s_1,s_2,\sigma_3 \\ \sigma_4,r_1,r_2,\sigma_5 \end{pmatrix} \odot \Gamma_1, x : A, y : A, \Gamma_2 \vdash \mathcal{J}}{\Delta_1, \sigma_1, (\sigma_1, 0), \Delta_2 \odot \Gamma_1, x : A, y : A, \Gamma_2 \vdash} \text{ Lemma 3.3}}{\Delta_1, \sigma_1 \odot \Gamma_1, x : A \vdash} \text{ Lemma 3.4}}{(\Delta_1, \sigma_1 \mid \mathbf{0}^{|\Delta_1|}, 1 \mid \sigma_1, 0) \odot \Gamma_1, x : A \vdash x : A} \text{ T\_Var}$$

1. $(\Delta_1, \sigma_1 \mid \mathbf{0}^{|\Delta_1|}, 1 \mid \sigma_1, 0) \odot \Gamma_1, x : A \vdash x : A$ ($\mathcal{C}$)
2. $(\Delta_1, \sigma_1, (\sigma_1, 0), \Delta_2 \mid \sigma_2, s_1, s_2, \sigma_3 \mid \sigma_4, r_1, r_2, \sigma_5) \odot \Gamma_1, x : A, y : A, \Gamma_2 \vdash \mathcal{J}$ (premise)

3. $|\sigma_2, s_1| = |\sigma_4, r_1| = |\Gamma_1, x : A|$ (trivially)

Giving:

$$\begin{pmatrix} \Delta_1,\sigma_1,(\Delta_2\backslash\pi+(\Delta_2/\pi)*(\mathbf{0}^{|\Delta_1|},1)) \\ (\sigma_2,s_1+s_2*(\mathbf{0}^{|\Delta_1|},1)),\sigma_3 \\ (\sigma_4,r_1+r_2*(\mathbf{0}^{|\Delta_1|},1)),\sigma_5 \end{pmatrix} \odot \Gamma_1, x : A, [x/y]\Gamma_2 \vdash [x/y]\mathcal{J}$$

where $\pi = |\Gamma, x : A| = |\Delta_1| + 1$. Recall that $\mathsf{contr}(|\Delta_1| \,;\Delta_2) = \Delta_2\backslash(|\Delta_1| + 1) + (\Delta_2/(|\Delta_1| + 1)) * (\mathbf{0}^{|\Delta_1|}, 1) = \Delta_2\backslash\pi + (\Delta_2/\pi) * (\mathbf{0}^{|\Delta_1|}, 1)$. From sizing, we can see that $(\sigma_2, s_1 + s_2 * (\mathbf{0}^{|\Delta_1|}, 1)), \sigma_3 = (\sigma_2, s_1 + \mathbf{0}^{|\Delta_1|}, s_2), \sigma_3 = \sigma_2, (s_1 + s_2), \sigma_3$, and similarly for $(\sigma_4, r_1 + r_2 * (\mathbf{0}^{|\Delta_1|}, 1)), \sigma_5$. Therefore, we have:

$$\begin{pmatrix} \Delta_1,\sigma_1,\mathsf{contr}(|\Delta_1|;\Delta_2) \\ \sigma_2,(s_1+s_2),\sigma_3 \\ \sigma_4,(r_1+r_2),\sigma_5 \end{pmatrix} \odot \Gamma_1, x : A, [x/y]\Gamma_2 \vdash [x/y]\mathcal{J}$$

As required. □

A.2.2 Exchange

Lemma 3.23 (Exchange)**.** *The following rule is admissible:*

$$\frac{\begin{pmatrix} \Delta_1,\sigma_1,(\sigma_2,0),\Delta_2 \\ \sigma_3,s_1,s_2,\sigma_4 \\ \sigma_5,r_1,r_2,\sigma_6 \end{pmatrix} \odot \Gamma_1, x : A, y : B, \Gamma_2 \vdash \mathcal{J} \quad x \notin \mathsf{FV}\,(B) \quad |\Delta_1| = |\sigma_3| = |\sigma_5| = |\Gamma_1|}{\begin{pmatrix} \Delta_1,\sigma_2,(\sigma_1,0),\mathsf{exch}(|\Delta_1|;\Delta_2) \\ \sigma_3,s_2,s_1,\sigma_4 \\ \sigma_5,r_2,r_1,\sigma_6 \end{pmatrix} \odot \Gamma_1, y : B, x : A, \Gamma_2 \vdash \mathcal{J}} \textsc{Exchange}$$

Proof. For well-formed contexts: by induction on the form of $\Delta_1, \sigma_1, (\sigma_2, 0), \Delta_2 \odot \Gamma_1, x : A, y : B, \Gamma_2 \vdash$, for which it suffices to consider whether Γ_2 is empty or not. In the case that Γ_2 is empty, we have:

$$\frac{(\Delta_1, \sigma_1 \mid \sigma_2, 0 \mid \mathbf{0}) \odot \Gamma_1, x : A \vdash B : \mathsf{Type}_l}{\Delta_1, \sigma_1, (\sigma_2, 0) \odot \Gamma_1, x : A, y : B \vdash} \textsc{Wf\_Ext}$$

And we obtain our goal via the following derivation:

$$\frac{\dfrac{\dfrac{(\Delta_1, \sigma_1 \mid \sigma_2, 0 \mid \mathbf{0}) \odot \Gamma_1, x : A \vdash B : \mathsf{Type}_l}{(\Delta_1 \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma_1 \vdash A : \mathsf{Type}_{l'}} \text{L. 3.5} \quad \dfrac{(\Delta_1, \sigma_1 \mid \sigma_2, 0 \mid \mathbf{0}) \odot \Gamma_1, x : A \vdash B : \mathsf{Type}_l}{(\Delta_1 \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma_1 \vdash B : \mathsf{Type}_l} \text{L. 3.1}}{(\Delta_1, \sigma_2 \mid \sigma_1, 0 \mid \mathbf{0}) \odot \Gamma_1, y : B \vdash A : \mathsf{Type}_{l'}} \text{L. 3.24}}{\Delta_1, \sigma_2, (\sigma_1, 0) \odot \Gamma_1, y : B, x : A \vdash} \textsc{Wf\_Ext}$$

Now consider that $\Gamma_2 = \Gamma_2', z : C$, then we have:

$$\frac{(\Delta_1, \sigma_1, (\sigma_2, 0), \Delta_2' \mid \sigma_3, s, r, \sigma_3' \mid \mathbf{0}) \odot \Gamma_1, x : A, y : B, \Gamma_2' \vdash C : \mathsf{Type}_l}{\Delta_1, \sigma_1, (\sigma_2, 0), \Delta_2', (\sigma_3, s, r, \sigma_3') \odot \Gamma_1, x : A, y : B, \Gamma_2', z : C \vdash} \textsc{Wf\_Ext}$$

Where $|\sigma_3| = |\Gamma_1|$. We can then obtain our goal via the following derivation:

$$\frac{\dfrac{\dfrac{(\Delta_1, \sigma_1, (\sigma_2, 0), \Delta_2' \mid \sigma_3, s, r, \sigma_3' \mid \mathbf{0}) \odot \Gamma_1, x : A, y : B, \Gamma_2' \vdash C : \mathsf{Type}_l}{(\Delta_1, \sigma_2, (\sigma_1, 0), \mathsf{exch}(|\Delta_1| \,; \Delta_2') \mid \sigma_3, r, s, \sigma_3' \mid \mathbf{0}) \odot \Gamma_1, y : B, x : A, \Gamma_2' \vdash C : \mathsf{Type}_l} \text{L. 3.23}}{\Delta_1, \sigma_2, (\sigma_1, 0), \mathsf{exch}(|\Delta_1| \,; \Delta_2'), (\sigma_3, r, s, \sigma_3') \odot \Gamma_1, y : B, x : A, \Gamma_2', z : C \vdash} \textsc{Wf\_Ext}}{\Delta_1, \sigma_2, (\sigma_1, 0), \mathsf{exch}(|\Delta_1| \,; (\Delta_2', (\sigma_3, s, r, \sigma_3'))) \odot \Gamma_1, y : B, x : A, \Gamma_2', z : C \vdash} \equiv$$

For typing $(t : C)$: By induction on the form of $(\Delta_1, \sigma_1, (\sigma_2, 0), \Delta_2 \mid \sigma_3, s_1, s_2, \sigma_4 \mid \sigma_5, r_1, r_2, \sigma_6) \odot \Gamma_1, x : A, y : B, \Gamma_2 \vdash t : C$ as follows:

Case.

$$\dfrac{\Delta_1, \sigma_1, (\sigma_2, 0), \Delta_2 \odot \Gamma_1, x : A, y : B, \Gamma_2 \vdash}{(\Delta_1, \sigma_1, (\sigma_2, 0), \Delta_2 \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma_1, x : A, y : B, \Gamma_2 \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}}\ \textsc{T\_Type}$$

Then our goal holds by the following derivation:

$$\dfrac{\dfrac{\Delta_1, \sigma_1, (\sigma_2, 0), \Delta_2 \odot \Gamma_1, x : A, y : B, \Gamma_2 \vdash}{\Delta_1, \sigma_2, (\sigma_1, 0), \mathsf{exch}(|\Delta_1| ; \Delta_2) \odot \Gamma_1, y : B, x : A, \Gamma_2 \vdash}\ \textsc{i.h.}}{(\Delta_1, \sigma_2, (\sigma_1, 0), \mathsf{exch}(|\Delta_1| ; \Delta_2) \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma_1, y : B, x : A, \Gamma_2 \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}}\ \textsc{T\_Type}$$

Case.

$$\dfrac{\begin{pmatrix} \Delta_1,\sigma_1,(\sigma_2,0),\Delta_2 \\ \sigma_3,s_1,s_2,\sigma_4 \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, x : A, y : B, \Gamma_2 \vdash C : \mathsf{Type}_{l_1} \qquad \begin{pmatrix} \Delta_1,\sigma_1,(\sigma_2,0),\Delta_2,(\sigma_3,s_1,s_2,\sigma_4) \\ \sigma_3',s_1',s_2',\sigma_4',r \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, x : A, y : B, \Gamma_2, z : C \vdash D : \mathsf{Type}_{l_2}}{\begin{pmatrix} \Delta_1,\sigma_1,(\sigma_2,0),\Delta_2 \\ \sigma_3,s_1,s_2,\sigma_4+\sigma_3',s_1',s_2',\sigma_4' \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, x : A, y : B, \Gamma_2 \vdash (z :_{(s,r)} C) \to D : \mathsf{Type}_{l_1 \sqcup l_2}}\ \textsc{T\_Arrow}$$

Performing induction on both premises, and putting them back together via T_Arrow, we have:

$$\dfrac{\begin{pmatrix} \Delta_1,\sigma_2,(\sigma_1,0),\mathsf{exch}(|\Delta_1|;\Delta_2) \\ \sigma_3,s_2,s_1,\sigma_4 \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, y : B, x : A, \Gamma_2 \vdash C : \mathsf{Type}_{l_1} \qquad \begin{pmatrix} \Delta_1,\sigma_2,(\sigma_1,0),\mathsf{exch}(|\Delta_1|;\Delta_2),(\sigma_3,s_2,s_1,\sigma_4) \\ \sigma_3',s_2',s_1',\sigma_4',r \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, x : A, y : B, \Gamma_2, z : C \vdash D : \mathsf{Type}_{l_2}}{\begin{pmatrix} \Delta_1,\sigma_2,(\sigma_1,0),\mathsf{exch}(|\Delta_1|;\Delta_2) \\ \sigma_3,s_2,s_1,\sigma_4+\sigma_3',s_2',s_1',\sigma_4' \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, y : B, x : A, \Gamma_2 \vdash (z :_{(s,r)} C) \to D : \mathsf{Type}_{l_1 \sqcup l_2}}\ \textsc{T\_Arrow}$$

As required. The proofs for T_Ten, T_Fun, T_App, T_Pair, T_TenCut, T_Box, T_BoxI, T_BoxE, and T_TyConv proceed similarly.

Case.

$$\dfrac{\Delta_1, \sigma_1, (\sigma_2, 0), \Delta_2 \odot \Gamma_1, x : A, y : B, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \sigma_1, (\sigma_2, 0), \Delta_2 \mid \mathbf{0}^{|\Delta_1|}, 1, 0, \mathbf{0} \mid \sigma_1, 0, 0, \mathbf{0}) \odot \Gamma_1, x : A, y : B, \Gamma_2 \vdash x : A}\ \textsc{T\_Var}$$

Our goal holds by the following derivation:

$$\dfrac{\dfrac{\Delta_1, \sigma_1, (\sigma_2, 0), \Delta_2 \odot \Gamma_1, x : A, y : B, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1| \quad x \notin \mathsf{FV}(B)}{\Delta_1, \sigma_2, (\sigma_1, 0), \mathsf{exch}(|\Delta_1| ; \Delta_2) \odot \Gamma_1, y : B, x : A, \Gamma_2 \vdash}\ \textsc{i.h.} \qquad |\Delta_1, \sigma_2| = |\Gamma_1, y : B|}{(\Delta_1, \sigma_2, (\sigma_1, 0), \mathsf{exch}(|\Delta_1| ; \Delta_2) \mid \mathbf{0}^{|\Delta_1|}, 0, 1, \mathbf{0} \mid \sigma_1, 0, 0, \mathbf{0}) \odot \Gamma_1, y : B, x : A, \Gamma_2 \vdash x : A}\ \textsc{T\_Var}$$

Case.

$$\frac{\Delta_1, \sigma_1, (\sigma_2, 0), \Delta_2 \odot \Gamma_1, x : A, y : B, \Gamma_2 \vdash \quad |\Delta_1, \sigma_1| = |\Gamma_1, x : A|}{(\Delta_1, \sigma_1, (\sigma_2, 0), \Delta_2 \mid \mathbf{0}^{|\Delta_1|}, 0, 1, \mathbf{0} \mid \sigma_2, 0, 0, \mathbf{0}) \odot \Gamma_1, x : A, y : B, \Gamma_2 \vdash y : B}\ \textsc{T\_Var}$$

Our goal holds by the following derivation:

$$\frac{\dfrac{\Delta_1, \sigma_1, (\sigma_2, 0), \Delta_2 \odot \Gamma_1, x : A, y : B, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1| \quad x \notin \mathsf{FV}\,(B)}{\Delta_1, \sigma_2, (\sigma_1, 0), \mathsf{exch}(|\Delta_1|\,; \Delta_2) \odot \Gamma_1, y : B, x : A, \Gamma_2 \vdash}\ \textsc{i.h.} \qquad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \sigma_2, (\sigma_1, 0), \mathsf{exch}(|\Delta_1|\,; \Delta_2) \mid \mathbf{0}^{|\Delta_1|}, 1, 0, \mathbf{0} \mid \sigma_2, 0, 0, \mathbf{0}) \odot \Gamma_1, y : B, x : A, \Gamma_2 \vdash y : B}\ \textsc{T\_Var}$$

Case.

$$\frac{\hat{\Delta_1}, \sigma, \hat{\Delta_1}', \sigma_1, (\sigma_2, 0), \Delta_2 \odot \hat{\Gamma_1}, z : C, \hat{\Gamma_1}', x : A, y : B, \Gamma_2 \vdash \quad \left|\hat{\Delta_1}\right| = \left|\hat{\Gamma_1}\right|}{(\hat{\Delta_1}, \sigma, \hat{\Delta_1}', \sigma_1, (\sigma_2, 0), \Delta_2 \mid \mathbf{0}^{|\hat{\Delta_1}|}, 1, \mathbf{0} \mid \sigma, \mathbf{0}) \odot \hat{\Gamma_1}, z : C, \hat{\Gamma_1}', x : A, y : B, \Gamma_2 \vdash z : C}\ \textsc{T\_Var}$$

Our goal holds by the following derivation:

$$\frac{\dfrac{\begin{array}{c}\left|\hat{\Delta_1}, \sigma, \hat{\Delta_1}'\right| = \left|\hat{\Gamma_1}, z : C, \hat{\Gamma_1}'\right| \quad x \notin \mathsf{FV}\,(B) \\ \hat{\Delta_1}, \sigma, \hat{\Delta_1}', \sigma_1, (\sigma_2, 0), \Delta_2 \odot \hat{\Gamma_1}, z : C, \hat{\Gamma_1}', x : A, y : B, \Gamma_2 \vdash\end{array}}{\hat{\Delta_1}, \sigma, \hat{\Delta_1}', \sigma_2, (\sigma_1, 0), \mathsf{exch}(\left|\hat{\Delta_1}, \sigma, \hat{\Delta_1}'\right|; \Delta_2) \odot \hat{\Gamma_1}, z : C, \hat{\Gamma_1}', y : B, x : A, \Gamma_2 \vdash}\ \textsc{i.h.} \qquad \left|\hat{\Delta_1}\right| = \left|\hat{\Gamma_1}\right|}{(\hat{\Delta_1}, \sigma, \hat{\Delta_1}', \sigma_2, (\sigma_1, 0), \mathsf{exch}(\left|\hat{\Delta_1}, \sigma, \hat{\Delta_1}'\right|; \Delta_2) \mid \mathbf{0}^{|\hat{\Delta_1}|}, 1, \mathbf{0} \mid \sigma, \mathbf{0}) \odot \hat{\Gamma_1}, z : C, \hat{\Gamma_1}', y : B, x : A, \Gamma_2 \vdash z : C}\ \textsc{T\_Var}$$

Case.

$$\frac{\Delta_1, \sigma_1, (\sigma_2, 0), \hat{\Delta_2}, \sigma, \hat{\Delta_2}' \odot \Gamma_1, x : A, y : B, \hat{\Gamma_2}, z : C, \hat{\Gamma_2}' \vdash \quad \left|\Delta_1, \sigma_1, (\sigma_2, 0), \hat{\Delta_2}\right| = \left|\Gamma_1, x : A, y : B, \hat{\Gamma_2}\right|}{(\Delta_1, \sigma_1, (\sigma_2, 0), \hat{\Delta_2}, \sigma, \hat{\Delta_2}' \mid \mathbf{0}^{|\hat{\Delta_1}|}, 1, \mathbf{0} \mid \sigma, \mathbf{0}) \odot \Gamma_1, x : A, y : B, \hat{\Gamma_2}, z : C, \hat{\Gamma_2}' \vdash z : C}\ \textsc{T\_Var}$$

Where $\sigma = \hat{\sigma}, s, r, \hat{\sigma}'$ with $|\hat{\sigma}| = |\Delta_1|$. Our goal holds by the following derivation:

$$\frac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\left|\Delta_1, \sigma_1, (\sigma_2, 0), \mathsf{exch}(|\Delta_1|\,; \hat{\Delta_2})\right| = \left|\Gamma_1, y : B, x : A, \hat{\Gamma_2}\right| \\ |\Delta_1| = |\Gamma_1| \quad x \notin \mathsf{FV}\,(B) \\ \Delta_1, \sigma_1, (\sigma_2, 0), \hat{\Delta_2}, \sigma, \hat{\Delta_2}' \odot \Gamma_1, x : A, y : B, \hat{\Gamma_2}, z : C, \hat{\Gamma_2}' \vdash\end{array}}{\Delta_1, \sigma_2, (\sigma_1, 0), \mathsf{exch}(|\Delta_1|\,; (\hat{\Delta_2}, \sigma, \hat{\Delta_2}')) \odot \Gamma_1, y : B, x : A, \hat{\Gamma_2}, z : C, \hat{\Gamma_2}' \vdash}\ \textsc{i.h.}}{\Delta_1, \sigma_2, (\sigma_1, 0), \mathsf{exch}(|\Delta_1|\,; \hat{\Delta_2}), (\hat{\sigma}, r, s, \hat{\sigma}'), \mathsf{exch}(|\Delta_1|\,; \hat{\Delta_2}') \odot \Gamma_1, y : B, x : A, \hat{\Gamma_2}, z : C, \hat{\Gamma_2}' \vdash}\ \equiv}{(\Delta_1, \sigma_2, (\sigma_1, 0), \mathsf{exch}(|\Delta_1|\,; \hat{\Delta_2}), (\hat{\sigma}, r, s, \hat{\sigma}'), \mathsf{exch}(|\Delta_1|\,; \hat{\Delta_2}') \mid \mathbf{0}^{|\hat{\Delta_1}|}, 1, \mathbf{0} \mid \hat{\sigma}, r, s, \hat{\sigma}', \mathbf{0}) \odot \Gamma_1, y : B, x : A, \hat{\Gamma_2}, z : C, \hat{\Gamma_2}' \vdash z : C}\ \textsc{T\_Var}}{(\Delta_1, \sigma_2, (\sigma_1, 0), \mathsf{exch}(|\Delta_1|\,; (\hat{\Delta_2}, \sigma, \hat{\Delta_2}')) \mid \mathbf{0}^{|\hat{\Delta_1}|}, 1, \mathbf{0} \mid \hat{\sigma}, r, s, \hat{\sigma}', \mathbf{0}) \odot \Gamma_1, y : B, x : A, \hat{\Gamma_2}, z : C, \hat{\Gamma_2}' \vdash z : C}\ \equiv$$

For equality, by standard induction and re-application (see Section A.8).
For subtyping, by standard induction and re-application (see Section A.8). ☐

Corollary 3.23.1 (Exchange (general))**.** *As a corollary to Lemma 3.23, the following rule is admissible:*

$$\frac{\begin{pmatrix} \Delta_1,\Delta_2,(\sigma_1,\mathbf{0}^{|\Delta_2|}),\Delta_3 \\ \sigma_2,\sigma_3,s,\sigma_4 \\ \sigma_5,\sigma_6,r,\sigma_7 \end{pmatrix} \odot \Gamma_1,\Gamma_2,x:A,\Gamma_3 \vdash \mathcal{J} \qquad \mathsf{Dom}(\Gamma_2)\cap\mathsf{FV}(A)=\emptyset \quad |\sigma_1|=|\Delta_1|=|\sigma_2|=|\sigma_5|=|\Gamma_1|}{\begin{pmatrix} \Delta_1,\sigma_1,\mathsf{ins}(|\Delta_1|;0;\Delta_2),\mathsf{mv}(|\Delta_1,\Delta_2|;|\Delta_1|;\Delta_3) \\ \sigma_2,s,\sigma_3,\sigma_4 \\ \sigma_5,r,\sigma_6,\sigma_7 \end{pmatrix} \odot \Gamma_1,x:A,\Gamma_2,\Gamma_3 \vdash \mathcal{J}} \ \textsc{ExchangeGen}$$

Where $\mathsf{Dom}(\Gamma)$ *is the domain of* Γ.

Proof. Consider Γ_2 is empty, then by premise we have $(\Delta_1,\sigma_1,\Delta_3 \mid \sigma_2,s,\sigma_4 \mid \sigma_5,r,\sigma_7)\odot\Gamma_1,x:A,\Gamma_3\vdash\mathcal{J}$, and need to show $(\Delta_1,\sigma_1,\Delta_3 \mid \sigma_2,s,\sigma_4 \mid \sigma_5,r,\sigma_7)\odot\Gamma_1,x:A,\Gamma_3\vdash\mathcal{J}$, which holds by premise. Now consider $\Gamma_2=\Gamma_2',y:C$, then by premise we have:

$$\begin{pmatrix} \Delta_1,(\Delta_2',\sigma_8),(\sigma_1,\mathbf{0}^{|\Delta_2',\sigma_8|}),\Delta_3 \\ \sigma_2,\sigma_3',s',s,\sigma_4 \\ \sigma_5,\sigma_6',r',r,\sigma_7 \end{pmatrix} \odot \Gamma_1,\Gamma_2',y:C,x:A,\Gamma_3 \vdash \mathcal{J}$$

And our goal is:

$$\begin{pmatrix} \Delta_1,\sigma_1,\mathsf{ins}(|\Delta_1|;0;(\Delta_2',\sigma_8)),\mathsf{mv}(\left|\Delta_1,(\Delta_2',\sigma_8)\right|;|\Delta_1|;\Delta_3) \\ \sigma_2,s,\sigma_3',s',\sigma_4 \\ \sigma_5,r,\sigma_6',r',\sigma_7 \end{pmatrix} \odot \Gamma_1,x:A,\Gamma_2',y:C,\Gamma_3 \vdash \mathcal{J}$$

Doing some trivial vector rewriting, and then applying our premise in the following derivation, we have:

$$\frac{\begin{pmatrix} \Delta_1,\Delta_2',\sigma_8,((\sigma_1,\mathbf{0}^{|\Delta_2'|}),0),\Delta_3 \\ \sigma_2,\sigma_3',s',s,\sigma_4 \\ \sigma_5,\sigma_6',r',r,\sigma_7 \end{pmatrix} \odot \Gamma_1,\Gamma_2',y:C,x:A,\Gamma_3 \vdash \mathcal{J} \qquad y\notin\mathsf{FV}(A) \quad |\Delta_1,\Delta_2'|=|\sigma_2,\sigma_3'|=|\sigma_5,\sigma_6'|=|\Gamma_1,\Gamma_2'|}{\begin{pmatrix} \Delta_1,\Delta_2',(\sigma_1,\mathbf{0}^{|\Delta_2'|}),(\sigma_8,0),\mathsf{exch}(\left|\Delta_1,\Delta_2'\right|;\Delta_3) \\ \sigma_2,\sigma_3',s,s',\sigma_4 \\ \sigma_5,\sigma_6',r,r',\sigma_7 \end{pmatrix} \odot \Gamma_1,\Gamma_2',x:A,y:C,\Gamma_3 \vdash \mathcal{J}} \ \textsc{Lemma 3.23}$$

Applying this inductively, we have:

$$\frac{\begin{pmatrix} \Delta_1,\Delta_2',(\sigma_1,\mathbf{0}^{|\Delta_2'|}),(\sigma_8,0),\mathsf{exch}(\left|\Delta_1,\Delta_2'\right|;\Delta_3) \\ \sigma_2,\sigma_3',s,s',\sigma_4 \\ \sigma_5,\sigma_6',r,r',\sigma_7 \end{pmatrix} \odot \Gamma_1,\Gamma_2',x:A,y:C,\Gamma_3 \vdash \mathcal{J} \qquad \mathsf{Dom}(\Gamma_2')\cap\mathsf{FV}(A)=\emptyset \quad |\sigma_1|=|\Delta_1|=|\sigma_2|=|\sigma_5|=|\Gamma_1|}{\begin{pmatrix} \Delta_1,\sigma_1,\mathsf{ins}(|\Delta_1|;0;\Delta_2'),\mathsf{mv}(\left|\Delta_1,\Delta_2'\right|;|\Delta_1|;((\sigma_8,0),\mathsf{exch}(\left|\Delta_1,\Delta_2'\right|;\Delta_3))) \\ \sigma_2,s,\sigma_3',s',\sigma_4 \\ \sigma_5,r,\sigma_6',r',\sigma_7 \end{pmatrix} \odot \Gamma_1,x:A,\Gamma_2',y:C,\Gamma_3 \vdash \mathcal{J}} \ \text{I.H.}$$

By Lemma 3.21, we have $\mathsf{mv}(|\Delta_1,\Delta_2'|\,;|\Delta_1|\,;(\mathsf{exch}(|\Delta_1,\Delta_2'|\,;\Delta_3)))=\mathsf{mv}(|\Delta_1,(\Delta_2',\sigma_8)|\,;|\Delta_1|\,;\Delta_3)$, and as by Lemma 3.2 we know $|\sigma_8|=|\Delta_1,\Delta_2'|$, we have:

$$\mathsf{mv}(|\Delta_1,\Delta_2'|\,;|\Delta_1|\,;((\sigma_8,0),\mathsf{exch}(|\Delta_1,\Delta_2'|\,;\Delta_3)))=(\sigma_8',0,\sigma_8''),\mathsf{mv}(|\Delta_1,(\Delta_2',\sigma_8)|\,;|\Delta_1|\,;\Delta_3)$$

for some σ_8' and σ_8'', with $|\sigma_8'|=|\Delta_1|$. Therefore, we have $\mathsf{ins}(|\Delta_1|\,;0;\Delta_2'),(\sigma_8',0,\sigma_8'')=\mathsf{ins}(|\Delta_1|\,;0;(\Delta_2',\sigma_8))$. Rewriting the conclusion by this information, we have:

$$\begin{pmatrix} \Delta_1,\sigma_1,\mathsf{ins}(|\Delta_1|;0;(\Delta_2',\sigma_8)),\mathsf{mv}(\left|\Delta_1,(\Delta_2',\sigma_8)\right|;|\Delta_1|;\Delta_3) \\ \sigma_2,s,\sigma_3',s',\sigma_4 \\ \sigma_5,r,\sigma_6',r',\sigma_7 \end{pmatrix} \odot \Gamma_1,x:A,\Gamma_2',y:C,\Gamma_3 \vdash \mathcal{J}$$

Which matches our goal. ☐

Corollary 3.23.2 (Exchange from end). *As a corollary to Corollary 3.23.1, the following rule is admissible:*

$$\frac{\begin{pmatrix} \Delta_1, \Delta_2, (\sigma_1, \boldsymbol{0}^{|\Delta_2|}) \\ \sigma_2, \sigma_3, s \\ \sigma_4, \sigma_5, r \end{pmatrix} \odot \Gamma_1, \Gamma_2, x : A \vdash \mathcal{J} \quad \mathsf{Dom}(\Gamma_2) \cap \mathsf{FV}\,(A) = \emptyset \quad |\Delta_1| = |\sigma_2| = |\sigma_4| = |\Gamma_1|}{\begin{pmatrix} \Delta_1, \sigma_1, \mathsf{ins}(|\Delta_1|; 0; \Delta_2) \\ \sigma_2, s, \sigma_3 \\ \sigma_4, r, \sigma_5 \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2 \vdash \mathcal{J}} \textsc{ ExchangeEnd}$$

Proof. This is Corollary 3.23.1 with Γ_3 being empty. □

A.2.3 Weakening

Lemma 3.24 (Weakening). *The following rule is admissible:*

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \mathcal{J} \quad (\Delta \mid \sigma_3 \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_l}{(\Delta, \sigma_3 \mid \sigma_1, 0 \mid \sigma_2, 0) \odot \Gamma, x : A \vdash \mathcal{J}} \textsc{ Weak}$$

Where $\mathcal{J}$ is typing, equality, or subtyping.

Proof. For typing: By induction on the form of $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : B$ as follows:

Case.

$$\frac{\Delta \odot \Gamma \vdash}{(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}} \textsc{ T\_Type}$$

Then our goal is to show:

$$(\Delta, \sigma_3 \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma, x : A \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}$$

We we obtain through the following derivation:

$$\frac{\dfrac{(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l'}}{\Delta, \sigma_3 \odot \Gamma, x : A \vdash} \textsc{ Wf\_Ext}}{(\Delta, \sigma_3 \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma, x : A \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}} \textsc{ T\_Type}$$

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash C : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, y : C \vdash D : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (y :_{(s,r)} C) \to D : \mathsf{Type}_{l_1 \sqcup l_2}} \textsc{ T\_Arrow}$$

Then our goal is to show:

$$(\Delta, \sigma_3 \mid (\sigma_1 + \sigma_2), 0 \mid \mathbf{0}) \odot \Gamma, x : A \vdash (y :_{(s,r)} C) \to D : \mathsf{Type}_{l_1 \sqcup l_2}$$

We we obtain through the following derivation:

$$\mathcal{C}$$
$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash C : \mathsf{Type}_{l_1} \quad (\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l}{(\Delta, \sigma_3 \mid \sigma_1, 0 \mid \mathbf{0}) \odot \Gamma, x : A \vdash C : \mathsf{Type}_{l_1}} \textsc{ i.h.}$$

$\mathcal{D}$

$$\dfrac{(\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, y : C \vdash D : \mathsf{Type}_{l_2} \quad \dfrac{(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l \quad (\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash C : \mathsf{Type}_{l_1}}{(\Delta, \sigma_1 \mid \sigma_3, 0 \mid \mathbf{0}) \odot \Gamma, y : C \vdash A : \mathsf{Type}_l}\ \text{I.H.}}{\dfrac{(\Delta, \sigma_1, (\sigma_3, 0) \mid \sigma_2, r, 0 \mid \mathbf{0}) \odot \Gamma, y : C, x : A \vdash D : \mathsf{Type}_{l_2}}{(\Delta, \sigma_3, (\sigma_1, 0) \mid \sigma_2, 0, r \mid \mathbf{0}) \odot \Gamma, x : A, y : C \vdash D : \mathsf{Type}_{l_2}}\ \text{L. 3.23}}\ \text{I.H.}$$

$$\dfrac{\dfrac{\mathcal{C} \quad \mathcal{D}}{(\Delta, \sigma_3 \mid \sigma_1, 0 + \sigma_2, 0 \mid \mathbf{0}) \odot \Gamma, x : A \vdash (y :_{(s,r)} C) \to D : \mathsf{Type}_{l_1 \sqcup l_2}}\ \text{T\_Arrow}}{(\Delta, \sigma_3 \mid (\sigma_1 + \sigma_2), 0 \mid \mathbf{0}) \odot \Gamma, x : A \vdash (y :_{(s,r)} C) \to D : \mathsf{Type}_{l_1 \sqcup l_2}}\ \equiv$$

The proofs for T_Ten, T_Fun, T_App, T_Pair, T_TenCut, T_Box, T_BoxI, T_BoxE, and T_TyConv proceed similarly, using induction and Lemma 3.23.

Case.

$$\dfrac{\Delta_1, \sigma, \Delta_2 \odot \Gamma_1, y : C, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \sigma, \Delta_2 \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \sigma, 0, \mathbf{0}) \odot \Gamma_1, y : C, \Gamma_2 \vdash y : C}\ \text{T\_Var}$$

Then our goal is to show:

$$(\Delta_1, \sigma, \Delta_2, \sigma_3 \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \sigma, \mathbf{0}) \odot \Gamma_1, y : C, \Gamma_2, x : A \vdash y : C$$

Which holds by the following derivation:

$$\dfrac{\dfrac{(\Delta_1, \sigma, \Delta_2 \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma_1, y : C, \Gamma_2 \vdash A : \mathsf{Type}_l}{\Delta_1, \sigma, \Delta_2, \sigma_3 \odot \Gamma_1, y : C, \Gamma_2, x : A \vdash}\ \text{Wf\_Ext} \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \sigma, \Delta_2, \sigma_3 \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \sigma, \mathbf{0}) \odot \Gamma_1, y : C, \Gamma_2, x : A \vdash y : C}\ \text{T\_Var}$$

For equality, by standard induction and re-application (see Section A.8).
For subtyping, by standard induction and re-application (see Section A.8). □

Lemma 3.25 (Weakening for well-formed contexts)**.** *The following rule is admissible:*

$$\dfrac{\Delta_1, \Delta_2 \odot \Gamma_1, \Gamma_2 \vdash \quad (\Delta_1 \mid \sigma_1 \mid \sigma_2) \odot \Gamma_1 \vdash t : A}{(\Delta_1, \Delta_2 \mid \sigma_1, \boldsymbol{0} \mid \sigma_2, \boldsymbol{0}) \odot \Gamma_1, \Gamma_2 \vdash t : A}\ \text{WeakWF}$$

Proof. By induction on the form of $(\Delta_1 \mid \sigma_1 \mid \sigma_2) \odot \Gamma_1 \vdash t : A$, and $\Delta_1, \Delta_2 \odot \Gamma_1, \Gamma_2 \vdash$, as follows: First, consider that Γ_2 is empty, then the typing premise for t matches the goal, and we're done. Now consider that $\Delta_1, \Delta_2 \odot \Gamma_1, \Gamma_2 \vdash$ is:

$$\dfrac{\mathcal{C} = (\Delta_1, \Delta_2' \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma_1, \Gamma_2' \vdash B : \mathsf{Type}_l}{\Delta_1, \Delta_2', \sigma_3 \odot \Gamma_1, \Gamma_2', x : B \vdash}\ \text{Wf\_Ext}$$

Then our goal is to show:

$$(\Delta_1, \Delta_2', \sigma_3 \mid \sigma_1, \mathbf{0} \mid \sigma_2, \mathbf{0}) \odot \Gamma_1, \Gamma_2', x : B \vdash t : A$$

Which we obtain by the following derivation:

$$\dfrac{\dfrac{\dfrac{\mathcal{C}}{\Delta_1, \Delta_2' \odot \Gamma_1, \Gamma_2' \vdash}\ \text{L. 3.3} \qquad (\Delta_1 \mid \sigma_1 \mid \sigma_2) \odot \Gamma_1 \vdash t : A}{(\Delta_1, \Delta_2' \mid \sigma_1, \mathbf{0} \mid \sigma_2, \mathbf{0}) \odot \Gamma_1, \Gamma_2' \vdash t : A}\ \text{I.H.} \qquad \mathcal{C}}{(\Delta_1, \Delta_2', \sigma_3 \mid \sigma_1, \mathbf{0} \mid \sigma_2, \mathbf{0}) \odot \Gamma_1, \Gamma_2', x : B \vdash t : A}\ \text{L. 3.24}$$

□

Lemma 3.26 (Weakening (general))**.** *The following rule is admissible (for $\mathcal{J}$ is typing, equality, or subtyping):*

$$\dfrac{(\Delta_1, \Delta_2 \mid \sigma_1, \sigma_1' \mid \sigma_2, \sigma_2') \odot \Gamma_1, \Gamma_2 \vdash \mathcal{J} \quad (\Delta_1 \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma_1 \vdash A : \mathsf{Type}_l \quad |\sigma_1| = |\sigma_2| = |\Gamma_1|}{(\Delta_1, \sigma_3, \mathsf{ins}(|\Delta_1|\,; 0; \Delta_2) \mid \sigma_1, 0, \sigma_1' \mid \sigma_2, 0, \sigma_2') \odot \Gamma_1, x : A, \Gamma_2 \vdash \mathcal{J}}\ \textsc{WeakGen}$$

Proof. By Lemma 3.3 with the premise for $\mathcal{J}$, we have $(\Delta_1, \Delta_2) \odot \Gamma_1, \Gamma_2 \vdash$. Therefore, using this and the typing premise for A with Lemma 3.25, we have $(\Delta_1, \Delta_2 \mid \sigma_3, \mathbf{0}^{|\Gamma_2|} \mid \mathbf{0}) \odot \Gamma_1, \Gamma_2 \vdash A : \mathsf{Type}_l$. Applying this, and the premise for $\mathcal{J}$, in the following derivation, we have:

$$\dfrac{(\Delta_1, \Delta_2 \mid \sigma_1, \sigma_1' \mid \sigma_2, \sigma_2') \odot \Gamma_1, \Gamma_2 \vdash \mathcal{J} \quad (\Delta_1, \Delta_2 \mid \sigma_3, \mathbf{0}^{|\Gamma_2|} \mid \mathbf{0}) \odot \Gamma_1, \Gamma_2 \vdash A : \mathsf{Type}_l}{(\Delta_1, \Delta_2, (\sigma_3, \mathbf{0}^{|\Gamma_2|}) \mid \sigma_1, \sigma_1', 0 \mid \sigma_2, \sigma_2', 0) \odot \Gamma_1, \Gamma_2, x : A \vdash \mathcal{J}}\ \textsc{Lemma 3.24}$$

As $(\Delta_1 \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma_1 \vdash A : \mathsf{Type}_l$, by disjointness of context extension we therefore have $\mathsf{Dom}(\Gamma_2) \cap \mathsf{FV}(A) = \emptyset$. By Lemma 3.2 we have $|\Gamma_2| = |\Delta_2|$ and $|\Delta_1| = |\Gamma_1|$. We can put this information together via the following derivation, to obtain the goal:

$$\dfrac{\begin{pmatrix} \Delta_1, \Delta_2, (\sigma_3, \mathbf{0}^{|\Delta_2|}) \\ \sigma_1, \sigma_1', 0 \\ \sigma_2, \sigma_2', 0 \end{pmatrix} \odot \Gamma_1, \Gamma_2, x : A \vdash \mathcal{J} \quad \mathsf{Dom}(\Gamma_2) \cap \mathsf{FV}(A) = \emptyset \quad |\Delta_1| = |\sigma_1| = |\sigma_2| = |\Gamma_1|}{\begin{pmatrix} \Delta_1, \sigma_3, \mathsf{ins}(|\Delta_1|; 0; \Delta_2) \\ \sigma_1, 0, \sigma_1' \\ \sigma_2, 0, \sigma_2' \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2 \vdash \mathcal{J}}\ \textsc{Corollary 3.23.2}$$

□

A.2.4 Substitution

Lemma 3.27 (Substitution for judgments)**.** *If the following premises hold:*

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$
2. $(\Delta, \sigma_1, \Delta' \mid \sigma_3, s, \sigma_4 \mid \sigma_5, r, \sigma_6) \odot \Gamma_1, x : A, \Gamma_2 \vdash \mathcal{J}$
3. $|\sigma_3| = |\sigma_5| = |\Gamma_1|$

Then $\begin{pmatrix} \Delta, (\Delta' \backslash \pi + (\Delta'/\pi) * \sigma_2) \\ (\sigma_3 + s * \sigma_2), \sigma_4 \\ (\sigma_5 + r * \sigma_2), \sigma_6 \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]\mathcal{J}$*, where* $\pi = |\Gamma_1|$.

Proof. Throughout the proof, we make implicit use of the following size information derived from the premises (and largely Lemma 3.2), further size calculations are trivial, and we typically do not bring attention to them:

- $|\Delta| = |\sigma_1| = |\sigma_2| = |\sigma_3| = |\sigma_5| = |\Gamma_1|$
- $|\Delta'| = |\sigma_4| = |\sigma_6| = |\Gamma_2|$

For well-formed contexts, we proceed by induction on the structure of $(\Delta, \sigma_1, \Delta') \odot \Gamma_1, x : A, \Gamma_2 \vdash$, as follows:

Case.

$$\frac{}{\emptyset \odot \emptyset \vdash}\ \textsc{Wf\_Empty}$$

Trivial as it does not match the form of the typing premise.

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma_1 \vdash A : \mathsf{Type}_l}{\Delta, \sigma_1 \odot \Gamma_1, x : A \vdash}\ \textsc{Wf\_Ext}$$

Then our goal is:

$$\Delta \odot \Gamma_1 \vdash$$

Which holds by Lemma 3.3 on the typing premise for A.

Case.

$$\frac{(\Delta, \sigma_1, \Delta' \mid \sigma_3, s, \sigma_4 \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2' \vdash B : \mathsf{Type}_l}{\Delta, \sigma_1, \Delta', (\sigma_3, s, \sigma_4) \odot \Gamma_1, x : A, \Gamma_2', y : B \vdash}\ \textsc{Wf\_Ext}$$

With $|\sigma_3| = |\Gamma_1|$. Then our goal is:

$$\Delta, ((\Delta', (\sigma_3, s, \sigma_4))\backslash\pi + ((\Delta', (\sigma_3, s, \sigma_4))/\pi) * \sigma_2) \odot \Gamma_1, [t/x]\Gamma_2', y : [t/x]B \vdash$$

As $|\sigma_3| = |\Gamma_1|$, we can see that by Lemma 3.17, our goal becomes:

$$\Delta, (\Delta'\backslash\pi + (\Delta'/\pi) * \sigma_2), ((\sigma_3 + s * \sigma_2), \sigma_4) \odot \Gamma_1, [t/x]\Gamma_2', y : [t/x]B \vdash$$

We can form the following premises, which we then apply to Lemma 3.27:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $(\Delta, \sigma_1, \Delta' \mid \sigma_3, s, \sigma_4 \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2' \vdash B : \mathsf{Type}_l$ (Wf_Ext typing premise for B)
3. $|\sigma_3| = |\mathbf{0}| = |\Gamma_1|$ (trivially, by assumption of sizing for judgments)

Giving:

$$\begin{pmatrix} \Delta, (\Delta'\backslash\pi + (\Delta'/\pi) * \sigma_2) \\ (\sigma_3 + s * \sigma_2), \sigma_4 \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2' \vdash [t/x]B : \mathsf{Type}_l$$

We can apply the result of this substitution to Wf_Ext, giving:

$$\frac{\begin{pmatrix} \Delta, (\Delta'\backslash\pi + (\Delta'/\pi) * \sigma_2) \\ (\sigma_3 + s * \sigma_2), \sigma_4 \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2' \vdash [t/x]B : \mathsf{Type}_l}{\Delta, (\Delta'\backslash\pi + (\Delta'/\pi) * \sigma_2), ((\sigma_3 + s * \sigma_2), \sigma_4) \odot \Gamma_1, [t/x]\Gamma_2', y : [t/x]B \vdash}\ \textsc{Wf\_Ext}$$

Which matches our goal.

For typing, we proceed by induction on the structure of $\begin{pmatrix} \Delta,\sigma_1,\Delta' \\ \sigma_3,s,\sigma_4 \\ \sigma_5,r,\sigma_6 \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2 \vdash t' : B$, as follows:

Case.

$$\frac{\Delta, \sigma_1, \Delta' \odot \Gamma_1, x : A, \Gamma_2 \vdash}{(\Delta, \sigma_1, \Delta' \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}} \text{ T\_TYPE}$$

Therefore we have $t' = \mathsf{Type}_l$ and $B = \mathsf{Type}_{\mathsf{suc}\ l}$, and our goal is:

$$\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}$$

We can form the following inductive premises:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $\Delta, \sigma_1, \Delta' \odot \Gamma_1, x : A, \Gamma_2 \vdash$ (T_TYPE well-formedness premise)

Giving:

$$\Delta, (\Delta'\backslash\pi + (\Delta'/\pi) * \sigma_2) \odot \Gamma_1, [t/x]\Gamma_2 \vdash$$

We can then apply this information to T_TYPE, to achieve our goal:

$$\frac{\Delta, (\Delta'\backslash\pi + (\Delta'/\pi) * \sigma_2) \odot \Gamma_1, [t/x]\Gamma_2 \vdash}{(\Delta, (\Delta'\backslash\pi + (\Delta'/\pi) * \sigma_2) \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma_1, [t/x]\Gamma_2 \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}} \text{ T\_TYPE}$$

Case.

$$\frac{\begin{pmatrix} \Delta,\sigma_1,\Delta' \\ \sigma_7 \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2 \vdash C : \mathsf{Type}_{l_1} \qquad \begin{pmatrix} \Delta,\sigma_7 \\ \sigma_8,r' \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2, y : C \vdash D : \mathsf{Type}_{l_2}}{(\Delta, \sigma_1, \Delta' \mid \sigma_7 + \sigma_8 \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash (y :_{(s',r')} C) \to D : \mathsf{Type}_{l_1 \sqcup l_2}} \text{ T\_ARROW}$$

Therefore we have $\sigma_3, s, \sigma_4 = \sigma_7 + \sigma_8$; $\sigma_5, r, \sigma_6 = \mathbf{0}$; $t' = (y :_{(s',r')} C) \to D$; and $B = \mathsf{Type}_{l_1 \sqcup l_2}$. Our full goal is therefore:

$$\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\sigma_3+s*\sigma_2),\sigma_4 \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]((y :_{(s',r')} C) \to D) : [t/x]\mathsf{Type}_{l_1 \sqcup l_2}$$

As $\sigma_3, s, \sigma_4 = \sigma_7 + \sigma_8$, we therefore have $\sigma_7 = \hat{\sigma_3}, \hat{s}, \hat{\sigma_4}$ and $\sigma_8 = \hat{\sigma_3}', \hat{s}', \hat{\sigma_4}'$ where $\hat{\sigma_3} + \hat{\sigma_3}' = \sigma_3$; $\hat{s} + \hat{s}' = s$; and $\hat{\sigma_4} + \hat{\sigma_4}' = \sigma_4$. Rewriting the first and second premises using the above information, we have the following inductive premises:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $(\Delta, \sigma_1, \Delta' \mid \hat{\sigma_3}, \hat{s}, \hat{\sigma_4} \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash C : \mathsf{Type}_{l_1}$ (first premise of T_ARROW)
3. $|\hat{\sigma_3}| = |\mathbf{0}| = |\Gamma_1|$ (trivially, and by premise (3))

And:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))

2. $\left(\begin{smallmatrix}\Delta,\sigma_1,\Delta',(\hat{\sigma_3},\hat{s},\hat{\sigma_4})\\ \hat{\sigma_3}',\hat{s}',\hat{\sigma_4}',r'\\ \mathbf{0}\end{smallmatrix}\right) \odot \Gamma_1, x : A, \Gamma_2, y : C \vdash D : \mathsf{Type}_{l_2}$ (second premise of T_Arrow)

3. $\left|\hat{\sigma_3}'\right| = |\mathbf{0}| = |\Gamma_1|$ (trivially, and by premise (3))

Giving, respectively:

$$\left(\begin{smallmatrix}\Delta,((\Delta'\backslash\pi)+(\Delta'/\pi)*\sigma_2)\\ (\hat{\sigma_3}+\hat{s}*\sigma_2),\hat{\sigma_4}\\ \mathbf{0}\end{smallmatrix}\right) \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]C : \mathsf{Type}_{l_1}$$

And:

$$\left(\begin{smallmatrix}\Delta,(((\Delta',(\hat{\sigma_3},\hat{s},\hat{\sigma_4}))\backslash\pi)+((\Delta',(\hat{\sigma_3},\hat{s},\hat{\sigma_4}))/\pi)*\sigma_2)\\ (\hat{\sigma_3}'+\hat{s}'*\sigma_2),\hat{\sigma_4}',r'\\ \mathbf{0}\end{smallmatrix}\right) \odot \Gamma_1, [t/x]\Gamma_2, y : [t/x]C \vdash [t/x]D : \mathsf{Type}_{l_2}$$

Rewriting $((\Delta',(\hat{\sigma_3},\hat{s},\hat{\sigma_4}))\backslash\pi)+((\Delta',(\hat{\sigma_3},\hat{s},\hat{\sigma_4}))/\pi)*\sigma_2$ by Lemma 3.17, we have $(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2),((\hat{\sigma_3}+\hat{s}*\sigma_2),\hat{\sigma_4})$, and can thus apply our two inductive hypotheses via T_Arrow, giving:

$$\frac{\left(\begin{smallmatrix}\Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2)\\ (\hat{\sigma_3}+\hat{s}*\sigma_2),\hat{\sigma_4}\\ \mathbf{0}\end{smallmatrix}\right) \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]C : \mathsf{Type}_{l_1} \qquad \left(\begin{smallmatrix}\Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2),((\hat{\sigma_3}+\hat{s}*\sigma_2),\hat{\sigma_4})\\ ((\hat{\sigma_3}'+\hat{s}'*\sigma_2),\hat{\sigma_4}'),r'\\ \mathbf{0}\end{smallmatrix}\right) \odot \Gamma_1, [t/x]\Gamma_2, y : [t/x]C \vdash [t/x]D : \mathsf{Type}_{l_2}}{\left(\begin{smallmatrix}\Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2)\\ ((\hat{\sigma_3}+\hat{s}*\sigma_2),\hat{\sigma_4})+((\hat{\sigma_3}'+\hat{s}'*\sigma_2),\hat{\sigma_4}')\\ \mathbf{0}\end{smallmatrix}\right) \odot \Gamma_1, [t/x]\Gamma_2 \vdash (y :_{(s',r')} [t/x]C) \to [t/x]D : \mathsf{Type}_{l_1 \sqcup l_2}} \text{T\_Arrow}$$

Rewriting the subject grades of this conclusion by Lemma 3.18, we have $((\hat{\sigma_3}+\hat{s}*\sigma_2),\hat{\sigma_4}) + ((\hat{\sigma_3}'+\hat{s}'*\sigma_2),\hat{\sigma_4}') = (\sigma_3+s*\sigma_2),\sigma_4$. As $x \neq y$ (by disjointness of concatenation), we have $(y :_{(s',r')} [t/x]C) \to [t/x]D = [t/x]((y :_{(s',r')} C) \to D)$. We have $[t/x]\mathsf{Type}_{l_1 \sqcup l_2} = \mathsf{Type}_{l_1 \sqcup l_2}$ trivially, as $\mathsf{Type}_{l_1 \sqcup l_2}$ is a constant. Using this information to rewrite the conclusion of T_Arrow derived above, we have:

$$\left(\begin{smallmatrix}\Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2)\\ (\sigma_3+s*\sigma_2),\sigma_4\\ \mathbf{0}\end{smallmatrix}\right) \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]((y :_{(s',r')} C) \to D) : [t/x]\mathsf{Type}_{l_1 \sqcup l_2}$$

Which matches the goal.

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_r A) \otimes B : \mathsf{Type}_{l_1 \sqcup l_2}} \text{T\_Ten}$$

This proceeds similarly to the case for T_Arrow.

Case.

$$\frac{\Delta, \sigma_1, \Delta' \odot \Gamma_1, x : A, \Gamma_2 \vdash \quad |\Delta| = |\Gamma_1|}{(\Delta, \sigma_1, \Delta' \mid \mathbf{0}^{|\Delta|}, 1, \mathbf{0} \mid \sigma_1, 0, \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash x : A} \text{T\_Var}$$

Therefore we have $t' = x$; and $B = A$. Our full goal is therefore:

$$\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ \sigma_2,\mathbf{0} \\ \sigma_1,\mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash t : A$$

We can form the following inductive premises:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $\Delta, \sigma_1, \Delta' \odot \Gamma_1, x : A, \Gamma_2 \vdash$ (T_Var well-formedness premise)

Giving:

$$\Delta, (\Delta'\backslash\pi + (\Delta'/\pi) * \sigma_2) \odot \Gamma_1, [t/x]\Gamma_2 \vdash$$

We can then form the following derivation, to achieve our goal:

$$\frac{\Delta, (\Delta'\backslash\pi + (\Delta'/\pi) * \sigma_2) \odot \Gamma_1, [t/x]\Gamma_2 \vdash \quad (\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A}{\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ \sigma_2,\mathbf{0} \\ \sigma_1,\mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash t : A} \text{ Lemma 3.25}$$

Case.

$$\frac{\Delta, \sigma_1, \hat{\Delta}', (\sigma_5, r, \hat{\sigma_6}), \hat{\Delta}'' \odot \Gamma_1, x : A, \hat{\Gamma_2}, y : C, \hat{\Gamma_2}' \vdash \quad \left|\Delta, \sigma_1, \hat{\Delta}'\right| = \left|\Gamma_1, x : A, \hat{\Gamma_2}\right|}{(\Delta, \sigma_1, \hat{\Delta}', (\sigma_5, r, \hat{\sigma_6}), \hat{\Delta}'' \mid \mathbf{0}^{\left|\Delta, (\sigma_5, r, \hat{\sigma_6}), \hat{\Delta}'\right|}, 1, \mathbf{0} \mid \sigma_5, r, \hat{\sigma_6}, 0, \mathbf{0}) \odot \Gamma_1, x : A, \hat{\Gamma_2}, y : C, \hat{\Gamma_2}' \vdash y : C} \text{ T\_Var}$$

Therefore we have $t' = y$; and $B = C$. Our full goal is therefore:

$$\begin{pmatrix} \Delta,((\hat{\Delta}',(\sigma_5,r,\hat{\sigma_6}),\hat{\Delta}'')\backslash\pi+((\hat{\Delta}',(\sigma_5,r,\hat{\sigma_6}),\hat{\Delta}'')/\pi)*\sigma_2) \\ \mathbf{0}^{|\Delta|},1,\mathbf{0} \\ (\sigma_5+r*\sigma_2),\hat{\sigma_6},\mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\hat{\Gamma_2}, y : [t/x]C, [t/x]\hat{\Gamma_2}' \vdash y : [t/x]C$$

We can form the following inductive premises:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $\Delta, \sigma_1, \hat{\Delta}', (\sigma_5, r, \hat{\sigma_6}), \hat{\Delta}'' \odot \Gamma_1, x : A, \hat{\Gamma_2}, y : C, \hat{\Gamma_2}' \vdash$ (T_Var well-formedness premise)

Giving:

$$\Delta, ((\hat{\Delta}', (\sigma_5, r, \hat{\sigma_6}), \hat{\Delta}'')\backslash\pi + ((\hat{\Delta}', (\sigma_5, r, \hat{\sigma_6}), \hat{\Delta}'')/\pi * \sigma_2)) \odot \Gamma_1, [t/x]\hat{\Gamma_2}, y : [t/x]C, [t/x]\hat{\Gamma_2}' \vdash$$

By Lemma 3.17, we have:

$$(\hat{\Delta}', (\sigma_5, r, \hat{\sigma_6}), \hat{\Delta}'')\backslash\pi + (\hat{\Delta}', (\sigma_5, r, \hat{\sigma_6}))/\pi * \sigma_2 =$$
$$(\hat{\Delta}'\backslash\pi + \hat{\Delta}'/\pi * \sigma_2), ((\sigma_5 + r * \sigma_2), \hat{\sigma_6}), (\hat{\Delta}''\backslash\pi + \hat{\Delta}''/\pi * \sigma_2)$$

We can apply this information to T_Var, and use the context grade vector equality derived above to rewrite the conclusion, to attain our goal:

$$\frac{|\Delta| = |\Gamma_1| \qquad \Delta, (\hat{\Delta}'\backslash\pi + \hat{\Delta}'/\pi * \sigma_2), ((\sigma_5 + r * \sigma_2), \hat{\sigma_6}), (\hat{\Delta}''\backslash\pi + \hat{\Delta}''/\pi * \sigma_2) \odot \Gamma_1, [t/x]\hat{\Gamma_2}, y : [t/x]C, [t/x]\hat{\Gamma_2}' \vdash}{\begin{pmatrix} \Delta,((\hat{\Delta}',(\sigma_5,r,\hat{\sigma_6}),\hat{\Delta}'')\backslash\pi+((\hat{\Delta}',(\sigma_5,r,\hat{\sigma_6}),\hat{\Delta}'')/\pi*\sigma_2)) \\ \mathbf{0}^{|\Delta|},1,\mathbf{0} \\ (\sigma_5+r*\sigma_2),\hat{\sigma_6},\mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\hat{\Gamma_2}, y : [t/x]C, [t/x]\hat{\Gamma_2}' \vdash y : [t/x]C} \text{ T\_Var}$$

Case.

$$\frac{\hat{\Delta}, \hat{\sigma_5}, \hat{\Delta}', \sigma_1, \Delta' \odot \hat{\Gamma_1}, y : C, \hat{\Gamma_1}', x : A, \Gamma_2 \vdash \quad \left|\hat{\Delta}\right| = \left|\hat{\Gamma_1}\right|}{(\hat{\Delta}, \hat{\sigma_5}, \hat{\Delta}', \sigma_1, \Delta' \mid \mathbf{0}^{|\hat{\Delta}|}, 1, \mathbf{0} \mid \hat{\sigma_5}, \mathbf{0}) \odot \hat{\Gamma_1}, y : C, \hat{\Gamma_1}', x : A, \Gamma_2 \vdash y : C} \text{ T\_Var}$$

Therefore we have $t' = y$; and $B = C$. Our full goal is therefore:

$$\begin{pmatrix} \hat{\Delta}, \hat{\sigma_5}, \hat{\Delta}', (\Delta' \backslash \pi + \Delta'/\pi * \sigma_2) \\ \mathbf{0}^{|\hat{\Delta}|}, 1, \mathbf{0} \\ \hat{\sigma_5}, \mathbf{0} \end{pmatrix} \odot \hat{\Gamma_1}, y : C, [t/x]\hat{\Gamma_1}', [t/x]\Gamma_2 \vdash y : C$$

We can form the following inductive premises:

1. $(\hat{\Delta}, \hat{\sigma_5}, \hat{\Delta}' \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $\hat{\Delta}, \hat{\sigma_5}, \hat{\Delta}', \sigma_1, \Delta' \odot \hat{\Gamma_1}, y : C, \hat{\Gamma_1}', x : A, \Gamma_2 \vdash$ (T_Var well-formedness premise)

Giving:

$$\hat{\Delta}, \hat{\sigma_5}, \hat{\Delta}', (\Delta' \backslash \pi + \Delta'/\pi * \sigma_2) \odot \hat{\Gamma_1}, y : C, \hat{\Gamma_1}', [t/x]\Gamma_2 \vdash$$

We can apply this information in the following derivation, to attain our goal:

$$\frac{\hat{\Delta}, \hat{\sigma_5}, \hat{\Delta}', (\Delta' \backslash \pi + \Delta'/\pi * \sigma_2) \odot \hat{\Gamma_1}, y : C, \hat{\Gamma_1}', [t/x]\Gamma_2 \vdash \quad \left|\hat{\Delta}\right| = \left|\hat{\Gamma_1}\right|}{\begin{pmatrix} (\hat{\Delta}, \hat{\sigma_5}, \hat{\Delta}', (\Delta' \backslash \pi + \Delta'/\pi * \sigma_2)) \\ \mathbf{0}^{|\hat{\Delta}|}, 1, \mathbf{0} \\ \hat{\sigma_5}, \mathbf{0} \end{pmatrix} \odot \hat{\Gamma_1}, y : C, \hat{\Gamma_1}', [t/x]\Gamma_2 \vdash y : C} \text{ T\_Var}$$

Case.

$$\frac{\begin{pmatrix} \Delta, \sigma_1, \Delta', \sigma_7 \\ \sigma_8, r' \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2, y : C \vdash D : \mathsf{Type}_l \qquad \begin{pmatrix} \Delta, \sigma_1, \Delta', \sigma_7 \\ \sigma_3, s, \sigma_4, s' \\ \sigma_8, r' \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2, y : C \vdash t'' : D}{\begin{pmatrix} (\Delta, \sigma_1), \Delta' \\ (\sigma_3, s), \sigma_4 \\ \sigma_7 + \sigma_8 \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2 \vdash \lambda y.t'' : (y :_{(s', r')} C) \to D} \text{ T\_Fun}$$

Therefore we have $\sigma_5, r, \sigma_6 = \sigma_7 + \sigma_8$; $t' = \lambda y.t''$; and $B = (y :_{(s', r')} C) \to D$. Our goal is therefore:

$$\begin{pmatrix} \Delta, (\Delta' \backslash \pi + (\Delta'/\pi) * \sigma_2) \\ (\sigma_3 + s * \sigma_2), \sigma_4 \\ (\sigma_5 + r * \sigma_2), \sigma_6 \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x](\lambda y.t'') : [t/x]((y :_{(s', r')} C) \to D)$$

As $\sigma_5, r, \sigma_6 = \sigma_7 + \sigma_8$, we therefore have $\sigma_7 = \hat{\sigma_5}, \hat{r}, \hat{\sigma_6}$ and $\sigma_8 = \hat{\sigma_5}', \hat{r}', \hat{\sigma_6}'$ where $\hat{\sigma_5} + \hat{\sigma_5}' = \sigma_5$; $\hat{r} + \hat{r}' = r$; and $\hat{\sigma_6} + \hat{\sigma_6}' = \sigma_6$. Rewriting the premise of T_Fun using the above information, we have the following inductive premises:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $\begin{pmatrix} \Delta, \sigma_1, \Delta', (\hat{\sigma_5}, \hat{r}, \hat{\sigma_6}) \\ \hat{\sigma_5}', \hat{r}', \hat{\sigma_6}', r' \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2, y : C \vdash D : \mathsf{Type}_l$ (T_Fun premise for D)
3. $\left|\hat{\sigma_5}'\right| = \left|\hat{\sigma_5}'\right| = |\Gamma_1|$ (trivially, and by premise (3))

And:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $\begin{pmatrix} \Delta,\sigma_1,\Delta',(\hat{\sigma_5},\hat{r},\hat{\sigma_6}) \\ \sigma_3,s,\sigma_4,s' \\ \hat{\sigma_5}',\hat{r}',\hat{\sigma_6}',r' \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2, y : C \vdash t'' : D$ (T_Fun premise for t'')
3. $|\hat{\sigma_3}| = \left|\hat{\sigma_5}'\right| = |\Gamma_1|$ (trivially, and by premise (3))

Giving, respectively:

$$\begin{pmatrix} \Delta,(((\Delta',(\hat{\sigma_5},\hat{r},\hat{\sigma_6}))\backslash\pi)+((\Delta',(\hat{\sigma_5},\hat{r},\hat{\sigma_6}))/\pi)*\sigma_2) \\ (\hat{\sigma_5}'+\hat{r}'*\sigma_2),\hat{\sigma_6}',r' \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2, y : [t/x]C \vdash [t/x]D : \mathsf{Type}_l$$

And:

$$\begin{pmatrix} \Delta,(((\Delta',(\hat{\sigma_5},\hat{r},\hat{\sigma_6}))\backslash\pi)+((\Delta',(\hat{\sigma_5},\hat{r},\hat{\sigma_6}))/\pi)*\sigma_2) \\ (\sigma_3+s*\sigma_2),\sigma_4,s' \\ (\hat{\sigma_5}'+\hat{r}'*\sigma_2),\hat{\sigma_6}',r' \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2, y : [t/x]C \vdash [t/x]t'' : [t/x]D$$

Rewriting $((\Delta', (\hat{\sigma_5}, \hat{r}, \hat{\sigma_6}))\backslash\pi) + ((\Delta', (\hat{\sigma_5}, \hat{r}, \hat{\sigma_6}))/\pi) * \sigma_2$ by Lemma 3.17, we have $(\Delta'\backslash\pi + (\Delta'/\pi) * \sigma_2), ((\hat{\sigma_5} + \hat{r} * \sigma_2), \hat{\sigma_6})$, and can thus apply our inductive hypotheses to T_Fun, giving:

$$\dfrac{\begin{pmatrix} \Delta,((\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2),((\hat{\sigma_5}+\hat{r}*\sigma_2),\hat{\sigma_6})) \\ (\hat{\sigma_5}'+\hat{r}'*\sigma_2),\hat{\sigma_6}',r' \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2, y : [t/x]C \vdash [t/x]D : \mathsf{Type}_l \quad \begin{pmatrix} \Delta,((\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2),((\hat{\sigma_5}+\hat{r}*\sigma_2),\hat{\sigma_6})) \\ (\sigma_3+s*\sigma_2),\sigma_4,s' \\ (\hat{\sigma_5}'+\hat{r}'*\sigma_2),\hat{\sigma_6}',r' \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2, y : [t/x]C \vdash [t/x]t'' : [t/x]D}{\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\sigma_3+s*\sigma_2),\sigma_4 \\ ((\hat{\sigma_5}+\hat{r}*\sigma_2),\hat{\sigma_6})+((\hat{\sigma_5}'+\hat{r}'*\sigma_2),\hat{\sigma_6}') \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash \lambda y.[t/x]t'' : (y :_{(s',r')} [t/x]C) \to [t/x]D} \text{T\_Fun}$$

Rewriting the subject grades of this conclusion by Lemma 3.18, we have $((\hat{\sigma_5} + \hat{r} * \sigma_2), \hat{\sigma_6}) + ((\hat{\sigma_5}' + \hat{r}' * \sigma_2), \hat{\sigma_6}') = (\sigma_5 + r * \sigma_2), \sigma_6$. As $x \neq y$ (by disjointness of concatenation), we have $\lambda y.[t/x]t'' = [t/x](\lambda y.t'')$ and $(y :_{(s',r')} [t/x]C) \to [t/x]D = [t/x]((y :_{(s',r')} C) \to D)$. Using this information to rewrite the conclusion of T_Fun derived above, we have:

$$\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\sigma_3+s*\sigma_2),\sigma_4 \\ (\sigma_5+r*\sigma_2),\sigma_6 \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x](\lambda y.t'') : [t/x]((y :_{(s',r')} C) \to D)$$

Which matches the goal.

Case.

$$\dfrac{\begin{array}{c} (\Delta, \sigma_1, \Delta', \sigma_7 \mid \sigma_9, r' \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2, y : C \vdash D : \mathsf{Type}_l \\ (\Delta, \sigma_1, \Delta' \mid \sigma_8 \mid \sigma_7 + \sigma_9) \odot \Gamma_1, x : A, \Gamma_2 \vdash t_1 : (y :_{(s',r')} C) \to D \\ (\Delta, \sigma_1, \Delta' \mid \sigma_{10} \mid \sigma_7) \odot \Gamma_1, x : A, \Gamma_2 \vdash t_2 : C \end{array}}{(\Delta, \sigma_1, \Delta' \mid \sigma_8 + s' * \sigma_{10} \mid \sigma_9 + r' * \sigma_{10}) \odot \Gamma_1, x : A, \Gamma_2 \vdash t_1\, t_2 : [t_2/y]D} \text{T\_App}$$

Therefore we have $\sigma_3, s, \sigma_4 = \sigma_8 + s' * \sigma_{10}$; $\sigma_8 = \hat{\sigma_3}, \hat{s}, \hat{\sigma_4}$ (where $|\hat{\sigma_3}| = |\Gamma_1|$); $\sigma_{10} = \hat{\sigma_3}', \hat{s}', \hat{\sigma_4}'$ (where $\left|\hat{\sigma_3}'\right| = |\Gamma_1|$); $\sigma_5, r, \sigma_6 = \sigma_7 + r' * \sigma_{10}$; $\sigma_7 = \hat{\sigma_5}, \hat{r}, \hat{\sigma_6}$ (where $|\hat{\sigma_5}| = |\Gamma_1|$); $\sigma_9 = \hat{\sigma_5}', \hat{r}', \hat{\sigma_6}'$ (where $\left|\hat{\sigma_5}'\right| = |\Gamma_1|$); $t' = t_1\, t_2$; and $B = [t_2/y]D$. We have $\hat{\sigma_3} + s' * \hat{\sigma_3}' = \sigma_3$, $\hat{\sigma_5} + r' * \hat{\sigma_5}' = \sigma_5$, and likewise for other components. Our goal is therefore:

$$\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\sigma_3+s*\sigma_2),\sigma_4 \\ (\sigma_5+r*\sigma_2),\sigma_6 \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x](t_1\, t_2) : [t/x][t_2/y]D$$

Rewriting by Lemma 3.20, we have $\sigma_7 + \sigma_9 = (\hat{\sigma_5} + \hat{\sigma_5}'), (\hat{r} + \hat{r}'), (\hat{\sigma_6} + \hat{\sigma_6}')$. Rewriting the premises of T_App using the above information, we have the following inductive premises:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $\begin{pmatrix} \Delta,\sigma_1,\Delta',(\hat{\sigma_5},\hat{r},\hat{\sigma_6}) \\ \hat{\sigma_5}',\hat{r}',\hat{\sigma_6}',r' \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2, y : C \vdash D : \mathsf{Type}_l$ (T_App premise for D)
3. $\left|\hat{\sigma_5}'\right| = \left|\hat{\sigma_5} + \hat{\sigma_5}'\right| = |\Gamma_1|$ (trivially, and by premise (3))

And:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $\begin{pmatrix} \Delta,\sigma_1,\Delta' \\ \hat{\sigma_3},\hat{s},\hat{\sigma_4} \\ (\hat{\sigma_5}+\hat{\sigma_5}'),(\hat{r}+\hat{r}'),(\hat{\sigma_6}+\hat{\sigma_6}') \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2 \vdash t_1 : (y :_{(s',r')} C) \to D$ (T_App premise for t_1)
3. $|\hat{\sigma_3}| = \left|\hat{\sigma_5} + \hat{\sigma_5}'\right| = |\Gamma_1|$ (trivially, and by premise (3))

And:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $\begin{pmatrix} \Delta,\sigma_1,\Delta' \\ \hat{\sigma_3}',\hat{s}',\hat{\sigma_4}' \\ \hat{\sigma_5},\hat{r},\hat{\sigma_6} \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2 \vdash t_2 : C$ (T_App premise for t_2)
3. $\left|\hat{\sigma_3}'\right| = |\hat{\sigma_5}| = |\Gamma_1|$ (trivially, and by premise (3))

Giving, respectively:

$$\begin{pmatrix} \Delta,(((\Delta',(\hat{\sigma_5},\hat{r},\hat{\sigma_6}))\backslash\pi)+((\Delta',(\hat{\sigma_5},\hat{r},\hat{\sigma_6}))/\pi)*\sigma_2) \\ (\hat{\sigma_5}'+\hat{r}'*\sigma_2),\hat{\sigma_6}',r' \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2, y : [t/x]C \vdash [t/x]D : \mathsf{Type}_l$$

And:

$$\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\hat{\sigma_3}+\hat{s}*\sigma_2),\hat{\sigma_4} \\ (\hat{\sigma_5}+\hat{\sigma_5}'+(\hat{r}+\hat{r}')*\sigma_2),(\hat{\sigma_6}+\hat{\sigma_6}') \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]t_1 : (y :_{(s',r')} [t/x]C) \to [t/x]D$$

And:

$$\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\hat{\sigma_3}'+\hat{s}'*\sigma_2),\hat{\sigma_4}' \\ (\hat{\sigma_5}+\hat{r}*\sigma_2),\hat{\sigma_6} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]t_2 : [t/x]C$$

By vector rewriting $(\hat{\sigma_5} + \hat{\sigma_5}' + (\hat{r} + \hat{r}') * \sigma_2), (\hat{\sigma_6} + \hat{\sigma_6}')$ we have $(\hat{\sigma_5} + \hat{r} * \sigma_2), \hat{\sigma_6} + (\hat{\sigma_5}' + \hat{r}' * \sigma_2), \hat{\sigma_6}'$. Rewriting $((\Delta', (\hat{\sigma_5}, \hat{r}, \hat{\sigma_6}))\backslash\pi) + ((\Delta', (\hat{\sigma_5}, \hat{r}, \hat{\sigma_6}))/\pi) * \sigma_2$ by Lemma 3.17, we have $(\Delta'\backslash\pi + (\Delta'/\pi) * \sigma_2), ((\hat{\sigma_5} + \hat{r} * \sigma_2), \hat{\sigma_6})$, and can thus apply our inductive hypotheses to T_App, giving:

$$\dfrac{\begin{array}{c} \begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2),((\hat{\sigma_5}+\hat{r}*\sigma_2),\hat{\sigma_6}) \\ (\hat{\sigma_5}'+\hat{r}'*\sigma_2),\hat{\sigma_6}',r' \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2, y : [t/x]C \vdash [t/x]D : \mathsf{Type}_l \\ \begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\hat{\sigma_3}+\hat{s}*\sigma_2),\hat{\sigma_4} \\ (\hat{\sigma_5}+\hat{r}*\sigma_2),\hat{\sigma_6}+(\hat{\sigma_5}'+\hat{r}'*\sigma_2),\hat{\sigma_6}' \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]t_1 : (y :_{(s',r')} [t/x]C) \to [t/x]D \\ \begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\hat{\sigma_3}'+\hat{s}'*\sigma_2),\hat{\sigma_4}' \\ (\hat{\sigma_5}+\hat{r}*\sigma_2),\hat{\sigma_6} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]t_2 : [t/x]C \end{array}}{\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\hat{\sigma_3}+\hat{s}*\sigma_2),\hat{\sigma_4}+s'*(\hat{\sigma_3}'+\hat{s}'*\sigma_2),\hat{\sigma_4}' \\ (\hat{\sigma_5}'+\hat{r}'*\sigma_2),\hat{\sigma_6}'+r'*(\hat{\sigma_3}'+\hat{s}'*\sigma_2),\hat{\sigma_4}' \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2 \vdash ([t/x]t_1)\,([t/x]t_2) : [t_2/y][t/x]D} \text{T\_App}$$

Rewriting the subject and subject type grades of this conclusion by Lemma 3.18 (and expanding the scaling), we have $(\hat{\sigma_3}+\hat{s}*\sigma_2), \hat{\sigma_4}+(s'*\hat{\sigma_3}'+s'*\hat{s}'*\sigma_2), (s'*\hat{\sigma_4}') = (\sigma_3+s*\sigma_2), \sigma_4$ and $(\hat{\sigma_5}+\hat{r}*\sigma_2), \hat{\sigma_6}+(r'*\hat{\sigma_5}'+r'*\hat{r}'*\sigma_2), (r'*\hat{\sigma_6}') = (\sigma_5+r*\sigma_2), \sigma_6$. By definition we have $([t/x]t_1)\,([t/x]t_2) = [t/x](t_1\,t_2)$. As $x \neq y$ (by disjointness of concatenation), we have $[t_2/y][t/x]D = [t/x][t_2/y]D$. Using this information to rewrite the conclusion of T_App derived above, we have:

$$\begin{pmatrix} \Delta, (\Delta' \backslash \pi + (\Delta'/\pi) * \sigma_2) \\ (\sigma_3+s*\sigma_2), \sigma_4 \\ (\sigma_5+r*\sigma_2), \sigma_6 \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x](t_1\,t_2) : [t/x][t_2/y]D$$

Which matches the goal.

Case.

$$\frac{\begin{array}{c}(\Delta, \sigma_1, \Delta', \sigma_7 \mid \sigma_9, r' \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2, y : C \vdash D : \mathsf{Type}_l \\ (\Delta, \sigma_1, \Delta' \mid \sigma_8 \mid \sigma_7) \odot \Gamma_1, x : A, \Gamma_2 \vdash t_1 : C \\ (\Delta, \sigma_1, \Delta' \mid \sigma_{10} \mid \sigma_9 + r' * \sigma_8) \odot \Gamma_1, x : A, \Gamma_2 \vdash t_2 : [t_1/y]D\end{array}}{(\Delta, \sigma_1, \Delta' \mid \sigma_8 + \sigma_{10} \mid \sigma_7 + \sigma_9) \odot \Gamma_1, x : A, \Gamma_2 \vdash (t_1, t_2) : (y :_{r'} C) \otimes D} \text{ T\_Pair}$$

Therefore we have $\sigma_3, s, \sigma_4 = \sigma_8 + \sigma_{10}$; $\sigma_8 = \hat{\sigma_3}, \hat{s}, \hat{\sigma_4}$ (where $|\hat{\sigma_3}| = |\Gamma_1|$); $\sigma_{10} = \hat{\sigma_3}', \hat{s}', \hat{\sigma_4}'$ (where $|\hat{\sigma_3}'| = |\Gamma_1|$); $\sigma_5, r, \sigma_6 = \sigma_7 + \sigma_9$; $\sigma_7 = \hat{\sigma_5}, \hat{r}, \hat{\sigma_6}$ (where $|\hat{\sigma_5}| = |\Gamma_1|$); $\sigma_9 = \hat{\sigma_5}', \hat{r}', \hat{\sigma_6}'$ (where $|\hat{\sigma_5}'| = |\Gamma_1|$); $t' = (t_1, t_2)$; and $B = (y :_{r'} C) \otimes D$. We have $\hat{\sigma_3} + \hat{\sigma_3}' = \sigma_3$, and likewise for other components. Our goal is therefore:

$$\begin{pmatrix} \Delta, (\Delta' \backslash \pi + (\Delta'/\pi) * \sigma_2) \\ (\sigma_3+s*\sigma_2), \sigma_4 \\ (\sigma_5+r*\sigma_2), \sigma_6 \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x](t_1, t_2) : [t/x]((y :_{r'} C) \otimes D)$$

As $\sigma_9 = \hat{\sigma_5}', \hat{r}', \hat{\sigma_6}'$, $\sigma_8 = \hat{\sigma_3}, \hat{s}, \hat{\sigma_4}$, and $|\hat{\sigma_5}'| = |\hat{\sigma_3}|$, we can rewrite $\sigma_9 + r' * \sigma_8$ by Lemma 3.20 to obtain $(\hat{\sigma_5}' + r' * \hat{\sigma_3}), (\hat{r}' + r' * \hat{s}), (\hat{\sigma_6}' + r' * \hat{\sigma_4})$. Rewriting the premises of T_Pair using the above information, we have the following inductive premises:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $\begin{pmatrix} \Delta, \sigma_1, \Delta', (\hat{\sigma_5}, \hat{r}, \hat{\sigma_6}) \\ \hat{\sigma_5}', \hat{r}', \hat{\sigma_6}', r' \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2, y : C \vdash D : \mathsf{Type}_l$ (first premise of T_Pair)
3. $|\hat{\sigma_5}'| = |\mathbf{0}| = |\Gamma_1|$ (trivially, and by premise (3))

And:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $\begin{pmatrix} \Delta, \sigma_1, \Delta' \\ \hat{\sigma_3}, \hat{s}, \hat{\sigma_4} \\ \hat{\sigma_5}, \hat{r}, \hat{\sigma_6} \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2 \vdash t_1 : C$ (second premise of T_Pair)
3. $|\hat{\sigma_3}| = |\hat{\sigma_5}| = |\Gamma_1|$ (trivially, and by premise (3))

And:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $\begin{pmatrix} \Delta, \sigma_1, \Delta' \\ \hat{\sigma_3}', \hat{s}', \hat{\sigma_4}' \\ (\hat{\sigma_5}'+r'*\hat{\sigma_3}), (\hat{r}'+r'*\hat{s}), (\hat{\sigma_6}'+r'*\hat{\sigma_4}) \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2 \vdash t_2 : [t_1/y]D$ (third premise of T_Pair)
3. $|\hat{\sigma_3}'| = |\hat{\sigma_5}' + r' * \hat{\sigma_3}| = |\Gamma_1|$ (trivially, and by premise (3))

Giving, respectively:

$$\begin{pmatrix} \Delta,(((\Delta',(\hat{\sigma_5},\hat{r},\hat{\sigma_6}))\backslash\pi)+((\Delta',(\hat{\sigma_5},\hat{r},\hat{\sigma_6}))/\pi)*\sigma_2) \\ (\hat{\sigma_5}'+\hat{r}'*\sigma_2),\hat{\sigma_6}',r' \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2, y : [t/x]C \vdash [t/x]D : \mathsf{Type}_l$$

And:

$$\begin{pmatrix} \Delta,\sigma_1,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\hat{\sigma_3}+\hat{s}*\sigma_2),\hat{\sigma_4} \\ (\hat{\sigma_5}+\hat{r}*\sigma_2),\hat{\sigma_6} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]t_1 : [t/x]C$$

And:

$$\begin{pmatrix} \Delta,\sigma_1,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\hat{\sigma_3}'+\hat{s}'*\sigma_2),\hat{\sigma_4}' \\ ((\hat{\sigma_5}'+r'*\hat{\sigma_3})+(\hat{r}'+r'*\hat{s})*\sigma_2),(\hat{\sigma_6}'+r'*\hat{\sigma_4}) \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]t_2 : [t/x][t_1/y]D$$

By vector rewriting $((\hat{\sigma_5}' + r' * \hat{\sigma_3}) + (\hat{r}' + r' * \hat{s}) * \sigma_2), (\hat{\sigma_6}' + r' * \hat{\sigma_4})$ we have $(\hat{\sigma_5}' + \hat{r}' * \sigma_2), \hat{\sigma_6}' + r' * (\hat{\sigma_3} + \hat{s} * \sigma_2), \hat{\sigma_4}$. As $y \notin \mathsf{FV}(t)$ and $x \neq y$, we have $[t/x][t_1/y]D = [[t/x]t_1/y][t/x]D$. Rewriting $((\Delta', (\hat{\sigma_5}, \hat{r}, \hat{\sigma_6}))\backslash\pi) + ((\Delta', (\hat{\sigma_5}, \hat{r}, \hat{\sigma_6}))/\pi) * \sigma_2$ by Lemma 3.17, we have $(\Delta'\backslash\pi + (\Delta'/\pi) * \sigma_2), ((\hat{\sigma_5} + \hat{r} * \sigma_2), \hat{\sigma_6})$. Thus we can apply our inductive hypotheses via T_Pair, giving:

$$\frac{\begin{array}{c} \begin{pmatrix} \Delta,((\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2),((\hat{\sigma_5}+\hat{r}*\sigma_2),\hat{\sigma_6})) \\ (\hat{\sigma_5}'+\hat{r}'*\sigma_2),\hat{\sigma_6}',r' \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2, y : [t/x]C \vdash [t/x]D : \mathsf{Type}_l \\ \begin{pmatrix} \Delta,\sigma_1,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\hat{\sigma_3}+\hat{s}*\sigma_2),\hat{\sigma_4} \\ (\hat{\sigma_5}+\hat{r}*\sigma_2),\hat{\sigma_6} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]t_1 : [t/x]C \\ \begin{pmatrix} \Delta,\sigma_1,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\hat{\sigma_3}'+\hat{s}'*\sigma_2),\hat{\sigma_4}' \\ (\hat{\sigma_5}'+\hat{r}'*\sigma_2),\hat{\sigma_6}'+r'*((\hat{\sigma_3}+\hat{s}*\sigma_2),\hat{\sigma_4}) \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]t_2 : [[t/x]t_1/y][t/x]D \end{array}}{\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\hat{\sigma_3}+\hat{s}*\sigma_2),\hat{\sigma_4}+(\hat{\sigma_3}'+\hat{s}'*\sigma_2),\hat{\sigma_4}' \\ (\hat{\sigma_5}+\hat{r}*\sigma_2),\hat{\sigma_6}+(\hat{\sigma_5}'+\hat{r}'*\sigma_2),\hat{\sigma_6}' \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash ([t/x]t_1, [t/x]t_2) : (y :_{r'} [t/x]C) \otimes [t/x]D} \text{ T\_PAIR}$$

Rewriting the subject and subject type grades of this conclusion by Lemma 3.18, we have $(\hat{\sigma_3} + \hat{s} * \sigma_2), \hat{\sigma_4} + (\hat{\sigma_3}' + \hat{s}' * \sigma_2), \hat{\sigma_4}' = (\sigma_3 + s * \sigma_2), \sigma_4$ and $(\hat{\sigma_5} + \hat{r} * \sigma_2), \hat{\sigma_6} + (\hat{\sigma_5}' + \hat{r}' * \sigma_2), \hat{\sigma_6}' = (\sigma_5 + r * \sigma_2), \sigma_6$. By definition we have $([t/x]t_1, [t/x]t_2) = [t/x](t_1, t_2)$. As $x \neq y$ (by disjointness of concatenation), we have $(y :_{r'} [t/x]C) \otimes [t/x]D = [t/x]((y :_{r'} C) \otimes D)$. Using this information to rewrite the conclusion of T_Pair derived above, we have:

$$\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\sigma_3+s*\sigma_2),\sigma_4 \\ (\sigma_5+r*\sigma_2),\sigma_6 \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x](t_1, t_2) : [t/x]((y :_{r'} C) \otimes D)$$

Which matches the goal.

Case.

$$\frac{\begin{array}{c} (\Delta, \sigma_1, \Delta' \mid \sigma_9 \mid \sigma_7 + \sigma_8) \odot \Gamma_1, x : A, \Gamma_2 \vdash t_1 : (y :_{r'} C) \otimes D \\ \begin{pmatrix} \Delta,\sigma_1,\Delta',\sigma_7,(\sigma_8,r') \\ \sigma_{10},s',s' \\ \sigma_{11},r'',r'' \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2, y : C, z : D \vdash t_2 : [(y, z)/w]E \\ (\Delta, \sigma_1, \Delta', (\sigma_7 + \sigma_8) \mid \sigma_{11}, r'' \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2, w : (y :_{r'} C) \otimes D \vdash E : \mathsf{Type}_l \end{array}}{(\Delta, \sigma_1, \Delta' \mid \sigma_{10} + s' * \sigma_9 \mid \sigma_{11} + r'' * \sigma_9) \odot \Gamma_1, x : A, \Gamma_2 \vdash \mathsf{let}\,(y, z) = t_1 \,\mathsf{in}\, t_2 : [t_1/w]E} \text{ T\_TENCUT}$$

Therefore we have $\sigma_3, s, \sigma_4 = \sigma_{10} + s' * \sigma_9$; $\sigma_5, r, \sigma_6 = \sigma_{11} + r'' * \sigma_9$; $\sigma_7 = \hat{\sigma_7}, s_7, \hat{\sigma_7}'$ (where $|\hat{\sigma_7}| = |\Gamma_1|$); $\sigma_8 = \hat{\sigma_8}, s_8, \hat{\sigma_8}'$ (where $|\hat{\sigma_8}| = |\Gamma_1|$); $t' = \mathsf{let}\,(y, z) = t_1 \,\mathsf{in}\, t_2$; and $B = [t_1/w]E$. Our goal is therefore:

$$\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\sigma_3+s*\sigma_2),\sigma_4 \\ (\sigma_5+r*\sigma_2),\sigma_6 \end{pmatrix} \odot \Gamma_1,[t/x]\Gamma_2 \vdash [t/x](\mathsf{let}\,(y,z)=t_1\,\mathsf{in}\,t_2) : [t/x][t_1/w]E$$

As $\sigma_3, s, \sigma_4 = \sigma_{10}+s'*\sigma_9$, we therefore have $\sigma_{10} = \hat{\sigma_3}, \hat{s}, \hat{\sigma_4}$ and $\sigma_9 = \hat{\sigma_3}', \hat{s}', \hat{\sigma_4}'$ where $\hat{\sigma_3}+s'*\hat{\sigma_3}' = \sigma_3$; $\hat{s}+s'*\hat{s}' = s$; and $\hat{\sigma_4}+s'*\hat{\sigma_4}' = \sigma_4$. As $\sigma_5, r, \sigma_6 = \sigma_{11}+r''*\sigma_9$, we therefore have $\sigma_{11} = \hat{\sigma_5}, \hat{r}, \hat{\sigma_6}$ where $\hat{\sigma_5} + r'' * \hat{\sigma_3}' = \sigma_5$; $\hat{r} + r'' * \hat{s}' = r$; and $\hat{\sigma_6} + r'' * \hat{\sigma_4}' = \sigma_6$. As $\sigma_7 = \hat{\sigma_7}, s_7, \hat{\sigma_7}'$ and $\sigma_8 = \hat{\sigma_8}, s_8, \hat{\sigma_8}'$, we have $\sigma_7 + \sigma_8 = (\hat{\sigma_7} + \hat{\sigma_8}), (s_7 + s_8), (\hat{\sigma_7}' + \hat{\sigma_8}')$. Rewriting the premises using the above information, we have the following inductive premises:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $\begin{pmatrix} \Delta,\sigma_1,\Delta' \\ \hat{\sigma_3}',\hat{s}',\hat{\sigma_4}' \\ (\hat{\sigma_7}+\hat{\sigma_8}),(s_7+s_8),(\hat{\sigma_7}'+\hat{\sigma_8}') \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2 \vdash t_1 : (y :_{r'} C) \otimes D$ (t_1 premise of T_TenCut)
3. $\left|\hat{\sigma_3}'\right| = |\hat{\sigma_7} + \hat{\sigma_8}| = |\Gamma_1|$ (trivially, and by premise (3))

And:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $\begin{pmatrix} \Delta,\sigma_1,\Delta',(\hat{\sigma_7},s_7,\hat{\sigma_7}'),(\hat{\sigma_8},s_8,\hat{\sigma_8}',r') \\ \hat{\sigma_3},\hat{s},\hat{\sigma_4},s',s' \\ \hat{\sigma_5},\hat{r},\hat{\sigma_6},r'',r'' \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2, y : C, z : D \vdash t_2 : [(y,z)/w]E$ (t_2 premise of T_TenCut)
3. $|\hat{\sigma_3}| = |\hat{\sigma_5}| = |\Gamma_1|$ (trivially, and by premise (3))

And:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $\begin{pmatrix} \Delta,\sigma_1,\Delta',((\hat{\sigma_7}+\hat{\sigma_8}),(s_7+s_8),(\hat{\sigma_7}'+\hat{\sigma_8}')) \\ \hat{\sigma_5},\hat{r},\hat{\sigma_6},r'' \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2, w : (y :_{r'} C) \otimes D \vdash E : \mathsf{Type}_l$ (E premise of T_TenCut)
3. $|\hat{\sigma_3}| = |\sigma_5| = |\Gamma_1|$ (trivially, and by premise (3))

Giving, respectively:

$$\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\hat{\sigma_3}'+\hat{s}'*\sigma_2),\hat{\sigma_4}' \\ (\hat{\sigma_7}+\hat{\sigma_8}+(s_7+s_8)*\sigma_2),(\hat{\sigma_7}'+\hat{\sigma_8}') \end{pmatrix} \odot \Gamma_1,[t/x]\Gamma_2 \vdash [t/x]t_1 : (y :_{r'} [t/x]C) \otimes [t/x]D$$

And:

$$\begin{pmatrix} \Delta,(((\Delta',(\hat{\sigma_7},s_7,\hat{\sigma_7}'),(\hat{\sigma_8},s_8,\hat{\sigma_8}',r'))\backslash\pi)+((\Delta',(\hat{\sigma_7},s_7,\hat{\sigma_7}'),(\hat{\sigma_8},s_8,\hat{\sigma_8}',r'))/\pi)*\sigma_2) \\ (\hat{\sigma_3}+\hat{s}*\sigma_2),\hat{\sigma_4},s',s' \\ (\hat{\sigma_5}+\hat{r}*\sigma_2),\hat{\sigma_6},r'',r'' \end{pmatrix} \odot \Gamma_1,[t/x]\Gamma_2, y : [t/x]C, z : [t/x]D$$
$$\vdash [t/x]t_2 : [t/x][(y,z)/w]E$$

And:

$$\begin{pmatrix} \Delta,(((\Delta',(\hat{\sigma_7}+\hat{\sigma_8}),(s_7+s_8),(\hat{\sigma_7}'+\hat{\sigma_8}'))\backslash\pi)+((\Delta',(\hat{\sigma_7}+\hat{\sigma_8}),(s_7+s_8),(\hat{\sigma_7}'+\hat{\sigma_8}'))/\pi)*\sigma_2) \\ (\hat{\sigma_5}+\hat{r}*\sigma_2),\hat{\sigma_6},r'' \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1,[t/x]\Gamma_2, w : (y :_{r'} [t/x]C) \otimes [t/x]D$$
$$\vdash [t/x]E : \mathsf{Type}_l$$

As $w \neq x$ (by disjointness of concatenation), we have $[t/x][(y,z)/w]E = [(y,z)/w][t/x]E$. By vector rewriting $(\hat{\sigma_7} + \hat{\sigma_8} + (s_7 + s_8) * \sigma_2), (\hat{\sigma_7}' + \hat{\sigma_8}')$ we have $(\hat{\sigma_7} + s_7 * \sigma_2), \hat{\sigma_7}' + (\hat{\sigma_8} + s_8 * \sigma_2), \hat{\sigma_8}'$.

Rewriting $((\Delta', (\hat{\sigma_7}, s_7, \hat{\sigma_7}'), (\hat{\sigma_8}, s_8, \hat{\sigma_8}'))\backslash\pi) + ((\Delta', (\hat{\sigma_7}, s_7, \hat{\sigma_7}'), (\hat{\sigma_8}, s_8, \hat{\sigma_8}'))/\pi) * \sigma_2$ by Lemma 3.17, we have $(\Delta'\backslash\pi + (\Delta'/\pi) * \sigma_2), ((\hat{\sigma_7} + s_7 * \sigma_2), \hat{\sigma'_7}), ((\hat{\sigma_8} + s_8 * \sigma_2), \hat{\sigma_8}', r')$. By vector rewriting $(((\Delta', (\hat{\sigma_7} + \hat{\sigma_8}), (s_7 + s_8), (\hat{\sigma_7}' + \hat{\sigma_8}'))\backslash\pi) + ((\Delta', (\hat{\sigma_7} + \hat{\sigma_8}), (s_7 + s_8), (\hat{\sigma_7}' + \hat{\sigma_8}'))/\pi) * \sigma_2)$ we have $(\Delta'\backslash\pi + (\Delta'/\pi) * \sigma_2), (((\hat{\sigma_7} + s_7 * \sigma_2), \hat{\sigma_7}') + ((\hat{\sigma_8} + s_8 * \sigma_2), \hat{\sigma_8}'))$. Therefore we can apply our inductive hypotheses via T_TenCut, giving:

$$\dfrac{\begin{matrix} \begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\hat{\sigma_3}'+\hat{s}'*\sigma_2),\hat{\sigma_4}' \\ ((\hat{\sigma_7}+s_7*\sigma_2),\hat{\sigma_7}')+((\hat{\sigma_8}+s_8*\sigma_2),\hat{\sigma_8}') \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]t_1 : (y :_{r'} [t/x]C) \otimes [t/x]D \\ \begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2),((\hat{\sigma_7}+s_7*\sigma_2),\hat{\sigma'_7}),((\hat{\sigma_8}+s_8*\sigma_2),\hat{\sigma_8}',r') \\ (\hat{\sigma_3}+\hat{s}*\sigma_2),\hat{\sigma_4},s',s' \\ (\hat{\sigma_5}+\hat{r}*\sigma_2),\hat{\sigma_6},r'',r'' \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2, y : [t/x]C, z : [t/x]D \vdash [t/x]t_2 : [(y,z)/w][t/x]E \\ \begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2),(((\hat{\sigma_7}+s_7*\sigma_2),\hat{\sigma_7}')+((\hat{\sigma_8}+s_8*\sigma_2),\hat{\sigma_8}')) \\ (\hat{\sigma_5}+\hat{r}*\sigma_2),\hat{\sigma_6},r'' \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2, w : (y :_{r'} [t/x]C) \otimes [t/x]D \vdash [t/x]E : \mathsf{Type}_l \end{matrix}}{\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ ((\hat{\sigma_3}+\hat{s}*\sigma_2),\hat{\sigma_4})+s'*((\hat{\sigma_3}'+\hat{s}'*\sigma_2),\hat{\sigma_4}') \\ ((\hat{\sigma_5}+\hat{r}*\sigma_2),\hat{\sigma_6})+r''*((\hat{\sigma_3}'+\hat{s}'*\sigma_2),\hat{\sigma_4}') \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash \mathsf{let}\,(y,z) = [t/x]t_1 \,\mathsf{in}\, [t/x]t_2 : [t_1/w][t/x]E} \text{ T\_TenCut}$$

Rewriting the subject and subject-type grades of this conclusion by Lemma 3.18 (after expanding the scaling), we have $((\hat{\sigma_3} + \hat{s} * \sigma_2), \hat{\sigma_4}) + ((s' * \hat{\sigma_3}' + s' * \hat{s}' * \sigma_2), (s' * \hat{\sigma_4}')) = (\sigma_3 + s * \sigma_2), \sigma_4$, and $((\hat{\sigma_5} + \hat{r} * \sigma_2), \hat{\sigma_6}) + ((r'' * \hat{\sigma_3}' + r'' * \hat{s}' * \sigma_2), (r'' * \hat{\sigma_4}')) = (\sigma_5 + r * \sigma_2), \sigma_6$. As $x \neq y$ and $x \neq z$ (by disjointness of concatenation), we have $\mathsf{let}\,(y,z) = [t/x]t_1 \,\mathsf{in}\, [t/x]t_2 = [t/x](\mathsf{let}\,(y,z) = t_1 \,\mathsf{in}\, t_2)$. As $w \neq x$, and $w \notin \mathsf{FV}\,(t)$, we have $[t_1/w][t/x]E = [t/x][t_1/w]E$. Using this information to rewrite the conclusion of T_TenCut derived above, we have:

$$\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\sigma_3+s*\sigma_2),\sigma_4 \\ (\sigma_5+r*\sigma_2),\sigma_6 \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x](\mathsf{let}\,(y,z) = t_1 \,\mathsf{in}\, t_2) : [t/x][t_1/w]E$$

Which matches the goal.

Case.

$$\dfrac{(\Delta, \sigma_1, \Delta' \mid \sigma_3, s, \sigma_4 \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash C : \mathsf{Type}_{l_1}}{(\Delta, \sigma_1, \Delta' \mid \sigma_3, s, \sigma_4 \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash \Box_{s'} C : \mathsf{Type}_l} \text{ T\_Box}$$

Therefore we have $\sigma_5 = \mathbf{0}$, $r = 0$, $\sigma_6 = \mathbf{0}$, $t' = \Box_{s'} C$, and $B = \mathsf{Type}_l$. Our full goal is therefore:

$$\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\sigma_3+s*\sigma_2),\sigma_4 \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x](\Box_{s'} C) : [t/x]\mathsf{Type}_l$$

Rewriting the premise using the above information, we have the following inductive premises:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $(\Delta, \sigma_1, \Delta' \mid \sigma_3, s, \sigma_4 \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash C : \mathsf{Type}_l$ (premise of T_Box)
3. $|\sigma_3| = |\mathbf{0}| = |\Gamma_1|$ (trivially, and by premise (3))

Giving:

$$\begin{pmatrix} \Delta,((\Delta'\backslash\pi)+(\Delta'/\pi)*\sigma_2) \\ (\sigma_3+s*\sigma_2),\sigma_4 \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]C : [t/x]\mathsf{Type}_l$$

Rewriting $[t/x]\mathsf{Type}_l = \mathsf{Type}_l$ and applying the inductive hypothesis to T_Box, we have:

$$\frac{(\Delta, ((\Delta'\backslash\pi) + (\Delta'/\pi) * \sigma_2) \mid (\sigma_3 + s * \sigma_2), \sigma_4 \mid \mathbf{0}) \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]C : \mathsf{Type}_{l_1}}{(\Delta, ((\Delta'\backslash\pi) + (\Delta'/\pi) * \sigma_2) \mid (\sigma_3 + s * \sigma_2), \sigma_4 \mid \mathbf{0}) \odot \Gamma_1, [t/x]\Gamma_2 \vdash \Box_{s'}[t/x]C : \mathsf{Type}_l} \text{ T\_Box}$$

As $[t/x]\mathsf{Type}_l = \mathsf{Type}_l$ and $[t/x](\Box_{s'}C) = \Box_{s'}[t/x]C$, we have:

$$\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\sigma_3+s*\sigma_2),\sigma_4 \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x](\Box_{s'}C) : [t/x]\mathsf{Type}_l$$

Which matches the goal.

Case.

$$\frac{(\Delta, \sigma_1, \Delta' \mid \sigma_7 \mid \sigma_5, r, \sigma_6) \odot \Gamma_1, x : A, \Gamma_2 \vdash t'' : C}{(\Delta, \sigma_1, \Delta' \mid s' * \sigma_7 \mid \sigma_5, r, \sigma_6) \odot \Gamma_1, x : A, \Gamma_2 \vdash \Box t'' : \Box_{s'}C} \text{ T\_BoxI}$$

Therefore we have $\sigma_3, s, \sigma_4 = s' * \sigma_7$; $\sigma_7 = \hat{\sigma_3}, \hat{s}, \hat{\sigma_4}$ (where $|\hat{\sigma_3}| = |\Gamma_1|$) ; $t' = \Box t''$; and $B = \Box_{s'}C$. We have $s' * \hat{\sigma_3} = \sigma_3$, and likewise for other components. Our goal is therefore:

$$\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\sigma_3+s*\sigma_2),\sigma_4 \\ (\sigma_5+r*\sigma_2),\sigma_6 \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x](\Box t'') : [t/x](\Box_{s'}C)$$

Rewriting the premise of T_BoxI using the above information, we have the following inductive premises:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $\begin{pmatrix} \Delta,\sigma_1,\Delta' \\ \hat{\sigma_3},\hat{s},\hat{\sigma_4} \\ \sigma_5,r,\sigma_6 \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2 \vdash t'' : C$ (premise of T_BoxI)
3. $|\hat{\sigma_3}| = |\sigma_5| = |\Gamma_1|$ (trivially, and by premise (3))

Giving:

$$\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\hat{\sigma_3}+\hat{s}*\sigma_2),\hat{\sigma_4} \\ (\sigma_5+r*\sigma_2),\sigma_6 \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]t'' : [t/x]C$$

We can apply our inductive hypothesis to T_BoxI, giving:

$$\frac{\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\hat{\sigma_3}+\hat{s}*\sigma_2),\hat{\sigma_4} \\ (\sigma_5+r*\sigma_2),\sigma_6 \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]t'' : [t/x]C}{\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ s'*(\hat{\sigma_3}+\hat{s}*\sigma_2),\hat{\sigma_4} \\ (\sigma_5+r*\sigma_2),\sigma_6 \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2 \vdash \Box[t/x]t'' : \Box_{s'}[t/x]C} \text{ T\_BoxI}$$

Rewriting the subject and subject type grades of this conclusion by Lemma 3.18 (and expanding the scaling), we have $(s' * \hat{\sigma_3}) + (s' * \hat{s} * \sigma_2), (s' * \hat{\sigma_4}) = (\sigma_3 + s * \sigma_2), \sigma_4$. By definition we have $\Box[t/x]t'' = [t/x](\Box t'')$ and $\Box_{s'}[t/x]C = [t/x](\Box_{s'}C)$. Using this information to rewrite the conclusion of T_BoxI derived above, we have:

$$\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\sigma_3+s*\sigma_2),\sigma_4 \\ (\sigma_5+r*\sigma_2),\sigma_6 \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x](\Box t'') : [t/x](\Box_{s'}C)$$

Which matches the goal.

Case.

$$\frac{\begin{array}{c}(\Delta, \sigma_1, \Delta', \sigma_8 \mid \sigma_{10}, r' \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2, z : \Box_{s'} C \vdash D : \mathsf{Type}_l \\ (\Delta, \sigma_1, \Delta' \mid \sigma_7 \mid \sigma_8) \odot \Gamma_1, x : A, \Gamma_2 \vdash t_1 : \Box_{s'} C \\ (\Delta, \sigma_1, \Delta', \sigma_8 \mid \sigma_9, s' \mid \sigma_{10}, (s' * r')) \odot \Gamma_1, x : A, \Gamma_2, y : C \vdash t_2 : [\Box y/z]D\end{array}}{(\Delta, \sigma_1, \Delta' \mid \sigma_7 + \sigma_9 \mid \sigma_{10} + r' * \sigma_7) \odot \Gamma_1, x : A, \Gamma_2 \vdash \mathsf{let}\,\Box y = t_1\,\mathsf{in}\,t_2 : [t_1/z]D}\ \text{T\_BoxE}$$

Therefore we have $\sigma_3, s, \sigma_4 = \sigma_7 + \sigma_9$; $\sigma_7 = \hat{\sigma_3}, \hat{s}, \hat{\sigma_4}$ (where $|\hat{\sigma_3}| = |\Gamma_1|$); $\sigma_8 = \hat{\sigma_8}, s'', \hat{\sigma_8}'$ (where $|\hat{\sigma_8}| = |\Gamma_1|$); $\sigma_9 = \hat{\sigma_3}', \hat{s}', \hat{\sigma_4}'$ (where $\left|\hat{\sigma_3}'\right| = |\Gamma_1|$); $\sigma_5, r, \sigma_6 = \sigma_{10} + r' * \sigma_7$; $\sigma_{10} = \hat{\sigma_5}, \hat{r}, \hat{\sigma_6}$ (where $|\hat{\sigma_5}| = |\Gamma_1|$); $t' = \mathsf{let}\,\Box y = t_1\,\mathsf{in}\,t_2$; and $B = [t_1/z]D$. We have $\hat{\sigma_3} + \hat{\sigma_3}' = \sigma_3$, $\hat{\sigma_5} + r' * \hat{\sigma_3} = \sigma_5$, and likewise for other components. Our goal is therefore:

$$\begin{pmatrix} \Delta, (\Delta' \backslash \pi + (\Delta'/\pi) * \sigma_2) \\ (\sigma_3 + s * \sigma_2), \sigma_4 \\ (\sigma_5 + r * \sigma_2), \sigma_6 \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x](\mathsf{let}\,\Box y = t_1\,\mathsf{in}\,t_2) : [t/x]([t_2/z]D)$$

We have the following inductive premises:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $\begin{pmatrix} \Delta, \sigma_1, \Delta', (\hat{\sigma_8}, s'', \hat{\sigma_8}') \\ \hat{\sigma_5}, \hat{r}, \hat{\sigma_6}, r' \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2, z : \Box_{s'} C \vdash D : \mathsf{Type}_l$ (T_BoxE premise for D)
3. $|\hat{\sigma_5}| = |\mathbf{0}| = |\Gamma_1|$ (trivially, and by premise (3))

And:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $\begin{pmatrix} \Delta, \sigma_1, \Delta' \\ \hat{\sigma_3}, \hat{s}, \hat{\sigma_4} \\ \hat{\sigma_8}, s'', \hat{\sigma_8}' \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2 \vdash t_1 : \Box_{s'} C$ (T_BoxE premise for t_1)
3. $|\hat{\sigma_3}| = |\hat{\sigma_8}| = |\Gamma_1|$ (trivially, and by premise (3))

And:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $\begin{pmatrix} \Delta, \sigma_1, \Delta', (\hat{\sigma_8}, s'', \hat{\sigma_8}') \\ \hat{\sigma_3}', \hat{s}', \hat{\sigma_4}', s' \\ \hat{\sigma_5}, \hat{r}, \hat{\sigma_6}, (s' * r') \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2, y : C \vdash t_2 : [\Box y/z]D$ (T_BoxE premise for t_2)
3. $\left|\hat{\sigma_3}'\right| = |\hat{\sigma_5}| = |\Gamma_1|$ (trivially, and by premise (3))

Giving, respectively:

$$\begin{pmatrix} \Delta, (((\Delta', (\hat{\sigma_8}, s'', \hat{\sigma_8}')) \backslash \pi) + ((\Delta', (\hat{\sigma_8}, s'', \hat{\sigma_8}'))/\pi) * \sigma_2) \\ (\hat{\sigma_5} + \hat{r} * \sigma_2), \hat{\sigma_6}, r' \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2, z : \Box_{s'}[t/x]C \vdash [t/x]D : \mathsf{Type}_l$$

And:

$$\begin{pmatrix} \Delta, (\Delta' \backslash \pi + (\Delta'/\pi) * \sigma_2) \\ (\hat{\sigma_3} + \hat{s} * \sigma_2), \hat{\sigma_4} \\ (\hat{\sigma_8} + s'' * \sigma_2), \hat{\sigma_8}' \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]t_1 : \Box_{s'}[t/x]C$$

And:

$$\begin{pmatrix} \Delta, (((\Delta', (\hat{\sigma_8}, s'', \hat{\sigma_8}')) \backslash \pi) + ((\Delta', (\hat{\sigma_8}, s'', \hat{\sigma_8}'))/\pi) * \sigma_2) \\ (\hat{\sigma_3}' + \hat{s}' * \sigma_2), \hat{\sigma_4}', s' \\ (\hat{\sigma_5} + \hat{r} * \sigma_2), \hat{\sigma_6}, (s' * r') \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2, y : [t/x]C \vdash [t/x]t_2 : [t/x]D$$

Rewriting $((\Delta', (\hat{\sigma}_8, s'', \hat{\sigma}_8{}'))\backslash\pi) + ((\Delta', (\hat{\sigma}_8, s'', \hat{\sigma}_8{}'))/\pi) * \sigma_2$ by Lemma 3.17, we have $(\Delta'\backslash\pi + (\Delta'/\pi) * \sigma_2), ((\hat{\sigma}_8 + s'' * \sigma_2), \hat{\sigma}_8{}')$.

We can thus apply our inductive hypotheses via T_BoxE, giving:

$$\dfrac{\begin{pmatrix} \Delta,((\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2),((\hat{\sigma}_8+s''*\sigma_2),\hat{\sigma}_8{}')) \\ (\hat{\sigma}_5+\hat{r}*\sigma_2),\hat{\sigma}_6,r' \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2, z : \Box_{s'}[t/x]C \vdash [t/x]D : \mathsf{Type}_l \quad \begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\hat{\sigma}_3+\hat{s}*\sigma_2),\hat{\sigma}_4 \\ (\hat{\sigma}_8+s''*\sigma_2),\hat{\sigma}_8{}' \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]t_1 : \Box_{s'}[t/x]C \quad \begin{pmatrix} \Delta,((\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2),((\hat{\sigma}_8+s''*\sigma_2),\hat{\sigma}_8{}')) \\ (\hat{\sigma}_3{}'+\hat{s}'*\sigma_2),\hat{\sigma}_4{}',s' \\ (\hat{\sigma}_5+\hat{r}*\sigma_2),\hat{\sigma}_6,(s'*r') \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2, y : [t/x]C \vdash [t/x]t_2 : [t/x]D}{\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\hat{\sigma}_3+\hat{s}*\sigma_2),\hat{\sigma}_4+(\hat{\sigma}_3{}'+\hat{s}'*\sigma_2),\hat{\sigma}_4{}' \\ (\hat{\sigma}_5+\hat{r}*\sigma_2),\hat{\sigma}_6+r'*(\hat{\sigma}_3+\hat{s}*\sigma_2),\hat{\sigma}_4 \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2 \vdash \mathsf{let}\,\Box y = [t/x]t_1 \,\mathsf{in}\, [t/x]t_2 : [t_2/z]([t/x]D)}\ \text{T\_BoxE}$$

Rewriting the subject and subject type grades of this conclusion by Lemma 3.18, we have $(\hat{\sigma}_3 + \hat{s} * \sigma_2), \hat{\sigma}_4 + (\hat{\sigma}_3{}' + \hat{s}' * \sigma_2), \hat{\sigma}_4{}' = (\sigma_3 + s * \sigma_2), \sigma_4$ and $(\hat{\sigma}_5 + \hat{r} * \sigma_2), \hat{\sigma}_6 + r' * (\hat{\sigma}_3 + \hat{s} * \sigma_2), \hat{\sigma}_4 = (\sigma_5 + r * \sigma_2), \sigma_6$. By definition we have $(\mathsf{let}\,\Box y = [t/x]t_1 \,\mathsf{in}\, [t/x]t_2) = [t/x](\mathsf{let}\,\Box y = t_1 \,\mathsf{in}\, t_2)$ (as $x \neq y$ by disjointness of concatenation), and similarly, we have $[t_2/z][t/x]D = [t/x]([t_2/z]D)$. Using this information to rewrite the conclusion of T_BoxE derived above, we have:

$$\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\sigma_3+s*\sigma_2),\sigma_4 \\ (\sigma_5+r*\sigma_2),\sigma_6 \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x](\mathsf{let}\,\Box y = t_1 \,\mathsf{in}\, t_2) : [t/x]([t_2/z]D)$$

Which matches the goal.

Case.

$$\dfrac{\begin{array}{c}(\Delta, \sigma_1, \Delta' \mid \sigma_3, s, \sigma_4 \mid \sigma_5, r, \sigma_6) \odot \Gamma_1, x : A, \Gamma_2 \vdash t' : C \\ (\Delta, \sigma_1, \Delta' \mid \sigma_5, r, \sigma_6 \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash C = B : \mathsf{Type}_l\end{array}}{(\Delta, \sigma_1, \Delta' \mid \sigma_3, s, \sigma_4 \mid \sigma_5, r, \sigma_6) \odot \Gamma_1, x : A, \Gamma_2 \vdash t' : B}\ \text{T\_TyConv}$$

Our full goal is therefore:

$$\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\sigma_3+s*\sigma_2),\sigma_4 \\ (\sigma_5+r*\sigma_2),\sigma_6 \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]t' : [t/x]B$$

Rewriting the first and second premises using the above information, we have the following inductive premises:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $(\Delta, \sigma_1, \Delta' \mid \sigma_3, s, \sigma_4 \mid \sigma_5, r, \sigma_6) \odot \Gamma_1, x : A, \Gamma_2 \vdash t' : C$ (T_TyConv premise for t')
3. $|\sigma_3| = |\sigma_5| = |\Gamma_1|$ (premise (3))

And:

1. $(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma_1 \vdash t : A$ (premise (1))
2. $\begin{pmatrix} \Delta,\sigma_1,\Delta' \\ \sigma_5,r,\sigma_6 \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2 \vdash C = B : \mathsf{Type}_l$ (T_TyConv premise for $C = B$)
3. $|\sigma_5| = |\mathbf{0}| = |\Gamma_1|$ (trivially, and by premise (3))

Giving, respectively:

$$\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\sigma_3+s*\sigma_2),\sigma_4 \\ (\sigma_5+r*\sigma_2),\sigma_6 \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]t' : [t/x]C$$

And:

$$\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\sigma_5+r*\sigma_2),\sigma_6 \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]C = [t/x]B : \mathsf{Type}_l$$

Which we can apply to T_TyConv, giving:

$$\dfrac{\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\sigma_3+s*\sigma_2),\sigma_4 \\ (\sigma_5+r*\sigma_2),\sigma_6 \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]t' : [t/x]C \qquad \begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\sigma_5+r*\sigma_2),\sigma_6 \\ \mathbf{0} \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]C = [t/x]B : \mathsf{Type}_l}{\begin{pmatrix} \Delta,(\Delta'\backslash\pi+(\Delta'/\pi)*\sigma_2) \\ (\sigma_3+s*\sigma_2),\sigma_4 \\ (\sigma_5+r*\sigma_2),\sigma_6 \end{pmatrix} \odot \Gamma_1, [t/x]\Gamma_2 \vdash [t/x]t' : [t/x]B} \text{ T\_TyConv}$$

Which matches the goal.

For equality, we proceed by induction and re-application on the structure of $\begin{pmatrix} \Delta,\sigma_1,\Delta' \\ \sigma_3,s,\sigma_4 \\ \sigma_5,r,\sigma_6 \end{pmatrix} \odot \Gamma_1, x : A, \Gamma_2 \vdash t' = t'' : B$ (in some cases substitutions need to be rewritten, to see how to rewrite substitutions, refer to the typing cases). See Section A.8. For subtyping, by standard induction and re-application (see Section A.8). $\square$

Lemma 3.28 (Equality through substitution)**.** *If the following premises hold:*

1. $(\Delta_1 \mid \sigma_1 \mid \sigma_2) \odot \Gamma_1 \vdash t_1 = t_2 : A$
2. $(\Delta_1, \sigma_2, \Delta_2 \mid \sigma_3, s, \sigma_4 \mid \sigma_5, r, \sigma_6) \odot \Gamma_1, x : A, \Gamma_2 \vdash t_3 : B$
3. $|\sigma_3| = |\sigma_5| = |\Gamma_1|$

Then $\begin{pmatrix} \Delta_1,(\Delta_2\backslash\pi+(\Delta_2/\pi)*\sigma_1) \\ (\sigma_3+s*\sigma_1),\sigma_4 \\ (\sigma_5+r*\sigma_1),\sigma_6 \end{pmatrix} \odot \Gamma_1, [t_1/x]\Gamma_2 \vdash [t_1/x]t_3 = [t_2/x]t_3 : [t_1/x]B$*, where* $\pi = |\Gamma_1|$.

Proof. Cases proceed very similarly to the proof for Lemma 3.27, inducting on the form of $(\Delta_1, \sigma_2, \Delta_2 \mid \sigma_3, s, \sigma_4 \mid \sigma_5, r, \sigma_6) \odot \Gamma_1, x : A, \Gamma_2 \vdash t_3 : B$, and building results using appropriate equality rules. As such, we omit most cases, but provide a couple for demonstration, as follows:

Case.

$$\dfrac{(\Delta_1, \sigma_2, \Delta_2 \mid \sigma_3, s, \sigma_4 \mid \mathbf{0}) \odot \Gamma, x : A \vdash C : \mathsf{Type}_l}{(\Delta_1, \sigma_2, \Delta_2 \mid \sigma_3, s, \sigma_4 \mid \mathbf{0}) \odot \Gamma, x : A \vdash \square_{s'} C : \mathsf{Type}_l} \text{ T\_Box}$$

Then our goal holds by the following derivation:

$$\dfrac{\dfrac{\dfrac{(\Delta, \sigma_2, \Delta_2 \mid \sigma_3, s, \sigma_4 \mid \mathbf{0}) \odot \Gamma, x : A \vdash C : \mathsf{Type}_l}{(\Delta_1, (\Delta_2\backslash\pi + (\Delta_2/\pi) * \sigma_1) \mid (\sigma_3 + s * \sigma_1), \sigma_4 \mid \mathbf{0}) \odot \Gamma_1, [t_1/x]\Gamma_2 \vdash [t_1/x]C = [t_2/x]C : \mathsf{Type}_l} \text{ I.H.}}{(\Delta_1, (\Delta_2\backslash\pi + (\Delta_2/\pi) * \sigma_1) \mid (\sigma_3 + s * \sigma_1), \sigma_4 \mid \mathbf{0}) \odot \Gamma_1, [t_1/x]\Gamma_2 \vdash \square_{s'}[t_1/x]C = \square_{s'}[t_2/x]C : \mathsf{Type}_l} \text{ TEQ\_Box}}{(\Delta_1, (\Delta_2\backslash\pi + (\Delta_2/\pi) * \sigma_1) \mid (\sigma_3 + s * \sigma_1), \sigma_4 \mid \mathbf{0}) \odot \Gamma_1, [t_1/x]\Gamma_2 \vdash [t_1/x](\square_{s'} C) = [t_2/x](\square_{s'} C) : \mathsf{Type}_l} \equiv$$

Case.

$$\frac{\begin{array}{c}(\Delta_1, \sigma_2, \Delta_2 \mid \hat{\sigma_3}, \hat{s}, \hat{\sigma_4} \mid \hat{\sigma_7}, r_1, \hat{\sigma_7}') \odot \Gamma_1, x : A, \Gamma_2 \vdash t_4 : \Box_{s'} C \\ (\Delta_1, \sigma_2, \Delta_2, (\hat{\sigma_7}, r_1, \hat{\sigma_7}') \mid \hat{\sigma_5}, \hat{r}, \hat{\sigma_6}, r \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2, z : \Box_{s'} C \vdash D : \mathsf{Type}_l \\ (\Delta_1, \sigma_2, \Delta_2, (\hat{\sigma_7}, r_1, \hat{\sigma_7}') \mid \hat{\sigma_3}', \hat{s}', \hat{\sigma_4}', s' \mid \hat{\sigma_5}, \hat{r}, \hat{\sigma_6}, (s' * r')) \odot \Gamma_1, x : A, \Gamma_2, y : C \vdash t_5 : [\Box y/z]D\end{array}}{(\Delta_1, \sigma_2, \Delta_2 \mid \hat{\sigma_3}, \hat{s}, \hat{\sigma_4} + \hat{\sigma_3}', \hat{s}', \hat{\sigma_4}' \mid \hat{\sigma_5}, \hat{r}, \hat{\sigma_6} + r' * \hat{\sigma_3}, \hat{s}, \hat{\sigma_4}) \odot \Gamma_1, x : A, \Gamma_2 \vdash \mathsf{let}\, \Box y = t_4 \,\mathsf{in}\, t_5 : [t_4/z]D} \text{ T\_BOXE}$$

By Lemma 3.32 we have $(\Delta_1 \mid \sigma_1 \mid \sigma_2) \odot \Gamma_1 \vdash t_1 : A$, and therefore by induction on the premises of T_BoxE for t_4 and t_5, and by Lemma 3.27 on the premise for D, we have the following derivation:

$$\dfrac{\dfrac{\begin{array}{c}\begin{pmatrix}\Delta_1, (\Delta_2 \backslash \pi + (\Delta_2/\pi) * \sigma_1), ((\hat{\sigma_7} + r_1 * \sigma_1), \hat{\sigma_7}') \\ (\hat{\sigma_5} + \hat{r} * \sigma_1), \hat{\sigma_6}, r \\ \mathbf{0}\end{pmatrix} \odot \Gamma_1, [t_1/x]\Gamma_2, z : \Box_{s'} [t_1/x]C \vdash [t_1/x]D : \mathsf{Type}_l \\ \begin{pmatrix}\Delta_1 \\ \hat{\sigma_3} + \hat{s} * \sigma_1 \\ \sigma_5 + r_1 * \sigma_1\end{pmatrix} \odot \Gamma \vdash [t_1/x]t_3 = [t_2/x]t_3 : \Box_{s'} [t_1/x]C \\ \begin{pmatrix}\Delta_1, (\Delta_2 \backslash \pi + (\Delta_2/\pi) * \sigma_1), (\hat{\sigma_7} + r_1 * \hat{\sigma_7}') \\ (\hat{\sigma_3}' + \hat{s}' * \hat{\sigma_4}'), s' \\ (\hat{\sigma_4} + \hat{r} * \sigma_1), (s' * r')\end{pmatrix} \odot \Gamma_1, [t_1/x]\Gamma_2, y : C \vdash [t_1/x]t_4 = [t_2/x]t_4 : [\Box y/z]([t_1/x]D)\end{array}}{\begin{pmatrix}\Delta_1, (\Delta_2 \backslash \pi + (\Delta_2/\pi) * \sigma_1) \\ (\sigma_3 + s * \sigma_1), \sigma_4 \\ (\sigma_5 + r * \sigma_1), \sigma_6\end{pmatrix} \odot \Gamma_1, [t_1/x]\Gamma_2 \vdash (\mathsf{let}\, \Box y = [t_1/x]t_4 \,\mathsf{in}\, [t_1/x]t_5) = (\mathsf{let}\, \Box y = [t_2/x]t_4 \,\mathsf{in}\, [t_2/x]t_5) : [t_4/z]([t_1/x]D)} \text{ TEQ\_BOXE}}{\begin{pmatrix}\Delta_1, (\Delta_2 \backslash \pi + (\Delta_2/\pi) * \sigma_1) \\ (\sigma_3 + s * \sigma_1), \sigma_4 \\ (\sigma_5 + r * \sigma_1), \sigma_6\end{pmatrix} \odot \Gamma_1, [t_1/x]\Gamma_2 \vdash [t_1/x](\mathsf{let}\, \Box y = t_4 \,\mathsf{in}\, t_5) = [t_2/x](\mathsf{let}\, \Box y = t_4 \,\mathsf{in}\, t_5) : [t_1/x]([t_4/z]D)} \equiv$$

Giving us our goal.

$\square$

A.3 Proofs for vector manipulation

Lemma 3.17 (Factoring choose and discard)**.** *If* $|\sigma_1| = |\sigma_3| = \pi$*, then* $(\Delta_1, (\sigma_1, r, \sigma_2), \Delta_2) \backslash \pi + ((\Delta_1, (\sigma_1, r, \sigma_2), \Delta_2)/\pi) * \sigma_3 = (\Delta_1 \backslash \pi + \Delta_1/\pi * \sigma_3), ((\sigma_1 + r * \sigma_3), \sigma_2), (\Delta_2 \backslash \pi + \Delta_2/\pi * \sigma_3)$.

Proof. We show this by rewriting equationally, as follows:

$$\begin{array}{lr} & (\Delta_1, (\sigma_1, s, \sigma_2), \Delta_2) \backslash \pi + (\Delta_1, (\sigma_1, s, \sigma_2), \Delta_2)/\pi * \sigma_3 \\ \{\text{defn. 2.10 and defn. 2.8}\} & = (\Delta_1 \backslash \pi), ((\sigma_1, r, \sigma_2) \backslash \pi), (\Delta_2 \backslash \pi) + ((\Delta_1/\pi), ((\sigma_1, r, \sigma_2)/\pi), (\Delta_2/\pi)) * \sigma_3 \\ \{\text{defn. 2.9 and defn. 2.7, using } |\sigma_1| = \pi\} & = (\Delta_1 \backslash \pi), (\sigma_1, \sigma_2), (\Delta_2 \backslash \pi) + ((\Delta_1/\pi), r, (\Delta_2/\pi)) * \sigma_3 \\ \{\text{defn. 2.11}\} & = (\Delta_1 \backslash \pi), (\sigma_1, \sigma_2), (\Delta_2 \backslash \pi) + (\Delta_1/\pi * \sigma_3), (r * \sigma_3), (\Delta_2/\pi * \sigma_3) \\ \{\text{defn. 2.13}\} & = (\Delta_1 \backslash \pi + \Delta_1/\pi * \sigma_3), (\sigma_1, \sigma_2 + r * \sigma_3), (\Delta_2 \backslash \pi + \Delta_2/\pi * \sigma_3) \\ \{\text{L. 3.19 with } |\sigma_1| = |\sigma_3|\} & = (\Delta_1 \backslash \pi + \Delta_1/\pi * \sigma_3), ((\sigma_1 + r * \sigma_3), \sigma_2), (\Delta_2 \backslash \pi + \Delta_2/\pi * \sigma_3) \end{array}$$

As required. $\square$

Lemma 3.18 (Factoring vector addition)**.** *If* $\hat{\sigma_1} + \hat{\sigma_1}' = \sigma_1$*;* $\hat{s} + \hat{s'} = s$*; and* $\hat{\sigma_3} + \hat{\sigma_3}' = \sigma_3$*, then* $((\hat{\sigma_1} + \hat{s} * \sigma_2), \hat{\sigma_3}) + ((\hat{\sigma_1}' + \hat{s}' * \sigma_2), \hat{\sigma_3}') = (\sigma_1 + s * \sigma_2), \sigma_3$ *for all* σ_2.

Proof. This holds equationally, by laws of the semiring, and vector operations, as follows:

$$\begin{aligned} & ((\hat{\sigma_1} + \hat{s} * \sigma_2), \hat{\sigma_3}) + ((\hat{\sigma_1}' + \hat{s}' * \sigma_2), \hat{\sigma_3}') \\ & = ((\hat{\sigma_1} + \hat{s} * \sigma_2) + (\hat{\sigma_1}' + \hat{s}' * \sigma_2)), (\hat{\sigma_3} + \hat{\sigma_3}') \\ & = ((\hat{\sigma_1} + \hat{\sigma_1}') + (\hat{s} + \hat{s'}) * \sigma_2), \sigma_3 \\ & = (\sigma_1 + s * \sigma_2), \sigma_3 \end{aligned}$$

□

Lemma 3.19 (Addition across same-sized components)**.** *If* $|\sigma_1| = |\sigma_3|$ *then* $\sigma_1, \sigma_2 + \sigma_3 = (\sigma_1 + \sigma_3), \sigma_2$.

Proof. Trivially, by the implicit padding of the right-vector by zeros. □

Lemma 3.20 (Vector addition across components)**.** *If* $|\sigma_1| = |\sigma_4|$ *and* $|\sigma_2| = |\sigma_5|$*, then* $\sigma_1, \sigma_2, \sigma_3 + \sigma_4, \sigma_5, \sigma_6 = (\sigma_1 + \sigma_4), (\sigma_2 + \sigma_5), (\sigma_3 + \sigma_6)$.

Proof. This holds trivially, by definition of addition on vectors. □

Lemma 3.21 (Moving an exchange)**.** *We have* $\mathsf{mv}((\pi_1 + \pi_2); \pi_3; (\mathsf{exch}((\pi_1 + \pi_2); \Delta))) = \mathsf{mv}((\pi_1 + \pi_2 + 1); \pi_3; \Delta)$.

Proof. By definition, we have $\mathsf{exch}((\pi_1 + \pi_2); \Delta) = \mathsf{mv}((\pi_1 + \pi_2 + 1); (\pi_1 + \pi_2); \Delta)$. Therefore, the index $\pi_1 + \pi_2 + 1$ now corresponds to index $\pi_1 + \pi_2$ in $\mathsf{exch}((\pi_1 + \pi_2); \Delta)$, and thus $\mathsf{mv}((\pi_1 + \pi_2 + 1); \pi_3; \Delta)$ and $\mathsf{mv}((\pi_1 + \pi_2); \pi_3; (\mathsf{exch}((\pi_1 + \pi_2); \Delta)))$ are the same. □

A.4 Proofs for properties of operations

Lemma 3.29 (Properties of insertion)**.** *The following properties hold for any valid insertion (i.e., where for all* $i < |\Delta|$*,* $\pi \leq |\Delta[i]|$*, for context grade vectors; and* $\pi \leq |\sigma|$*, for grade vectors):*

1. *(insPreservesSize)* $|\mathsf{ins}(\pi; R; \Delta)| = |\Delta|$;
2. *(insIncSizes) if* $\pi \leq |\Delta[i]|$*, then* $|(\mathsf{ins}(\pi; R; \Delta))[i]| = |\Delta[i]| + 1$;
3. *(insIncSizeGV)* $|\mathsf{ins}(\pi; R; \sigma)| = |\sigma| + 1$;
4. *(insCVthenGV)* $\mathsf{ins}(\pi; R; \Delta), \mathsf{ins}(\pi; R; \sigma) = \mathsf{ins}(\pi; R; (\Delta, \sigma))$

Proof. Trivial by definition. □

A.5 Proofs for equality and conversion

Lemma 3.30 (Equality is an equivalence relation)**.** *For all, we have:*

- *(reflexivity) if* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$ *then* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t = t : A$;
- *(transitivity) if* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A$ *and* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_2 = t_3 : A$*, then* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_3 : A$;
- *(symmetry) if* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A$*, then* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_2 = t_1 : A$

Proof. Reflexivity holds by TEQ_REFL, transitivity holds by TEQ_TRANS, and symmetry holds by TEQ_SYM. □

Lemma 3.31 (Reduction implies equality)**.** *If* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : A$ *and* $t_1 \rightsquigarrow t_2$*, then* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A$.

Proof. By induction on the form of $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : A$, as follows:

Case.

$$\frac{\Delta \odot \Gamma \vdash}{(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}}\ \text{T\_TYPE}$$

No reductions possible, we are done.

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : \mathsf{Type}_{l_1 \sqcup l_2}} \text{T\_ARROW}$$

Case.

$$\frac{A \rightsquigarrow A'}{(x :_{(s,r)} A) \to B \rightsquigarrow (x :_{(s,r)} A') \to B} \text{SEM\_CONGARROW1}$$

By induction we have $(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A = A' : \mathsf{Type}_{l_1}$. Therefore, we can form the following derivation:

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A = A' : \mathsf{Type}_{l_1} \quad \dfrac{(\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l_2}}{(\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B = B : \mathsf{Type}_{l_2}} \text{TEQ\_REFL}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B = (x :_{(s,r)} A') \to B : \mathsf{Type}_{l_1 \sqcup l_2}} \text{TEQ\_ARROW}$$

Thus obtaining our goal.

Case.

$$\frac{B \rightsquigarrow B'}{(x :_{(s,r)} A) \to B \rightsquigarrow (x :_{(s,r)} A) \to B'} \text{SEM\_CONGARROW2}$$

By induction we have $(\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B = B' : \mathsf{Type}_{l_2}$. Therefore, we can form the following derivation:

$$\frac{\dfrac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l_1}}{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A = A : \mathsf{Type}_{l_1}} \text{TEQ\_REFL} \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma \vdash B = B' : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B = (x :_{(s,r)} A) \to B' : \mathsf{Type}_{l_1 \sqcup l_2}} \text{TEQ\_ARROW}$$

Thus obtaining our goal.

The case for T_TEN proceeds similarly, using SEM_CONGTEN1, SEM_CONGTEN2, and TEQ_TEN.

Case.

$$\frac{\begin{array}{c}(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma \vdash t_1 : A \quad (\Delta \mid \sigma_4 \mid \sigma_3 + r * \sigma_2) \odot \Gamma \vdash t_2 : [t_1/x]B\end{array}}{(\Delta \mid \sigma_2 + \sigma_4 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash (t_1, t_2) : (x :_r A) \otimes B} \text{T\_PAIR}$$

Proceeds similarly to the case for T_ARROW, but using SEM_CONGPAIR, SEM_CONGPAIRTWO, and TEQ_PAIR.

Case.

$$\frac{\Delta_1, \sigma, \Delta_2 \odot \Gamma_1, x : A, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \sigma, \Delta_2 \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \sigma, 0, \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash x : A} \text{T\_VAR}$$

No reductions possible, we are done.

Case.

$$\frac{\begin{array}{c}(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t_1 : (x :_{(s,r)} A) \to B \qquad (\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : A\end{array}}{(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash (\lambda x.t_1)\, t_2 : [t_2/x]B} \text{T\_APP}$$

With:

$$\frac{}{(\lambda x.t_1)\, t_2 \rightsquigarrow [t_2/x]t_1} \text{SEM\_BETAFUN}$$

By Lemma 3.11 we have $(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t_1 : B$. Then we have the following derivation:

$$\frac{(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t_1 : B \quad (\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : A}{(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash (\lambda x.t_1)\, t_2 = [t_2/x]t_1 : [t_2/x]B} \text{TEQ\_ARROWCOMP}$$

Giving us our goal.

Case.

$$\frac{\begin{array}{l}(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t_1 : (x :_{(s,r)} A) \to B \quad (\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : A\end{array}}{(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash t_1\, t_2 : [t_2/x]B} \text{T\_APP}$$

With:

$$\frac{t_1 \rightsquigarrow t_1'}{t_1\, t_2 \rightsquigarrow t_1'\, t_2} \text{SEM\_CONGFUNONE}$$

By induction we have $(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t_1 = t_1' : (x :_{(s,r)} A) \to B$. Therefore, we have the following derivation:

$$\frac{\begin{array}{c}(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t_1 = t_1' : (x :_{(s,r)} A) \to B \quad \dfrac{(\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : A}{(\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 = t_2 : A} \text{TEQ\_REFL}\end{array}}{(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash t_1\, t_2 = t_1'\, t_2 : [t_2/x]B} \text{TEQ\_APP}$$

Giving us our goal.

Case.

$$\frac{\begin{array}{c}(\Delta \mid \sigma_3 \mid \sigma_1 + \sigma_2) \odot \Gamma \vdash (t_1, t_2) : (x :_r A) \otimes B \\ (\Delta, (\sigma_1 + \sigma_2) \mid \sigma_5, r' \mid \mathbf{0}) \odot \Gamma, z : (x :_r A) \otimes B \vdash C : \mathsf{Type}_l \\ (\Delta, \sigma_1, (\sigma_2, r) \mid \sigma_4, s, s \mid \sigma_5, r', r') \odot \Gamma, x : A, y : B \vdash t_3 : [(x, y)/z]C\end{array}}{(\Delta \mid \sigma_4 + s * \sigma_3 \mid \sigma_5 + r' * \sigma_3) \odot \Gamma \vdash \mathsf{let}\,(x, y) = (t_1, t_2)\,\mathsf{in}\,t_3 : [(t_1, t_2)/z]C}\ \text{T\_TenCut}$$

With:

$$\frac{}{\mathsf{let}\,(x, y) = (t_1, t_2)\,\mathsf{in}\,t_3 \rightsquigarrow [t_1/x][t_2/y]t_3}\ \text{Sem\_BetaTen}$$

Then we obtain our goal by the following derivation:

$$\frac{\begin{array}{l}\mathcal{A} \\ (\Delta, \sigma_1, (\sigma_2, r) \mid \sigma_4, s, s \mid \sigma_5, r', r') \odot \Gamma, x : A, y : B \vdash t_3 : [(x, y)/z]C\end{array}}{(\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l}\ \text{L. 3.5}$$

By Lemma 3.12 with $\mathcal{A}$ (and premise), we have:

- $(\mathcal{B})$ $(\Delta \mid \hat{\sigma_3} \mid \sigma_1) \odot \Gamma \vdash t_1 : A$;
- $(\mathcal{C})$ $(\Delta \mid \hat{\sigma_3}' \mid \sigma_2 + r * \hat{\sigma_3}) \odot \Gamma \vdash t_2 : [t_1/x]B$;
- $\hat{\sigma_3} + \hat{\sigma_3}' = \sigma_3$

$$\frac{\begin{array}{c}\mathcal{B} \quad \mathcal{C} \\ (\Delta, (\sigma_1 + \sigma_2) \mid \sigma_5, r' \mid \mathbf{0}) \odot \Gamma, z : (x :_r A) \otimes B \vdash C : \mathsf{Type}_l \\ (\Delta, \sigma_1, (\sigma_2, r) \mid \sigma_4, s, s \mid \sigma_5, r', r') \odot \Gamma, x : A, y : B \vdash t_3 : [(x, y)/z]C\end{array}}{(\Delta \mid \sigma_4 + s * \sigma_3 \mid \sigma_5 + r' * \sigma_3) \odot \Gamma \vdash \mathsf{let}\,(x, y) = (t_1, t_2)\,\mathsf{in}\,t_3 = [t_1/x][t_2/y]t_3 : [(t_1, t_2)/z]C}\ \text{TEQ\_TenComp}$$

Case.

$$\frac{\begin{array}{l}(\Delta \mid \sigma_3 \mid \sigma_1 + \sigma_2) \odot \Gamma \vdash t_1 : (x :_r A) \otimes B \\ (\Delta, (\sigma_1 + \sigma_2) \mid \sigma_5, r' \mid \mathbf{0}) \odot \Gamma, z : (x :_r A) \otimes B \vdash C : \mathsf{Type}_l \\ (\Delta, \sigma_1, (\sigma_2, r) \mid \sigma_4, s, s \mid \sigma_5, r', r') \odot \Gamma, x : A, y : B \vdash t_2 : [(x, y)/z]C\end{array}}{(\Delta \mid \sigma_4 + s * \sigma_3 \mid \sigma_5 + r' * \sigma_3) \odot \Gamma \vdash \mathsf{let}\,(x, y) = t_1\,\mathsf{in}\,t_2 : [t_1/z]C}\ \text{T\_TenCut}$$

Remaining cases proceed similarly to those for T_App, using Sem_CongTenCut1, and TEQ_TenCut.

Case.

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A : \mathsf{Type}_l}\ \text{T\_Box}$$

With:

$$\frac{A \rightsquigarrow A'}{\Box_s A \rightsquigarrow \Box_s A'}\ \textsc{Sem\_CongBox4}$$

Then by induction we have $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A = A' : \mathsf{Type}_l$, and obtain our goal by application to TEQ_Box.

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box t_1 : \Box_s A \qquad \begin{array}{c}(\Delta, \sigma_2 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, z : \Box_s A \vdash B : \mathsf{Type}_l \\ (\Delta, \sigma_2 \mid \sigma_3, s \mid \sigma_4, (s * r)) \odot \Gamma, x : A \vdash t_2 : [\Box x/z]B\end{array}}{(\Delta \mid \sigma_1 + \sigma_3 \mid \sigma_4 + r * \sigma_1) \odot \Gamma \vdash \mathsf{let}\, \Box x = \Box t_1\, \mathsf{in}\, t_2 : [\Box t_1/z]B}\ \textsc{T\_BoxE}$$

With:

$$\frac{}{\mathsf{let}\, \Box x = \Box t_1\, \mathsf{in}\, t_2 \rightsquigarrow [t_1/x]t_2}\ \textsc{Sem\_BetaBox}$$

By Lemma 3.13 we have $(\Delta \mid \hat{\sigma_1} \mid \sigma_2) \odot \Gamma \vdash t_1 : A$ with $s * \hat{\sigma_1} = \sigma_1$, therefore we have the following derivation:

$$\frac{(\Delta \mid \hat{\sigma_1} \mid \sigma_2) \odot \Gamma \vdash t_1 : A \quad \begin{pmatrix}\Delta, \sigma_2 \\ \sigma_3, s \\ \sigma_4, (s*r)\end{pmatrix} \odot \Gamma, x : A \vdash t_2 : [\Box x/z]B \quad (\Delta, \sigma_2 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, z : \Box_s A \vdash B : \mathsf{Type}_l}{(\Delta \mid \sigma_3 + \sigma_1 \mid \sigma_4 + r * \sigma_1) \odot \Gamma \vdash (\mathsf{let}\, \Box x = \Box t_1\, \mathsf{in}\, t_2) = [t_1/x]t_2 : [\Box t_1/z]B}\ \textsc{TEQ\_BoxB}$$

Giving us our goal.

Case.

$$\frac{\begin{array}{l}(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : \Box_s A \\ (\Delta, \sigma_2 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, z : \Box_s A \vdash B : \mathsf{Type}_l \\ (\Delta, \sigma_2 \mid \sigma_3, s \mid \sigma_4, (s * r)) \odot \Gamma, x : A \vdash t_2 : [\Box x/z]B\end{array}}{(\Delta \mid \sigma_1 + \sigma_3 \mid \sigma_4 + r * \sigma_1) \odot \Gamma \vdash \mathsf{let}\, \Box x = t_1\, \mathsf{in}\, t_2 : [t_1/z]B}\ \textsc{T\_BoxE}$$

Remaining cases proceed similarly to those for T_App, using Sem_CongBox1, and TEQ_BoxE.

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash A \leq B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : B}\ \textsc{T\_TyConv}$$

With $t \rightsquigarrow t'$. By induction, we have $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t = t' : A$, therefore we have the following derivation:

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t = t' : A \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash A \leq B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t = t' : B}\ \textsc{TEQ\_ConvTy}$$

Giving us our goal.

$\square$

Lemma 3.32 (Equality inversion)**.** *If* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A$*, then* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : A$ *and* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_2 : A$.

Proof. By induction on the structure of $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A$, as follows:

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t = t : A}\ \text{TEQ\_R}\textsc{efl}$$

Then our goal holds by premise.

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A \quad (\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_2 = t_3 : A}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_3 : A}\ \text{TEQ\_T}\textsc{rans}$$

Then our goal holds by induction.

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_2 = t_1 : A}\ \text{TEQ\_S}\textsc{ym}$$

Then our goal holds by induction.

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash A \leq B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : B}\ \text{TEQ\_C}\textsc{onv}\text{T}\textsc{y}$$

Then our goals hold by the following derivations:

$$\frac{\dfrac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : A}\ \textsc{i.h.} \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash A \leq B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : B}\ \text{T\_T}\textsc{y}\text{C}\textsc{onv}$$

$$\frac{\dfrac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_2 : A}\ \textsc{i.h.} \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash A \leq B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_2 : B}\ \text{T\_T}\textsc{y}\text{C}\textsc{onv}$$

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A = C : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B = D : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B = (x :_{(s,r)} C) \to D : \mathsf{Type}_{l_1 \sqcup l_2}} \text{TEQ\_ARROW}$$

Then our goals hold by the following derivations:

$$\frac{\dfrac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A = C : \mathsf{Type}_{l_1}}{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l_1}} \text{I.H.} \quad \dfrac{(\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B = D : \mathsf{Type}_{l_2}}{(\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l_2}} \text{I.H.}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : \mathsf{Type}_{l_1 \sqcup l_2}} \text{T\_ARROW}$$

$$\frac{\dfrac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A = C : \mathsf{Type}_{l_1}}{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash C : \mathsf{Type}_{l_1}} \text{I.H.} \quad \dfrac{\dfrac{(\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B = D : \mathsf{Type}_{l_2}}{(\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash D : \mathsf{Type}_{l_2}} \text{I.H.}}{(\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : C \vdash D : \mathsf{Type}_{l_2}} \text{L. 3.33}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} C) \to D : \mathsf{Type}_{l_1 \sqcup l_2}} \text{T\_ARROW}$$

The case for TEQ_TEN proceeds similarly.

Case.

$$\frac{(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t_1 : B \quad (\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : A}{(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash (\lambda x.t_1)\, t_2 = [t_2/x]t_1 : [t_2/x]B} \text{TEQ\_ARROWCOMP}$$

Then our goals hold by the following derivations:

$$\mathcal{A}\quad \frac{(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t_1 : B}{(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l} \text{L. 3.6}$$

$$\mathcal{C}\quad \frac{\mathcal{A} \quad (\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t_1 : B}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t_1 : (x :_{(s,r)} A) \to B} \text{T\_FUN}$$

$$\frac{\mathcal{C} \quad \mathcal{A} \quad (\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : A}{(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash (\lambda x.t_1)\, t_2 : [t_2/x]B} \text{T\_APP}$$

$$\frac{(\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : A \quad (\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t_1 : B}{(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash [t_2/x]t_1 : [t_2/x]B} \text{L. 3.27}$$

The case for TEQ_TENCOMP proceeds similarly, using double substitution. The case for TEQ_BOXB proceeds similarly.

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : (x :_{(s,r)} A) \to B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t = \lambda x.(t\, x) : (x :_{(s,r)} A) \to B}\ \textsc{TEQ\_ArrowUniq}$$

Then our first goal holds by premise, and our second goal holds by the following derivation:

$$\mathcal{A}\quad \frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : (x :_{(s,r)} A) \to B}{(\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : \mathsf{Type}_l}\ \text{L. 3.6}$$

By Lemma 3.8 with $\mathcal{A}$, we have:

- $(\mathcal{B})$ $(\Delta \mid \hat{\sigma_2} \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l'}$;
- $(\mathcal{C})$ $(\Delta, \hat{\sigma_2} \mid \hat{\sigma_2}', r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l''}$;
- $\hat{\sigma_2} + \hat{\sigma_2}' = \sigma_2$

$$\mathcal{D}\quad \frac{\dfrac{\mathcal{B}}{\Delta, \hat{\sigma_2} \odot \Gamma, x' : A \vdash}\ \textsc{Wf\_Ext}}{(\Delta, \hat{\sigma_2} \mid \mathbf{0}, 1 \mid \hat{\sigma_2}, 0) \odot \Gamma, x' : A \vdash x' : A}\ \textsc{T\_Var}$$

$$\mathcal{E}\quad \frac{\mathcal{C} \quad \mathcal{B}}{(\Delta, \hat{\sigma_2}, (\hat{\sigma_2}, 0) \mid \hat{\sigma_2}', 0, r \mid \mathbf{0}) \odot \Gamma, x' : A, x : A \vdash B : \mathsf{Type}_{l''}}\ \text{L. 3.26}$$

$$\mathcal{F}\quad \frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : (x :_{(s,r)} A) \to B \quad \mathcal{B}}{(\Delta, \hat{\sigma_2} \mid \sigma_1, 0 \mid \sigma_2, 0) \odot \Gamma, x' : A \vdash t : (x :_{(s,r)} A) \to B}\ \text{L. 3.26}$$

$$\mathcal{G}\quad \frac{\dfrac{\dfrac{\mathcal{E} \quad \mathcal{F} \quad \mathcal{D}}{(\Delta, \hat{\sigma_2} \mid \sigma_1, s \mid \hat{\sigma_2}', r) \odot \Gamma, x' : A \vdash t\, x' : [x'/x]B}\ \textsc{T\_App} \qquad \mathcal{B}}{(\Delta, \hat{\sigma_2}, (\hat{\sigma_2}, 0) \mid \sigma_1, 0, s \mid \hat{\sigma_2}', 0, r) \odot \Gamma, x : A, x' : A \vdash t\, x' : [x'/x]B}\ \text{L. 3.26}}{(\Delta, \hat{\sigma_2} \mid \sigma_1, s \mid \hat{\sigma_2}', r) \odot \Gamma, x : A \vdash t\, x : B}\ \text{L. 3.22}$$

$$\frac{\mathcal{C} \quad \mathcal{G}}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \lambda x.(t\, x) : (x :_{(s,r)} A) \to B}\ \textsc{T\_Fun}$$

Case.

$$\dfrac{(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t_1 = t_2 : B}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t_1 = \lambda x.t_2 : (x :_{(s,r)} A) \to B}\;\text{TEQ\_Fun}$$

Then our goals hold by the following derivations:

$$\mathcal{A}$$

$$\dfrac{(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t_1 = t_2 : B}{(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l}\;\text{L. 3.6}$$

$$\dfrac{\mathcal{A} \qquad \dfrac{(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t_1 = t_2 : B}{(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t_1 : B}\;\text{I.H.}}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t_1 : (x :_{(s,r)} A) \to B}\;\text{T\_Fun}$$

$$\dfrac{\mathcal{A} \qquad \dfrac{(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t_1 = t_2 : B}{(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t_2 : B}\;\text{I.H.}}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t_2 : (x :_{(s,r)} A) \to B}\;\text{T\_Fun}$$

Case.

$$\dfrac{\begin{array}{l}(\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_1 \mid \sigma_3 + \sigma_4) \odot \Gamma \vdash t_1 = t_2 : (x :_{(s,r)} A) \to B \quad (\Delta \mid \sigma_2 \mid \sigma_3) \odot \Gamma \vdash t_3 = t_4 : A\end{array}}{(\Delta \mid \sigma_1 + s * \sigma_2 \mid \sigma_4 + r * \sigma_2) \odot \Gamma \vdash t_1\, t_3 = t_2\, t_4 : [t_3/x]B}\;\text{TEQ\_App}$$

Then our goals hold by the following derivations:

$$\dfrac{\begin{array}{c}(\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ \dfrac{(\Delta \mid \sigma_1 \mid \sigma_3 + \sigma_4) \odot \Gamma \vdash t_1 = t_2 : (x :_{(s,r)} A) \to B}{(\Delta \mid \sigma_1 \mid \sigma_3 + \sigma_4) \odot \Gamma \vdash t_1 : (x :_{(s,r)} A) \to B}\;\text{I.H.} \quad \dfrac{(\Delta \mid \sigma_2 \mid \sigma_3) \odot \Gamma \vdash t_3 = t_4 : A}{(\Delta \mid \sigma_2 \mid \sigma_3) \odot \Gamma \vdash t_3 : A}\;\text{I.H.}\end{array}}{(\Delta \mid \sigma_1 + s * \sigma_2 \mid \sigma_4 + r * \sigma_2) \odot \Gamma \vdash t_1\, t_3 : [t_3/x]B}\;\text{T\_App}$$

$$\mathcal{C}$$

$$\dfrac{\begin{array}{c}(\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ \dfrac{(\Delta \mid \sigma_1 \mid \sigma_3 + \sigma_4) \odot \Gamma \vdash t_1 = t_2 : (x :_{(s,r)} A) \to B}{(\Delta \mid \sigma_1 \mid \sigma_3 + \sigma_4) \odot \Gamma \vdash t_2 : (x :_{(s,r)} A) \to B}\;\text{I.H.} \quad \dfrac{(\Delta \mid \sigma_2 \mid \sigma_3) \odot \Gamma \vdash t_3 = t_4 : A}{(\Delta \mid \sigma_2 \mid \sigma_3) \odot \Gamma \vdash t_4 : A}\;\text{I.H.}\end{array}}{(\Delta \mid \sigma_1 + s * \sigma_2 \mid \sigma_4 + r * \sigma_2) \odot \Gamma \vdash t_2\, t_4 : [t_4/x]B}\;\text{T\_App}$$

$$\dfrac{\mathcal{C} \qquad \dfrac{\dfrac{\dfrac{(\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \quad (\Delta \mid \sigma_2 \mid \sigma_3) \odot \Gamma \vdash t_3 = t_4 : A}{(\Delta \mid \sigma_4 + r * \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash [t_3/x]B = [t_4/x]B : \mathsf{Type}_l}\;\text{L. 3.28}}{(\Delta \mid \sigma_4 + r * \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash [t_4/x]B = [t_3/x]B : \mathsf{Type}_l}\;\text{TEQ\_Sym}}{(\Delta \mid \sigma_4 + r * \sigma_2) \odot \Gamma \vdash [t_4/x]B \leq [t_3/x]B}\;\text{ST\_Eq}}{(\Delta \mid \sigma_1 + s * \sigma_2 \mid \sigma_4 + r * \sigma_2) \odot \Gamma \vdash t_2\, t_4 : [t_3/x]B}\;\text{T\_TyConv}$$

The cases for TEQ_TenCut and TEQ_BoxE proceed similarly.

Case.

$$\dfrac{\begin{array}{c}(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma \vdash t_1 = t_1' : A \quad (\Delta \mid \sigma_4 \mid \sigma_3 + r * \sigma_2) \odot \Gamma \vdash t_2 = t_2' : [t_1/x]B\end{array}}{(\Delta \mid \sigma_2 + \sigma_4 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash (t_1, t_2) = (t_1', t_2') : (x :_r A) \otimes B} \text{TEQ\_PAIR}$$

Then our goals hold by the following derivations:

$$\dfrac{\dfrac{(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma \vdash t_1 = t_1' : A}{(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma \vdash t_1 : A} \text{I.H.} \quad \begin{array}{c}(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ \dfrac{(\Delta \mid \sigma_4 \mid \sigma_3 + r * \sigma_2) \odot \Gamma \vdash t_2 = t_2' : [t_1/x]B}{(\Delta \mid \sigma_4 \mid \sigma_3 + r * \sigma_2) \odot \Gamma \vdash t_2 : [t_1/x]B} \text{I.H.}\end{array}}{(\Delta \mid \sigma_2 + \sigma_4 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash (t_1, t_2) : (x :_r A) \otimes B} \text{T\_PAIR}$$

$$\begin{array}{l}\mathcal{C} \\ \dfrac{\dfrac{(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \quad (\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma \vdash t_1 = t_1' : A}{(\Delta \mid \sigma_3 + r * \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash [t_1/x]B = [t_1'/x]B : \mathsf{Type}_l} \text{L. 3.28}}{(\Delta \mid \sigma_3 + r * \sigma_2) \odot \Gamma \vdash [t_1/x]B \leq [t_1'/x]B} \text{ST\_EQ}\end{array}$$

$$\begin{array}{l}\mathcal{D} \\ \dfrac{\dfrac{(\Delta \mid \sigma_4 \mid \sigma_3 + r * \sigma_2) \odot \Gamma \vdash t_2 = t_2' : [t_1/x]B}{(\Delta \mid \sigma_4 \mid \sigma_3 + r * \sigma_2) \odot \Gamma \vdash t_2' : [t_1/x]B} \text{I.H.} \quad \mathcal{C}}{(\Delta \mid \sigma_4 \mid \sigma_3 + r * \sigma_2) \odot \Gamma \vdash t_2' : [t_1'/x]B} \text{T\_TYCONV}\end{array}$$

$$\dfrac{\dfrac{(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma \vdash t_1 = t_1' : A}{(\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma \vdash t_1' : A} \text{I.H.} \quad (\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \quad \mathcal{D}}{(\Delta \mid \sigma_2 + \sigma_4 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash (t_1', t_2') : (x :_r A) \otimes B} \text{T\_PAIR}$$

Case.

$$\dfrac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : (x :_r A) \otimes B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t = (\mathsf{let}\,(x, y) = t\,\mathsf{in}\,(x, y)) : (x :_r A) \otimes B} \text{TEQ\_TENU}$$

Then our first goal holds by premise, and our second goal holds by the following derivation:

$$\begin{array}{l}\mathcal{A} \\ \dfrac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : (x :_r A) \otimes B}{(\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_r A) \otimes B : \mathsf{Type}_l} \text{L. 3.6}\end{array}$$

By Lemma 3.9 with $\mathcal{A}$, we have:

- $(\mathcal{B})$ $(\Delta \mid \hat{\sigma_2} \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l'}$;
- $(\mathcal{C})$ $(\Delta, \hat{\sigma_2} \mid \hat{\sigma_2}', r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l''}$;
- $\hat{\sigma_2} + \hat{\sigma_2}' = \sigma_2$

$\mathcal{D}$
$$\dfrac{\dfrac{\mathcal{C} \quad \mathcal{B}}{(\Delta, \hat{\sigma_2}, (\hat{\sigma_2}, 0) \mid \hat{\sigma_2}', r, 0 \mid \mathbf{0}) \odot \Gamma, x' : A, x : A \vdash B : \mathsf{Type}_{l''}} \text{ L. 3.26}}{(\Delta, \hat{\sigma_2} \mid \hat{\sigma_2}', r \mid \mathbf{0}) \odot \Gamma, x' : A \vdash [x'/x]B : \mathsf{Type}_{l''}} \text{ L. 3.22}$$

$\mathcal{E}$
$$\dfrac{\dfrac{\mathcal{D}}{\Delta, \hat{\sigma_2}, (\hat{\sigma_2}', r) \odot \Gamma, x' : A, y : [x'/x]B \vdash} \text{ WF\_EXT}}{(\Delta, \hat{\sigma_2}, (\hat{\sigma_2}', r) \mid \mathbf{0}, 1, 0 \mid \hat{\sigma_2}, 0, 0) \odot \Gamma, x' : A, y : [x'/x]B \vdash x' : A} \text{ T\_VAR}$$

$\mathcal{F}$
$$\dfrac{\dfrac{\mathcal{C} \quad \mathcal{B}}{(\Delta, \hat{\sigma_2}, (\hat{\sigma_2}, 0) \mid \hat{\sigma_2}', 0, r \mid \mathbf{0}) \odot \Gamma, x' : A, x : A \vdash B : \mathsf{Type}_{l''}} \text{ L. 3.26} \quad \mathcal{D}}{(\Delta, \hat{\sigma_2}, (\hat{\sigma_2}, r), (\hat{\sigma_2}', 0, 0) \mid \hat{\sigma_2}', 0, 0, r \mid \mathbf{0}) \odot \Gamma, x' : A, y : [x'/x]B, x : A \vdash B : \mathsf{Type}_{l''}} \text{ L. 3.26}$$

$\mathcal{G}$
$$\dfrac{\mathcal{E} \quad \dfrac{\dfrac{\mathcal{F}}{\Delta, \hat{\sigma_2}, (\hat{\sigma_2}', r), (\hat{\sigma_2}, 0, 0) \odot \Gamma, x' : A, y : [x'/x]B, x : A \vdash} \text{ L. 3.3}}{(\Delta, \hat{\sigma_2}, (\hat{\sigma_2}', r), (\hat{\sigma_2}, 0, 0) \mid \mathbf{0}, 0, 1, 0 \mid \hat{\sigma_2}', r, 0, 0) \odot \Gamma, x' : A, y : [x'/x]B, x : A \vdash y : [x'/x]B} \text{ T\_VAR}}{(\Delta, \hat{\sigma_2}, (\hat{\sigma_2}', r) \mid \mathbf{0}, 0, 1 \mid \hat{\sigma_2}', r, 0) \odot \Gamma, x' : A, y : [x'/x]B \vdash y : [x'/x]B} \text{ L. 3.27}$$

$\mathcal{H}$
$$\dfrac{\dfrac{\dfrac{\mathcal{F} \quad \mathcal{E} \quad \mathcal{G}}{(\Delta, \hat{\sigma_2}, (\hat{\sigma_2}', r) \mid \mathbf{0}, 1, 1 \mid \sigma_2, 0, 0) \odot \Gamma, x' : A, y : [x'/x]B \vdash (x', y) : (x :_r A) \otimes B} \text{ T\_PAIR} \quad \mathcal{B}}{(\Delta, \hat{\sigma_2}, (\hat{\sigma_2}, 0), (\hat{\sigma_2}', 0, r) \mid \mathbf{0}, 0, 1, 1 \mid \sigma_2, 0, 0, 0) \odot \Gamma, x : A, x' : A, y : [x'/x]B \vdash (x', y) : (x :_r A) \otimes B} \text{ L. 3.26}}{(\Delta, \hat{\sigma_2}, (\hat{\sigma_2}', 0, r) \mid \mathbf{0}, 1, 1 \mid \sigma_2, 0, 0) \odot \Gamma, x : A, y : B \vdash (x, y) : (x :_r A) \otimes B} \text{ L. 3.22}$$

$$\dfrac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : (x :_r A) \otimes B \quad \dfrac{\mathcal{A} \quad \mathcal{A}}{(\Delta, \sigma_2 \mid \sigma_2, 0 \mid \mathbf{0}) \odot \Gamma, z : (x :_r A) \otimes B \vdash (x :_r A) \otimes B : \mathsf{Type}_l} \text{ L. 3.24} \quad \mathcal{H}}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \mathsf{let}\,(x, y) = t\,\mathsf{in}\,(x, y) : (x :_r A) \otimes B} \text{ T\_TENCUT}$$

Case.

$$\dfrac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A = B : \mathsf{Type}_l}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A = \Box_s B : \mathsf{Type}_l} \text{ TEQ\_BOX}$$

Then our goals hold by the following derivations:

$$\dfrac{\dfrac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A = B : \mathsf{Type}_l}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l} \text{ I.H.}}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A : \mathsf{Type}_l} \text{ T\_BOX}$$

$$\dfrac{\dfrac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A = B : \mathsf{Type}_l}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash B : \mathsf{Type}_l}\ \text{I.H.}}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s B : \mathsf{Type}_l}\ \text{T\_Box}$$

Case.

$$\dfrac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A}{(\Delta \mid s * \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box t_1 = \Box t_2 : \Box_s A}\ \text{TEQ\_BoxI}$$

Then our goals hold by the following derivations:

$$\dfrac{\dfrac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : A}\ \text{I.H.}}{(\Delta \mid s * \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box t_1 : \Box_s A}\ \text{T\_BoxI}$$

$$\dfrac{\dfrac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_2 : A}\ \text{I.H.}}{(\Delta \mid s * \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box t_2 : \Box_s A}\ \text{T\_BoxI}$$

Case.

$$\dfrac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : \Box_s A}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t = (\mathsf{let}\, \Box x = t \,\mathsf{in}\, \Box x) : \Box_s A}\ \text{TEQ\_BoxU}$$

Then our first goal holds by premise, and our second goal holds by the following derivation:

$$\mathcal{C}\quad \dfrac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : \Box_s A}{(\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A : \mathsf{Type}_l}\ \text{L. 3.6}$$

$$\mathcal{D}\quad \dfrac{\dfrac{\dfrac{\dfrac{\mathcal{C}}{(\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l'}}\ \text{L. 3.10}}{\Delta, \sigma_2 \odot \Gamma, x : A \vdash}\ \text{Wf\_Ext}}{(\Delta, \sigma_2 \mid \mathbf{0}, 1 \mid \sigma_2, 0) \odot \Gamma, x : A \vdash x : A}\ \text{T\_Var}}{(\Delta, \sigma_2 \mid \mathbf{0}, s \mid \sigma_2, 0) \odot \Gamma, x : A \vdash \Box x : \Box_s A}\ \text{T\_BoxI}$$

$$\dfrac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : \Box_s A \qquad \dfrac{\mathcal{C} \quad \mathcal{C}}{(\Delta, \sigma_2 \mid \sigma_2, 0 \mid \mathbf{0}) \odot \Gamma, z : \Box_s A \vdash \Box_s A : \mathsf{Type}_l}\ \text{L. 3.24} \qquad \mathcal{D}}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \mathsf{let}\, \Box x = t \,\mathsf{in}\, \Box x : \Box_s A}\ \text{T\_BoxE}$$

$\square$

Lemma 3.33 (Deriving judgments under equal contexts)**.** *If* $(\Delta_1, \sigma_1, \Delta_2 \mid \sigma_2, s, \sigma_3 \mid \sigma_4, r, \sigma_5) \odot \Gamma_1, x : A, \Gamma_2 \vdash \mathcal{J}$ *and* $(\Delta_1 \mid \sigma_1 \mid \boldsymbol{0}) \odot \Gamma \vdash A = A' : \mathsf{Type}_l$*, then* $(\Delta_1, \sigma_1, \Delta_2 \mid \sigma_2, s, \sigma_3 \mid \sigma_4, r, \sigma_5) \odot \Gamma_1, x : A', \Gamma_2 \vdash \mathcal{J}$*.*

Proof. For well-formed contexts:

Case.

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l}{\Delta, \sigma \odot \Gamma, x : A \vdash} \text{ Wf\_Ext}$$

With $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A = A' : \mathsf{Type}_l$. Then our goal holds by the following derivation:

$$\frac{\dfrac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A = A' : \mathsf{Type}_l}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A' : \mathsf{Type}_l} \text{ L. 3.32}}{\Delta, \sigma \odot \Gamma, x : A' \vdash} \text{ Wf\_Ext}$$

Case.

$$\frac{(\Delta_1, \sigma_1, \Delta_2 \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash B : \mathsf{Type}_{l'}}{\Delta_1, \sigma_1, \Delta_2, \sigma_2 \odot \Gamma_1, x : A, \Gamma_2, y : B \vdash} \text{ Wf\_Ext}$$

Then our goal holds by the following derivation:

$$\frac{\dfrac{(\Delta_1, \sigma_1, \Delta_2 \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash B : \mathsf{Type}_{l'} \quad (\Delta_1 \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma_1 \vdash A = A' : \mathsf{Type}_l}{(\Delta_1, \sigma_1, \Delta_2 \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma_1, x : A', \Gamma_2 \vdash B : \mathsf{Type}_{l'}} \text{ I.H.}}{\Delta_1, \sigma_1, \Delta_2, \sigma_2 \odot \Gamma_1, x : A', \Gamma_2, y : B \vdash} \text{ Wf\_Ext}$$

For typing, all cases proceed by induction then re-application to respective rules. For equality, all cases proceed by induction then re-application to respective rules. For subtyping, all cases proceed by induction then re-application to respective rules. $\square$

A.6 Proofs for subtyping

Lemma 3.34 (Subtyping inversion to typing)**.** *If* $(\Delta \mid \sigma) \odot \Gamma \vdash A \leq B$*, then* $(\Delta \mid \sigma \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_l$ *and* $(\Delta \mid \sigma \mid \boldsymbol{0}) \odot \Gamma \vdash B : \mathsf{Type}_l$ *for some level* l*.*

Proof. By induction on the form of $(\Delta \mid \sigma) \odot \Gamma \vdash A \leq B$, as follows:

Case.

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A = B : \mathsf{Type}_l}{(\Delta \mid \sigma) \odot \Gamma \vdash A \leq B} \text{ ST\_Eq}$$

By Lemma 3.32 we have $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l$ and $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash B : \mathsf{Type}_l$, as required.

Case.

$$\frac{(\Delta \mid \sigma) \odot \Gamma \vdash A \leq B \quad (\Delta \mid \sigma) \odot \Gamma \vdash B \leq C}{(\Delta \mid \sigma) \odot \Gamma \vdash A \leq C} \text{ ST\_TRANS}$$

Our goals hold by induction.

Case.

$$\frac{\Delta \odot \Gamma \vdash \quad l \leq l'}{(\Delta \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_l \leq \mathsf{Type}_{l'}} \text{ ST\_TY}$$

Our goals hold by the following derivations:

$$\frac{\Delta \odot \Gamma \vdash}{(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}} \text{ T\_TYPE}$$

$$\frac{\Delta \odot \Gamma \vdash}{(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_{l'} : \mathsf{Type}_{\mathsf{suc}\ l'}} \text{ T\_TYPE}$$

Case.

$$\frac{\begin{array}{l}(\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_1) \odot \Gamma \vdash A' \leq A \quad (\Delta, \sigma_1 \mid \sigma_2, r) \odot \Gamma, x : A' \vdash B \leq B'\end{array}}{(\Delta \mid \sigma_1 + \sigma_2) \odot \Gamma \vdash (x :_{(s,r)} A) \to B \leq (x :_{(s,r)} A') \to B'} \text{ ST\_ARROW}$$

Then we have our goals by the following derivations:

$$\mathcal{C}\ \frac{\dfrac{(\Delta \mid \sigma_1) \odot \Gamma \vdash A' \leq A}{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A' : \mathsf{Type}_{l'}} \text{ I.H.} \qquad \dfrac{(\Delta, \sigma_1 \mid \sigma_2, r) \odot \Gamma, x : A' \vdash B \leq B'}{(\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A' \vdash B' : \mathsf{Type}_{l''}} \text{ I.H.}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A') \to B' : \mathsf{Type}_{l' \sqcup l''}} \text{ T\_ARROW}$$

$$\mathcal{D}\ \frac{\dfrac{(\Delta \mid \sigma_1) \odot \Gamma \vdash A' \leq A}{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l'}} \text{ I.H.} \qquad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : \mathsf{Type}_{l' \sqcup l}} \text{ T\_ARROW}$$

$$\frac{\mathcal{C} \qquad \dfrac{\dfrac{(\Delta \mid \sigma_1) \odot \Gamma \vdash A' \leq A}{\Delta \odot \Gamma \vdash} \text{ L. 3.3} \qquad l' \sqcup l'' \leq l \sqcup l' \sqcup l''}{(\Delta \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_{l' \sqcup l''} \leq \mathsf{Type}_{l \sqcup l' \sqcup l''}} \text{ ST\_TY}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A') \to B' : \mathsf{Type}_{l \sqcup l' \sqcup l''}} \text{ T\_TYCONV}$$

$$
\dfrac{\mathcal{D} \quad \dfrac{\dfrac{(\Delta \mid \sigma_1) \odot \Gamma \vdash A' \leq A}{\Delta \odot \Gamma \vdash} \text{ L. 3.3} \quad l' \sqcup l \leq l \sqcup l' \sqcup l''}{(\Delta \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_{l' \sqcup l} \leq \mathsf{Type}_{l \sqcup l' \sqcup l''}} \text{ ST\_TY}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : \mathsf{Type}_{l \sqcup l' \sqcup l''}} \text{ T\_TYCONV}
$$

The case for ST_TEN proceeds similarly, using T_TEN.

Case.

$$
\dfrac{(\Delta, \sigma_1 \mid \sigma_2, r) \odot \Gamma, x : A \vdash B \leq B'}{(\Delta \mid \sigma_1 + \sigma_2) \odot \Gamma \vdash (x :_r A) \otimes B \leq (x :_r A) \otimes B'} \text{ ST\_TEN}
$$

Then we have our goals by the following derivations:

$$
\dfrac{\dfrac{(\Delta, \sigma_1 \mid \sigma_2, r) \odot \Gamma, x : A \vdash B \leq B'}{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l} \text{ L. 3.6} \quad \dfrac{(\Delta, \sigma_1 \mid \sigma_2, r) \odot \Gamma, x : A \vdash B \leq B'}{(\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l'}} \text{ I.H.}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_r A) \otimes B : \mathsf{Type}_{l \sqcup l'}} \text{ T\_TEN}
$$

$$
\dfrac{\dfrac{(\Delta, \sigma_1 \mid \sigma_2, r) \odot \Gamma, x : A \vdash B \leq B'}{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l} \text{ L. 3.6} \quad \dfrac{(\Delta, \sigma_1 \mid \sigma_2, r) \odot \Gamma, x : A \vdash B \leq B'}{(\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B' : \mathsf{Type}_{l'}} \text{ I.H.}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_r A) \otimes B : \mathsf{Type}_{l \sqcup l'}} \text{ T\_TEN}
$$

Case.

$$
\dfrac{(\Delta \mid \sigma) \odot \Gamma \vdash A \leq A'}{(\Delta \mid \sigma) \odot \Gamma \vdash \Box_s A \leq \Box_s A'} \text{ ST\_BOX}
$$

Then we have our goals by the following derivations:

$$
\dfrac{\dfrac{(\Delta \mid \sigma) \odot \Gamma \vdash A' \leq A}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A' : \mathsf{Type}_l} \text{ I.H.}}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A' : \mathsf{Type}_l} \text{ T\_BOX}
$$

$$
\dfrac{\dfrac{(\Delta \mid \sigma) \odot \Gamma \vdash A' \leq A}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l'}} \text{ I.H.}}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A : \mathsf{Type}_l} \text{ T\_BOX}
$$

$\square$

A.7 Proof for type preservation

Lemma 3.35 (Type preservation)**.** *If* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$ *and* $t \rightsquigarrow t'$*, then* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t' : A$*.*

Proof. By Lemma 3.31 we have $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t = t' : A$, and therefore by Lemma 3.32 we have $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t' : A$, as required. □

A.8 Standard results

For inductive results on multi-judgment lemmas, refer to the following lists for how to obtain goals:
For subtyping:

- ST_Eq This holds by induction then re-application to the rule.
- ST_Trans This holds by induction then re-application to the rule.
- ST_Ty This holds by induction then re-application to the rule.
- ST_Arrow This holds by induction then re-application to the rule (see the T_Arrow case for how to handle the typing premise for $[[B]]$).
- ST_Ten This holds by induction then re-application to the rule (see the T_Fun case for how to handle the extended context).
- ST_Box This holds by induction then re-application to the rule.

For equality:

- TEQ_Refl This holds by induction then re-application to the rule.
- TEQ_Trans Holds similarly to the case for TEQ_Refl.
- TEQ_Sym Holds similarly to the case for TEQ_Refl.
- TEQ_ConvTy Holds similarly to the case for TEQ_Refl.
- TEQ_Arrow Holds similarly to the case for T_Arrow.
- TEQ_ArrowComp Holds similarly to the case for T_App (induction then re-application).
- TEQ_ArrowUniq Holds by induction then re-application.
- TEQ_Fun Holds similarly to the case for T_Fun.
- TEQ_App Holds similarly to the case for T_App.
- TEQ_Ten Holds similarly to the case for T_Ten.
- TEQ_TenComp Holds by induction then re-application.
- TEQ_Pair Holds similarly to the case for T_Pair.
- TEQ_TenCut Holds similarly to the case for T_TenCut.
- TEQ_TenU Holds by induction then re-application.
- TEQ_Box Holds similarly to the case for T_Box.
- TEQ_BoxI Holds similarly to the case for T_BoxI.
- TEQ_BoxB Holds by induction then re-application.
- TEQ_BoxE Holds similarly to the case for T_BoxE.
- TEQ_BoxU Holds by induction then re-application.

B Encoding

B.1 Simply-typed Lambda Calculus

A subset of GrTT encodes STLC We define the following subset of GRTT typing judgments by the predicate $\textsc{Stlc}(\vdash)$:

$$\textsc{Stlc}((\emptyset \mid \emptyset \mid \emptyset) \odot \emptyset \vdash t : A)$$
$$\textsc{Stlc}((\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A) \implies \textsc{Stlc}((\Delta, \mathbf{0} \mid \sigma_1, r \mid \sigma_2, 0) \odot \Gamma, x : B \vdash t : A)$$

$$\textsc{Stlc}(\emptyset \odot \emptyset \vdash)$$
$$\textsc{Stlc}(\Delta \odot \Gamma \vdash) \implies \textsc{Stlc}(\Delta, \mathbf{0} \odot \Gamma, x : A \vdash)$$

Then we define an inductive encoding $[\![-]\!]$ from of a subset of GRTT syntax (ignoring Type and tensors) to STLC. We define a partial inductive encoding on terms $[\![-]\!]_t$ and contexts $[\![-]\!]_\Gamma$:

$$[\![x]\!]_t = x$$
$$[\![\lambda x.t]\!]_t = \lambda x.[\![t]\!]_t$$
$$[\![t_1\, t_2]\!]_t = [\![t_1]\!]_t\ [\![t_2]\!]_t$$

$$[\![(x :_{(r,0)} A) \to B]\!]_\tau = [\![A]\!]_\tau \to [\![B]\!]_\tau$$

$$[\![\emptyset]\!] = \cdot$$
$$[\![\Gamma, x : A]\!] = [\![\Gamma]\!], x : [\![A]\!]_\tau$$

B.1.1 Key lemmas on quantiative use

Some key lemmas for soundness

Lemma B.1. *Given a quantitative semiring, if* $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : C$*, with* $(\Delta \mid \mathbf{0}) \odot \Gamma \vdash C \leq \mathsf{Type}_l$ *for some level* l*, and* $[\![A]\!]_\tau$ *is defined, then:*

$$\mathsf{fv}(A) = \{y \mid \Gamma[i] = y : B \wedge \sigma[i] \neq 0\}$$

Proof. By induction on the form of $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l$, as follows:

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l_1} \qquad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : \mathsf{Type}_{l_1 \sqcup l_2}}\ \textsc{T\_Arrow}$$

By induction we have:

$$\mathsf{fv}(A) = \{y \mid \Gamma[i] = y : B \wedge \sigma_1[i] \neq 0\} \quad \text{(ih1)}$$
$$\mathsf{fv}(B) = \{y \mid \Gamma[i] = y : B \wedge \sigma_2, r[i] \neq 0\} \quad \text{(ih2)}$$

The goal is then that:

$$\begin{aligned}\mathsf{fv}((x :_{(s,r)} A) \to B : \mathsf{Type}_{l_1 \sqcup l_2}) &= \mathsf{fv}(A) \cup (\mathsf{fv}(B) \setminus \{x\}) \\ &= \mathsf{fv}((x :_{(s,r)} A) \to B : \mathsf{Type}_{l_1 \sqcup l_2}) = \{y \mid \Gamma[i] = y : B \wedge \sigma_1 + \sigma_2[i] \neq 0\}\end{aligned}$$

We get that $\mathsf{fv}(B) \setminus \{x\} = \{y \mid \Gamma[i] = y : B \wedge \sigma_2[i] \neq 0\} = (\text{ih1}) \setminus \{x\}$.

Next we then need to prove by $((\sigma_2[i] \neq 0) \vee (\sigma_3[i] \neq 0)) \Leftrightarrow (\sigma_2 + \sigma_3)[i] \neq 0$:

- Left-to-right: Assuming $(\sigma_2[i] \neq 0 \vee (\sigma_3)[i] \neq 0)$
  By contradiction: Assume $(\sigma_2 + \sigma_3)[i] = 0$ then by positivity $\sigma_2[i] = 0$ and $\sigma_3[i] = 0$.
  Thus we can eliminate the assumption with these to get a contradiction.
- Right-to-left: Assuming $(\sigma_2 + \sigma_3)[i] \neq 0$
  The goal is $(\sigma_2[i] \neq 0 \vee (\sigma_3)[i] \neq 0)$ which is equivalent to $\neg(\sigma_2[i] = 0 \wedge \sigma_3[i] = 0)$ thus we go by contradiction:
  Assuming $(\sigma_2[i] = 0 \wedge \sigma_3[i] = 0)$ Therefore by monoid unit we have $(\sigma_2 + \sigma_3)[i] = 0$ which contradicts the assumption hence we by PBC that $(\sigma_2[i] \neq 0 \vee (\sigma_3)[i] \neq 0)$.

And we are done.

Case.

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : B \qquad (\Delta \mid \mathbf{0}) \odot \Gamma \vdash B \leq C}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : C}\ \textsc{T\_TyConv}$$

By ST_Trans we have $(\Delta \mid \mathbf{0}) \odot \Gamma \vdash B \leq \mathsf{Type}_l$, and therefore our goal holds by induction.

∎

Lemma B.2. *For a quantitative semiring, under the interpretation, if $\Delta \odot \Gamma \vdash$ then*

$$\forall i. 0 \leq i < |\Gamma| \;.\; \mathit{fv}(\Gamma[i]) = \{y \mid \forall j. \Gamma[i] = x : A \wedge j < i \wedge \Gamma[j] = y : B \wedge \Delta[i][j] \neq 0)\}$$

Proof.

- Case Wf_Empty

$$\frac{}{\emptyset \odot \emptyset \vdash}\ \textsc{Wf\_Empty}$$

Trivial since $\mathsf{fv}(\emptyset = \emptyset)$

- Case Wf_Ext

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l}{\Delta, \sigma \odot \Gamma, x : A \vdash}\ \textsc{Wf\_Ext}$$

By Lemma B.1 on $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l$ then we have that:

$$\mathsf{fv}(\Gamma[i]) = \{y \mid \forall j. \Gamma[i] = x : A \wedge j < i \wedge \Gamma[j] = y : B \wedge \Delta[i][j] \neq 0)\} \tag{1}$$
$$\mathsf{fv}(A) = \{x \mid \Gamma[i] = x : A' \wedge \sigma[i] \neq 0\} \tag{2}$$

Therefore we can use $\mathsf{fv}(\Gamma[i])$ to prove the goal for all i where $0 \leq i < |\Gamma|$ then for $i' = |\Gamma|$ we need to prove that:

$$\mathsf{fv}(\Gamma[i']) = \{y \mid \forall j. (\Gamma, x : A)[i'] = x : A \wedge j < i' \wedge \Gamma[j] = y : B \wedge \Delta, \sigma[i'][j] \neq 0)\} \tag{goal}$$

which holds since $(\Gamma, x : A)[i'] = x : A$ is true and $\Delta, \sigma[i'][j] = \sigma[j]$ then our goal reduces to:

$$\mathsf{fv}(\Gamma[i']) = \{y \mid \forall j. j < i' \wedge \Gamma[j] = y : B \wedge \sigma[j] \neq 0)\}$$

which follows from $\mathsf{fv}(A)$ (by alpha equivalence).

∎

B.1.2 Type soundness

Given a derivation of $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$ such that $\textsc{Stlc}((\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A)$ then $[\![\Gamma]\!] \vdash [\![t]\!] : [\![A]\!]$ in STLC.

Proof By induction of the typing relation.

- Case T_Type. Trivial since it is rejected by $\textsc{Stlc}(-)$
- Case T_Var, thus under $\textsc{Stlc}(-)$ we have:

$$\frac{(\Delta_1, \mathbf{0}, \Delta_2) \odot \Gamma_1, x : A, \Gamma_2 \vdash}{(\Delta_1, \mathbf{0}, \Delta_2 \mid \mathbf{0}, 1, \mathbf{0} \mid \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash x : A} \text{ T\_Var}$$

  We can thus form the derivation in STLC as follows since we know that A is closed by the typing (Corollary B.8.1), i.e., there does not occur any variables inside of A by the subject grade $\mathbf{0}$.

$$\frac{}{[\![\Gamma_1]\!], x : [\![A]\!], [\![\Gamma_2]\!] \vdash x : [\![A]\!]}$$

- Case T_Arrow, T_Ten are trivial since they do not satisfy the predicate
- Case T_Fun

$$\frac{(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \quad (\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t : B}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t : (x :_{(s,r)} A) \to B} \text{ T\_Fun}$$

  By induction on $[\![(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t : B]\!]$ we have:

$$[\![\Gamma]\!], x : [\![A]\!] \vdash [\![t]\!] : [\![B]\!]$$

  Thus we can form the STLC derivation:

$$\frac{[\![\Gamma]\!], x : [\![A]\!] \vdash [\![t]\!] : [\![B]\!]}{[\![\Gamma]\!] \vdash \lambda x.[\![t]\!] : [\![A]\!] \to [\![B]\!]}$$

- Case T_App

$$\frac{(\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash t_1 : (x :_{(s,0)} A) \to B \qquad (\Delta \mid \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash t_2 : A}{(\Delta \mid \sigma_2 + s * \mathbf{0} \mid \mathbf{0} + 0 * \sigma_4) \odot \Gamma \vdash t_1\, t_2 : [t_2/x]B} \text{ T\_App}$$

  By Corollary B.8.1, then B is closed and thus the conclusion here is equal to:

$$(\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash t_1\, t_2 : B$$

  Trivially by induction we get the STLC judgments:

$$[\![\Gamma]\!] \vdash [\![t_1]\!] : [\![A]\!] \to [\![B]\!] \quad \text{(ih1)}$$
$$[\![\Gamma]\!] \vdash [\![t_2]\!] : [\![A]\!] \quad \text{(ih2)}$$

  Thus we can form the typing: $[\![\Gamma]\!] \vdash [\![t_1]\!]\ [\![t_2]\!] : [\![B]\!]$.

- Rest of the rules are trivial since they are not part of the interpretation.

B.1.3 Soundness

Lemma B.3. $[\![[t_2/x]t_1]\!] = [[\![t_2]\!]/x][\![t_1]\!]$.

Proof. Straightforward by induction on the definition of syntactic substitution for GRTT. □

Lemma B.4 (Soundness)**.** *Given* STLC$((\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A)$ *then* $[\![(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A]\!]$ *implies* $[\![\Gamma]\!] \vdash [\![t]\!] : [\![A]\!]$ *in* GRTT *and whenever* $t \rightsquigarrow t'$ *then* $[\![t]\!] \rightsquigarrow [\![t']\!]$ *in the CBN STLC (+ application right congruence reduction).*

Proof. First we observe that by Type Preservation (Lemma 3.35) that reduction preserves typing thus if STLC$(-)$ holds for $[\![(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A]\!]$ then STLC$((\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t' : A)$. Then we show that reduction in GRTT is simulated in STLC by induction on the typing:

- Case T_TYPE trivial since it is rejected by STLC$(-)$.
- Case T_VAR does not have a reduction (normal form);
- Case T_ARROW, T_TEN are trivial since they do not satisfy the predicate
- Case T_FUN does not have a reduction (normal form);
- Case T_APP with

$$\frac{(\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash t_1 : (x :_{(s,0)} A) \to B \qquad (\Delta \mid \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash t_2 : A}{(\Delta \mid \sigma_2 + s * \mathbf{0} \mid \mathbf{0} + 0 * \sigma_4) \odot \Gamma \vdash t_1\, t_2 : [t_2/x]B}\ \text{T\_APP}$$

  which translates to the derivation:

$$\frac{[\![\Gamma]\!] \vdash [\![t_1]\!] : [\![A]\!] \to [\![B]\!] \qquad [\![\Gamma]\!] \vdash [\![t_2]\!] : [\![A]\!]}{[\![\Gamma]\!] \vdash [\![t_1]\!]\ [\![t_2]\!] : [\![B]\!]}$$

  There are then two possible reductions:

  - $(\lambda x.t_1)\, t_2 \rightsquigarrow [t_2/x]t_1$ Goal is that $[\![(\lambda x.t_1)\, t_2]\!] \to [\![[t_2/x]t_1]\!]$.
    By the translation the left-hand side is equivalent to:

$$(\lambda x.[\![t_1]\!])\ [\![t_2]\!]$$

    By β-reduction in STLC we get

$$(\lambda x.[\![t_1]\!])\ [\![t_2]\!] \rightsquigarrow [[\![t_2]\!]/x][\![t_1]\!]$$

    which via Lemma B.3 provides the goal
  - The other case is the congruence:

$$\frac{t_1 \rightsquigarrow t_1'}{t_1\, t_2 \rightsquigarrow t_1'\, t_2}\ \text{SEM\_CONGFUNONE}$$

    By induction we have $[\![t_1]\!] \rightsquigarrow [\![t_1']\!]$ which using application congruence for STLC yields the reduction in STLC:

$$\frac{[\![t_1]\!] \rightsquigarrow [\![t_1']\!]}{[\![t_1]\!][\![t_2]\!] \rightsquigarrow [\![t_1']\!][\![t_2]\!]}$$

    which satisfies the goal since $[\![t_1\, t_2]\!] = [\![t_1]\!][\![t_2]\!]$.
- Case T_PAIR and case T_TENCUT are trivial since we exclude tensor types.

- Case T_TyConv trivial by induction because it introduced no additional syntax.
- Case T_Box trivial as we don't translate type formation rules
- Case T_BoxI trivial since it has no reduction
- Case T_BoxE

$$\frac{\begin{array}{c}(\Delta \mid \sigma_1 \mid \sigma_5) \odot \Gamma \vdash t_1 : \Box_s A \\ (\Delta, \sigma_5 \mid \sigma_4, 0 \mid \mathbf{0}) \odot \Gamma, z : \Box_s A \vdash B : \mathsf{Type}_l \qquad (\Delta, \sigma_5 \mid \sigma_3, s \mid \sigma_4, 0) \odot \Gamma, x : A \vdash t_2 : B\end{array}}{(\Delta \mid \sigma_1 + \sigma_3 \mid \sigma_4) \odot \Gamma \vdash \mathsf{let}\,\Box x = t_1\,\mathsf{in}\,t_2 : B}\ \textrm{T\_BoxE}$$

There are two possible case for reduction:

  –

$$\frac{}{\mathsf{let}\,\Box x = \Box t_1\,\mathsf{in}\,t_2 \rightsquigarrow [t_1/x]t_2}\ \textrm{Sem\_BetaBox}$$

  Thus the reduction here yields the translated term $[\![[t_1/x]t_2]\!]$.
  The translation gives that $[\![\mathsf{let}\,\Box x = \Box t_1\,\mathsf{in}\,t_2]\!] = (\lambda x.[\![t_2]\!])\ [\![t_1]\!]$. Which in the STLC then reduces by β-reduction to $[[\![t_1]\!]/x][\![t_2]\!]$ which matches the translated term by Lemma B.3.

  –

$$\frac{t_1 \rightsquigarrow t_1'}{\mathsf{let}\,\Box x = t_1\,\mathsf{in}\,t_2 \rightsquigarrow \mathsf{let}\,\Box x = t_1'\,\mathsf{in}\,t_2}\ \textrm{Sem\_CongBox1}$$

  The translation gives that $[\![\mathsf{let}\,\Box x = t_1\,\mathsf{in}\,t_2,]\!] = (\lambda x.[\![t_2]\!])\ [\![t_1]\!]$. Thus the reduction here yields the encoded term $(\lambda x.[\![t_2]\!])\ [\![t_1']\!]$.
  By induction we have $[\![t_1]\!] \rightsquigarrow [\![t_1']\!]$ from which we can then apply the STLC reduction (right congruence).

$$\frac{[\![t_1]\!] \rightsquigarrow [\![t_1']\!]}{(\lambda x.[\![t_2]\!])\ [\![t_1]\!] \rightsquigarrow (\lambda x.[\![t_2]\!])\ [\![t_1']\!]}$$

□

B.1.4 Completeness

Lemma B.5 (Completeness)**.** *Given* Stlc$((\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A)$ *and* $[\![(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A]\!]$ *and* $[\![\Gamma]\!] \vdash [\![t]\!] : [\![A]\!]$ *and* $[\![t]\!] \rightsquigarrow t_a$ *in the full beta STLC then* $t \rightsquigarrow t'$ *and* $[\![t']\!] \equiv_{\beta\eta} t_a$ *in* Grtt*.*

Proof. By induction on typing.

- Case T_Type trivial since it is rejected by Stlc$(-)$.
- Case T_Var does not have a reduction (normal form);
- Case T_Arrow, T_Ten are trivial since they do not satisfy the predicate
- Case T_Fun with

$$\frac{(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \quad (\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t : B}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t : (x :_{(s,r)} A) \to B}\ \textrm{T\_Fun}$$

which translates to the derivation:

$$\frac{[\![\Gamma]\!] \vdash [\![t]\!] : [\![B]\!]}{[\![\Gamma]\!] \vdash \lambda x.[\![t]\!] : [\![A]\!] \to [\![B]\!]}$$

We then reduce under the lambda to get:

$$\frac{[\![t]\!] \rightsquigarrow t_a}{\lambda x.[\![t]\!] \rightsquigarrow \lambda x.t_a}$$

By induction on the typing with the premise we then habe the GRTT reduction $t \rightsquigarrow t'$ where $[\![t']\!] = t_a$. Thue we can construct the GRTT reduction:

$$\frac{t \rightsquigarrow t'}{\lambda x.t \rightsquigarrow \lambda x.t'}$$

for which the interepretation $[\![\lambda x.t']\!] = \lambda x.t'_a$.

- Case T_APP with

$$\frac{(\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash t_1 : (x :_{(s,0)} A) \to B \qquad (\Delta \mid \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash t_2 : A}{(\Delta \mid \sigma_2 + s * \mathbf{0} \mid \mathbf{0} + 0 * \sigma_4) \odot \Gamma \vdash t_1\, t_2 : [t_2/x]B} \text{ T\_APP}$$

which translates to the derivation:

$$\frac{[\![\Gamma]\!] \vdash [\![t_1]\!] : [\![A]\!] \to [\![B]\!] \qquad [\![\Gamma]\!] \vdash [\![t_2]\!] : [\![A]\!]}{[\![\Gamma]\!] \vdash [\![t_1]\!]\ [\![t_2]\!] : [\![B]\!]}$$

Then depending on t_1 there are three possible reductions:

  – (application left congruence)

$$\frac{[\![t_1]\!] \rightsquigarrow t_b}{[\![t_1]\!]\ [\![t_2]\!] \rightsquigarrow t_b\ [\![t_2]\!]}$$

  where $t_a = t_b\ [\![t_2]\!]$
  By induction on typing with the premise here then we have that $t_1 \rightsquigarrow t'_1$ and $[\![t'_1]\!] \equiv t_b$. Thus we can construct the GRTT reduction (congruence on left):

$$\frac{t_1 \rightsquigarrow t'_1}{t_1\, t_2 \rightsquigarrow t'_1\, t_2} \text{ SEM\_CONGFUNONE}$$

  For which the interpretation $[\![t'_1\, t_2]\!] = [\![t'_1]\!]\ [\![t_2]\!]$
  By congruence on ($\beta\eta$-equality for STLC) with $[\![t'_1]\!] \equiv t_b$ then $[\![t'_1]\!]\ [\![t_2]\!] \equiv t_b\ [\![t_2]\!]$ thus matching our goal

  – (application right congruence)

$$\frac{[\![t_2]\!] \rightsquigarrow t_b}{[\![t_1]\!]\ [\![t_2]\!] \rightsquigarrow [\![t_1]\!]\ t_b}$$

  where $t_a = [\![t_1]\!]\ t_b$.
  By induction on typing with the premise here then we have that $t_2 \rightsquigarrow t'_2$ and $[\![t'_2]\!] \equiv t_b$. Thus we can construct the GRTT reduction (congruence on right).

  – (beta) The only way for the translation to yield a λ abstraction on the left is by translation a λ, thus we must have $t_1 = \lambda x.t'_1$ and thus:

$$\frac{}{[\![\lambda x.t'_1]\!]\ [\![t_2]\!] \rightsquigarrow [[\![t_2]\!]/x][\![t'_1]\!]}$$

  Therefore in GRTT we have $(\lambda x.t'_1)t_2$ which reduces by β to $[t_2/x](t'_1)$ whose interpretation $[\![[t_2/x](t'_1)]\!] = [[\![t_2]\!]/x][\![t'_1]\!]$ by Lemma B.3 matching the goal here.

$\square$

B.2 Stratified System F

We define the following subset of GRTT typing judgments by the predicate $\textsc{Ssf}(\vdash)$:

$$\begin{aligned}
&\textsc{Ssf}((\emptyset \mid \emptyset \mid \emptyset) \odot \emptyset \vdash t : A) \\
&\textsc{Ssf}((\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A) \implies \textsc{Ssf}((\Delta, \mathbf{0} \mid \sigma_1, 0 \mid \sigma_2, r) \odot \Gamma, x : \mathsf{Type}_l \vdash t : A) \\
&\textsc{Ssf}((\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A) \implies \mathsf{Type}_l \not\in^{+ve} B \wedge \textsc{Ssf}((\Delta, \sigma_3 \mid \sigma_1, s \mid \sigma_2, 0) \odot \Gamma, x : B \vdash t : A)
\end{aligned}$$

By $\mathsf{Type}_l \not\in^{+ve} B$ we mean that Type_l is not a positive subterm of B, so that we avoid higher-order typing terms (like type constructors) which do not exist in SSF.

We give a type directed encoding mapping from typing derivations of GRTT to SSF. Thus given a GRTT derivation of judgment $[\![(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A]\!]$ we have that $\exists t'$ (an SSF term) such that there is a derivation of judgment $[\![\Gamma]\!] \vdash t' : [\![A]\!]$ in SSF where we interpret A as:

$$\begin{aligned}
[\![x]\!]_\tau &= x \\
[\![\mathsf{Type}_l]\!]_\tau &= \star_l \\
[\![(x :_{(0,r)} \mathsf{Type}_l) \to B_\tau]\!] &= \forall x : \star_l . [\![B]\!]_\tau \\
[\![(x :_{(s,0)} A) \to B]\!]_\tau &= [\![A]\!]_\tau \to [\![B]\!]_\tau \qquad (\textit{where } \mathsf{Type}_l \not\in^{+ve} A)
\end{aligned}$$

A few key lemmas are needed to prove this is sound and complete. Which we develop first.

B.2.1 Key lemmas for soundness of encoding

Lemma B.6. *Given a quantitative semiring, if* $(\Delta \mid \sigma \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_l$ *and* $[\![A]\!]$ *is defined, then:*

$$\mathit{fv}(A) = \{y \mid \Gamma[i] = y : B \wedge \sigma[i] \neq 0\}$$

Proof. Given the restriction of $[\![-]\!]$, the only two possibilities are:

- Case T_VAR

$$\frac{\Delta_1, \sigma, \Delta_2 \odot \Gamma_1, x : A, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \sigma, \Delta_2 \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \sigma, 0, \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash x : A}\ \textsc{T\_Var}$$

  (where $A = \mathsf{Type}_l$)

  Goal is to prove $\mathsf{fv}(x) = \{y \mid \Gamma[i] = y : B \wedge \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0}[i] \neq 0\} = \{x\}$.

  Since x is at position $j = |M1|$ then then $\mathbf{0}^{|\Delta_1|}, 1, \mathbf{0}[j] \neq 0$ is true and otherwise $\mathbf{0}^{|\Delta_1|}, 1, \mathbf{0}[i] = 0$ for all other values therefore,

$$\mathsf{fv}(x) = \{x\}$$

  matching the lemma goal.

- Case T_TYPE

$$\frac{\Delta \odot \Gamma \vdash}{(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}}\ \textsc{T\_Type}$$

  The goal follows since $\neg(\sigma[i] \neq 0)$ for all i thus:

$$\mathsf{fv}(\mathsf{Type}_l) = \emptyset = \{x \mid \Gamma[i] = y : B \wedge \sigma[i] \neq 0\}$$

- Case T_ARROW

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : \mathsf{Type}_{l_1 \sqcup l_2}}\ \textsc{T\_Arrow}$$

By induction on the interpretation, we then have $[\![A]\!]$ and $[\![B]\!]$ must be defined. By induction of this lemma on the premises we get:

$$\mathsf{fv}(A) = \{x \mid \Gamma[i] = y : B \wedge \sigma_1[i] \neq 0\}$$
$$\mathsf{fv}(B) = \{x \mid \Gamma, x : A[i] = y : B \wedge \sigma_2, r[i] \neq 0\}$$

Then the goal is:

$$\mathsf{fv}((x :_{(s,r)} A) \to B) = \mathsf{fv}(A) \cup (\mathsf{fv}(B) \setminus \{x\}) = \{x \mid \Gamma[i] = y : B \wedge \sigma_1 + \sigma_2[i] \neq 0\}$$

We get $\mathsf{fv}(B) \setminus \{x\}) = \{x \mid \Gamma[i] = y : B \wedge \sigma_2[i] \neq 0\}$.

Then we can get the above goal if: $(\sigma_1[i] \neq 0 \vee \sigma_2[i] \neq 0) \Leftrightarrow \sigma_1 + \sigma_2[i] \neq 0$.

–
- Left-to-right: Assuming $(\sigma_1[i] \neq 0 \vee (\sigma_2)[i] \neq 0)$
  By contradiction: Assume $(\sigma_1 + \sigma_2)[i] = 0$ then by positivity $\sigma_1[i] = 0$ and $\sigma_2[i] = 0$.
  Thus we can eliminate the assumption with these to get a contradiction.
- Right-to-left: Assuming $(\sigma_1 + \sigma_2)[i] \neq 0$
  The goal is $(\sigma_1[i] \neq 0 \vee (\sigma_2)[i] \neq 0)$ which is equivalent to $\neg(\sigma_1[i] = 0 \wedge \sigma_2[i] = 0)$ thus we go by contradiction:
  Assuming $(\sigma_1[i] = 0 \wedge \sigma_2[i] = 0)$ Therefore by monoid unit we have $(\sigma_1 + \sigma_2)[i] = 0$ which contradicts the assumption hence we by PBC that $(\sigma_1[i] \neq 0 \vee (\sigma_2)[i] \neq 0)$.

□

Lemma B.7. *Given a type* $(\Delta \mid \sigma \mid \boldsymbol{0}) \odot \Gamma \vdash (x :_{(s,0)} A) \to B : C$ *where* $(\Delta \mid \boldsymbol{0}) \odot \Gamma \vdash C \leq \mathsf{Type}_l$*, then when* $[\![B]\!]$ *is defined, then* $x \notin \mathsf{fv}(B)$*.*

Proof. By inversion on T_ARROW we have:

Case.

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l_1} \qquad (\Delta, \sigma_1 \mid \sigma_2, 0 \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,0)} A) \to B : \mathsf{Type}_{l_1 \sqcup l_2}} \text{ T\_ARROW}$$

Such that $\sigma = \sigma_1 + \sigma_2$.

By Lemma B.6 on the second premise then we have that that $\mathsf{fv}(B) = \{y \mid (\Gamma, x : A)[i] = y : B \wedge (\sigma_2, 0)[i] \neq 0\}$.

Since at the position where $(\Gamma, x : A)[i] = x : A$ then $(\sigma_2, 0)[i] = 0$ therefore $x \notin \mathsf{fv}(B)$.

Case.

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,0)} A) \to B : D \qquad (\Delta \mid \mathbf{0}) \odot \Gamma \vdash D \leq C}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,0)} A) \to B : C} \text{ T\_TYCONV}$$

By ST_TRANS we have $(\Delta \mid \mathbf{0}) \odot \Gamma \vdash D \leq \mathsf{Type}_l$, and therefore our goal holds by induction.

$\square$

Lemma B.8. *Given a quantitative semiring, if* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : B$ *and* $[\![B]\!]$ *is defined and for all* $x : A \in \Gamma$ *such that* $[\![A]\!]$ *is defined, then:*

- $\mathit{fv}(B) = \{x \mid \Gamma[i] = x : A \wedge \sigma_2[i] \neq 0\}$
- $\forall i.0 \leq i < |\Gamma|$.
  $\mathit{fv}(\Gamma[i]) = \{y \mid \forall j.\Gamma[i] = x : A \wedge j < i \wedge \Gamma[j] = y : C \wedge \Delta[i][j] \neq 0)\}$

Proof.

- Case T_Type

$$\frac{\Delta \odot \Gamma \vdash}{(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}}\ \text{T\_Type}$$

By Lemma B.2 on the premise (under the encodability restriction) then we get (3).

Conclusion (1) follows since $\neg\sigma_1[i] \neq 0$ for all i thus:

$$\mathsf{fv}(\mathsf{Type}_l) = \emptyset = \{x \mid \Gamma[i] = x : A \wedge \sigma_2[i] \neq 0\}$$

- Case T_Var

$$\frac{\Delta_1, \sigma, \Delta_2 \odot \Gamma_1, x : A, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \sigma, \Delta_2 \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \sigma, 0, \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash x : A}\ \text{T\_Var}$$

By Lemma 3.4 we get that $\Delta_1, \sigma \odot \Gamma_1, x : A \vdash$ then by inversion we have that:

$$(\Delta_1 \mid \sigma \mid \mathbf{0}) \odot \Gamma_1 \vdash A : \mathsf{Type}_l$$

Applying, Lemma B.6 then we know that:

$$(*) \quad \mathsf{fv}(A) = \{y \mid \Gamma[i] = y : B \wedge \sigma[i] \neq 0\}$$

We can thus prove the two goals as follows:

1. Goal is $\mathsf{fv}(A) = \{x \mid (\Gamma_1, x : A, \Gamma_2)[i] = y : A' \wedge \sigma, 0[i] \neq 0\}$ which is provided by (*) since for all of $i > |G1|$ we have that we have that $\sigma, 0 = 0$ thus, the goal collapse to (*).
2. This follows by Lemma B.2 applied to the premise then we have the goal:

$$\begin{aligned}
&\forall i.0 \leq i < |\Gamma_1, x : A, \Gamma_2| \,. \\
&\mathsf{fv}(\Gamma_1, x : A, \Gamma_2[i]) = \\
&\quad \{y \mid \forall j.(\Gamma_1, x : A, \Gamma_2)[i] = x : A \wedge j < i \wedge (\Gamma_1, x : A, \Gamma_2)[j] = y : C \wedge (\Delta_1, \sigma, \Delta_2)[i][j] \neq 0)\}
\end{aligned}$$

- Case T_Fun

$$\frac{(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \quad (\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t : B}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t : (x :_{(s,r)} A) \to B}\ \text{T\_Fun}$$

By induction we have:

$$\begin{aligned}
&(1)\mathsf{fv}(B) = \{y \mid \Gamma, x : A[i] = y : C \wedge (\sigma_3, r)[i] \neq 0\} \\
&(2)\forall i.0 \leq i < |\Gamma, x : A| \,. \\
&\quad \mathsf{fv}(\Gamma, x : A[i]) = \{y \mid \forall j.(\Gamma, x : A)[i] = x' : A' \wedge j < i \wedge (\Gamma, x : A)[j] = y' : B' \wedge (\Delta, \sigma_1)[i][j] \neq 0)\}
\end{aligned}$$

Two goals are then:

1.
$$
\begin{aligned}
&\mathsf{fv}((x :_{(s,r)} A) \to B) \\
&\qquad = \mathsf{fv}(A) \cup (\mathsf{fv}(B) \setminus \{x\}) \\
&\qquad = \{y \mid \Gamma[i] = y : C \wedge (\sigma_1 + \sigma_3)[i] \neq 0\}
\end{aligned}
$$

By Lemma 3.3 and inversion of well-formedness we have that $(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l$ which we apply Lemma B.6 to, yielding:

$$\mathsf{fv}(A) = \{y \mid \Gamma[i] = y : B \wedge \sigma_1[i] \neq 0\}$$

From (1) we can get $\mathsf{fv}(B) = \{y \mid \Gamma, x : A[i] = y : C \wedge (\sigma_3, r)[i] \neq 0\} \setminus \{x\} = \{y \mid \Gamma[i] = y : C \wedge (\sigma_3)[i] \neq 0\}$.
We then prove the goal by showing $((\sigma_1[i] \neq 0) \vee (\sigma_3[i] \neq 0)) \Leftrightarrow (\sigma_1 + \sigma_3)[i] \neq 0$. Which follows by positivity (see above arguments for similar reasoning).

2. And the second goal follows by the inductive hypothesis (2) removing $\{x\}$.

- Case T_App

$$\frac{\begin{array}{l}(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t_1 : (x :_{(s,r)} A) \to B \quad (\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : A\end{array}}{(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash t_1\, t_2 : [t_2/x]B}\ \textsc{T\_App}$$

By induction we have:

(A1) $\mathsf{fv}((x :_{(s,r)} A) \to B) = \{x \mid \Gamma[i] = x : A \wedge (\sigma_1 + \sigma_3)[i] \neq 0\}$

(A2) $\forall i. 0 \leq i < |\Gamma|$.
$\mathsf{fv}(\Gamma[i]) = \{y \mid \forall j. \Gamma[i] = x : A \wedge j < i \wedge \Gamma[j] = y : B \wedge \Delta[i][j] \neq 0)\}$

(B1) $\mathsf{fv}(A) = \{x \mid \Gamma[i] = x : A \wedge \sigma_1[i] \neq 0\}$

(B2) (same as (A3))

The second goal follows from A2 or B2 trivially.

To get the first goal, there are two cases depending on the grades:

- $r = 0$
  Goal is $\mathsf{fv}([t_2/x]B) == \{y \mid \Gamma[i] = y : C \wedge (\sigma_3)[i] \neq 0\}$ since $\sigma_3 + 0 * \sigma_4 = \sigma_3$ by absorption.
  By Lemma B.7 then we know that $x \notin \mathsf{fv}(B)$, therefore $\mathsf{fv}([t_2/x]B) = \mathsf{fv}(B)$, which follows by inversion on the typing (and since we cannot do promotion here under the translation) (Lemma 3.6 with no box).
- $s = 0$. And in this case $A = \mathsf{Type}_l$ therefore t_2 is a type term A' subject to the interpretation thus we apply Lemma B.6 to it, yielding:

  $$\mathsf{fv}(A) = \{y \mid \Gamma[i] = y : B \wedge \sigma_4[i] \neq 0\}$$

  By inversion on the first premise (Lemma 3.6) and since we have no promotion then we get:

  $$\mathsf{fv}(B) = \{y \mid \Gamma[i] = y : B \wedge \sigma_3[i] \neq 0\}$$

  Now the goal is that $\mathsf{fv}([A/x]B) = \{x \mid \Gamma[i] = x : A \wedge \sigma_3 + r * \sigma_4[i] \neq 0\}$.
  We consider two cases depending on whether r is 0 or not.
  $r = 0$ $\mathsf{fv}([[A/x]B]) = \{x \mid \Gamma[i] = x : A \wedge \sigma_3[i] \neq 0\}$ and by Lemma B.7 and the same reasoning about for $r = 0$ (previous case) then we have the goal.
  $r \neq 0$
  We can prove the goal by showing that $(\sigma_3 + r * \sigma_4)[i] \neq 0 \leftrightarrow (\sigma_3[i] \neq 0) \vee (\sigma_4[i] \neq 0)$

* Going left-to-right. Assume: $(\sigma_3 + r * \sigma_4)[i] \neq 0$.
  The goal is equivalent to $\neg(\sigma_3[i] = 0 \wedge \sigma_4[i] = 0)$ (De Morgan's). Then we prove by contradiction by assuming $\sigma_4[i] = 0 \wedge \sigma_4[i] = 0$ By absorption then $(r * \sigma_3)[i] = 0$. Then by monoid unit we have $(r * \sigma_3 + \sigma_4)[i] = 0$. which contradicts the premise. Thus we have $(\sigma_3[i] \neq 0) \vee (\sigma_4[i] \neq 0)$.
  If $\sigma_3[i] = 0$ then $r * \sigma_4[i] = 0$ by absorb therefore
* Going right-to-right. Assume: $(\sigma_3[i] \neq 0) \vee (\sigma_4[i] \neq 0)$
  Proof by contradiction. Assume $(\sigma_3 + r * \sigma_4)[i] = 0$ then from positivity we have $\sigma_3[i] = 0$ and $(r * \sigma_4)[i] = 0$ and from zero-product we have that $r[i] = 0 \vee \sigma_4[i] = 0$.
  We eliminate the assumption via:
  · $(\sigma_3[i] \neq 0)$ this contradicts $\sigma_3[i] = 0$.
  · $(\sigma_4[i] \neq 0)$ since $r[i] \neq 0$ then we must have $\sigma_4[i] = 0$ which contradicts this assumption.

  Thus we have reached falsum, and so we prove the goal by contradiction.

□

Corollary B.8.1. *For a quantitative semiring, with $[\![B]\!]$ defined then $(\Delta \mid \sigma_1 \mid \boldsymbol{0}) \odot \Gamma \vdash t : B$ implies that B is closed:*

Corollary B.8.2. *For a quantitaitve semiring, with $[\![A]\!]$ define then $(\Delta, \boldsymbol{0}, \Delta' \mid \sigma_1 \mid \sigma_2) \odot \Gamma, x : A, \Gamma' \vdash t : B$ implies that A is closed:*

B.2.2 Interpretation of well-formedness

* Case Wf_Empty

$$\frac{}{\emptyset \odot \emptyset \vdash}\ \textsc{Wf\_Empty}$$

  We can thus construct the SSF well-formedness derivation:

$$\frac{}{\cdot\ \mathit{Ok}}$$

* Case Wf_Ext

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l}{\Delta, \sigma \odot \Gamma, x : A \vdash}\ \textsc{Wf\_Ext}$$

  By Lemma 3.3 then we have $\Delta \odot \Gamma \vdash$ and thus by induction we have $[\![\Gamma]\!]\ \mathit{Ok}$.

  – Then if $A = \mathsf{Type}_l$ then we can derfine the derivation:

$$\frac{[\![\Gamma]\!]\ \mathit{Ok}}{[\![\Gamma, x : \star_l]\!]\ \mathit{Ok}}$$

  – Otherwise, we apply the main encoding to get some term T such that $[\![\Gamma]\!] \vdash T : \star_l$ Thus we can build the derivation:

$$\frac{[\![\Gamma]\!] \vdash T : \star_l \qquad [\![\Gamma]\!]\ \mathit{Ok}}{[\![\Gamma]\!], x : T\ \mathit{Ok}}$$

B.2.3 Interpretation of derivations

* Case T_Type

$$\frac{\Delta \odot \Gamma \vdash}{(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}}\ \textsc{T\_Type}$$

  This has no analogue in SSF since $\star_l$ is not a type itself in SSF.

- Case T_VAR

$$\frac{\Delta_1, \sigma, \Delta_2 \odot \Gamma_1, x : A, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \sigma, \Delta_2 \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \sigma, 0, \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash x : A} \text{T\_VAR}$$

By the interepretation of well-formendess we have that $[\![\Gamma_1]\!], x : [\![A]\!], [\![\Gamma_2]\!]\ \mathit{Ok}$. Then we form the derivation:

$$\frac{[\![\Gamma_1]\!], x : [\![A]\!], [\![\Gamma_2]\!]\ \mathit{Ok}}{[\![\Gamma_1]\!], x : [\![A]\!], [\![\Gamma_2]\!] \vdash x : [\![A]\!]}$$

- Case T_ARROW

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : \mathsf{Type}_{l_1 \sqcup l_2}} \text{T\_ARROW}$$

We have two cases depending on A.

  - $A = \mathsf{Type}_l$ then by induction we have SSF term B' such that:

$$[\![\Gamma]\!] \vdash \star_l : \star_{l_1} \quad \text{(ih1)}$$
$$[\![\Gamma]\!], x : \star_l \vdash B' : \star_{l_2} \quad \text{(ih2)}$$

    where $l < l_1$ by the GRTT universes. We can thus form the following SSF derivation:

$$\frac{\text{(ih2)}}{[\![\Gamma]\!] \vdash \forall (X : \star_l).B' : \star_{(l+1) \sqcup l_2}}$$

    and then by either $(l+1) \sqcup l_2 = l_1 \cup l_2$ and we are done, or if $l + 1 \leq l_1$ then by monotonicity we have $(l+1) \sqcup l_2 \leq l_1 \sqcup l_2$ and we use Eades and Stump [20] (Lemma 3) to get:

$$\frac{\text{(ih2)}}{[\![\Gamma]\!] \vdash \forall (X : \star_l).B' : l_1 \sqcup l_2}$$

  - $\mathsf{Type}_l \not\in^{+ve} A$
    By the Corlloary B.8.2 and the definition of the SSF$(-)$, we know that B (and in this case $r = 0$) here must in fact be closed, so we can *strengthen* then premise to:

$$(\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash B : \mathsf{Type}_{l_2}$$

    Then by induction we have SSF terms A' and B' such that:

$$[\![\Gamma]\!] \vdash A' : \star_{l_1} \quad \text{(ih1)}$$
$$[\![\Gamma]\!] \vdash B' : \star_{l_2} \quad \text{(ih2)}$$

    We can thus form the following SSF derivation:

$$\frac{\text{(ih1)} \qquad \text{(ih2)}}{[\![\Gamma]\!] \vdash A' \to B' : \star_{l_1 \sqcup l_2}}$$

- Case T_TEN trivial since we ignore products

- Case T_FUN

$$\frac{(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \quad (\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t : B}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t : (x :_{(s,r)} A) \to B} \text{T\_FUN}$$

We define two cases depending on A:

– $\mathsf{Type}_l = A$

$$\frac{(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : \mathsf{Type}_l \vdash t : B}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t : (x :_{(0,r)} \mathsf{Type}_l) \to B} \text{ T\_FUN}$$

By induction on the premise we have some SSF term t' such that:

$$[\![\Gamma]\!], x : \star_l \vdash t' : [\![B]\!]$$

Then we construct the following derivation in SSF:

$$\frac{[\![\Gamma]\!], x : \star_l \vdash t' : [\![B]\!]}{[\![\Gamma]\!] \vdash \Lambda(x : \star_l).t' : \forall x : \star_l.[\![B]\!]}$$

– $\mathsf{Type}_l \notin^{+ve} A$ (no positive occurences of Type_l)

$$\frac{(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t : B}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t : (x :_{(0,r)} A) \to B} \text{ T\_FUN}$$

By induction on the premise we have that exists some SSF term t' such that:

$$[\![\Gamma]\!], x : [\![A]\!] \vdash t' : [\![B]\!]$$

Then we construct the following derivation in SSF:

$$\frac{[\![\Gamma]\!], x : [\![A]\!] \vdash t' : [\![B]\!]}{[\![\Gamma]\!] \vdash \lambda(x : [\![A]\!]).t' : [\![A]\!] \to [\![B]\!]}$$

- Case T_APP

$$\frac{\begin{array}{l}(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t_1 : (x :_{(s,r)} A) \to B \quad (\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : A\end{array}}{(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash t_1\, t_2 : [t_2/x]B} \text{ T\_APP}$$

We define two cases depending on A:

– $\mathsf{Type}_l = A$ By induction we have terms t' and T' such that:

$$[\![\Gamma]\!] \vdash t' : \forall(x : \star_l).[\![B]\!] \tag{3}$$

$$[\![\Gamma]\!] \vdash T : \star_l \tag{4}$$

Then we can form the SSF derivation:

$$\frac{ih1 \quad ih2}{[\![\Gamma]\!] \vdash t'[T] : [T/x][\![B]\!]}$$

where $[T/x][\![B]\!]$ matches $[\![[t_2/x]B]\!]$ by Lemma B.9.

– $\mathsf{Type}_l \notin^{+ve} A$ (no positive occurences of Type_l) thus $r = 0$.
By induction we have two terms t_1' and t_2'

$$[\![\Gamma]\!] \vdash t_1' : [\![A]\!] \to [\![B]\!] \tag{5}$$

$$[\![\Gamma]\!] \vdash t_2' : [\![A]\!] \tag{6}$$

Then we can form the SSF derivation:

$$\frac{\text{(ih1)} \qquad \text{(ih2)}}{[\![\Gamma]\!] \vdash t_1'\ t_2' : [\![B]\!]}$$

which is equal to the goal here by since by Lemma B.7 since $[t_2/x]B = B$.

- Case T_PAIR and case T_TENCUT are trivial since we exclude tensor types.
- Case T_BOX, T_BOXI, T_BOXE trivial as we don't translate graded modalities

B.2.4 Soundness

We first need some further auxiliary lemmas

Lemma B.9 (Preservation for substitution on terms)**.**

$$[\![[A/x]B]\!] = [[\![A]\!]/x][\![B]\!]$$

Proof. By induction on the syntax:

- $[\![[A/x]\mathsf{Type}_l]\!] = [\![\mathsf{Type}_l]\!] = \star_l = [[\![A]\!]/x]\star_l$
- $[\![[A/x]((y :_{(0,r)} \mathsf{Type}_l) \to B)]\!] = [\![((y :_{(0,r)} \mathsf{Type}_l) \to [A/x]B)]\!] = \forall x : \star_l.[\![[A/x]B]\!] = \forall x : \star_l.[[\![A]\!]/x][\![B]\!] = [[\![A]\!]/x](\forall x : \star_l.[\![B]\!])$ (penulimate step by induction)
- $[\![[A/x]((y :_{(s,0)} A') \to B)]\!] = [\![((y :_{(s,0)} [A/x]A') \to [A/x]B)]\!] = [\![[A/x]A']\!] \to [\![[A/x]B]\!] = [[\![A]\!]/x][\![A']\!] \to [[\![A]\!]/x][\![B]\!] = [[\![A]\!]/x]([\![A']\!] \to [\![B]\!])$ (penultimate step is by induction)

  (*where* $\mathsf{Type}_l \not\in^{+ve} A$)

∎

Lemma B.10 (Substitution soundness)**.** *For all* $\Delta, \sigma_1, \sigma_2, \sigma_3, \sigma_4, t_1, t_2, l, B,\ \Gamma,\ T,\ y,\ t_0,$ *where* $A \neq \mathsf{Type}_l$, *if:*

$$[\![(\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : A]\!] = [\![\Gamma]\!] \vdash t_b : [\![A]\!] \tag{1}$$

$$[\![(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, y : A \vdash t_1 : B]\!] = [\![\Gamma]\!], y : [\![A]\!] \vdash t_a : B \tag{2}$$

then

$$[\![(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash [t_2/y]t_1 : [t_2/y]B]\!] = [\![\Gamma]\!] \vdash [t_b/y]t_a : [\![B]\!]$$

Proof. By induction on the typing of t_1 under the restriction of the translation

- Case T_Var. Depends on whether the variable x introduced by the T_Var is equal to y or not:
  - $y = x$ then $t_1 = x$ and $[t_2/y]t_1 = t_2$ the interpretation of which is t_b by premise (1). In this case then $B = A$ (from that fact that $y = x$). By the encoding then $t_a = y$ and thus $[t_b/y]t_a = t_b$ satisfying the goal here.
  - $y \neq x$ then $t_1 = y$ and $[t_2/y]t_1 = y = x$.
- Case T_Arrow thus $t_1 = (x :_{(s,r)} A') \to B'$ with:

$$\frac{\begin{array}{c}(\Delta, \sigma_1 \mid \sigma'_1 \mid \mathbf{0}) \odot \Gamma, y : A \vdash A' : \mathsf{Type}_{l_1} \\ (\Delta, \sigma_1, \sigma'_1 \mid \sigma'_2, r' \mid \mathbf{0}) \odot \Gamma, y : A, x : A' \vdash B' : \mathsf{Type}_{l_2}\end{array}}{(\Delta, \sigma_1 \mid \sigma'_1 + \sigma'_2 \mid \mathbf{0}) \odot \Gamma, y : A \vdash (x :_{(s',r')} A') \to B' : \mathsf{Type}_{l_1 \sqcup l_2}}\ \text{T\_Arrow}$$

  where $\sigma_2, s = \sigma'_1 + \sigma'_2$ and $\mathbf{0} = \sigma_3, \sigma$.

  - Case $A' = \mathsf{Type}_{l'}$
    Thus, $t_a = \forall(X : \star_{l'}).B''$ by the interpretation on function types.
    By induction, then we know:

$$[\![[t_2/y]\mathsf{Type}_l]\!] = [t_b/y]\star_{l'}$$

$$[\![[t_2/y]B']\!] = [t_b/y]B''$$

    Therefore, $[\![[t_2/y]((x :_{(s',r')} A') \to B')]\!] = [\![(x :_{(s',r')} [t_2/y]A') \to [t_2/y]B']\!] = \forall(X : \star_{l'}).[t_b/y]B'' = [t_b/y](\forall(X : \star_{l'}).B'')$ Satisfying the goal here.

– Case $\mathsf{Type}_l \not\in^{+ve} B$
Thus, $t_a = A'' \to B''$ by the interprettion.
By induction we know:

$$[\![[t_2/y]A']\!] = [t_b/y]A''$$
$$[\![[t_2/y]B']\!] = [t_b/y]B''$$

Therefore, $[\![[t_2/y]((x :_{(s',r')} A') \to B')]\!] = [\![(x :_{(s',r')} [t_2/y]\mathsf{Type}_l) \to [t_2/y]B']\!] = [t_b/y]A'' \to [t_b/y]B''$ $= [t_b/y](A'' \to B'')$ Satisfying the goal here.

- Case T_Fun We define two cases depending on A':
  – $A' = \mathsf{Type}_{l'}$

$$\frac{(\Delta, \sigma'_1 \mid \sigma'_2, s' \mid \sigma'_3, r') \odot \Gamma, y : A, x : \mathsf{Type}_{l'} \vdash t : B'}{(\Delta \mid \sigma'_2 \mid \sigma'_1 + \sigma'_3) \odot \Gamma, y : A \vdash \lambda x.t : (x :_{(0,r)} \mathsf{Type}_{l'}) \to B'} \text{ T\_Fun}$$

By induction on the premise we have

$$[\![[t_2/y]t]\!] = [t_b/y]t'_a$$

Therefore $[\![[t_2/y](\lambda x.t)]\!] = \Lambda(x : \star_{l'}).[t_2/y]t'_a = [t_2/y](\Lambda(x : \star_{l'}).t'_a)$ satisfying the goal.
  – $\mathsf{Type}_{l'} \not\in^{+ve} A$ (no positive occurences of $\mathsf{Type}_{l'}$)

$$\frac{(\Delta, \sigma_1, \sigma'_1 \mid \sigma'_2, s, s' \mid \sigma'_3, r, r') \odot \Gamma, y : A, x : A' \vdash t : B'}{(\Delta, \sigma_1 \mid \sigma'_2, s \mid \sigma'_1 + \sigma'_3, r') \odot \Gamma, y : A \vdash \lambda x.t : (x :_{(0,r)} A') \to B'} \text{ T\_Fun}$$

Let $\sigma_5, r, \sigma_6 = \sigma'_1 + (\sigma''_3, r)$
By induction on the premise we have that exists some SSF term t' such that:

$$[\![\begin{pmatrix} \Delta,(\emptyset,\sigma'_1\backslash\pi+(\emptyset,\sigma'_1/\pi)*\sigma_4) \\ (\sigma'_2+s*\sigma_4),s' \\ (\sigma'_3+r*\sigma_4),\sigma_6 \end{pmatrix} \odot \Gamma, y : \mathsf{Type}_l, x : [t_2/y]A' \vdash [t_2/y]t : [t_2/y]B']\!] = [t_a/y]t'_0$$

Therefore

$$[\![[t_2/y](\lambda x.t)]\!] = \lambda(x : [\![[t_2/y]A']\!]).[t_a/y]t'_0$$

Then by Lemma B.9 this is equal to:

$$\begin{aligned} &(\lambda(x : [t_a/y][\![A']\!]).[t_a/y]t'_0) \\ =&[t_a/y](\lambda(x : [\![A']\!]).t'_0) \end{aligned}$$

satisfying the goal.

- Case T_App

$$\frac{(\Delta \mid \sigma'_2 \mid \sigma'_1 + \sigma'_3) \odot \Gamma, y : A \vdash t'_1 : (x :_{(s',r')} A') \to B' \qquad (\Delta \mid \sigma'_4 \mid \sigma'_1) \odot \Gamma, y : A \vdash t'_2 : A'}{(\Delta \mid \sigma'_2 + s' * \sigma'_4 \mid \sigma'_3 + r' * \sigma'_4) \odot \Gamma, y : A \vdash t'_1\, t'_2 : [t_2/x]B'} \text{ T\_App}$$

We define two cases depending on A':

  – $A' = \mathsf{Type}_l$
Thus $t_a = t'_a[T']$ (where the intermrediate of t'_2 is interpreted as to T')
By induction, then we know:

$$[\![[t2/y]t1']\!] = [t_b/y]T'$$
$$[\![[t2/y]t2']\!] = [t_b/y]t'_a$$

Therefore

$$[\![[t_2/y](t'_1\, t'_2)]\!] = [t_b/y]t'_a[[t_b/y]T'] = [t_b/y](t'_a[T'])$$

satisfying the goal.

– $\mathsf{Type}_{l'} \not\in^{+ve} A'$ (no positive occurences of $\mathsf{Type}_{l'}$)
Thus $t_a = t'_a\ t''_a$.
By induction we know that:

$$[\![[t_2/y]t1']\!] = [t_b/y]t'_a$$
$$[\![[t_2/y]t2']\!] = [t_b/y]t''_a$$

Therefore
$$[\![[t_2/y](t'_1\,t'_2)]\!] = [t_b/y]t'_a\ [t_b/y]t''_a = [t_b/y](t'_a\ t''_a)$$
satisfying the goal.

□

Lemma B.11 (Type substitution soundness)**.** *For all* $\Delta, \sigma_1, \sigma_2, \sigma_3, \sigma_4, t_1, t_2, l, B,\ \Gamma,\ T,\ y,\ t_0$*, if:*

$$[\![(\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : \mathsf{Type}_l]\!] = [\![\Gamma]\!] \vdash T : \star_l \tag{1}$$
$$[\![(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, y : \mathsf{Type}_l \vdash t_1 : B]\!] = [\![\Gamma]\!], y : \star_l \vdash t_0 : [\![B]\!] \tag{2}$$

then:

$$[\![(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash [t_2/y]t_1 : [t_2/y]B]\!] = [\![\Gamma]\!] \vdash [T/y]t_0 : [T/y][\![B]\!]$$

Proof. By induction on the typing of t_1 under the restriction of the translation

- Case T_Var. Depends on whether the variable x introduced by the T_Var is equal to y or not:
  - $y = x$ then $t_1 = x$ and $[t_2/y]t_1 = t_2$ the interpretation of which is T by premise (1).
    In this case then $B = \mathsf{Type}_l$ (from that fact that $y = x$) and thus (2) is a kinding derivation of SSF.
    By the encoding then $t_0 = y$ and thus $[T/y]t_0 = T$ satisfying the goal here.
  - $y \neq x$ then $t_1 = y$ and $[t_2/y]t_1 = y = x$.
    By the encoding then $t_0 = x$ and thus $[T/y]t_0 = x$ satisfying the goal here.
- Case T_Arrow thus $t_1 = (x :_{(s,r)} A') \to B'$ with:

$$\frac{\begin{array}{c}(\Delta, \sigma_1 \mid \sigma'_1 \mid \mathbf{0}) \odot \Gamma, y : \mathsf{Type}_l \vdash A' : \mathsf{Type}_{l_1} \\ (\Delta, \sigma_1, \sigma'_1 \mid \sigma'_2, r' \mid \mathbf{0}) \odot \Gamma, y : \mathsf{Type}_l, x : A' \vdash B' : \mathsf{Type}_{l_2}\end{array}}{(\Delta, \sigma_1 \mid \sigma'_1 + \sigma'_2 \mid \mathbf{0}) \odot \Gamma, y : \mathsf{Type}_l \vdash (x :_{(s',r')} A') \to B' : \mathsf{Type}_{l_1 \sqcup l_2}}\ \text{T\_Arrow}$$

  where $\sigma_2, s = \sigma'_1 + \sigma'_2$ and $\mathbf{0} = \sigma_3, \sigma$.

  - Case $A' = \mathsf{Type}_{l'}$
    Thus, $t_0 = \forall(X : \star_{l'}).B''$ by the interpretation on function types.
    By induction, then we know:

$$[\![[t_2/y]\mathsf{Type}_l]\!] = [T/y]\star_{l'}$$
$$[\![[t_2/y]B']\!] = [T/y]B''$$

    Therefore, $[\![[t_2/y]((x :_{(s',r')} A') \to B')]\!] = [\![(x :_{(s',r')} [t_2/y]\mathsf{Type}_l) \to [t_2/y]B']\!] = \forall(X : \star_{l'}).[T/y]B''$ $= [T/y](\forall(X : \star_{l'}).B'')$ Satisfying the goal here.

– Case $\mathsf{Type}_l \not\in^{+ve} B$
Thus, $t_0 = A'' \to B''$ by the interpretation.
By induction we know:

$$\llbracket [t_2/y]A' \rrbracket = [T/y]A''$$
$$\llbracket [t_2/y]B' \rrbracket = [T/y]B''$$

Therefore, $\llbracket [t_2/y]((x :_{(s',r')} A') \to B') \rrbracket = \llbracket (x :_{(s',r')} [t_2/y]\mathsf{Type}_l) \to [t_2/y]B' \rrbracket = [T/y]A'' \to [T/y]B''$ $= [T/y](A'' \to B'')$ Satisfying the goal here.

- Case T_Fun We define two cases depending on A':

– $A' = \mathsf{Type}_{l'}$

$$\frac{(\Delta, \sigma_1' \mid \sigma_2', s' \mid \sigma_3', r') \odot \Gamma, y : \mathsf{Type}_l, x : \mathsf{Type}_{l'} \vdash t : B'}{(\Delta \mid \sigma_2' \mid \sigma_1' + \sigma_3') \odot \Gamma, y : \mathsf{Type}_l \vdash \lambda x.t : (x :_{(0,r)} \mathsf{Type}_{l'}) \to B'} \text{ T\_Fun}$$

By induction on the premise we have

$$\llbracket [t_2/y]t \rrbracket = [T/y]t_0'$$

Therefore $\llbracket [t_2/y](\lambda x.t) \rrbracket = \Lambda(x : \star_{l'}).[T/y]t0' = [T/y](\Lambda(x : \star_l).t0')$ satisfying the goal.

– $\mathsf{Type}_l \not\in^{+ve} A$ (no positive occurences of Type_l)

$$\frac{(\Delta, \sigma_1, \sigma_1' \mid \sigma_2', s, s' \mid \sigma_3', r, r') \odot \Gamma, y : \mathsf{Type}_l, x : A' \vdash t : B'}{(\Delta, \sigma_1 \mid \sigma_2', s \mid \sigma_1' + \sigma_3', r') \odot \Gamma, y : \mathsf{Type}_l \vdash \lambda x.t : (x :_{(0,r)} A') \to B'} \text{ T\_Fun}$$

Let $\sigma_5, r, \sigma_6 = \sigma_1' + (\sigma_3'', r)$
By induction on the premise we have that exists some SSF term t' such that:

$$\llbracket \begin{pmatrix} \Delta, (\emptyset, \sigma_1' \setminus \pi + (\emptyset, \sigma_1'/\pi) * \sigma_4) \\ (\sigma_2' + s * \sigma_4), s' \\ (\sigma_3' + r * \sigma_4), \sigma_6 \end{pmatrix} \odot \Gamma, y : \mathsf{Type}_l, x : [t_2/y]A' \vdash [t_2/y]t : [t_2/y]B' \rrbracket = [T/y]t_0'$$

Therefore

$$\llbracket [t_2/y](\lambda x.t) \rrbracket = \lambda(x : \llbracket [t_2/y]A' \rrbracket).[T/y]t_0'$$

Then by Lemma B.9 this is equal to:

$$\begin{aligned} &(\lambda(x : [T/y]\llbracket A' \rrbracket).[T/y]t_0') \\ =&[T/y](\lambda(x : \llbracket A' \rrbracket).t_0') \end{aligned}$$

satisfying the goal.

– Case T_App

$$\frac{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t_1' : (x :_{(s,r)} A') \to B \qquad (\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2' : A'}{(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash t_1' \, t_2' : [t_2/x]B} \text{ T\_App}$$

We define two cases depending on A:

∗ $A' = \mathsf{Type}_l$
Thus $t_0 = t'[T']$ by induction.
By induction, then we know:

$$\llbracket [t2/y]t1' \rrbracket = [T/y]t'$$
$$\llbracket [t2/y]t2' \rrbracket = [T/y]T'$$

Therefore

$$\llbracket [t_2/y](t_1' \, t_2') \rrbracket = [T/y]t'[[T/y]T'] = [T/y](t'[T'])$$

satisfying the goal.

* $\mathsf{Type}_{l'} \notin^{+ve} A'$ (no positive occurences of $\mathsf{Type}_{l'}$)
  Thus $t_0 = t'_1\ t'_2$.
  By induction we know that:

$$\llbracket [t_2/y]t1' \rrbracket = [T/y]t'_1$$
$$\llbracket [t_2/y]t2' \rrbracket = [T/y]t'_2$$

  Therefore
$$\llbracket [t_2/y](t'_1\, t'_2) \rrbracket = [T/y]t'_1\ [T/y]t'_2 = [T/y](t'_1\ t'_2)$$
  satisfying the goal.

□

Lemma B.12 (Soundness)**.** *Given* $\textsc{Ssf}((\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : A)$ *and* $\llbracket (\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : A \rrbracket = \llbracket \Gamma \rrbracket \vdash t' : \llbracket A \rrbracket$ *in* GRTT *and* $t_1 \rightsquigarrow t'_1$ *and* $\llbracket (\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t'_1 : A \rrbracket = \llbracket \Gamma \rrbracket \vdash t'' : \llbracket A \rrbracket$ *then* $t' \rightsquigarrow t''$ *in the CBN SSF*

Proof. First we observe that by Type Preservation (Lemma 3.35) that reduction preserves typing thus if $\textsc{Ssf}(-)$ holds for $\llbracket (\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : A \rrbracket$ then $\textsc{Ssf}((\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t'_1 : A)$. Then we show that reduction in GRTT is simulated in SSF by induction on the typing:

- Case T_TYPE, T_VAR, T_ARROW,are all trivial as they already in normal forms (no reduction)
- Case T_FUN.

$$\frac{(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \quad (\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t : B}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t : (x :_{(s,r)} A) \to B}\ \textsc{T\_Fun}$$

  If $A = \mathsf{Type}_l$ then there are no such reductions possible in GRTT under the interpretation (just arrows, vars, and Type) Otherwise we have the encoding: $\dfrac{\llbracket \Gamma \rrbracket, x : \llbracket A \rrbracket \vdash t' : \llbracket B \rrbracket}{\llbracket \Gamma \rrbracket \vdash \lambda(x : \llbracket A \rrbracket).t' : \llbracket A \rrbracket \to \llbracket B \rrbracket}$ for which there is no possible reduction.

- Case T_APP

$$\frac{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t_1 : (x :_{(s,r)} A) \to B \qquad (\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : A}{(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash t_1\, t_2 : [t_2/x]B}\ \textsc{T\_App}$$

  We define two cases depending on A:

  - $\mathsf{Type}_l = A$ Then the encoding yields

$$\frac{\llbracket \Gamma \rrbracket \vdash t' : \forall(x : \star_l).\llbracket B \rrbracket \quad \llbracket \Gamma \rrbracket \vdash T : \star_l}{\llbracket \Gamma \rrbracket \vdash t'[T] : [T/x]\llbracket B \rrbracket}$$

    There are then three possible GRTT reductions:

    * $(\lambda x.t'_1)\, t_2 \rightsquigarrow [t_2/x]t'_1$ Thus:

$$\begin{aligned} & \llbracket (\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash [t_2/x]t_1 : [t_2/x]B \rrbracket \\ = \ & \llbracket \Gamma \rrbracket \vdash t'' : \llbracket [t_2/x]B \rrbracket \end{aligned}$$

$$\frac{\llbracket (\Delta \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : \mathsf{Type}_l \vdash t'_1 : B \rrbracket = \llbracket \Gamma \rrbracket, x : \star_l \vdash t'_0 : \llbracket B \rrbracket}{\llbracket (\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash (\lambda x.t'_1) : (x :_{(s,r)} \mathsf{Type}_l) \to B \rrbracket = \llbracket \Gamma \rrbracket \vdash \Lambda(x : \star_l).t'_0 : \forall(x : \star_l).\llbracket B \rrbracket}\ \textsc{T\_Fun} \tag{7}$$

By type-application β in SSF we then get:

$$(\Lambda(x:\star_l).t'_0)[T] \rightsquigarrow [T/x]t'_0$$

Applying Lemma B.10 with the translation here with $(\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : \mathsf{Type}_l = [\![\Gamma]\!] \vdash T : \star_l$ and the premise of (7) above gives us:

$$[\![(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash [t_2/x]t'_1 : [t_2/x]B]\!] = [\![\Gamma]\!] \vdash [T/x]t'_0 : [T/x][\![B]\!]$$

Thus satisfying the goal here.

* The other case is the congruence:

$$\frac{t_1 \rightsquigarrow t'_1}{t_1\, t_2 \rightsquigarrow t'_1\, t_2}\ \textsc{Sem\_CongFunOne}$$

By type preservation (Lemma 3.35) on the premise we know that

$$(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t'_1 : (x :_{(s,r)} A) \to B$$

and thus we can apply the encoding again to get some term t'' in SSF:

$$(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t'_1 : (x :_{(s,r)} A) \to B = [\![\Gamma]\!] \vdash t'' : [\![(x :_{(s,r)} A) \to B]\!]$$

Thus the interpretation of the result is:

$$[\![(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash t'_1\, t_2 : [t_2/x]B]\!] = [\![\Gamma]\!] \vdash t''[T] : [T/x][\![B]\!]$$

By induction on the premise, with the encoding of t_1 and $t_1 \rightsquigarrow t'_1$ and (8) then we have that the SSF reduction:

$$t' \rightsquigarrow t''$$

Then we can construct the SSF reduction:

$$\frac{t' \rightsquigarrow t''}{t'[T] \rightsquigarrow t''[T]}$$

which satisfies the goal here.

– $\mathsf{Type}_l \not\in^{+ve} B$ then the encoding yields:

$$\frac{[\![\Gamma]\!] \vdash t_a : [\![A]\!] \to [\![B]\!] \quad [\![\Gamma]\!] \vdash t_b : [\![A]\!]}{[\![\Gamma]\!] \vdash t_a\ t_b : [\![B]\!]} \tag{8}$$

There are then two possible Grtt reductions:

* $(\lambda x.t'_1)\, t_2 \rightsquigarrow [t_2/x]t'_1$ Thus we have the encoding:

$$\frac{[\![(\Delta \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t'_1 : B]\!] = [\![\Gamma]\!], x : [\![A]\!] \vdash t'_a : [\![B]\!]}{[\![(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash (\lambda x.t'_1) : (x :_{(s,r)} A) \to B]\!] = [\![\Gamma]\!] \vdash \lambda(x : [\![A]\!]).t'_a : [\![A]\!] \to [\![B]\!]}\ \textsc{T\_Fun} \tag{9}$$

Applying Lemma B.10 with the translation here of t_2 to t_b and the premise of (9) then gives:

$$[\![(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash [t_2/x]t'_1 : [t_2/x]B]\!] = [\![\Gamma]\!] \vdash [t_b/x]t'_a : [\![B]\!]$$

By β in SSF we then get:

$$(\lambda(x : [\![A]\!]).t'_a)t_b \rightsquigarrow [t_b/x]t'_a$$

Thus showing soundness of this reduction.

* The other case is the congruence:

$$\frac{t_1 \rightsquigarrow t_1'}{t_1\,t_2 \rightsquigarrow t_1'\,t_2}\ \textsc{Sem\_CongFunOne}$$

By type preservation (Lemma 3.35) on the premise we know that

$$(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t_1' : (x :_{(s,r)} A) \to B$$

and thus we can apply the encoding again to get some term t_a' in SSF:

$$(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t_1' : (x :_{(s,r)} A) \to B = [\![\Gamma]\!] \vdash t_a' : [\![(x :_{(s,r)} A) \to B]\!]$$

Thus the interpretation of the result is:

$$[\![(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash t_1'\,t_2 : [t_2/x]B]\!] = [\![\Gamma]\!] \vdash t_a'\ t_b : [\![B]\!]$$

By induction on the premise, with the encoding of t_1 and $t_1 \rightsquigarrow t_1'$ and (8) then we have that the SSF reduction:

$$t_a \rightsquigarrow t_a'$$

Then we can construct the SSF reduction:

$$\frac{t_a \rightsquigarrow t_a'}{t_a\ t_b \rightsquigarrow t_a'\ t_b}$$

which satisfies the goal here.

□

B.2.5 Completeness

Lemma B.13 (Completeness)**.** *Given* $\textsc{Ssf}((\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A)$ *and* $[\![(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A]\!] = [\![\Gamma]\!] \vdash t_s : [\![A]\!]$ *in* $\textsc{Grtt}$ *and* $t_s \rightsquigarrow^{\mathsf{ssf}} t_s'$ *then* $t \rightsquigarrow t'$ *and* $[\![(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t' : A]\!] = [\![\Gamma]\!] \vdash t_s'' : [\![A]\!]$ *with* $t_s'' = t_s'$

Proof. By induction on typing derivations in the image of the encoding:

- Case T_Var

$$\frac{\Delta_1, \sigma, \Delta_2 \odot \Gamma_1, x : A, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \sigma, \Delta_2 \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \sigma, 0, \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash x : A}\ \textsc{T\_Var}$$

  Give us the derivation:

$$\frac{[\![\Gamma_1]\!], x : [\![A]\!], [\![\Gamma_2]\!]\ \ Ok}{[\![\Gamma_1]\!], x : [\![A]\!], [\![\Gamma_2]\!] \vdash x : [\![A]\!]}$$

  Since variables cannot reduce the premise of the lemma here is false, thus trivially holds.

- Case T_Arrow maps to the type language of SSF and so has no reduction.

- Case T_Fun is interpreted as an SSF function which has no reduction (under the CBN semantics) so this case trivially holds similar to the T_Var case.

- Case T_App with $t = t_1 t_2$:

$$\frac{\begin{array}{l}(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t_1 : (x :_{(s,r)} A) \to B \quad (\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : A\end{array}}{(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash t_1\,t_2 : [t_2/x]B}\ \textsc{T\_App}$$

  There are two cases depending on A:

– $\mathsf{Type}_l = A$ By induction we have terms t_0 and T' such that:

$$\llbracket\Gamma\rrbracket \vdash t_0 : \forall(x : \star_l).\llbracket B\rrbracket \tag{10}$$
$$\llbracket\Gamma\rrbracket \vdash T : \star_l \tag{11}$$

Then with the SSF derivation:

$$\frac{\mathit{ih1} \quad \mathit{ih2}}{\llbracket\Gamma\rrbracket \vdash t_0[T] : [T/x]\llbracket B\rrbracket}$$

where $[T/x]\llbracket B\rrbracket$ matches $\llbracket[t_2/x]B\rrbracket$ by Lemma B.9.
There are two possible reductions depending on the form of t_0:

* $$\frac{t_0 \rightsquigarrow^{\mathsf{ssf}} t_0'}{t_0[T] \rightsquigarrow^{\mathsf{ssf}} t_0'[T]}$$

  Thus by induction on t_1 (ih1) with the premise of this reduction here we have:

  $t_1 \rightsquigarrow t_1' \quad \wedge \quad \llbracket(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t_1' : (x :_{(s,r)} A) \to B\rrbracket = \llbracket\Gamma\rrbracket \vdash t_1'' : \llbracket A\rrbracket \to \llbracket B\rrbracket \quad \wedge \quad t_1'' = t_0'$

  Thus we can derive the reduction in GRTT of:

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

  and we have that $t_1''[T] = t_0'[T]$ satisfying the goal.
* The remaining case is of a β-reduction and thus we must have that $t_1 = \lambda x.t_1'$. By the interpretation we then get $(\Lambda(x : \llbracket A\rrbracket).t_s') = t_0$ in SSF (thus $\llbracket t1'\rrbracket = t_s'$) thus here we have reduction:

  $$(\Lambda(x : \llbracket A\rrbracket).t_s')[T] \rightsquigarrow^{\text{SSF}} [T/x]t_s'$$

  GrTT can then make a β reduction itself as:

  $$(\lambda x.t_1')t_2 \rightsquigarrow [t_2/x]t_1'$$

  The goal is then that $\llbracket(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash [t_2/x]t_1' : A\rrbracket = \llbracket\Gamma\rrbracket \vdash t_s'' : \llbracket A\rrbracket$ and $t_s'' = [T/x]t_s'$ which we get from Lemma B.11.

– $\mathsf{Type}_l \not\in^{+ve} A$ (no positive occurences of Type_l) thus $r = 0$.
Which inductively has two interpeted terms:

$$\llbracket\Gamma\rrbracket \vdash t_{s1} : \llbracket A\rrbracket \to \llbracket B\rrbracket \tag{12}$$
$$\llbracket\Gamma\rrbracket \vdash t_{s2} : \llbracket A\rrbracket \tag{13}$$

and the SSF derivation:

$$\frac{\text{(ih1)} \qquad \text{(ih2)}}{\llbracket\Gamma\rrbracket \vdash t_{s1}\ t_{s2} : \llbracket B\rrbracket}$$

There are then two possible redctions depending on the form of t_1/t_{s1}:

* $$\frac{t_{s1} \rightsquigarrow^{\mathsf{ssf}} t_{s1}'}{t_{s1}\ t_{s2} \rightsquigarrow^{\mathsf{ssf}} t_{s1}'\ t_{s2}}$$

  Thus, by induction on t_1 with the premise of this reduction we have:

  $t_1 \rightsquigarrow t_1' \quad \wedge \quad \llbracket(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t_1' : (x :_{(s,r)} A) \to B\rrbracket = \llbracket G\rrbracket \vdash t_{s1}'' : \llbracket A\rrbracket \to \llbracket B\rrbracket \quad \wedge \quad t_{s1}'' = t_{s1}'$

Thus we can derive the GRTT reduction of:

$$\frac{t_1 \rightsquigarrow t_1'}{t_1\ t_2 \rightsquigarrow t_1'\ t_2}$$

with $t''_{s1}\ t_{s2} = t'_{s1}\ t_{s2}$

∗ β-reduction with $t_1 = t_1 = \lambda x.t_1'$. By the interpretation we then get $(\lambda(x : [\![A]\!]).t_s) = t_{s1}$ in SSF (thus $[\![t1']\!] = t_s$) thus here we have reduction:

$$(\lambda(x : [\![A]\!]).t_s)\ t_{s2} \rightsquigarrow^{\text{SSF}} [t_{s2}/x]t_s$$

GrTT can then make a β reduction itself as:

$$(\lambda x.t_1')t_2 \rightsquigarrow [t_2/x]t_1'$$

The goal is then that $[\![(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash [t_2/x]t_1' : A]\!] = [\![\Gamma]\!] \vdash t_s'' : [\![A]\!]$ and $t_s'' = [t_{s2}/x]t_s$ which we get from Lemma B.10.

□

C Strong Normalization

Definition C.1. *Typing can be broken up into the following stages:*

$$\begin{aligned}
\mathsf{Kind} &:= \{A \mid \exists \Delta, \sigma_1, \Gamma.(\Delta \mid \sigma_1 \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_1\} \\
\mathsf{Type} &:= \{A \mid \exists \Delta, \sigma_1, \Gamma.(\Delta \mid \sigma_1 \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_0\} \\
\mathsf{Const} &:= \{t \mid \exists \Delta, \sigma_1, \sigma_2, \Gamma, A.(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A \wedge \\
&\qquad (\Delta \mid \sigma_2 \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_1\} \\
\mathsf{Term} &:= \{t \mid \exists \Delta, \sigma_1, \sigma_2, \Gamma, A.(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A \wedge \\
&\qquad (\Delta \mid \sigma_2 \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_0\}
\end{aligned}$$

Lemma C.2 (Classification)**.** *It is the case that* $\mathsf{Kind} \cap \mathsf{Type} = \emptyset$ *and* $\mathsf{Const} \cap \mathsf{Term} = \emptyset$.

Definition C.3. *The set of base terms $\mathcal{B}$ is defined by:*

1. *For x a variable, $x \in \mathcal{B}$,*
2. $\mathsf{Type}_0, \mathsf{Type}_1 \in \mathcal{B}$,
3. *If $t_1 \in \mathcal{B}$ and $t_2 \in \mathsf{SN}$, then $(t_1\, t_2) \in \mathcal{B}$,*
4. *If $t_2 \in \mathcal{B}$ and $t_1 \in \mathsf{SN}$, then $(\mathsf{let}\,(x, y) = t_1 \,\mathsf{in}\, t_2) \in \mathcal{B}$,*
5. *If $t_2 \in \mathcal{B}$ and $t_1 \in \mathsf{SN}$, then $(\mathsf{let}\,\Box x = t_1 \,\mathsf{in}\, t_2) \in \mathcal{B}$,*
6. *If $A, B \in \mathsf{SN}$, then $((x :_{(r,s)} A) \to B) \in \mathcal{B}$ for any $r, s \in R$,*
7. *If $A, B \in \mathsf{SN}$, then $((x :_r A) \otimes B) \in \mathcal{B}$ for any $r \in R$,*
8. *If $A \in \mathsf{SN}$, then $(\Box_r A) \in \mathcal{B}$ for any $r \in R$.*

Definition C.4. *The* key redex *of a term is defined by:*

1. *If t is a redex, then t is its own key redex,*
2. *If t_1 has key redex t, then $(t_1\, t_2)$ has key redex t,*

3. *If* t_1 *has key redex* t*, then* $(\mathsf{let}\,(x,y) = t_1 \,\mathsf{in}\, t_2)$ *has key redex* t*,*

4. *If* t_1 *has key redex* t*, then* $(\mathsf{let}\,\Box x = t_1 \,\mathsf{in}\, t_2)$ *has key redex* t*.*

The term obtained from t *by contracting its key redex is denoted by* $\mathsf{red_k}\, t$*.*

Lemma C.5. *The following are both true:*

1. $\mathcal{B} \subseteq \mathsf{SN}$

2. *The key redex of a term is unique and a head redex.*

Definition C.6. *A set of terms* X *is* saturated *if:*

1. $X \subset \mathsf{SN}$*,*

2. $\mathcal{B} \subset X$*,*

3. *If* $\mathsf{red_k}\, t \in X$ *and* $t \in \mathsf{SN}$*, then* $t \in X$*.*

The collection of saturated sets is denoted by SAT*.*

Lemma C.7 (SN is saturated)**.** *Every saturated set is non-empty and* SN *is saturated.*

Proof. By definition. □

Definition C.8. *For* $T \in \mathsf{Kind}$*, the kind interpretation,* $\mathcal{K}[\![T]\!]$*, is defined inductively as follows:*

$$\begin{array}{l}
\mathcal{K}[\![\mathsf{Type}_0]\!] = \mathsf{SAT}\\
\mathcal{K}[\![(x :_{(r,s)} A) \to B]\!] = \{f \mid f : \mathcal{K}[\![A]\!] \to \mathcal{K}[\![B]\!]\},\ \textit{if}\ A, B \in \mathsf{Kind}\\
\mathcal{K}[\![(x :_{(r,s)} A) \to B]\!] = \mathcal{K}[\![A]\!],\ \textit{if}\ A \in \mathsf{Kind}, B \in \mathsf{Type}\\
\mathcal{K}[\![(x :_{(r,s)} A) \to B]\!] = \mathcal{K}[\![B]\!],\ \textit{if}\ A \in \mathsf{Type}, B \in \mathsf{Kind}\\
\mathcal{K}[\![(x :_{s} A) \otimes B]\!] = \mathcal{K}[\![A]\!] \times \mathcal{K}[\![B]\!],\ \textit{if}\ A, B \in \mathsf{Kind}\\
\mathcal{K}[\![(x :_{s} A) \otimes B]\!] = \mathcal{K}[\![A]\!],\ \textit{if}\ A \in \mathsf{Kind}, B \in \mathsf{Type}\\
\mathcal{K}[\![(x :_{s} A) \otimes B]\!] = \mathcal{K}[\![B]\!],\ \textit{if}\ A \in \mathsf{Type}, B \in \mathsf{Kind}\\
\mathcal{K}[\![\Box_{s} A]\!] = \mathcal{K}[\![A]\!]
\end{array}$$

Definition C.9. Type valuations*,* $\Delta \odot \Gamma \models \varepsilon$*, are defined as follows:*

$$\frac{}{\emptyset \odot \emptyset \models \emptyset}\ \textsc{Ep\_Empty} \qquad \frac{\begin{array}{c} X \in \mathcal{K}[\![A]\!] \\ \Delta \odot \Gamma \models \varepsilon \quad (\Delta \mid \sigma_2 \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_1 \end{array}}{(\Delta, \sigma_2) \odot (\Gamma, x : A) \models \varepsilon[x \mapsto X]}\ \textsc{Ep\_ExtTy}$$

$$\frac{\Delta \odot \Gamma \models \varepsilon \quad (\Delta \mid \sigma_2 \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_0}{(\Delta, \sigma_2) \odot (\Gamma, x : A) \models \varepsilon}\ \textsc{Ep\_ExtTm}$$

Definition C.10. *Given a type valuation* $\Delta \odot \Gamma \models \varepsilon$ *and a type* $T \in (\mathsf{Kind} \cup \mathsf{Type} \cup \mathsf{Con})$ *with* T *typable in*

$\Delta \odot \Gamma$, *we define the interpretation of types,* $[\![T]\!]_\varepsilon$ *inductively as follows:*

$$
\begin{array}{rcl}
[\![\mathsf{Type}_1]\!]_\varepsilon & = & \mathsf{SN} \\
[\![\mathsf{Type}_0]\!]_\varepsilon & = & \lambda X \in \mathsf{SAT}.\mathsf{SN} \\
[\![x]\!]_\varepsilon & = & \varepsilon\, x,\ \textit{if } x \in \mathsf{Con} \\
[\![(x :_{(r,s)} A) \to B]\!]_\varepsilon & = & \lambda X \in \mathcal{K}[\![A]\!] \to \mathcal{K}[\![B]\!].\bigcap_{Y \in \mathcal{K}[\![A]\!]}(([\![A]\!]_\varepsilon)\,(Y) \to ([\![B]\!]_{\varepsilon[x\mapsto Y]})\,(X\,(Y))), \\
 & & \textit{if } A, B \in \mathsf{Kind} \\
[\![(x :_{(r,s)} A) \to B]\!]_\varepsilon & = & \bigcap_{Y \in \mathcal{K}[\![A]\!]}(([\![A]\!]_\varepsilon)\,(Y) \to ([\![B]\!]_{\varepsilon[x\mapsto Y]})), \\
 & & \textit{if } A \in \mathsf{Kind}, B \in \mathsf{Type} \\
[\![(x :_{(r,s)} A) \to B]\!]_\varepsilon & = & \lambda X \in \mathcal{K}[\![B]\!].[\![A]\!]_\varepsilon \to ([\![B]\!]_\varepsilon)\,(X),\ \textit{if } A \in \mathsf{Type}, B \in \mathsf{Kind} \\
[\![(x :_{(r,s)} A) \to B]\!]_\varepsilon & = & [\![A]\!]_\varepsilon \to [\![B]\!]_\varepsilon,\ \textit{if } A, B \in \mathsf{Type} \\
[\![\lambda x : A.B]\!]_\varepsilon & = & \lambda X \in \mathcal{K}[\![A]\!].[\![B]\!]_{\varepsilon[x\mapsto X]},\ \textit{if } A \in \mathsf{Kind}, B \in \mathsf{Con} \\
[\![\lambda x : A.B]\!]_\varepsilon & = & [\![A]\!]_\varepsilon,\ \textit{for any } A \in \mathsf{Kind}, B \in \mathsf{Term} \\
[\![\lambda x : A.B]\!]_\varepsilon & = & [\![B]\!]_\varepsilon,\ \textit{for any } A \in \mathsf{Type}, B \in \mathsf{Con} \\
[\![A\,B]\!]_\varepsilon & = & [\![A]\!]_\varepsilon([\![B]\!]_\varepsilon),\ \textit{if } B \in \mathsf{Con} \\
[\![A\,t]\!]_\varepsilon & = & [\![A]\!]_\varepsilon,\ \textit{if } t \in \mathsf{Term} \\
[\![(x :_s A) \otimes B]\!]_\varepsilon & = & \lambda X \in \mathcal{K}[\![A]\!] \times \mathcal{K}[\![B]\!].([\![A]\!]_\varepsilon)\,(\pi_1\,X) \times ([\![B]\!]_{\varepsilon[x\mapsto \pi_1\,X]})\,(\pi_2\,X), \\
 & & \textit{if } A, B \in \mathsf{Kind} \\
[\![(x :_s A) \otimes B]\!]_\varepsilon & = & \lambda X \in \mathcal{K}[\![B]\!].[\![A]\!]_\varepsilon \times ([\![B]\!]_\varepsilon)\,(X),\ \textit{if } A \in \mathsf{Type}, B \in \mathsf{Kind} \\
[\![(x :_s A) \otimes B]\!]_\varepsilon & = & \lambda X \in \mathcal{K}[\![A]\!].([\![A]\!]_\varepsilon)\,(X) \times [\![B]\!]_{\varepsilon[x\mapsto X]},\ \textit{if } A \in \mathsf{Kind}, B \in \mathsf{Type} \\
[\![(x :_s A) \otimes B]\!]_\varepsilon & = & [\![A]\!]_\varepsilon \times [\![B]\!]_\varepsilon,\ \textit{if } A, B \in \mathsf{Type} \\
[\![(A, B)]\!]_\varepsilon & = & ([\![A]\!]_\varepsilon, [\![B]\!]_\varepsilon) \\
[\![(A, t)]\!]_\varepsilon & = & [\![A]\!]_\varepsilon,\ \textit{if } t \in \mathsf{Term} \\
[\![(t, B)]\!]_\varepsilon & = & [\![B]\!]_\varepsilon,\ \textit{if } t \in \mathsf{Term} \\
[\![\mathsf{let}\,(x : A, y : B) = t\,\mathsf{in}\,C]\!]_\varepsilon & = & [\![C]\!]_{\varepsilon[x\mapsto \pi_1\,[\![t]\!]_\varepsilon, y\mapsto \pi_2\,[\![t]\!]_\varepsilon]},\ \textit{if } A, B \in \mathsf{Kind} \\
[\![\mathsf{let}\,(x : A, y : B) = t\,\mathsf{in}\,C]\!]_\varepsilon & = & [\![C]\!]_{\varepsilon[x\mapsto [\![t]\!]_\varepsilon]},\ \textit{if } A \in \mathsf{Kind}, B \in \mathsf{Type} \\
[\![\mathsf{let}\,(x : A, y : B) = t\,\mathsf{in}\,C]\!]_\varepsilon & = & [\![C]\!]_{\varepsilon[y\mapsto [\![t]\!]_\varepsilon]},\ \textit{if } A \in \mathsf{Type}, B \in \mathsf{Kind} \\
[\![\mathsf{let}\,(x : A, y : B) = t\,\mathsf{in}\,C]\!]_\varepsilon & = & [\![C]\!]_\varepsilon,\ \textit{if } A, B \in \mathsf{Type} \\
[\![\Box_{(r,s)} A]\!]_\varepsilon & = & [\![A]\!]_\varepsilon \\
[\![\Box A]\!]_\varepsilon & = & [\![A]\!]_\varepsilon \\
[\![\mathsf{let}\,\Box(x : A) = t\,\mathsf{in}\,B]\!]_\varepsilon & = & [\![B]\!]_{\varepsilon[x\mapsto [\![t]\!]_\varepsilon]},\ \textit{if } A \in \mathsf{Kind} \\
[\![\mathsf{let}\,\Box(x : A) = t\,\mathsf{in}\,B]\!]_\varepsilon & = & [\![B]\!]_\varepsilon,\ \textit{if } A \in \mathsf{Type}
\end{array}
$$

Definition C.11. *Suppose* $\Delta \odot \Gamma \models \varepsilon$. *Then* Valid term valuations, $\Delta \odot \Gamma \models_\varepsilon \rho$, *are defined as follows:*

$$
\dfrac{}{\emptyset \odot \emptyset \models_\emptyset \emptyset}\ \textsc{Rho\_Empty}
\qquad
\dfrac{\begin{array}{l} t \in ([\![A]\!]_\varepsilon)\,(\varepsilon\,x) \\ \Delta \odot \Gamma \models_\varepsilon \rho \quad (\Delta \mid \sigma_2 \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_1 \end{array}}{(\Delta, \sigma_2) \odot (\Gamma, x : A) \models_\varepsilon \rho[x \mapsto t]}\ \textsc{Rho\_ExtTy}
$$

$$
\dfrac{\begin{array}{l} t \in [\![A]\!]_\varepsilon \\ \Delta \odot \Gamma \models_\varepsilon \rho \quad (\Delta \mid \sigma_2 \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_0 \end{array}}{(\Delta, \sigma_2) \odot (\Gamma, x : A) \models_\varepsilon \rho[x \mapsto t]}\ \textsc{Rho\_ExtTm}
$$

Definition C.12. *Suppose* $\Delta \odot \Gamma \models_\varepsilon \rho$. *Then the* interpretation of a term t *typable in* $\Delta \odot \Gamma$ *is* $(\!|t|\!)_\rho = \rho\, t$, *but where all let-expressions are translated into substitutions, and all graded modalities are erased.*

Definition C.13. *Suppose* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$. *Then* semantic typing, $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \models t : A$, *is defined as follows:*

1. *If* $(\Delta \mid \sigma \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_1$, *then for every* $\Delta \odot \Gamma \models_\varepsilon \rho$, $(\!|t|\!)_\rho \in [\![A]\!]_\varepsilon\,([\![t]\!]_\varepsilon)$.

2. *If* $(\Delta \mid \sigma \mid \boldsymbol{0}) \odot \Gamma \vdash A : \mathsf{Type}_0$, *then for every* $\Delta \odot \Gamma \models_\varepsilon \rho$, $(\!|t|\!)_\rho \in [\![A]\!]_\varepsilon$.

Lemma C.14 (Substitution for Typing Interpretation)**.** *Suppose* $\Delta \odot \Gamma \models \varepsilon$ *and we have types* $T_2 \in (\mathsf{Kind} \cup \mathsf{Con})$ *and* $T_1 \in \mathsf{Con}$ *with* T_1 *and* T_2 *typable in* $\Delta \odot \Gamma$, *and a term* $t \in \mathsf{Term}$ *typable in* $\Delta \odot \Gamma$. *Then:*

1. $[\![T_2]\!]_{\varepsilon[x\mapsto [\![T_1]\!]_\varepsilon]} = [\![[T_1/x]T_2]\!]_\varepsilon$.

2. $[\![T_2]\!]_\varepsilon = [\![[t/x]T_2]\!]_\varepsilon$.

Proof. By straightforward induction on T_2 with the fact that substitutions disappear in the kind interpretation. $\square$

Lemma C.15 (Equality of Interpretations)**.** *Suppose* $\Delta \odot \Gamma \models \varepsilon$*. Then:*

1. *if* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash T_1 = T_2 : \mathsf{Type}_1$*, then* $\mathcal{K}[\![T_1]\!] = \mathcal{K}[\![T_2]\!]$*.*
2. *if* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash T_1 = T_2 : \mathsf{Type}_l$*, then* $[\![T_1]\!]_\varepsilon = [\![T_2]\!]_\varepsilon$*.*

Proof. Part one follows by induction on $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash T_1 = T_2 : \mathsf{Type}_l$, and so does part two, but it also depends on part one and the previous lemma. □

Lemma C.16 (Interpretation Soundness)**.** *Suppose* $\Delta \odot \Gamma \models \varepsilon$ *and* $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_1$*. Then:*

1. *If* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$*, then* $[\![t]\!]_\varepsilon \in \mathcal{K}[\![A]\!]$
2. $[\![A]\!]_\varepsilon \in \mathcal{K}[\![A]\!] \to \mathsf{SAT}$

Proof. This is a proof by simultaneous induction over $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$ and $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_1$. We consider part 1 assuming 2, and vice versa.

Proof of part 1:

Case 1:

$$\frac{\Delta \odot \Gamma \vdash}{(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_0 : \mathsf{Type}_1}\ \textsc{T\_Type}$$

This case holds trivially, because Type_1 cannot be of type Type_1.

Case 2:

$$\frac{\Delta_1, \sigma, \Delta_2 \odot \Gamma_1, x : A, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \sigma, \Delta_2 \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \sigma, 0, \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash x : A}\ \textsc{T\_Var}$$

In this case we have:

$$\begin{aligned}
&\Delta = (\Delta_1, \sigma, \Delta_2)\\
&\sigma_1 = (\mathbf{0}^{|\Delta_1|}, 1, \mathbf{0})\\
&\sigma_2 = (\sigma, 0, \mathbf{0})\\
&t = x
\end{aligned}$$

Thus, we must show that:

$$[\![x]\!]_\varepsilon \in \mathcal{K}[\![A]\!]$$

We know by Definition C.10 and Definition C.9 that $[\![x]\!]_\varepsilon = \varepsilon\, x \in \mathcal{K}[\![A]\!]$; thus, we obtain our result.

Case 3:

$$\frac{(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, x : B_1 \vdash B_2 : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_3 + \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} B_1) \to B_2 : \mathsf{Type}_{l_1 \sqcup l_2}}\ \textsc{T\_Arrow}$$

In this case we know:

$$\begin{aligned}
&\sigma_1 = (\sigma_3 + \sigma_4)\\
&\sigma_2 = \mathbf{0}\\
&t = (x :_{(s,r)} B_1) \to B_2\\
&A = \mathsf{Type}_{l_1 \sqcup l_2}
\end{aligned}$$

However, by assumption we know that $(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_{l_1 \sqcup l_2} : \mathsf{Type}_1$, and hence, $\mathsf{Type}_{l_1 \sqcup l_2} = \mathsf{Type}_0$ which implies that $l_1 = l_2 = 0$. This all implies that $B_1, B_2 \in \mathsf{Type}$. Furthermore, we know that $(\Delta, \sigma_3) \odot (\Gamma, x : B_1) \models \varepsilon$, because $B_1 \in \mathsf{Type}$.

Thus, by Definition C.8 we must show that:

$$[\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon \in \mathcal{K}[\![\mathsf{Type}_0]\!] = \mathsf{SAT}$$

By Definition C.10 we know that:

$$[\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon = [\![B_1]\!]_\varepsilon \to [\![B_2]\!]_\varepsilon$$

By the IH:

IH(1): $[\![B_1]\!]_\varepsilon \in \mathcal{K}[\![\mathsf{Type}_0]\!] = \mathsf{SAT}$

IH(2): $[\![B_2]\!]_\varepsilon \in \mathcal{K}[\![\mathsf{Type}_0]\!] = \mathsf{SAT}$

Therefore, $[\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon = ([\![B_1]\!]_\varepsilon \to [\![B_2]\!]_\varepsilon) = (\mathsf{SAT} \to \mathsf{SAT}) \in \mathsf{SAT}$ which holds by the closer of SAT under function spaces.

Case 4:

$$\frac{(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, x : B_1 \vdash B_2 : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_3 + \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash (x :_r B_1) \otimes B_2 : \mathsf{Type}_{l_1 \sqcup l_2}} \text{ T\_TEN}$$

This case follows nearly exactly as the previous case, but ending with a cartesian product, $\mathsf{SAT} \times \mathsf{SAT}$, rather than the function space.

Case 5:

$$\frac{(\Delta, \sigma_3 \mid \sigma_4, s \mid \sigma_5, r) \odot \Gamma, x : B_1 \vdash t' : B_2}{(\Delta \mid \sigma_4 \mid \sigma_3 + \sigma_5) \odot \Gamma \vdash \lambda x : B_1.t' : (x :_{(s,r)} B_1) \to B_2} \text{ T\_FUN}$$

In this case we know:

$$\begin{aligned}&\sigma_1 = \sigma_4\\&\sigma_2 = \sigma_3 + \sigma_5\\&t = \lambda x : B_1.t'\\&A = (x :_{(s,r)} B_1) \to B_2\end{aligned}$$

We also know that $(\Delta \mid \sigma_3 + \sigma_5 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} B_1) \to B_2 : \mathsf{Type}_1$ by assumption. This implies by inversion that $B_1, B_2 \in \mathsf{Kind}$, $B_1 \in \mathsf{Kind}$ and $B_2 \in \mathsf{Type}$, or $B_1 \in \mathsf{Type}$ and $B_2 \in \mathsf{Kind}$. We consider each case in turn:

Subcase 1: Suppose $B_1, B_2 \in \mathsf{Kind}$. Then we must show that:

$$[\![\lambda x : B_1.t']\!]_\varepsilon \in \mathcal{K}[\![(x :_{(s,r)} B_1) \to B_2]\!] = \{f \mid f : \mathcal{K}[\![B_1]\!] \to \mathcal{K}[\![B_2]\!]\}$$

By Definition C.10 we know that:

$$[\![\lambda x : B_1.t']\!]_\varepsilon = \lambda X \in \mathcal{K}[\![B_1]\!].[\![t']\!]_{\varepsilon[x \mapsto X]}$$

Suppose $X \in \mathcal{K}[\![B_1]\!]$. Then we will show that $[\![t']\!]_{\varepsilon[x \mapsto X]} \in \mathcal{K}[\![B_2]\!]$. We know by assumption that $\Delta \odot \Gamma \models \varepsilon$ and $(\Delta, \sigma_3 \mid \sigma_4, s \mid \sigma_5, r) \odot \Gamma, x : B_1 \vdash t' : B_2$. Hence, by Lemma 3.5, we know that $(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_1$. Thus, we know that $(\Delta, \sigma_3) \odot (\Gamma, x : B_1) \models \varepsilon[x \mapsto X]$.

Therefore, by the IH:

IH(1): $[\![t']\!]_{\varepsilon[x \mapsto X]} \in \mathcal{K}[\![B_2]\!]$

which is what was to be shown.

Subcase 2: Suppose $B_1 \in \mathsf{Kind}$ and $B_2 \in \mathsf{Type}$. Then we must show that:

$$[\![\lambda x : B_1.t']\!]_\varepsilon \in \mathcal{K}[\![(x :_{(s,r)} B_1) \to B_2]\!] = \mathcal{K}[\![B_1]\!]$$

By Definition C.10 we know that:

$$[\![\lambda x : B_1.t']\!]_\varepsilon = \mathcal{K}[\![B_1]\!]$$

which is what was to be shown.

Subcase 3: Suppose $B_1 \in \mathsf{Type}$ and $B_2 \in \mathsf{Kind}$. Then we must show that:

$$[\![\lambda x : B_1.t']\!]_\varepsilon \in \mathcal{K}[\![(x :_{(s,r)} B_1) \to B_2]\!] = \mathcal{K}[\![B_2]\!]$$

By Definition C.10 we know that:

$$[\![\lambda x : B_1.t']\!]_\varepsilon = \mathcal{K}[\![B_2]\!]$$

which is what was to be shown.

Case 6:

$$\frac{(\Delta \mid \sigma_4 \mid \sigma_3 + \sigma_5) \odot \Gamma \vdash t_1 : (x :_{(s,r)} B_1) \to B_2 \quad (\Delta \mid \sigma_6 \mid \sigma_3) \odot \Gamma \vdash t_2 : B_1}{(\Delta \mid \sigma_4 + s * \sigma_6 \mid \sigma_5 + r * \sigma_6) \odot \Gamma \vdash t_1\, t_2 : [t_2/x]B_2} \text{ T\_App}$$

In this case we know that:

$$\begin{array}{l} \sigma_1 = \sigma_4 + s * \sigma_6 \\ \sigma_2 = \sigma_5 + r * \sigma_6 \\ t = t_1\, t_2 \\ A = [t_2/x]B_2 \end{array}$$

It suffices to show that:

$$[\![t_1\, t_2]\!]_\varepsilon \in \mathcal{K}[\![[t_2/x]B_2]\!] = \mathcal{K}[\![B_2]\!]$$

In this case we know by assumption that $[t_2/x]B_2 \in \mathsf{Kind}$ which implies that $B_2 \in \mathsf{Kind}$, and this further implies that $((x :_{(s,r)} B_1) \to B_2) \in \mathsf{Kind}$. However, there are two cases for B_1, either $B_1 \in \mathsf{Kind}$ or $B_1 \in \mathsf{Type}$. We consider each case in turn.

Subcase 1: Suppose $B_1 \in \mathsf{Kind}$. Then we know by assumption that $\Delta \odot \Gamma \models \varepsilon$. Thus, we apply the IH to obtain:

IH(1): $[\![t_1]\!]_\varepsilon \in \mathcal{K}[\![(x :_{(s,r)} B_1) \to B_2]\!] = \{f \mid f : \mathcal{K}[\![B_1]\!] \to \mathcal{K}[\![B_2]\!]\}$
IH(2): $[\![t_2]\!]_\varepsilon \in \mathcal{K}[\![B_1]\!]$

Using the IH's and Definition C.10 we know:

$$[\![t_1\, t_2]\!]_\varepsilon = [\![t_1]\!]_\varepsilon\, ([\![t_2]\!]_\varepsilon) \in \mathcal{K}[\![B_2]\!]$$

which was what was to be shown.

Subcase 2: Suppose $B_1 \in \mathsf{Type}$. Then we know by assumption that $\Delta \odot \Gamma \models \varepsilon$. Thus, we apply the IH to obtain:

IH(1): $[\![t_1]\!]_\varepsilon \in \mathcal{K}[\![(x :_{(s,r)} B_1) \to B_2]\!] = \mathcal{K}[\![B_2]\!]$

Using the IH's and Definition C.10 we know:

$$[\![t_1\, t_2]\!]_\varepsilon = [\![t_1]\!]_\varepsilon \in \mathcal{K}[\![B_2]\!]$$

which was what was to be shown.

Case 7:

$$\frac{\begin{array}{l} (\Delta, \sigma_3 \mid \sigma_5, r \mid \mathbf{0}) \odot \Gamma, x : B_1 \vdash B_2 : \mathsf{Type}_l \\ (\Delta \mid \sigma_4 \mid \sigma_3) \odot \Gamma \vdash t_1 : B_1 \\ (\Delta \mid \sigma_6 \mid \sigma_5 + r * \sigma_4) \odot \Gamma \vdash t_2 : [t_1/x]B_2 \end{array}}{(\Delta \mid \sigma_4 + \sigma_6 \mid \sigma_3 + \sigma_5) \odot \Gamma \vdash (t_1, t_2) : (x :_r B_1) \otimes B_2} \text{ T\_Pair}$$

In this case we know that:

$$\begin{array}{l} \sigma_1 = \sigma_4 + \sigma_6 \\ \sigma_2 = \sigma_3 + \sigma_5 \\ t = (t_1, t_2) \\ A = (x :_r B_1) \otimes B_2 \end{array}$$

It suffices to show that:

$$[\![(t_1, t_2)]\!]_\varepsilon \in \mathcal{K}[\![(x :_r B_1) \otimes B_2]\!]$$

By assumption we know that $((x :_r B_1) \otimes B_2) \in \mathsf{Kind}$. Thus, it must be the case that either $B_1, B_2 \in \mathsf{Kind}$, $B_1 \in \mathsf{Kind}$ and $B_2 \in \mathsf{Type}$, or $B_1 \in \mathsf{Type}$ and $B_2 \in \mathsf{Kind}$. We consider each of these cases in turn.

Subcase 1: Suppose $B_1, B_2 \in \mathsf{Kind}$ so $l = 1$. Then by Definition C.8 and Definition C.10 it suffices to show that:

$$\begin{aligned}[\![(t_1, t_2)]\!]_\varepsilon &= ([\![t_1]\!]_\varepsilon, [\![t_2]\!]_\varepsilon) \\ &\in \mathcal{K}[\![(x :_r B_1) \otimes B_2]\!] \\ &= \mathcal{K}[\![B_1]\!] \times \mathcal{K}[\![B_2]\!]\end{aligned}$$

We know by assumption that $\Delta \odot \Gamma \models \varepsilon$ and $(\Delta \mid \sigma_4 \mid \sigma_3) \odot \Gamma \vdash t_1 : B_1$. By kinding for typing (Lemma 3.6) we know that $(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_1$, and by assumption we know that $(\Delta, \sigma_3 \mid \sigma_5, r \mid \mathbf{0}) \odot (\Gamma, x : B_1) \vdash B_2 : \mathsf{Type}_1$. By substitution for typing (Lemma 3.27), we know that $(\Delta \mid \sigma_5 + r * \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash [t_1/x]B_2 : \mathsf{Type}_1$. We can now apply the IH to the premises for t_1 and t_2 using the fact that we know $(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_1$ and $(\Delta \mid \sigma_5 + r * \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash [t_1/x]B_2 : \mathsf{Type}_1$.

Thus, by the IH:

IH(1): $[\![t_1]\!]_\varepsilon \in \mathcal{K}[\![B_1]\!]$

IH(2): $[\![t_2]\!]_\varepsilon \in \mathcal{K}[\![[t_1/x]B_2]\!] = \mathcal{K}[\![B_2]\!]$

Therefore, $([\![t_1]\!]_\varepsilon, [\![t_2]\!]_\varepsilon) \in \mathcal{K}[\![B_1]\!] \times \mathcal{K}[\![B_2]\!]$.

Subcase 2: Suppose $B_1 \in \mathsf{Kind}$ and $B_2 \in \mathsf{Type}$. Then by Definition C.8 and Definition C.10 it suffices to show that:

$$\begin{aligned}[\![(t_1, t_2)]\!]_\varepsilon &= [\![t_1]\!]_\varepsilon \\ &\in \mathcal{K}[\![(x :_r B_1) \otimes B_2]\!] \\ &= \mathcal{K}[\![B_1]\!]\end{aligned}$$

We now by assumption that $\Delta \odot \Gamma \models \varepsilon$ and $(\Delta \mid \sigma_4 \mid \sigma_3) \odot \Gamma \vdash t_1 : B_1$. By kinding for typing (Lemma 3.6) we know that $(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_1$. Thus, by the IH:

IH(1): $[\![t_1]\!]_\varepsilon \in \mathcal{K}[\![B_1]\!]$

which is what was to be shown.

Subcase 3: Suppose $B_1 \in \mathsf{Type}$ and $B_2 \in \mathsf{Kind}$. Then by Definition C.8 and Definition C.10 it suffices to show that:

$$\begin{aligned}[\![(t_1, t_2)]\!]_\varepsilon &= [\![t_2]\!]_\varepsilon \\ &\in \mathcal{K}[\![(x :_r B_1) \otimes B_2]\!] \\ &= \mathcal{K}[\![B_2]\!]\end{aligned}$$

We know by assumption that $\Delta \odot \Gamma \models \varepsilon$ $(\Delta \mid \sigma_4 \mid \sigma_3) \odot \Gamma \vdash t_1 : B_1$, and $(\Delta, \sigma_3 \mid \sigma_5, r \mid \mathbf{0}) \odot (\Gamma, x : B_1) \vdash B_2 : \mathsf{Type}_1$. By substitution for typing (Lemma 3.27), we know that $(\Delta \mid \sigma_5 + r * \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash [t_1/x]B_2 : \mathsf{Type}_1$. We can now apply the IH to the premises for t_2 using the fact that we know $(\Delta \mid \sigma_5 + r * \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash [t_1/x]B_2 : \mathsf{Type}_1$.

Thus, by the IH:

IH(1): $[\![t_2]\!]_\varepsilon \in \mathcal{K}[\![[t_1/x]B_2]\!] = \mathcal{K}[\![B_2]\!]$

which is what was to be shown.

Case 8:

$$\frac{\begin{array}{l}(\Delta \mid \sigma_5 \mid \sigma_3 + \sigma_4) \odot \Gamma \vdash t_1 : (x :_r B_1) \otimes B_2 \\ (\Delta, (\sigma_3 + \sigma_4) \mid \sigma_7, r' \mid \mathbf{0}) \odot \Gamma, z : (x :_r B_1) \otimes B_2 \vdash C : \mathsf{Type}_1 \\ (\Delta, \sigma_3, (\sigma_4, r) \mid \sigma_6, s, s \mid \sigma_7, r', r') \odot \Gamma, x : B_1, y : B_2 \vdash t_2 : [(x, y)/z]C\end{array}}{(\Delta \mid \sigma_6 + s * \sigma_5 \mid \sigma_7 + r' * \sigma_5) \odot \Gamma \vdash \mathsf{let}\,(x : B_1, y : B_2) = t_1 \,\mathsf{in}\, t_2 : [t_1/z]C} \text{ T\_TenCut}$$

In this case we know that:

$$\begin{array}{l}\sigma_1 = \sigma_6 + s * \sigma_5\\ \sigma_2 = \sigma_7 + r' * \sigma_5\\ t = (\mathsf{let}\,(x : B_1, y : B_2) = t_1 \,\mathsf{in}\, t_2)\\ A = [t_1/z]C\end{array}$$

It suffices to show that:

$$[\![\mathsf{let}\,(x : B_1, y : B_2) = t_1 \,\mathsf{in}\, t_2]\!]_\varepsilon \in \mathcal{K}[\![[t_1/z]C]\!] = \mathcal{K}[\![C]\!]$$

Based on these assumptions, it must be the case that $C \in \mathsf{Kind}$ and $t_2 \in \mathsf{Type}$. First, we can conclude by kinding for typing that:

$$(\Delta, \sigma_3, (\sigma_4, r) \mid \sigma_{11} \mid \mathbf{0}) \odot (\Gamma, x : B_1, y : B_2) \vdash [(x, y)/z]C : \mathsf{Type}_1$$

for some vector σ_1. Then by well-formed contexts for typing we know that:

$$(\Delta, \sigma_3, (\sigma_4, r)) \odot (\Gamma, x : B_1, y : B_2) \vdash$$

This then implies that:

$$(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_{l_1} (\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot (\Gamma, x : B_1) \vdash B_2 : \mathsf{Type}_{l_2}$$

We will use these to apply the IH and define the typing evaluation below.

We must now consider cases for when $B_1, B_2 \in \mathsf{Kind}$, $B_1 \in \mathsf{Kind}$ and $B_2 \in \mathsf{Type}$, $B_1 \in \mathsf{Type}$ and $B_2 \in \mathsf{Kind}$, and $B_1, B_2 \in \mathsf{Type}$. We consider each case in turn.

Subcase 1: Suppose $B_1, B_2 \in \mathsf{Kind}$. Then we know:

$$\begin{array}{l}(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_1\\ (\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot (\Gamma, x : B_1) \vdash B_2 : \mathsf{Type}_1\\ (\Delta \mid \sigma_3 + \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash (x :_r B_1) \otimes B_2 : \mathsf{Type}_1\end{array}$$

By Definition C.8 and Definition C.10 it suffices to show:

$$\begin{array}{l}[\![\mathsf{let}\,(x : B_1, y : B_2) = t_1 \,\mathsf{in}\, t_2]\!]_\varepsilon\\ \quad = [\![t_2]\!]_{\varepsilon[x \mapsto \pi_1\, [\![t_1]\!]_\varepsilon, y \mapsto \pi_2\, [\![t_1]\!]_\varepsilon]}\\ \quad \in \mathcal{K}[\![[t_1/z]C]\!]\\ \quad = \mathcal{K}[\![C]\!]\end{array}$$

By the applying the IH to the premise for t_1 and Definition C.8:

IH(1): $[\![t_1]\!]_\varepsilon \in \mathcal{K}[\![(x :_r B_1) \otimes B_2]\!] = \mathcal{K}[\![B_1]\!] \times \mathcal{K}[\![B_2]\!]$

Hence, $\pi_1\, [\![t_1]\!]_\varepsilon \in \mathcal{K}[\![B_1]\!]$ and $\pi_2\, [\![t_1]\!]_\varepsilon \in \mathcal{K}[\![B_2]\!]$. This along with the kinding judgments given above imply that

$$(\Delta, \sigma_3, (\sigma_4, r)) \odot (\Gamma, x : B_1, y : B_2) \models \varepsilon[x \mapsto \pi_1\, [\![t_1]\!]_\varepsilon, y \mapsto \pi_2\, [\![t_1]\!]_\varepsilon]$$

given the assumption $\Delta \odot \Gamma \models \varepsilon$. We also know:

$$(\Delta, \sigma_3, (\sigma_4, r) \mid \sigma_{12} \mid \mathbf{0}) \odot (\Gamma, x : B_1, y : B_2) \vdash [(x, y)/z]C : \mathsf{Type}_1$$

by applying kinding for typing to the premise for t_2. We now have everything we need to apply the IH to the premise for t_2:

IH(2): $[\![t_2]\!]_{\varepsilon[x \mapsto \pi_1\, [\![t_1]\!]_\varepsilon, y \mapsto \pi_2\, [\![t_1]\!]_\varepsilon]} \in \mathcal{K}[\![[(x, y)/z]C]\!] = \mathcal{K}[\![C]\!]$

Now by Definition C.10 and the previous results:

$$\begin{aligned}&[\![\mathsf{let}\,(x:B_1,y:B_2)=t_1\,\mathsf{in}\,t_2]\!]_\varepsilon\\&\quad=[\![t_2]\!]_{\varepsilon[x\mapsto\pi_1\,[\![t_1]\!]_\varepsilon,y\mapsto\pi_2\,[\![t_1]\!]_\varepsilon]}\\&\quad\in K[|C|]\end{aligned}$$

which was what was to be shown.

Subcase 2: Suppose $B_1\in\mathsf{Kind}$ and $B_2\in\mathsf{Type}$. Then we know:

$$\begin{array}{l}(\Delta\mid\sigma_3\mid\mathbf{0})\odot\Gamma\vdash B_1:\mathsf{Type}_1\\(\Delta,\sigma_3\mid\sigma_4,r\mid\mathbf{0})\odot(\Gamma,x:B_1)\vdash B_2:\mathsf{Type}_0\\(\Delta\mid\sigma_3+\sigma_4\mid\mathbf{0})\odot\Gamma\vdash(x:_r B_1)\otimes B_2:\mathsf{Type}_1\end{array}$$

By Definition C.8 and Definition C.10 it suffices to show:

$$\begin{aligned}&[\![\mathsf{let}\,(x:B_1,y:B_2)=t_1\,\mathsf{in}\,t_2]\!]_\varepsilon\\&\quad=[\![t_2]\!]_{\varepsilon[x\mapsto[\![t_1]\!]_\varepsilon]}\\&\quad\in\mathcal{K}[\![[t_1/z]C]\!]\\&\quad=\mathcal{K}[\![C]\!]\end{aligned}$$

By the applying the IH to the premise for t_1 and Definition C.8:

IH(1): $[\![t_1]\!]_\varepsilon\in\mathcal{K}[\![(x:_r B_1)\otimes B_2]\!]=\mathcal{K}[\![B_1]\!]$

This along with the kinding judgments given above imply that

$$(\Delta,\sigma_3,(\sigma_4,r))\odot(\Gamma,x:B_1,y:B_2)\models\varepsilon[x\mapsto[\![t_1]\!]_\varepsilon]$$

given the assumption $\Delta\odot\Gamma\models\varepsilon$. We also know:

$$(\Delta,\sigma_3,(\sigma_4,r)\mid\sigma_{12}\mid\mathbf{0})\odot(\Gamma,x:B_1,y:B_2)\vdash[(x,y)/z]C:\mathsf{Type}_1$$

by applying kinding for typing to the premise for t_2. We now have everything we need to apply the IH to the premise for t_2:

IH(2): $[\![t_2]\!]_{\varepsilon[x\mapsto[\![t_1]\!]_\varepsilon]}\in\mathcal{K}[\![[(x,y)/z]C]\!]=\mathcal{K}[\![C]\!]$

Now by Definition C.10 and the previous results:

$$\begin{aligned}&[\![\mathsf{let}\,(x:B_1,y:B_2)=t_1\,\mathsf{in}\,t_2]\!]_\varepsilon\\&\quad=[\![t_2]\!]_{\varepsilon[x\mapsto[\![t_1]\!]_\varepsilon]}\\&\quad\in K[|C|]\end{aligned}$$

which was what was to be shown.

Subcase 3: Suppose $B_1\in\mathsf{Type}$ and $B_2\in\mathsf{Kind}$. This case is similar to the previous case using:

$$(\Delta,\sigma_3,(\sigma_4,r))\odot(\Gamma,x:B_1,y:B_2)\models\varepsilon[x\mapsto[\![t_1]\!]_\varepsilon]$$

Subcase 4: Suppose $B_1,B_2\in\mathsf{Type}$. This case is similar to the previous case using:

$$(\Delta,\sigma_3,(\sigma_4,r))\odot(\Gamma,x:B_1,y:B_2)\models\varepsilon$$

Case 9:

$$\frac{(\Delta\mid\sigma\mid\mathbf{0})\odot\Gamma\vdash B:\mathsf{Type}_0}{(\Delta\mid\sigma\mid\mathbf{0})\odot\Gamma\vdash\Box_s B:\mathsf{Type}_0}\ \text{T\_Box}$$

This case follows directly from the induction hypothesis.

Case 10:

$$\frac{(\Delta \mid \sigma_3 \mid \sigma_4) \odot \Gamma \vdash t' : B}{(\Delta \mid s * \sigma_3 \mid \sigma_4) \odot \Gamma \vdash \Box t' : \Box_s B}\ \text{T\_BoxI}$$

This case follows directly from the induction hypothesis.

Case 11:

$$\frac{\begin{array}{l}(\Delta \mid \sigma_3 \mid \sigma_7) \odot \Gamma \vdash t_1 : \Box_s B_1 \\ (\Delta, \sigma_7 \mid \sigma_5, s \mid \sigma_6, (s * r)) \odot \Gamma, x : B_1 \vdash t_2 : [\Box x/z]B_2 \\ (\Delta, \sigma_7 \mid \sigma_6, r \mid \mathbf{0}) \odot \Gamma, z : \Box_s B_1 \vdash B_2 : \mathsf{Type}_l\end{array}}{(\Delta \mid \sigma_3 + \sigma_5 \mid \sigma_6 + r * \sigma_3) \odot \Gamma \vdash \mathsf{let}\,\Box(x : B_1) = t_1 \,\mathsf{in}\, t_2 : [t_1/z]B_2}\ \text{T\_BoxE}$$

In this case we know that:

$$\begin{array}{l}\sigma_1 = \sigma_3 + \sigma_5 \\ \sigma_2 = \sigma_6 + r * \sigma_3 \\ t = (\mathsf{let}\,\Box(x : B_1) = t_1 \,\mathsf{in}\, t_2) \\ A = [t_1/z]B_2\end{array}$$

It suffices to show that:

$$[\![\mathsf{let}\,\Box(x : B_1) = t_1 \,\mathsf{in}\, t_2]\!]_\varepsilon \in \mathcal{K}[\![[t_1/z]B_2]\!] = \mathcal{K}[\![B_2]\!]$$

In this case we know that $([t_1/z]B_2) \in \mathsf{Kind}$, and thus, $B_2 \in \mathsf{Kind}$ and either $B_1 \in \mathsf{Kind}$ or $B_1 \in \mathsf{Type}$. We cover both of these cases in turn.

Subcase 1: Suppose $B_1 \in \mathsf{Kind}$. It suffices to show that:

$$\begin{array}{ll}[\![\mathsf{let}\,\Box(x : B_1) = t_1 \,\mathsf{in}\, t_2]\!]_\varepsilon & = [\![t_2]\!]_{\varepsilon[x \mapsto [\![t_1]\!]_\varepsilon]} \\ & \in \mathcal{K}[\![[t_1/z]B_2]\!] \\ & = \mathcal{K}[\![B_2]\!]\end{array}$$

As we have seen in the previous cases we can apply well-formed contexts for typing to obtain that:

$$(\Delta \mid \sigma_7 \mid \mathbf{0}) \odot \Gamma \vdash \Box_s B_1 : \mathsf{Type}_1$$

We can now apply the IH to the premise for t_1 to obtain:

IH(1): $[\![t_1]\!]_\varepsilon \in \mathcal{K}[\![\Box_s B_1]\!] = \mathcal{K}[\![B_1]\!]$

Using the previous two facts along with the assumption that $\Delta \odot \Gamma \models \varepsilon$ we may obtain

$$(\Delta, \sigma_7) \odot (\Gamma, x : B_1) \models \varepsilon[x \mapsto [\![t_1]\!]_\varepsilon]$$

In addition, we know that $(\Delta, \sigma_7 \mid \sigma_6, r \mid \mathbf{0}) \odot \Gamma, z : \Box_s B_1 \vdash B_2 : \mathsf{Type}_1$. Thus, we can now apply the induction hypothesis a second time.

IH(2): $[\![t_2]\!]_{\varepsilon[x \mapsto [\![t_1]\!]_\varepsilon]} \in \mathcal{K}[\![B_2]\!]$

which was what was to be shown.

Subcase 2: Suppose $B_1 \in \mathsf{Type}$. It suffices to show that:

$$\begin{array}{ll}[\![\mathsf{let}\,\Box(x : B_1) = t_1 \,\mathsf{in}\, t_2]\!]_\varepsilon & = [\![t_2]\!]_\varepsilon \\ & \in \mathcal{K}[\![[t_1/z]B_2]\!] \\ & = \mathcal{K}[\![B_2]\!]\end{array}$$

As we have seen in the previous cases we can apply well-formed contexts for typing to obtain that:

$$(\Delta \mid \sigma_7 \mid \mathbf{0}) \odot \Gamma \vdash \Box_s B_1 : \mathsf{Type}_0$$

Using this along with the assumption that $\Delta \odot \Gamma \models \varepsilon$ we may obtain

$$(\Delta, \sigma_7) \odot (\Gamma, x : B_1) \models \varepsilon$$

In addition, we know that $(\Delta, \sigma_7 \mid \sigma_6, r \mid \mathbf{0}) \odot \Gamma, z : \Box_s B_1 \vdash B_2 : \mathsf{Type}_1$. Thus, we can now apply the induction hypothesis a second time.

IH(2): $[\![t_2]\!]_\varepsilon \in \mathcal{K}[\![B_2]\!]$

which was what was to be shown.

Case 12:

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A' \quad (\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash A' = A : \mathsf{Type}_1}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A} \text{ T\_TyConv}$$

This case follows by first applying the induction hypothesis to the typing premise, and then applying Lemma C.15 to obtain that $\mathcal{K}[\![A]\!] = \mathcal{K}[\![A']\!]$ obtaining our result.

We now move onto the second part of this result assuming the first. In this part we will show:

If $\Delta \odot \Gamma \models \varepsilon$ and $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_1$, then $[\![A]\!]_\varepsilon \in \mathcal{K}[\![A]\!] \to \mathsf{SAT}$.

Recall that this is a proof by mutual induction on $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$ and $(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_1$.

Case 1:

$$\frac{\Delta \odot \Gamma \vdash}{(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_0 : \mathsf{Type}_1} \text{ T\_Type}$$

In this case we know that:

$$\sigma = \mathbf{0}$$
$$A = \mathsf{Type}_0$$

It suffices to show that:

$$[\![\mathsf{Type}_0]\!]_\varepsilon \in (\mathcal{K}[\![\mathsf{Type}_0]\!] \to \mathsf{SAT}) = (\mathsf{SAT} \to \mathsf{SAT})$$

But, $[\![\mathsf{Type}_0]\!]_\varepsilon = \lambda x \in \mathsf{SAT}.\mathsf{SN}$, and by Lemma C.7 $\mathsf{SN} \in \mathsf{SAT}$; hence, $(\lambda x \in \mathsf{SAT}.\mathsf{SN}) \in (\mathsf{SAT} \to \mathsf{SAT})$.

Case 2:

$$\frac{\Delta_1, \mathbf{0}, \Delta_2 \odot \Gamma_1, x : \mathsf{Type}_1, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \mathbf{0}, \Delta_2 \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \mathbf{0}) \odot (\Gamma_1, x : \mathsf{Type}_1, \Gamma_2) \vdash x : \mathsf{Type}_1} \text{ T\_Var}$$

This case is impossible, because the well-formed context premise fails, because Type_1 has no type.

Case 3:

$$\frac{\begin{array}{l}(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_{l_1} \\ (\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, x : B_1 \vdash B_2 : \mathsf{Type}_{l_2}\end{array}}{(\Delta \mid \sigma_3 + \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} B_1) \to B_2 : \mathsf{Type}_1} \text{ T\_Arrow}$$

In this case we know that:

$$\sigma = \sigma_3 + \sigma_4$$
$$A = (x :_{(s,r)} B_1) \to B_2$$

It suffices to show that:

$$[\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon \in (\mathcal{K}[\![(x :_{(s,r)} B_1) \to B_2]\!] \to \mathsf{SAT})$$

In this case either $B_1, B_2 \in \mathsf{Kind}$, $B_1 \in \mathsf{Kind}$ and $B_2 \in \mathsf{Type}$, or $B_1 \in \mathsf{Type}$ and $B_2 \in \mathsf{Kind}$. We consider each case in turn.

Subcase 1: Suppose $B_1, B_2 \in \mathsf{Kind}$. It suffices to show:

$$\begin{aligned}
&[\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon \\
&= \lambda X \in \mathcal{K}[\![B_1]\!] \to \mathcal{K}[\![B_2]\!].\bigcap_{Y \in \mathcal{K}[\![B_1]\!]}(([\![B_1]\!]_\varepsilon)\,(Y) \to ([\![B_2]\!]_{\varepsilon[x\mapsto Y]})\,(X\,(Y))) \\
&\in (\mathcal{K}[\![(x :_{(s,r)} B_1) \to B_2]\!] \to \mathsf{SAT}) \\
&= \{f \mid f : \mathcal{K}[\![B_1]\!] \to \mathcal{K}[\![B_2]\!]\} \to \mathsf{SAT}
\end{aligned}$$

Now suppose $X \in (\mathcal{K}[\![B_1]\!] \to \mathcal{K}[\![B_2]\!])$ and $Y \in \mathcal{K}[\![B_1]\!]$. We know by assumption that $(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_{l_1}$ and $\Delta \odot \Gamma \models \varepsilon$, and so we can apply the induction hypothesis to the premise for B_1 to obtain:

IH(1): $[\![B_1]\!]_\varepsilon \in (\mathcal{K}[\![B_1]\!] \to \mathsf{SAT})$

The previous facts now allow us, by Definition C.9, to obtain:

$$(\Delta, \sigma_3) \odot (\Gamma, x : B_1) \models \varepsilon[x \mapsto Y]$$

Thus, we can now apply the induction hypothesis to the premise for B_2 to obtain:

IH(2): $[\![B_2]\!]_{\varepsilon[x\mapsto Y]} \in (\mathcal{K}[\![B_2]\!] \to \mathsf{SAT})$

Then we know by IH(1) that $[\![B_1]\!]_\varepsilon\,(Y) \in \mathsf{SAT}$ and by IH(2) $[\![B_2]\!]_{\varepsilon[x\mapsto Y]}\,(X\,(Y)) \in \mathsf{SAT}$, thus:

$$([\![B_1]\!]_\varepsilon\,(Y) \to [\![B_2]\!]_{\varepsilon[x\mapsto Y]}\,(X\,(Y))) \in (\mathsf{SAT} \to \mathsf{SAT}) \in \mathsf{SAT}$$

Then by Lemma C.8:

$$\bigcap_{Y \in \mathcal{K}[\![B_1]\!]} ([\![B_1]\!]_\varepsilon\,(Y) \to [\![B_2]\!]_{\varepsilon[x\mapsto Y]}\,(X\,(Y))) \in \mathsf{SAT}$$

Therefore, we obtain our result.

Subcase 2: Suppose $B_1 \in \mathsf{Kind}$ and $B_2 \in \mathsf{Type}$. Similar to the previous case.

Subcase 3: Suppose $B_1 \in \mathsf{Type}$ and $B_2 \in \mathsf{Kind}$. It suffices to show:

$$\begin{aligned}
&[\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon \\
&= \lambda X \in \mathcal{K}[\![B_2]\!].([\![B_1]\!]_\varepsilon \to ([\![B_2]\!]_\varepsilon)\,(X)) \\
&\in (\mathcal{K}[\![(x :_{(s,r)} B_1) \to B_2]\!] \to \mathsf{SAT}) \\
&= (\mathcal{K}[\![B_2]\!] \to \mathsf{SAT})
\end{aligned}$$

Now suppose $X \in \mathcal{K}[\![B_2]\!]$. We know by assumption that $(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_0$ and $\Delta \odot \Gamma \models \varepsilon$, and so we can apply the first part of the induction hypothesis to the premise for B_1 to obtain:

IH(1): $[\![B_1]\!]_\varepsilon \in \mathcal{K}[\![\mathsf{Type}_0]\!] = \mathsf{SAT}$

Now we know by assumption that $(\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, x : B_1 \vdash B_2 : \mathsf{Type}_0$, and we can now show by Definition C.9 that $(\Delta, \sigma_3) \odot (\Gamma, x : B_1) \models \varepsilon$ holds. So we can apply the IH to the former judgment to obtain:

IH(2): $[\![B_2]\!]_\varepsilon \in (\mathcal{K}[\![B_2]\!] \to \mathsf{SAT})$

At this point we can see that $(\lambda X \in \mathcal{K}[\![B_2]\!].([\![B_1]\!]_\varepsilon \to [\![B_2]\!]_\varepsilon\,(X))) \in \mathsf{SAT}$ by the previous facts, and the fact that SAT is closed under function spaces.

Case 4:

$$\frac{\begin{array}{c}(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_{l_1} \\ (\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, x : B_1 \vdash B_2 : \mathsf{Type}_{l_2}\end{array}}{(\Delta \mid \sigma_3 + \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash (x :_r B_1) \otimes B_2 : \mathsf{Type}_{l_1 \sqcup l_2}}\ \text{T\_Ten}$$

This case is similar to the previous case.

Case 5:

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash B : \mathsf{Type}_1}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_{s} B : \mathsf{Type}_1} \text{ T\_Box}$$

This case follows from the IH.

Case 6:

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash A : A' \quad (\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash A' = \mathsf{Type}_1 : B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash A : \mathsf{Type}_1} \text{ T\_TyConv}$$

This case is impossible, because Type_1 has no type B.

$\square$

Theorem C.17 (Soundness for Semantic Typing)**.** *If* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$*, then* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \models t : A$*.*

Proof. This is a proof by induction on the assumed typing derivation.

Case 1:

$$\frac{\Delta \odot \Gamma \vdash}{(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_0 : \mathsf{Type}_1} \text{ T\_Type}$$

In this case we have:

$$\begin{array}{l} \sigma_1 = \mathbf{0} \\ \sigma_2 = \mathbf{0} \\ t = \mathsf{Type}_0 \\ A = \mathsf{Type}_1 \end{array}$$

We can now see that this case holds trivially, because Type_1 has no type.

Case 2:

$$\frac{\Delta_1, \sigma, \Delta_2 \odot \Gamma_1, x : A, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \sigma, \Delta_2 \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \sigma, 0, \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash x : A} \text{ T\_Var}$$

In this case we have:

$$\begin{array}{l} \Delta = (\Delta_1, \sigma, \Delta_2) \\ \sigma_1 = \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \\ \sigma_2 = \sigma, 0, \mathbf{0} \\ t = x \end{array}$$

Now either $A \in \mathsf{Kind}$ or $A \in \mathsf{Type}$. We consider both cases in turn.

Subcase 1: Suppose $A \in \mathsf{Kind}$ and $\Delta \odot \Gamma \models_{\varepsilon} \rho$. It suffices to show:

$$(\!|x|\!)_{\rho} = \rho\, x \in [\![A]\!]_{\varepsilon}\, ([\![x]\!]_{\varepsilon}) = [\![A]\!]_{\varepsilon}\, (\varepsilon\, x)$$

But, this holds by Definition C.12, because the well-formed context premise above implies the proper kinding of A.

Subcase 2: Suppose $A \in \mathsf{Type}$ and $\Delta \odot \Gamma \models_{\varepsilon} \rho$. It suffices to show:

$$(\!|x|\!)_{\rho} = \rho\, x \in [\![A]\!]_{\varepsilon}$$

But, this holds by Definition C.12, because the well-formed context premise above implies the proper kinding of A.

Case 3:

$$\frac{\begin{array}{c}(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_0 \\ (\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, x : B_1 \vdash B_2 : \mathsf{Type}_0\end{array}}{(\Delta \mid \sigma_3 + \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} B_1) \to B_2 : \mathsf{Type}_0} \text{T\_Arrow}$$

In this case we have:

$$\begin{aligned}&\sigma_1 = (\sigma_3 + \sigma_4)\\&\sigma_2 = \mathbf{0}\\&t = (x :_{(s,r)} B_1) \to B_2\\&A = \mathsf{Type}_0\end{aligned}$$

We only need to consider the first case of this theorem, because Type_1 has no type. So suppose $\Delta \odot \Gamma \models_\varepsilon \rho$. It suffices to show:

$$\begin{aligned}&(\!|(x :_{(s,r)} B_1) \to B_2|\!)_\rho\\&= ((x :_{(s,r)} (\!|B_1|\!)_\rho) \to (\!|B_2|\!)_\rho)\\&\in [\![\mathsf{Type}_0]\!]_\varepsilon\,([\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon)\\&= \mathsf{SN}\end{aligned}$$

We know by assumption that $\Delta \odot \Gamma \models_\varepsilon \rho$ so we can apply the IH to conclude:

IH(1): $(\!|B_1|\!)_\rho \in [\![\mathsf{Type}_0]\!]_\varepsilon\,([\![B_1]\!]_\varepsilon) = \mathsf{SN}$

Now suppose $t \in [\![B_1]\!]_\varepsilon$. Then we know by Definition C.12 that $(\Delta, \sigma_3) \odot (\Gamma, x : B_1) \models_\varepsilon \rho[x \mapsto t]$. Thus, by applying the IH to the premise for B_2 we may conclude that

IH(2): $(\!|B_2|\!)_{\rho[x \mapsto t]} \in [\![\mathsf{Type}_0]\!]_\varepsilon\,([\![B_2]\!]_\varepsilon) = \mathsf{SN}$

holds for every t. Therefore, we may conclude out result.

Case 4:

$$\frac{\begin{array}{c}(\Delta \mid \sigma_3 \mid \mathbf{0}) \odot \Gamma \vdash B_1 : \mathsf{Type}_0 \\ (\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, x : B_1 \vdash B_2 : \mathsf{Type}_0\end{array}}{(\Delta \mid \sigma_3 + \sigma_4 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{r} B_1) \otimes B_2 : \mathsf{Type}_0} \text{T\_Ten}$$

Similar to the previous case.

Case 5:

$$\frac{(\Delta, \sigma_3 \mid \sigma_4, s \mid \sigma_5, r) \odot \Gamma, x : B_1 \vdash t' : B_2}{(\Delta \mid \sigma_4 \mid \sigma_3 + \sigma_5) \odot \Gamma \vdash \lambda x : B_1.t' : (x :_{(s,r)} B_1) \to B_2} \text{T\_Fun}$$

In this case we have:

$$\begin{aligned}&\sigma_1 = \sigma_4\\&\sigma_2 = (\sigma_3 + \sigma_5)\\&t = \lambda x : B_1.t'\\&A = (x :_{(s,r)} B_1) \to B_2\end{aligned}$$

We have two cases to consider, either $((x :_{(s,r)} B_1) \to B_2) \in \mathsf{Kind}$ or $((x :_{(s,r)} B_1) \to B_2) \in \mathsf{Type}$. We cover both cases in turn.

Subcase 1: Suppose $((x :_{(s,r)} B_1) \to B_2) \in \mathsf{Kind}$. We now have three subcases depending on the typing for both B_1 and B_2.

Subcase 1: Suppose $B_1, B_2 \in \mathsf{Kind}$. It suffices to show:

$$\begin{aligned}&(\!|\lambda x : B_1.t'|\!)_\rho\\&= \lambda x : (\!|B_1|\!)_\rho.(\!|t'|\!)_\rho\\&\in [\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon\,([\![\lambda x : B_1.t']\!]_\varepsilon)\\&= \bigcap_{Y \in \mathcal{K}[\![B_1]\!]}(([\![B_1]\!]_\varepsilon)\,(Y) \to ([\![B_2]\!]_{\varepsilon[x \mapsto Y]})\,([\![\lambda x : B_1.t']\!]_\varepsilon\,(Y)))\\&= \bigcap_{Y \in \mathcal{K}[\![B_1]\!]}(([\![B_1]\!]_\varepsilon)\,(Y) \to ([\![B_2]\!]_{\varepsilon[x \mapsto Y]})\,([\![t']\!]_{\varepsilon[x \mapsto Y]}))\end{aligned}$$

So suppose we have a $Y \in \mathcal{K}[\![B_1]\!]$ and a $t \in [\![B_1]\!]_\varepsilon (Y) = [\![B_1]\!]_{(\varepsilon[x \mapsto Y])} (\varepsilon\, x)$ (since $x \notin \mathsf{FV}(B_2)$). Then we know that $(\Delta, \sigma_3) \odot (\Gamma, x : B_1) \models_{\varepsilon[x \mapsto Y]} \rho[x \mapsto t]$. Then by the IH:

IH(1): $(\!| t' |\!)_{\rho[x \mapsto t]} \in [\![B_2]\!]_{\varepsilon[x \mapsto Y]} ([\![t']\!]_{\varepsilon[x \mapsto Y]})$

Thus, we obtain our result.

Subcase 2: Suppose $B_1 \in \mathsf{Kind}$ and $B_2 \in \mathsf{Type}$. This case is similar to the previous case, except we will use part two of the IH.

Subcase 3: Suppose $B_1 \in \mathsf{Type}$ and $B_2 \in \mathsf{Kind}$. Similar to the previous case.

Subcase 2: Suppose $((x :_{(s,r)} B_1) \to B_2) \in \mathsf{Type}$. This case is similar to the above, but we will use the second part of the IH.

Case 6:

$$\frac{\begin{array}{c}(\Delta \mid \sigma_4 \mid \sigma_3 + \sigma_5) \odot \Gamma \vdash t_1 : (x :_{(s,r)} B_1) \to B_2 \\ (\Delta \mid \sigma_6 \mid \sigma_3) \odot \Gamma \vdash t_2 : B_1\end{array}}{(\Delta \mid \sigma_4 + s * \sigma_6 \mid \sigma_5 + r * \sigma_6) \odot \Gamma \vdash t_1\, t_2 : [t_2/x]B_2} \text{T\_App}$$

In this case we have:

$$\begin{array}{l}\sigma_1 = (\sigma_4 + s * \sigma_6) \\ \sigma_2 = (\sigma_5 + r * \sigma_6) \\ t = (t_1\, t_2) \\ A = [t_2/x]B_2\end{array}$$

We have several cases to consider.

Subcase 1: Suppose $B_1, [t_2/x]B_2 \in \mathsf{Kind}$. It suffices to show:

$$\begin{array}{l}(\!| t_1\, t_2 |\!)_\rho \\ = (\!| t_1 |\!)_\rho\, (\!| t_2 |\!)_\rho \\ \in [\![[t_2/x]B_2]\!]_\varepsilon ([\![t_1\, t_2]\!]_\varepsilon) \\ = [\![B_2]\!]_{\varepsilon[x \mapsto [\![t_2]\!]_\varepsilon]} ([\![t_1]\!]_\varepsilon ([\![t_2]\!]_\varepsilon))\end{array}$$

By the IH we have:

IH(1): $(\!| t_1 |\!)_\rho \in [\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon ([\![t_1]\!]_\varepsilon)$

IH(2): $(\!| t_2 |\!)_\rho \in [\![B_1]\!]_\varepsilon ([\![t_2]\!]_\varepsilon)$

Notice that by Lemma C.16 we know that $[\![t_2]\!]_\varepsilon \in \mathcal{K}[\![B_1]\!]$. So using C.10 we know:

$$\begin{array}{rl}(\!| t_1 |\!)_\rho & \in [\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon ([\![t_1]\!]_\varepsilon) \\ = & \bigcap_{Y \in \mathcal{K}[\![A]\!]} (([\![A]\!]_\varepsilon)(Y) \to ([\![B]\!]_{\varepsilon[x \mapsto Y]})([\![t_1]\!]_\varepsilon (Y)))\end{array}$$

Thus, $(\!| t_1 |\!)_\rho\, (\!| t_2 |\!)_\rho \in ([\![B]\!]_{\varepsilon[x \mapsto [\![t_2]\!]_\varepsilon]})([\![t_1]\!]_\varepsilon ([\![t_2]\!]_\varepsilon))$.

Subcase 2: Suppose $B_1 \in \mathsf{Kind}$ and $[t_2/x]B_2 \in \mathsf{Type}$. It suffices to show:

$$\begin{array}{l}(\!| t_1\, t_2 |\!)_\rho \\ = (\!| t_1 |\!)_\rho\, (\!| t_2 |\!)_\rho \\ \in [\![[t_2/x]B_2]\!]_\varepsilon \\ = [\![B_2]\!]_{\varepsilon[x \mapsto [\![t_2]\!]_\varepsilon]}\end{array}$$

By the IH we have:

IH(1): $(\!| t_1 |\!)_\rho \in [\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon ([\![t_1]\!]_\varepsilon)$

IH(2): $(\!| t_2 |\!)_\rho \in [\![B_1]\!]_\varepsilon ([\![t_2]\!]_\varepsilon)$

Notice that by Lemma C.16 we know that $[\![t_2]\!]_\varepsilon \in \mathcal{K}[\![B_1]\!]$. So using C.10 we know:

$$\begin{array}{l} (\!|t_1|\!)_\rho \in [\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon\, ([\![t_1]\!]_\varepsilon) \\ = \bigcap_{Y \in \mathcal{K}[\![B_1]\!]} (([\![B_1]\!]_\varepsilon)\,(Y) \to ([\![B_2]\!]_{\varepsilon[x \mapsto Y]})) \end{array}$$

Thus, $(\!|t_1|\!)_\rho\, (\!|t_2|\!)_\rho \in ([\![B_2]\!]_{\varepsilon[x \mapsto [\![t_2]\!]_\varepsilon]})$.

Subcase 3: Suppose $B_1 \in \mathsf{Type}$ and $[t_2/x]B_2 \in \mathsf{Type}$. It suffices to show:

$$\begin{array}{l} (\!|t_1\, t_2|\!)_\rho \\ = (\!|t_1|\!)_\rho\, (\!|t_2|\!)_\rho \\ \in [\![[t_2/x]B_2]\!]_\varepsilon \\ = [\![B_2]\!]_{\varepsilon[x \mapsto [\![t_2]\!]_\varepsilon]} \end{array}$$

By the IH we have:

IH(1): $(\!|t_1|\!)_\rho \in [\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon$

IH(2): $(\!|t_2|\!)_\rho \in [\![B_1]\!]_\varepsilon$

Using C.10 we know:

$$\begin{array}{l} (\!|t_1|\!)_\rho \in [\![(x :_{(s,r)} B_1) \to B_2]\!]_\varepsilon\, ([\![t_1]\!]_\varepsilon) \\ = [\![B_1]\!]_\varepsilon \to [\![B_2]\!]_\varepsilon \end{array}$$

Thus, $(\!|t_1|\!)_\rho\, (\!|t_2|\!)_\rho \in [\![B_2]\!]_\varepsilon$.

Case 7:

$$\frac{\begin{array}{c} (\Delta, \sigma_3 \mid \sigma_5, r \mid \mathbf{0}) \odot \Gamma, x : B_1 \vdash B_2 : \mathsf{Type}_l \\ (\Delta \mid \sigma_4 \mid \sigma_3) \odot \Gamma \vdash t_1 : B_1 \\ (\Delta \mid \sigma_6 \mid \sigma_5 + r * \sigma_4) \odot \Gamma \vdash t_2 : [t_1/x]B_2 \end{array}}{(\Delta \mid \sigma_4 + \sigma_6 \mid \sigma_3 + \sigma_5) \odot \Gamma \vdash (t_1, t_2) : (x :_r B_1) \otimes B_2}\ \text{T\_Pair}$$

This case is similar to the case for λ-abstraction above.

Case 8:

$$\frac{\begin{array}{c} (\Delta \mid \sigma_5 \mid \sigma_3 + \sigma_4) \odot \Gamma \vdash t_1 : (x :_r B_1) \otimes B_2 \\ (\Delta, (\sigma_3 + \sigma_4) \mid \sigma_7, r' \mid \mathbf{0}) \odot \Gamma, z : (x :_r B_1) \otimes B_2 \vdash C : \mathsf{Type}_1 \\ (\Delta, \sigma_3, (\sigma_4, r) \mid \sigma_6, s, s \mid \sigma_7, r', r') \odot \Gamma, x : B_1, y : B_2 \vdash t_2 : [(x,y)/z]C \end{array}}{(\Delta \mid \sigma_6 + s * \sigma_5 \mid \sigma_7 + r' * \sigma_5) \odot \Gamma \vdash \mathsf{let}\,(x : B_1, y : B_2) = t_1\,\mathsf{in}\,t_2 : [t_1/z]C}\ \text{T\_TenCut}$$

Similar to the application case above.

Case 9:

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash B : \mathsf{Type}_0}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s B : \mathsf{Type}_0}\ \text{T\_Box}$$

This case follows from the IH.

Case 10:

$$\frac{(\Delta \mid \sigma_3 \mid \sigma_2) \odot \Gamma \vdash t' : B}{(\Delta \mid s * \sigma_3 \mid \sigma_2) \odot \Gamma \vdash \Box t' : \Box_s B}\ \text{T\_BoxI}$$

In this case we have:

$$\begin{array}{l} \sigma_1 = (s * \sigma_3) \\ t = (\Box t') \\ A = \Box_s B \end{array}$$

We have two cases to consider.

Subcase 1: Suppose $B \in \mathsf{Kind}$. It suffices to show:

$$\begin{aligned} & (\!|\Box t'|\!)_\rho \\ = \ & (\!|t'|\!)_\rho \\ \in \ & [\![\Box_s B]\!]_\varepsilon\,([\![\Box t']\!]_\varepsilon) \\ = \ & [\![B]\!]_\varepsilon\,([\![t']\!]_\varepsilon) \end{aligned}$$

At this point, this case holds by the IH.

Subcase 2: Suppose $B \in \mathsf{Type}$. Similar to the previous case.

Case 11:

$$\frac{\begin{array}{l}(\Delta \mid \sigma_3 \mid \sigma_7) \odot \Gamma \vdash t_1 : \Box_s B_1 \\ (\Delta, \sigma_7 \mid \sigma_6, r \mid \mathbf{0}) \odot \Gamma, z : \Box_s B_1 \vdash B_2 : \mathsf{Type}_0 \\ (\Delta, \sigma_7 \mid \sigma_5, s \mid \sigma_6, (s * r)) \odot \Gamma, x : B_1 \vdash t_2 : [\Box x/z]B_2\end{array}}{(\Delta \mid \sigma_3 + \sigma_5 \mid \sigma_6 + r * \sigma_3) \odot \Gamma \vdash \mathsf{let}\,\Box(x : B_1) = t_1\,\mathsf{in}\,t_2 : [t_1/z]B_2}\ \text{T\_BoxE}$$

In this case we have:

$$\begin{aligned} & \sigma_1 = (\sigma_3 + \sigma_5) \\ & \sigma_2 = (\sigma_6 + r * \sigma_3) \\ & t = (\mathsf{let}\,\Box(x : B_1) = t_1\,\mathsf{in}\,t_2) \\ & A = [t_1/z]B_2 \end{aligned}$$

We have several cases to consider:

Subcase 1: Suppose $B_1, B_2 \in \mathsf{Kind}$. It suffices to show:

$$\begin{aligned} & (\!|\mathsf{let}\,\Box(x : B_1) = t_1\,\mathsf{in}\,t_2|\!)_\rho \\ = \ & (\!|t_2|\!)_{\rho[x \mapsto (\!|t_1|\!)_\rho]} \\ \in \ & [\![[t_1/z]B_2]\!]_\varepsilon\,([\![\mathsf{let}\,\Box(x : B_1) = t_1\,\mathsf{in}\,t_2]\!]_\varepsilon) \\ = \ & [\![B_2]\!]_{\varepsilon[x \mapsto [\![t_1]\!]_\varepsilon]}\,([\![t_2]\!]_{\varepsilon[x \mapsto [\![t_1]\!]_\varepsilon]}) \end{aligned}$$

By the IH:

IH(1): $(\!|t_1|\!)_\rho \in [\![\Box_s B_1]\!]_\varepsilon\,([\![t_1]\!]_\varepsilon) = [\![B_1]\!]_\varepsilon\,([\![t_1]\!]_\varepsilon)$

At this point we need:

$$(\Delta, \sigma_7) \odot (\Gamma, x : B_1) \models_{\varepsilon[x \mapsto [\![t_1]\!]_\varepsilon]} \rho[x \mapsto (\!|t_1|\!)_\rho]$$

But, this follows by definition and Lemma C.16.

By the IH:

IH(2): $(\!|t_2|\!)_{\rho[x \mapsto (\!|t_1|\!)_\rho]} \in [\![B_2]\!]_{\varepsilon[x \mapsto [\![t_1]\!]_\varepsilon]}\,([\![t_2]\!]_{\varepsilon[x \mapsto [\![t_1]\!]_\varepsilon]})$

Subcase 2: Suppose $B_1 \in \mathsf{Kind}$ and $B_2 \in \mathsf{Type}$. Similar to the previous case.

Subcase 3: Suppose $B_1 \in \mathsf{Type}$ and $B_2 \in \mathsf{Kind}$. It suffices to show:

$$\begin{aligned} & (\!|\mathsf{let}\,\Box(x : B_1) = t_1\,\mathsf{in}\,t_2|\!)_\rho \\ = \ & (\!|t_2|\!)_{\rho[x \mapsto (\!|t_1|\!)_\rho]} \\ \in \ & [\![[t_1/z]B_2]\!]_\varepsilon\,([\![\mathsf{let}\,\Box(x : B_1) = t_1\,\mathsf{in}\,t_2]\!]_\varepsilon) \\ = \ & [\![B_2]\!]_\varepsilon\,([\![t_2]\!]_\varepsilon) \end{aligned}$$

By the IH:

IH(1): $(\!|t_1|\!)_\rho \in [\![\Box_s B_1]\!]_\varepsilon = [\![B_1]\!]_\varepsilon$

At this point we need:

$$(\Delta, \sigma_7) \odot (\Gamma, x : B_1) \models_\varepsilon \rho[x \mapsto (\!|t_1|\!)_\rho]$$

But, this follows by definition.

By the IH:

IH(2): $(\!|t_2|\!)_{\rho[x \mapsto (\!|t_1|\!)_\rho]} \in [\![B_2]\!]_\varepsilon\,([\![t_2]\!]_\varepsilon)$

Subcase 4: Suppose $B_1, B_2 \in \mathsf{Type}$. Similar to the previous case.

Case 12:

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A' \quad (\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash A' = A : \mathsf{Type}_l}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A} \text{ T\_TyConv}$$

By the induction hypothesis we know $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \models t : A'$ holds, but this implies $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \models t : A'$ holds by Lemma C.15.

□

Corollary C.17.1 (Strong Normalization)**.** *For every* $(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A$, $t \in \mathsf{SN}$.

Proof. Similarly to CC, we can define a notion of canonical element in $\mathcal{K}[\![A]\!]$, and define a term valuation $\Delta \odot \Gamma \models_\varepsilon \rho$, and then conclude SN by the previous theorem. □

D Graded Modal Dependent Type Theory complete specification

$\boxed{\Delta \odot \Gamma \models_\varepsilon \rho}$ Valid Term Substitutions

$$\frac{}{\emptyset \odot \emptyset \models_\emptyset \emptyset} \text{ Rho\_Empty}$$

$$\frac{t \in [\![A]\!]_\varepsilon \quad \Delta \odot \Gamma \models_\varepsilon \rho \quad (\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_0}{(\Delta, \sigma_2) \odot (\Gamma, x : A) \models_\varepsilon \rho[x \mapsto t]} \text{ Rho\_ExtTm}$$

$$\frac{t \in ([\![A]\!]_\varepsilon)\,(\varepsilon\, x) \quad \Delta \odot \Gamma \models_\varepsilon \rho \quad (\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_1}{(\Delta, \sigma_2) \odot (\Gamma, x : A) \models_\varepsilon \rho[x \mapsto t]} \text{ Rho\_ExtTy}$$

$\boxed{\Delta \odot \Gamma \models \varepsilon}$ Valid Type Substitutions

$$\frac{}{\emptyset \odot \emptyset \models \emptyset} \text{ Ep\_Empty}$$

$$\frac{\Delta \odot \Gamma \models \varepsilon \quad (\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_0}{(\Delta, \sigma_2) \odot (\Gamma, x : A) \models \varepsilon} \text{ Ep\_ExtTm}$$

$$\frac{X \in \mathcal{K}[\![A]\!] \quad \Delta \odot \Gamma \models \varepsilon \quad (\Delta \mid \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_1}{(\Delta, \sigma_2) \odot (\Gamma, x : A) \models \varepsilon[x \mapsto X]} \text{ Ep\_ExtTy}$$

$\boxed{\Delta \odot \Gamma \vdash}$ Well-formed contexts

$$\frac{}{\emptyset \odot \emptyset \vdash} \text{ Wf\_Empty}$$

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l}{\Delta, \sigma \odot \Gamma, x : A \vdash} \text{ Wf\_Ext}$$

$\boxed{\Gamma \vdash t_1 \leq t_2}$ Subtyping

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A = B : \mathsf{Type}_l}{(\Delta \mid \sigma) \odot \Gamma \vdash A \leq B} \text{ ST\_Eq}$$

$$\frac{(\Delta \mid \sigma) \odot \Gamma \vdash A \leq B \quad (\Delta \mid \sigma) \odot \Gamma \vdash B \leq C}{(\Delta \mid \sigma) \odot \Gamma \vdash A \leq C}\ \textsc{ST\_Trans}$$

$$\frac{\Delta \odot \Gamma \vdash \quad l \leq l'}{(\Delta \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_l \leq \mathsf{Type}_{l'}}\ \textsc{ST\_Ty}$$

$$\frac{\begin{array}{l}(\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_1) \odot \Gamma \vdash A' \leq A \quad (\Delta, \sigma_1 \mid \sigma_2, r) \odot \Gamma, x : A' \vdash B \leq B'\end{array}}{(\Delta \mid \sigma_1 + \sigma_2) \odot \Gamma \vdash (x :_{(s,r)} A) \to B \leq (x :_{(s,r)} A') \to B'}\ \textsc{ST\_Arrow}$$

$$\frac{(\Delta, \sigma_1 \mid \sigma_2, r) \odot \Gamma, x : A \vdash B \leq B'}{(\Delta \mid \sigma_1 + \sigma_2) \odot \Gamma \vdash (x :_r A) \otimes B \leq (x :_r A) \otimes B'}\ \textsc{ST\_Ten}$$

$$\frac{(\Delta \mid \sigma) \odot \Gamma \vdash A \leq A'}{(\Delta \mid \sigma) \odot \Gamma \vdash \Box_s A \leq \Box_s A'}\ \textsc{ST\_Box}$$

$\boxed{\Gamma \vdash t_1 : t_2}$ Typing

$$\frac{\Delta \odot \Gamma \vdash}{(\Delta \mid \mathbf{0} \mid \mathbf{0}) \odot \Gamma \vdash \mathsf{Type}_l : \mathsf{Type}_{\mathsf{suc}\ l}}\ \textsc{T\_Type}$$

$$\frac{\Delta_1, \sigma, \Delta_2 \odot \Gamma_1, x : A, \Gamma_2 \vdash \quad |\Delta_1| = |\Gamma_1|}{(\Delta_1, \sigma, \Delta_2 \mid \mathbf{0}^{|\Delta_1|}, 1, \mathbf{0} \mid \sigma, 0, \mathbf{0}) \odot \Gamma_1, x : A, \Gamma_2 \vdash x : A}\ \textsc{T\_Var}$$

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B : \mathsf{Type}_{l_1 \sqcup l_2}}\ \textsc{T\_Arrow}$$

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_r A) \otimes B : \mathsf{Type}_{l_1 \sqcup l_2}}\ \textsc{T\_Ten}$$

$$\frac{(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \quad (\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t : B}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t : (x :_{(s,r)} A) \to B}\ \textsc{T\_Fun}$$

$$\frac{\begin{array}{l}(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash t_1 : (x :_{(s,r)} A) \to B \quad (\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : A\end{array}}{(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash t_1\, t_2 : [t_2/x]B}\ \textsc{T\_App}$$

$$\frac{\begin{array}{l}(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma \vdash t_1 : A \quad (\Delta \mid \sigma_4 \mid \sigma_3 + r * \sigma_2) \odot \Gamma \vdash t_2 : [t_1/x]B\end{array}}{(\Delta \mid \sigma_2 + \sigma_4 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash (t_1, t_2) : (x :_r A) \otimes B}\ \textsc{T\_Pair}$$

$$\frac{\begin{array}{l}(\Delta \mid \sigma_3 \mid \sigma_1 + \sigma_2) \odot \Gamma \vdash t_1 : (x :_r A) \otimes B \\ (\Delta, (\sigma_1 + \sigma_2) \mid \sigma_5, r' \mid \mathbf{0}) \odot \Gamma, z : (x :_r A) \otimes B \vdash C : \mathsf{Type}_l \\ (\Delta, \sigma_1, (\sigma_2, r) \mid \sigma_4, s, s \mid \sigma_5, r', r') \odot \Gamma, x : A, y : B \vdash t_2 : [(x, y)/z]C\end{array}}{(\Delta \mid \sigma_4 + s * \sigma_3 \mid \sigma_5 + r' * \sigma_3) \odot \Gamma \vdash \mathsf{let}\,(x, y) = t_1\,\mathsf{in}\,t_2 : [t_1/z]C}\ \textsc{T\_TenCut}$$

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A : \mathsf{Type}_l}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A : \mathsf{Type}_l}\ \textsc{T\_Box}$$

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A}{(\Delta \mid s * \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box t : \Box_s A}\ \textsc{T\_BoxI}$$

$$\frac{\begin{array}{l}(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : \Box_s A \\ (\Delta, \sigma_2 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, z : \Box_s A \vdash B : \mathsf{Type}_l \\ (\Delta, \sigma_2 \mid \sigma_3, s \mid \sigma_4, (s * r)) \odot \Gamma, x : A \vdash t_2 : [\Box x/z]B\end{array}}{(\Delta \mid \sigma_1 + \sigma_3 \mid \sigma_4 + r * \sigma_1) \odot \Gamma \vdash \mathsf{let}\,\Box x = t_1\,\mathsf{in}\,t_2 : [t_1/z]B}\ \textsc{T\_BoxE}$$

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash A \leq B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : B} \textsc{T\_TyConv}$$

$\boxed{t_1 \rightsquigarrow t_2}$ Reduction rules

$$\frac{}{(\lambda x.t_1)\, t_2 \rightsquigarrow [t_2/x]t_1} \textsc{Sem\_BetaFun}$$

$$\frac{}{\mathsf{let}\,(x, y) = (t_1, t_2)\,\mathsf{in}\,t_3 \rightsquigarrow [t_1/x][t_2/y]t_3} \textsc{Sem\_BetaTen}$$

$$\frac{}{\mathsf{let}\,\Box x = \Box t_1\,\mathsf{in}\,t_2 \rightsquigarrow [t_1/x]t_2} \textsc{Sem\_BetaBox}$$

$$\frac{t_1 \rightsquigarrow t_1'}{(t_1, t_2) \rightsquigarrow (t_1', t_2)} \textsc{Sem\_CongPair}$$

$$\frac{t_2 \rightsquigarrow t_2'}{(t_1, t_2) \rightsquigarrow (t_1, t_2')} \textsc{Sem\_CongPairTwo}$$

$$\frac{t_1 \rightsquigarrow t_1'}{t_1\, t_2 \rightsquigarrow t_1'\, t_2} \textsc{Sem\_CongFunOne}$$

$$\frac{t_1 \rightsquigarrow t_1'}{\mathsf{let}\,(x, y) = t_1\,\mathsf{in}\,t_2 \rightsquigarrow \mathsf{let}\,(x, y) = t_1'\,\mathsf{in}\,t_2} \textsc{Sem\_CongTenCut1}$$

$$\frac{t_1 \rightsquigarrow t_1'}{\mathsf{let}\,\Box x = t_1\,\mathsf{in}\,t_2 \rightsquigarrow \mathsf{let}\,\Box x = t_1'\,\mathsf{in}\,t_2} \textsc{Sem\_CongBox1}$$

$$\frac{A \rightsquigarrow A'}{\Box_{s} A \rightsquigarrow \Box_{s} A'} \textsc{Sem\_CongBox4}$$

$$\frac{A \rightsquigarrow A'}{(x :_{(s,r)} A) \to B \rightsquigarrow (x :_{(s,r)} A') \to B} \textsc{Sem\_CongArrow1}$$

$$\frac{B \rightsquigarrow B'}{(x :_{(s,r)} A) \to B \rightsquigarrow (x :_{(s,r)} A) \to B'} \textsc{Sem\_CongArrow2}$$

$$\frac{A \rightsquigarrow A'}{(x :_{r} A) \otimes B \rightsquigarrow (x :_{r} A') \otimes B} \textsc{Sem\_CongTen1}$$

$$\frac{B \rightsquigarrow B'}{(x :_{r} A) \otimes B \rightsquigarrow (x :_{r} A) \otimes B'} \textsc{Sem\_CongTen2}$$

$\boxed{\Delta; \Gamma \vdash t_1 \Leftarrow t_2; \sigma_1; \sigma_2}$ Checking

$$\frac{\begin{array}{l}\Delta; \Gamma \vdash A \Rightarrow \mathsf{Type}_{l}; \sigma_1; \mathbf{0} \\ \Delta, \sigma_1; \Gamma, x : A \vdash t \Leftarrow B; \sigma_2, s; \sigma_3, r\end{array}}{\Delta; \Gamma \vdash \lambda x.t \Leftarrow (x :_{(s,r)} A) \to B; \sigma_2; \sigma_1 + \sigma_3} \textsc{ChkAlg\_Fun}$$

$\boxed{\Delta; \Gamma \vdash t_1 \Rightarrow t_2; \sigma_1; \sigma_2}$ Inference

$$\frac{\begin{array}{l}\Delta; \Gamma \vdash t_1 \Rightarrow (x :_{(s,r)} A) \to B; \sigma_2; \sigma_{13} \\ \Delta; \Gamma \vdash t_2 \Leftarrow A; \sigma_4; \sigma_1 \\ \Delta, \sigma_1; \Gamma, x : A \vdash B \Rightarrow \mathsf{Type}_{l}; \sigma_3, r; \mathbf{0} \\ \sigma_{13} = \sigma_1 + \sigma_3\end{array}}{\Delta; \Gamma \vdash t_1\, t_2 \Rightarrow [t_2/x]B; \sigma_2 + s * \sigma_4; \sigma_3 + r * \sigma_4} \textsc{InfAlg\_App}$$

$$\frac{\begin{array}{l}\Delta = \Delta_1, \sigma, \Delta_2\\ \Gamma = \Gamma_1, x : A, \Gamma_2\\ |\Gamma_1| = |\Delta_1|\\ \Delta_1; \Gamma_1 \vdash A \Rightarrow \mathsf{Type}_l; \sigma; \mathbf{0}\end{array}}{\Delta; \Gamma \vdash x \Rightarrow A; (\mathbf{0}^{|\Delta_1|}, 1, \mathbf{0}^{|\Delta_2|}); (\sigma, 0, \mathbf{0}^{|\Delta_2|})}\ \textsc{InfAlg\_Var}$$

$\boxed{\Gamma \vdash t_1 = t_2 : t_3}$ Term equality

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : A}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t = t : A}\ \text{TEQ\_}\textsc{Refl}$$

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A \quad (\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_2 = t_3 : A}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_3 : A}\ \text{TEQ\_}\textsc{Trans}$$

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_2 = t_1 : A}\ \text{TEQ\_}\textsc{Sym}$$

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A \quad (\Delta \mid \sigma_2) \odot \Gamma \vdash A \leq B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : B}\ \text{TEQ\_}\textsc{ConvTy}$$

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A = C : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B = D : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_{(s,r)} A) \to B = (x :_{(s,r)} C) \to D : \mathsf{Type}_{l_1 \sqcup l_2}}\ \text{TEQ\_}\textsc{Arrow}$$

$$\frac{(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t_1 : B \quad (\Delta \mid \sigma_4 \mid \sigma_1) \odot \Gamma \vdash t_2 : A}{(\Delta \mid \sigma_2 + s * \sigma_4 \mid \sigma_3 + r * \sigma_4) \odot \Gamma \vdash (\lambda x.t_1)\, t_2 = [t_2/x]t_1 : [t_2/x]B}\ \text{TEQ\_}\textsc{ArrowComp}$$

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : (x :_{(s,r)} A) \to B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t = \lambda x.(t\, x) : (x :_{(s,r)} A) \to B}\ \text{TEQ\_}\textsc{ArrowUniq}$$

$$\frac{(\Delta, \sigma_1 \mid \sigma_2, s \mid \sigma_3, r) \odot \Gamma, x : A \vdash t_1 = t_2 : B}{(\Delta \mid \sigma_2 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash \lambda x.t_1 = \lambda x.t_2 : (x :_{(s,r)} A) \to B}\ \text{TEQ\_}\textsc{Fun}$$

$$\frac{\begin{array}{l}(\Delta, \sigma_3 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l\\ (\Delta \mid \sigma_1 \mid \sigma_3 + \sigma_4) \odot \Gamma \vdash t_1 = t_2 : (x :_{(s,r)} A) \to B \quad (\Delta \mid \sigma_2 \mid \sigma_3) \odot \Gamma \vdash t_3 = t_4 : A\end{array}}{(\Delta \mid \sigma_1 + s * \sigma_2 \mid \sigma_4 + r * \sigma_2) \odot \Gamma \vdash t_1\, t_3 = t_2\, t_4 : [t_3/x]B}\ \text{TEQ\_}\textsc{App}$$

$$\frac{(\Delta \mid \sigma_1 \mid \mathbf{0}) \odot \Gamma \vdash A = C : \mathsf{Type}_{l_1} \quad (\Delta, \sigma_1 \mid \sigma_2, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B = D : \mathsf{Type}_{l_2}}{(\Delta \mid \sigma_1 + \sigma_2 \mid \mathbf{0}) \odot \Gamma \vdash (x :_s A) \otimes B = (x :_s C) \otimes D : \mathsf{Type}_{l_1 \sqcup l_2}}\ \text{TEQ\_}\textsc{Ten}$$

$$\frac{\begin{array}{l}(\Delta, (\sigma_1 + \sigma_2) \mid \sigma_5, r' \mid \mathbf{0}) \odot \Gamma, z : (x :_r A) \otimes B \vdash C : \mathsf{Type}_l\\ (\Delta \mid \sigma_3 \mid \sigma_1) \odot \Gamma \vdash t_1 : A \quad (\Delta \mid \sigma_6 \mid \sigma_2 + r * \sigma_3) \odot \Gamma \vdash t_2 : [t_1/x]B\\ (\Delta, \sigma_1, (\sigma_2, r) \mid \sigma_4, s, s \mid \sigma_5, r', r') \odot \Gamma, x : A, y : B \vdash t_3 : [(x,y)/z]C\end{array}}{(\Delta \mid \sigma_4 + s * (\sigma_3 + \sigma_6) \mid \sigma_5 + r' * (\sigma_3 + \sigma_6)) \odot \Gamma \vdash \mathsf{let}\,(x,y) = (t_1, t_2)\,\mathsf{in}\, t_3 = [t_1, t_2/x, y]t_3 : [(t_1,t_2)/z]C}\ \text{TEQ\_}\textsc{TenComp}$$

$$\frac{\begin{array}{l}(\Delta, \sigma_1 \mid \sigma_3, r \mid \mathbf{0}) \odot \Gamma, x : A \vdash B : \mathsf{Type}_l\\ (\Delta \mid \sigma_2 \mid \sigma_1) \odot \Gamma \vdash t_1 = t_1' : A \quad (\Delta \mid \sigma_4 \mid \sigma_3 + r * \sigma_2) \odot \Gamma \vdash t_2 = t_2' : [t_1/x]B\end{array}}{(\Delta \mid \sigma_2 + \sigma_4 \mid \sigma_1 + \sigma_3) \odot \Gamma \vdash (t_1, t_2) = (t_1', t_2') : (x :_r A) \otimes B}\ \text{TEQ\_}\textsc{Pair}$$

$$\frac{\begin{array}{l}(\Delta \mid \sigma_3 \mid \sigma_1 + \sigma_2) \odot \Gamma \vdash t_1 = t_2 : (x :_r A) \otimes B\\ (\Delta, (\sigma_1 + \sigma_2) \mid \sigma_5, r' \mid \mathbf{0}) \odot \Gamma, z : (x :_r A) \otimes B \vdash C : \mathsf{Type}_l\\ (\Delta, \sigma_1, (\sigma_2, r) \mid \sigma_4, s, s \mid \sigma_5, r', r') \odot \Gamma, x : A, y : B \vdash t_3 = t_4 : [(x,y)/z]C\end{array}}{(\Delta \mid \sigma_4 + s * \sigma_3 \mid \sigma_5 + r' * \sigma_3) \odot \Gamma \vdash (\mathsf{let}\,(x,y) = t_1\,\mathsf{in}\, t_3) = (\mathsf{let}\,(x,y) = t_2\,\mathsf{in}\, t_4) : [t_1/z]C}\ \text{TEQ\_}\textsc{TenCut}$$

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : (x :_r A) \otimes B}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t = (\mathsf{let}\,(x,y) = t\,\mathsf{in}\,(x,y)) : (x :_r A) \otimes B}\ \text{TEQ\_}\textsc{TenU}$$

$$\frac{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash A = B : \mathsf{Type}_l}{(\Delta \mid \sigma \mid \mathbf{0}) \odot \Gamma \vdash \Box_s A = \Box_s B : \mathsf{Type}_l} \text{TEQ\_Box}$$

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : A}{(\Delta \mid s * \sigma_1 \mid \sigma_2) \odot \Gamma \vdash \Box t_1 = \Box t_2 : \Box_s A} \text{TEQ\_BoxI}$$

$$\frac{\begin{array}{l}(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 : A \\ (\Delta, \sigma_2 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, z : \Box_s A \vdash B : \mathsf{Type}_l \\ (\Delta, \sigma_2 \mid \sigma_3, s \mid \sigma_4, (s * r)) \odot \Gamma, x : A \vdash t_2 : [\Box x / z]B\end{array}}{(\Delta \mid \sigma_3 + s * \sigma_1 \mid \sigma_4 + s * r * \sigma_1) \odot \Gamma \vdash (\mathsf{let}\, \Box x = \Box t_1 \,\mathsf{in}\, t_2) = [t_1/x]t_2 : [\Box t_1 / z]B} \text{TEQ\_BoxB}$$

$$\frac{\begin{array}{l}(\Delta, \sigma_2 \mid \sigma_4, r \mid \mathbf{0}) \odot \Gamma, z : \Box_s A \vdash B : \mathsf{Type}_l \\ (\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t_1 = t_2 : \Box_s A \quad (\Delta, \sigma_2 \mid \sigma_3, s \mid \sigma_4, (s * r)) \odot \Gamma, x : A \vdash t_3 = t_4 : [\Box x / z]B\end{array}}{(\Delta \mid \sigma_1 + \sigma_3 \mid \sigma_4 + r * \sigma_1) \odot \Gamma \vdash (\mathsf{let}\, \Box x = t_1 \,\mathsf{in}\, t_3) = (\mathsf{let}\, \Box x = t_2 \,\mathsf{in}\, t_4) : [t_1/z]B} \text{TEQ\_BoxE}$$

$$\frac{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t : \Box_s A}{(\Delta \mid \sigma_1 \mid \sigma_2) \odot \Gamma \vdash t = (\mathsf{let}\, \Box x = t \,\mathsf{in}\, \Box x) : \Box_s A} \text{TEQ\_BoxU}$$