Du typage vectoriel

Alejandro Diaz Caro

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Alejandro DÍAZ-CARO

Thèse dirigée par **Pablo ARRIGHI**
et codirigée par **Frédéric PROST**

préparée au sein du **Laboratoire d'Informatique de Grenoble**
et de l'**École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

Du typage vectoriel

Thèse soutenue publiquement le **23 Septembre 2011**,
devant le jury composé de :

M. Philippe JORRAND
Directeur de Recherche Émérite CNRS, HDR, Président

M. Eduardo BONELLI
Profesor Asociado à l'UNQ et Investigador Adjunto au CONICET, Rapporteur

M. Gilles DOWEK
Directeur de Recherche INRIA, HDR, Rapporteur

M. Thomas EHRHARD
Directeur de Recherche CNRS, HDR, Rapporteur

M. Michele PAGANI
Maître de Conférences à l'Université de Paris Nord, Examinateur

M. Laurent REGNIER
Professeur à l'Université de la Méditerranée, HDR, Examinateur

M. Lionel VAUX
Maître de Conférences à l'Université de la Méditerranée, Examinateur

M. Pablo ARRIGHI
Maître de Conférences à l'Université de Grenoble, HDR, Directeur de thèse

a Nache
mi compañera de ruta

Résumé

L'OBJECTIF de cette thèse est de développer une théorie de types pour le λ-calcul linéaire-algébrique, une extension du λ-calcul motivé par l'informatique quantique. Cette extension algébrique comprend tous les termes du λ-calcul plus leurs combinaisons linéaires, donc si $\mathbf{t}$ et $\mathbf{r}$ sont des termes, $\alpha.\mathbf{t}+\beta.\mathbf{r}$ est aussi un terme, avec α et β des scalaires pris dans un anneau. L'idée principale et le défi de cette thèse était d'introduire un système de types où les types, de la même façon que les termes, constituent un espace vectoriel, permettant la mise en évidence de la structure de la forme normale d'un terme. Cette thèse présente le système *Lineal*, ainsi que trois systèmes intermédiaires, également intéressants en eux-même : *Scalar*, *Additive* et λ^{CA}, chacun avec leurs preuves de préservation de type et de normalisation forte.

Abstract

THE objective of this thesis is to develop a type theory for the linear-algebraic λ-calculus, an extension of λ-calculus motivated by quantum computing. This algebraic extension encompasses all the terms of λ-calculus together with their linear combinations, so if $\mathbf{t}$ and $\mathbf{r}$ are two terms, so is $\alpha.\mathbf{t}+\beta.\mathbf{r}$, with α and β being scalars from a given ring. The key idea and challenge of this thesis was to introduce a type system where the types, in the same way as the terms, form a vectorial space, providing the information about the structure of the normal form of the terms. This thesis presents the system *Lineal*, and also three intermediate systems, however interesting by themselves: *Scalar*, *Additive* and λ^{CA}, all of them with their subject reduction and strong normalisation proofs.

Acknowledgements

This thesis would not have been possible without the guidance and help of several individuals, who in one way or another have contributed and extended their valuable assistance in the preparation and completion of this study.

First, my utmost gratitude to Pablo Arrighi, my adviser, who trusted me from the beginning. He encouraged me at every step, not only in scientific matters but also for my integration in France. His motivation, enthusiasm and foremost, his confidence in me, are what made me always try harder.

I thank Frédéric Prost for sharing the task of co-advising this thesis.

The INS2I-PEPS project *QuAND* has financed me several times to go abroad and present the result of these works, for which I want to thank its coordinator Lionel Vaux. This work was also partially surpported by the ANR-JCJC project *CausaQ* and the European FP6-STREP project *QICS*.

I want to thank to Christine Tasson, with whom I have had the pleasure to work. I also thank Barbara Petit, who has been a source of brilliant ideas, and who has become a very good friend. My thanks also go to Benoît Valiron and Simon Perdrix, with whom I have had the privilege of working and sharing experiences; it has been always fruitful to work with them.

I extend my gratitude to my masters student, Pablo Buiras, and his co-adviser, Mauro Jaskelioff, with whom I have spent hours discussing via skype and email, which enriched me every time.

I am very grateful with Eduardo Bonelli, Gilles Dowek, Laurent Regnier, Lionel Vaux, Michele Pagani, Philippe Jorrand and Thomas Ehrhard for giving me the honour of evaluating this thesis and for their useful comments and enlightening discussions.

I am grateful to my fellow labmates, Jonathan Grattage, Renan Fargetton, Simon Perdrix and Mehdi Mhalla, for the stimulating discussions and for all the fun we have had over the last three years. In particular, I want to especially thank Jon and his wife Janine for their invaluable friendship, and for all the proofreading they have done over the past three years; and Simon for all his encouragement, and for being the first to make me learn French with his infinite patience. I also thank the whole CAPP team for having received me within the group.

The lack of typos and the improved English in this thesis are due to Barbara, Brian, Endo, Gabriela, Isolda, Janine, Jon, Rodrigo and Santi. Thanks! On the other hand, the enormous amount of typos and bad English that are still here, are my fault entirely.

I cannot forget to thank to all the $\tilde{\mathrm{N}}$ mailing list, my source of great amusement at a distance.

I want to express a special thanks to my mother, who has always supported me and encouraged me to follow my dreams. She always instilled me the love for knowledge, from an early age, when she encouraged me when I participated in the school science club, and bought me every book I asked for to satisfy my growing curiosity during my childhood.

To my brothers and sisters Félix, Diego, Mariano, Paty, Sol, Belén and Estefy, my brothers and sisters in law, my nephews and nieces, my uncles and aunts: thanks for always being there, for support me and for that much love.

Most importantly, I want to thank my wife Nache, who has given me her support, understanding and love, and was on my side during all the successes and failures over the past three years, and before. It would not have been possible to finish this thesis without her. To her I dedicate this thesis.

Contents

List of Figures

Chapter 1

Introduction

Résumé du Chapitre

Ce chapitre est à la fois une préface décrivant les motivations de cette thèse, et un résumé de notions préliminaires de base, y compris quelques notions d'informatique quantique avec une preuve du théorème de non-clonage [Wootters and Zurek, 1982]. Est également présenté ici le lambda-calcul linéaire algébrique non-typé, λ_{lin} [Arrighi and Dowek, 2008], et des exemples d'encodage de l'informatique quantique dans ce calcul. ■

The λ-calculus [Church, 1936] is a model for the definition of function. It can be seen as the most simple and universal programming language: it includes only a definition of variable and a variable substitution rule. The concept of computability itself can be defined in terms of λ-calculus: a function is computable if and only if there exists a way to write it in λ-calculus.

A way to characterise programs without executing them is to type them [Church, 1940]: a type system will statically classify the programs in types, thereby yielding certain information about the kind of output that the program will produce. Moreover, the Curry-Howard correspondence (see for example [Sørensen and Urzyczyn, 2006]) establishes a direct relation between the type of a program and a proof in constructive mathematics. This way, a typed program becomes a proof of a given logical formulae.

Two algebraic extensions of the λ-calculus arise independently in distinct contexts: the *algebraic λ-calculus* (λ_{alg}) [Vaux, 2007, 2009] and the *linear-algebraic λ-calculus* (λ_{lin}) [Arrighi and Dowek, 2008]. The former has been introduced in the context of linear logic as a fragment of the *differential λ-calculus* [Ehrhard and Regnier, 2003]: the algebraic structure allows to gather in a non deterministic manner different terms, *i.e.* each term represents one possible execution. The latter has been introduced as a candidate λ-calculus for quantum computation: in λ_{lin}, a linear combination of terms reflects the phenomenon of superposition, *i.e.* the capacity for a quantum system to be in two or more states at the same time.

These two languages are rather similar: they both merge higher-order computation, be it terminating or not, in its simplest and most general form (namely the untyped λ-calculus) together with linear algebra in its simplest and most general form also (the axioms of vector spaces). In fact they can simulate each other (*cf.* Chapter 2). Our starting point will be the second one: because its confluence proof allows arbitrary scalars and because we are interested in the possible applications to quantum computing.

The two languages are also reminiscent of other works in the literature: The functional style of programming is based on the λ-calculus together with a number of extensions, so as to make everyday programming more accessible. Hence since the birth of functional programming there has been several theoretical studies of extensions of the λ-calculus in order to account for basic algebra (see for example Dougherty's algebraic extension [Dougherty, 1992] for normalising terms of the λ-calculus) and other basic programming constructs such as pattern-matching, together with the sometimes non-trivial associated type theories (see for example Petit's λ-calculus extension and type system [Petit, 2009] with pattern matching). Whilst this was not the original motivation behind the algebraic λ-calculi, these languages could still be viewed as just an extension of the λ-calculus in order to handle operations over vector spaces, and make everyday programming more accessible upon them. The main difference in approach is that here the λ-calculus is not seen as a control structure which sits on top of the vector space data structure, controlling which operations to apply and when. Rather, the λ-calculus terms themselves can be summed and weighted, hence they actually are the basis of the vector space... upon which they can also act.

The above intertwining of concepts is essential if seeking to represent parallel or probabilistic computation as it is the computation itself which must be endowed with a vector space structure. The ability to superpose λ-terms in that sense takes us back to Bouldol's parallel λ-calculus [Boudol, 1994], and may also be viewed as taking part of a wave of probabilistic extensions of calculi, *e.g.* [Bournez and Hoyrup, 2003, Herescu and Palamidessi, 2000, Di Pierro, Hankin, and Wiklicky, 2005].

Hence algebraic λ-calculi can be seen as a platform for various applications, ranging from algebraic computation, probabilistic computation, quantum computation and resource-aware computation.

In the same way that the theory of vector spaces has many applications, but also many theoretical refinements that deserve to be studied in their own right, we take the view that the theory of vector spaces plus λ-calculus has got theoretical refinements that need to be studied in their own right. Moreover, these theoretical refinements are often necessary in order to address the applications, as is notoriously the case as instance with the notion of norm. For example if we want to be able to interpret a linear combination of terms $\sum \alpha_i.\mathbf{t}_i$ as a probability distribution, we will need to make sure that it has norm one. The same is true if we want to interpret $\sum \alpha_i.\mathbf{t}_i$ as quantum superposition, but with a different norm[1]. Yet the very definition of a norm is difficult in our context: deciding whether a term terminates is undecidable; but these terms produce infinities, hence convergence of the norm is undecidable. Related to this precise topic, Vaux has studied simply typed algebraic λ-calculus, ensuring convergence of the norm [Vaux, 2009]. Quite recently Tasson has studied some model-theoretic properties of the *barycentric* ($\sum \alpha_i = 1$) subset of this simply typed calculus [Tasson, 2009], whereas Ehrhard has proven the convergence of a Taylor series expansion of algebraic λ-calculus terms, via a System F typing system [Ehrhard, 2010].

Therefore, it can be said that standard type systems provide part of the solution: they ensure the convergence of (the existence of) the norm of a term. And indeed it is not so hard to define a simple extension of System F that fits λ_{lin} – just by providing the needed rules to type additions, scalar products and the null vector in some trivial manner (see Definition 3.3.1). However notice that the solution of a straightforward extension of a standard type system could be both, too restrictive in one sense and too

[1] Whereas it is clear already that λ_{lin} is a quantum λ-calculus, in the sense that any quantum algorithm can be expressed in this language (*cf.* Section 1.2), the converse, alas, is not true, in the sense that some programs in λ_{lin} express evolutions which are not valid quantum algorithms. This is precisely because λ_{lin} does not restrict its vectors to be normalised $\sum |\alpha_i|^2 = 1$ and its applications to be isometries.

permissive in another: for example to type a sum we can rely on a typing rule like

$$\frac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{r} : T}{\Gamma \vdash \mathbf{t} + \mathbf{r} : T}$$

which can be thought of as too restrictive: a sum will have type only when both terms have the same type. On the other hand, the rule for scaling terms could be

$$\frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : T}$$

which does not provide any way to control the scalar we are multiplying by, making hard to impose any kind of restriction such as a barycentric relation or a quantum normalisation.

Instead, having a theory where sums and scalar multiplications are reflected in the types, can give the answer: the term $\alpha.\mathbf{t} + \beta.\mathbf{r}$ would have the type $\alpha.T + \beta.R$, which provides the details on the vectorial structure of the term.

In addition, since we provide full-blown proofs of strong normalisation from all the type systems presented in this thesis, a byproduct of this result is that we are able to remove several conditions that were limiting the reduction rules of λ_{lin}, because their purpose was to keep indefinite form from reducing (such as $\mathbf{t} - \mathbf{t}$, with $\mathbf{t}$ not normal and hence potentially infinite). This makes λ_{lin} into a simpler language.

Basic definitions and notation

The usual notation regarding rewrite systems is used: given a rewrite system R, R^* is its *reflexive and transitive closure*. That is, $\mathbf{t} R^* \mathbf{r}$ is valid if $\mathbf{r} = \mathbf{t}$ or if there exists a rewrite sequence $\mathbf{t}\, R\, \mathbf{t}_1\, R \cdots R\, \mathbf{t}_n\, R\, \mathbf{r}$ linking $\mathbf{t}$ and $\mathbf{r}$. $R_{\leftrightarrow}$ is the *symmetric closure* of R, that is, the relation that satisfies $\mathbf{t}\, R_{\leftrightarrow}\, \mathbf{r}$ if and only if $\mathbf{t}\, R\, \mathbf{r}$ or $\mathbf{r}\, R\, \mathbf{t}$.

In some examples, we may use the notation $\mathbf{t} - \mathbf{r}$ as a shorthand for $\mathbf{t} + (-1).\mathbf{r}$, however there will not be any ambiguity, since $-$ is not a binary operation in any of the calculi developed in this thesis.

A rewrite system R is *locally confluent* if whenever $\mathbf{t} R \mathbf{r}$ and $\mathbf{t} R \mathbf{s}$ there is $\mathbf{t}'$ such that $\mathbf{r} R^* \mathbf{t}'$ and $\mathbf{s} R^* \mathbf{t}'$. In comparison, a rewrite system R is *confluent* if whenever $\mathbf{t} R^* \mathbf{r}$ and $\mathbf{t} R^* \mathbf{s}$ there is t such that $\mathbf{r} R^* \mathbf{t}'$ and $\mathbf{s} R^* \mathbf{t}'$. Notice that confluence implies local confluence whereas the inverse is not true.

1.1 A brief and fast introduction to the quantum notation

This section does not pretend to introduce a full description of quantum computing, the interested reader can find actual introductions to this area in many textbooks, *e.g.* [Nielsen and Chuang, 2000, Jaeger, 2007]. This section only pretends to introduce the basic notations and concepts used in the rest of this thesis.

In quantum computation, data is encoded on normalised vectors in Hilbert spaces. For our purpose, this means that the vector spaces are defined over complex numbers and come with a norm and a notion of orthogonality. The smallest space usually considered is the space of *qubits*. This space is the two-dimensional vector space $\mathbb{C}^2$, and it comes with a chosen orthonormal basis denoted by $\{|0\rangle, |1\rangle\}$. A general quantum bit (or qubit) is a normalised vector $\alpha|0\rangle + \beta|1\rangle$, where $|\alpha|^2 + |\beta|^2 = 1$. To denote an unknown qubit ϕ it is common to write $|\phi\rangle$. A two-qubits vector is a normalised vector in $\mathbb{C}^2 \otimes \mathbb{C}^2$, that is, a normalised vector generated by the orthonormal basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, where $|xy\rangle$ stands for

$|x\rangle \otimes |y\rangle$. In the same way, a n-qubits vector is a normalised vector in $(\mathbb{C}^2)^n$ (or $\mathbb{C}^N$ with $N = 2^n$). Also common is the notation $\langle\psi|$ for the transposed, conjugate of $|\psi\rangle$, *e.g.* if $|\psi\rangle = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T$, then $\langle\psi| = [\alpha_1^*, \alpha_2^*, \ldots, \alpha_n^*]$ where for any $\alpha \in \mathbb{C}$, α^* denotes its conjugated.

The operators on qubits that are considered in this thesis are the *quantum gates*, that is, isometric operators. A *isometric operator* is a linear function preserving the norm and the orthogonality of vectors. The *adjoint* of a given operator U is denoted by $U^\dagger$, and the isometric condition imposes that $U^\dagger U = Id$. These functions are linear, and so it is enough to describe their action on the base vectors. Another way to describe these functions would be by matrices, and then the adjoint is just its conjugate transpose. A set of universal quantum gates is the set *cnot*, $R_{\frac{\pi}{4}}$ and *had*, which can be defined as follows:

The *cnot* gate. The *controlled-not*, or *cnot*, is a two-qubits gate which only changes the second qubit if the first one is $|1\rangle$:

$$cnot|0x\rangle = |0x\rangle \qquad ; \qquad cnot|1x\rangle = |1\rangle \otimes not|x\rangle$$

where $not|0\rangle = |1\rangle$ and $not|1\rangle = |0\rangle$.

The $R_{\frac{\pi}{4}}$ gate. The $R_{\frac{\pi}{4}}$ gate is a single-qubit gate that modifies the phase of the qubit:

$$R_{\frac{\pi}{4}}|0\rangle = |0\rangle \qquad ; \qquad R_{\frac{\pi}{4}}|1\rangle = e^{i\frac{\pi}{4}}|1\rangle$$

where $\frac{\pi}{4}$ is the phase shift.

The *had* gate. The *Hadamard* gate, or *had*, is a single-qubit gate which produces a basis change:

$$had|0\rangle = \tfrac{1}{\sqrt{2}}|0\rangle + \tfrac{1}{\sqrt{2}}|1\rangle \qquad ; \qquad had|1\rangle = \tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle$$

To make these gates act in higher-dimension qubits, they can be put together with the bilinear symbol $\otimes$. For example, to make the Hadamard gate act only in the first qubit of a two-qubits register, it can be taken to $had \otimes Id$, and to apply a Hadamard gate to both qubits, just $had \otimes had$.

An important restriction, which has to be taken into account if a calculus pretends to encode quantum computing, is the so called *no-cloning theorem* [Wootters and Zurek, 1982]:

Theorem 1.1.1 (No cloning)**.** *There is no linear operator such that, given any qubit $|\phi\rangle \in \mathbb{C}^N$, it can clone it. That is, it does not exists any isometric operator U and fixed $|\psi\rangle \in \mathbb{C}^N$ such that $U|\psi\phi\rangle = |\phi\phi\rangle$.*

Proof. Assume there exists such an operator U, so given any $|\varphi\rangle, |\phi\rangle$ one has $U|\psi\varphi\rangle = |\varphi\varphi\rangle$ and also $U|\psi\phi\rangle = |\phi\phi\rangle$. Then

$$\langle U\varphi\psi|U\psi\phi\rangle = \langle\varphi\varphi|\phi\phi\rangle \tag{1.1}$$

where $\langle U\varphi\psi|$ is the conjugate transpose of $U|\psi\varphi\rangle$. However, notice that the left side of equation 1.1 can be rewritten as $\langle\varphi\psi|U^\dagger U|\psi\phi\rangle = \langle\varphi\psi|\psi\phi\rangle = \langle\varphi|\phi\rangle$.

On the other hand, the right side of equation 1.1 can be rewritten as $\langle\varphi|\phi\rangle\langle\varphi|\phi\rangle = \langle\varphi|\phi\rangle^2$.

So $\langle\varphi|\phi\rangle = \langle\varphi|\phi\rangle^2$, which implies either $\langle\varphi|\phi\rangle = 0$ or $\langle\varphi|\phi\rangle = 1$, none of which can be true in the general case, since $|\varphi\rangle$ and $|\phi\rangle$ were picked as random qubits. □

The implication of this theorem in the design choices of a calculus is that it must be forbidden to allow functions duplicating arbitrary arguments. However notice that this does not forbids cloning some specific qubit states. Indeed for example the $|0\rangle$ and $|1\rangle$ qubits can be cloned without much effort by using the *cnot* gate: $cnot|00\rangle = |00\rangle$ and $cnot|10\rangle = |11\rangle$. In this sense, the imposed restriction is not a 'resources aware' restriction *à la* linear logic [Girard, 1987]. It is a restriction that forbids us to create a 'universal cloning machine', but still allows us to clone any given known term.

1.2 The linear-algebraic λ-calculus (λ_{lin})

In this section we present the untyped *linear-algebraic λ-calculus* (λ_{lin}) [Arrighi and Dowek, 2008]. As a language of terms, λ_{lin} is just λ-calculus together with the possibility to make arbitrary linear combinations of terms ($\alpha.\mathbf{t} + \beta.\mathbf{r}$). In terms of operational semantics, λ_{lin} merges higher-order computation, be it terminating or not, in its simplest and most general form (the β-reduction of the untyped λ-calculus) together with linear algebra in its simplest and most general form also (the oriented axioms of vector spaces). However it is not a straightforward step to merge these two families of reduction rules.

To illustrate the kind of decisions that has to be taken, consider the term $(\lambda x.(x)\ x)\ (\alpha.\mathbf{t} + \beta.\mathbf{u})$ which may be thought of as reducing to $(\alpha.\mathbf{t} + \beta.\mathbf{u})\ (\alpha.\mathbf{t} + \beta.\mathbf{u})$ if it is chosen to β-reduce it first, or to $\alpha.((\mathbf{t})\ \mathbf{t}) + \beta.((\mathbf{u})\ \mathbf{u})$ if it chosen to first distribute the application and then do the β-reduction. The first option is a call-by-name strategy while the second can be seen as *call-by-base*, defining the variables and abstractions to be the *base terms*. Notice that the call-by-base strategy is compatible with the view that application should be bilinear (*cf. Application rules*, below). Leaving both options open would break confluence, the second option was chosen, which entails restricting the β-reduction to terms not containing sums or scalars in head position (*cf. Beta reduction* rule, below).

The call-by-base strategy is also compatible with the no-cloning theorem (*cf.* Theorem 1.1.1): assume that $\mathbf{t}$ and $\mathbf{u}$ are two "base terms" encoding $|0\rangle$ and $|1\rangle$ respectivelly, and let $C = \lambda x.(x)\ x$ be a valid abstraction (depending on the encoding of $|0\rangle/|1\rangle$ we will need different encoding for this abstraction, but for the sake of the example, assume this is an abstraction that just concatenates twice its argument). Then with a call-by-name strategy, the term $(C)\ (\alpha.\mathbf{t} + \beta.\mathbf{u}) \to^* (\alpha.\mathbf{t} + \beta.\mathbf{u})(\alpha.\mathbf{t} + \beta.\mathbf{u})$ which is clearly a cloning of the argument. Instead, in a call-by-base setting the same term leads to $\alpha.(\mathbf{t})\ \mathbf{t} + \beta.(\mathbf{u})\ \mathbf{u}$, which is not forbidden by Theorem 1.1.1.

Instead of introducing vector spaces via an oriented version of their axioms (*e.g.* $\alpha.\mathbf{u}+\beta.\mathbf{u} \to (\alpha+\beta).\mathbf{u}$), one could have decided to perform the β-reduction 'modulo equality in the theory of vector spaces' (*e.g.* $\alpha.\mathbf{u} + \beta.\mathbf{u} = (\alpha + \beta).\mathbf{u}$). But there is also a good reason not to do that: it is possible to define fixed point operators

$$\mathbf{Y} = \lambda y.(\lambda x.(y + (x)\ x))\ \lambda x.(y + (x)\ x)$$

and a term $\mathbf{b}$ such that $(\mathbf{Y})\ \mathbf{b}$ reduces to $\mathbf{b} + (\mathbf{Y})\ \mathbf{b}$ and so on. Modulo equality over vector spaces, the theory would be inconsistent, as the term $(\mathbf{Y})\ \mathbf{b} - (\mathbf{Y})\ \mathbf{b}$ would then be equal to $\mathbf{0}$, but would also reduce to $\mathbf{b} + (\mathbf{Y})\ \mathbf{b} - (\mathbf{Y})\ \mathbf{b}$ and hence also be equal to $\mathbf{b}$. Instead, this problem can be fixed by restricting rules such as $\alpha.\mathbf{u} + \beta.\mathbf{u} \to (\alpha + \beta).\mathbf{u}$ to terms that cannot reduce forever (*cf. Factorisation rules*, below), matching the old intuition that indefinite forms '$\infty - \infty$' must be left alone. Moreover, oriented axioms of vector spaces define vector spaces, and no more than vector spaces, just as well as the original axioms do, as was shown in [Arrighi and Dowek, 2008]. Plus the orientation serves a purpose: it presents the vector in its canonical form. A more in-depth discussion about these decisions: call-by-name vs. call-by-base and reduction vs. equalities, is delayed to Chapter 2, where the four possibilities are analysed and the simulations between them are proved.

The untyped λ_{lin} calculus, as defined in [Arrighi and Dowek, 2008], is presented in Fig. 1.1. Terms contain a subclass of *base terms*, that are the only ones that can be substituted for a variable in a β-reduction step. Terms are considered modulo associativity and commutativity of operator $+$ (that is an *AC-rewrite system* [Jouannaud and Kirchner, 1986]; *cf.* Chapter 2 for a full discussion about it). Scalars (notation: $\alpha, \beta, \gamma, \dots$) are members of a commutative ring $(\mathcal{S}, +, \times)$. The confluence of this calculus has

been formally proven in [Arrighi and Dowek, 2008]. The proof relies on the restrictions on the reduction rules. However since these restrictions are not longer needed when working with the stronger normalising subset of the calculus, they will not remain in any of the typed versions (*cf.* the confluence proofs in any of the typed versions in the following chapters). So this confluence proof is not developed here and is rather delayed to the typed versions.

The set of free variables of a term (notation: $FV(\mathbf{t})$) is defined as expected. The operation of substitution on terms (notation: $\mathbf{t}[\mathbf{b}/x]$) is defined as usual (*i.e.* taking care of renaming bound variables when needed in order to prevent variable capture), with $(\alpha.\mathbf{t} + \beta.\mathbf{u})[\mathbf{b}/x] = \alpha.(\mathbf{t}[\mathbf{b}/x]) + \beta.(\mathbf{u}[\mathbf{b}/x])$. Also, the notation $(\mathbf{t})\ \vec{\mathbf{r}} = (((\mathbf{t})\ \mathbf{r}_1)\dots)\ \mathbf{r}_n$ is used when needed.

Terms: $\quad \mathbf{t}, \mathbf{r}, \mathbf{u} ::= \mathbf{b} \mid (\mathbf{t})\ \mathbf{r} \mid \mathbf{0} \mid \alpha.\mathbf{t} \mid \mathbf{t} + \mathbf{r}$

Base terms: $\quad \mathbf{b} ::= x \mid \lambda x.\mathbf{t}$

Elementary rules:
$\mathbf{t} + \mathbf{0} \to \mathbf{t}$,
$0.\mathbf{t} \to \mathbf{0}$,
$1.\mathbf{t} \to \mathbf{t}$,
$\alpha.\mathbf{0} \to \mathbf{0}$,
$\alpha.(\beta.\mathbf{t}) \to (\alpha \times \beta).\mathbf{t}$,
$\alpha.(\mathbf{t} + \mathbf{r}) \to \alpha.\mathbf{t} + \alpha.\mathbf{r}$.

Factorisation rules:
$\alpha.\mathbf{t} + \beta.\mathbf{t} \to (\alpha + \beta).\mathbf{t}$ (*),
$\alpha.\mathbf{t} + \mathbf{t} \to (\alpha + 1).\mathbf{t}$ (*),
$\mathbf{t} + \mathbf{t} \to (1 + 1).\mathbf{t}$ (*).

Beta reduction:
$(\lambda x.\mathbf{t})\ \mathbf{b} \to \mathbf{t}[\mathbf{b}/x]$ (***).

Application rules:
$(\mathbf{t} + \mathbf{r})\ \mathbf{u} \to (\mathbf{t})\ \mathbf{u} + (\mathbf{r})\ \mathbf{u}$ (**),
$(\mathbf{u})\ (\mathbf{t} + \mathbf{r}) \to (\mathbf{u})\ \mathbf{t} + (\mathbf{u})\ \mathbf{r}$ (**),
$(\alpha.\mathbf{t})\ \mathbf{r} \to \alpha.(\mathbf{t})\ \mathbf{r}$ (*),
$(\mathbf{r})\ (\alpha.\mathbf{t}) \to \alpha.(\mathbf{r})\ \mathbf{t}$ (*),
$(\mathbf{0})\ \mathbf{t} \to \mathbf{0}$,
$(\mathbf{t})\ \mathbf{0} \to \mathbf{0}$.

Contextual rules: If $\mathbf{t} \to \mathbf{r}$, then for any term $\mathbf{u}$, scalar α and variable x,
$(\mathbf{t})\ \mathbf{u} \to (\mathbf{r})\ \mathbf{u}$, $\quad \mathbf{t} + \mathbf{u} \to \mathbf{r} + \mathbf{u}$, $\quad \alpha.\mathbf{t} \to \alpha.\mathbf{r}$ and
$(\mathbf{u})\ \mathbf{t} \to (\mathbf{u})\ \mathbf{r}$, $\quad \mathbf{u} + \mathbf{t} \to \mathbf{u} + \mathbf{r}$, $\quad \lambda x.\mathbf{t} \to \lambda x.\mathbf{r}$.

where $+$ is an associative-commutative (AC) symbol, $\alpha, \beta \in \mathcal{S}$, with $(\mathcal{S}, +, \times)$ a commutative ring and
(*) these rules apply only if $\mathbf{t}$ is a closed normal term.
(**) these rules apply only if $\mathbf{t} + \mathbf{r}$ is a closed normal term.
(***) the rule apply only when $\mathbf{b}$ is a base term.
Restriction (***) is the one that limits the β-reduction, whereas restrictions (*) and (**) are those that avoid confluence problems related to infinities and indefinite forms, as discussed above.

Figure 1.1: Syntax and reduction rules of λ_{lin}

1.2.1 Some technical remarks about λ_{lin}

In this section we provide some technical details that will be needed further. They are introduced here since we are introducing λ_{lin}, even though they do not form part of the introductory discourse to the thesis.

A call-by-base calculus. In the classical λ-calculus the call-by-value strategy can be defined by: *Only the outermost redexes are reduced: a redex is reduced only when its right hand side has reduced to a value (variable or lambda abstraction).* Notice that in λ_{lin}, the set of *variables and lambda abstractions* is exactly the set of base terms, in fact values are naturally extended to be base terms and their linear

combinations. So, it is not precise to say that this calculus is call-by-value, instead it can be called *call-by-base*.

The ring of scalars. In the original definition of λ_{lin}, the ring of scalars was defined using a term grammar and a rewrite system [Arrighi and Dowek, 2008, Section III – Definition 1]. For the purpose of this thesis it is sufficient to think of them as members of a ring.

Form of closed normal forms. As first stated in [Arrighi and Dowek, 2008, Proposition 2], closed normal forms are linear combinations of λ-abstractions. We reproduce the propositions and their proofs here for completeness. The first one is an auxiliary lemma [Arrighi and Dowek, 2008, Proposition 1]:

Lemma 1.2.1. *A closed normal form that is neither a sum, nor a product by scalar, or the term* $\mathbf{0}$*, is an abstraction.*

Proof. By induction over term structure. Let $\mathbf{t}$ be a closed normal term that is not a sum, a product by a scalar, or the term $\mathbf{0}$. The term $\mathbf{t}$ is not a variable because it is closed, hence it is either an abstraction in which case we are done, or an application. In this case it has the form $(((\mathbf{u})\ \mathbf{r}_1)\dots)\ \mathbf{r}_n$ where $\mathbf{u}, \mathbf{r}_1, \dots \mathbf{r}_n$ are normal and closed and n is different from 0. Neither $\mathbf{u}$ nor $\mathbf{r}_1$ is a sum, a product by a scalar, or the term $\mathbf{0}$ since the term being normal we then could apply application rules. Thus by induction hypothesis both terms are abstractions. Hence the term is not normal because of the β-reduction. □

Now the proposition states as follows:

Proposition 1.2.2 (Form of closed normal forms [Arrighi and Dowek, 2008])**.** *A closed normal form in* λ_{lin} *is either the null vector or of the form* $\sum_i \alpha_i.\lambda x.\mathbf{t}_i + \sum_i \lambda x.\mathbf{u}_i$*.*

Proof. If the term is not the term $\mathbf{0}$, it can be written as a sum of terms that are neither $\mathbf{0}$ nor sums. We partition these terms in order to group those which are weighted by a scalar and those which are not. Hence we obtain a term of the form $\sum_i \alpha_i.\mathbf{t}'_i + \sum_i \mathbf{u}'_i$ where the terms $\mathbf{u}'_i$ are neither $\mathbf{0}$, nor sums, nor weighted by a scalar. Hence by Lemma 1.2.1 they are abstractions. Because the whole term is normal the terms $\mathbf{t}'_i$ are themselves neither $\mathbf{0}$, nor sums, nor weighted by a scalar since we could apply the Elementary rules. Hence 1.2.1 also applies. □

From this proposition, we can define an infinite vectorial space of normal terms, over the ring of scalars, as the span of the base formed by all the variables and abstractions in normal form.

1.2.2 Encoding quantum computation in λ_{lin}

In this section it is shown how to encode the qubits and the set of universal quantum gates (*cf.* Section 1.1). Since λ_{lin} allows linear combinations of terms, it suffices to encode the basis of the vector space of qubits. For a single-qubit it will be used the terms **false** and **true** for $|0\rangle$ and $|1\rangle$ respectively, with its usual encoding in λ-calculus: $\mathbf{true} = \lambda x.\lambda y.x$ and $\mathbf{false} = \lambda x.\lambda y.y$. A classical example of use of this encoding is the following:

$$not = \lambda x.((x)\ \mathbf{false})\ \mathbf{true}$$

Notice that this term globally expresses a isometric operator, even if some subterms are non isometries.

To encode the Hadamard gate, the first naive choice would be

$$had_{naive} = \lambda x.((x)\ (\frac{1}{\sqrt{2}}.\mathbf{false} - \frac{1}{\sqrt{2}}.\mathbf{true}))\ (\frac{1}{\sqrt{2}}.\mathbf{false} + \frac{1}{\sqrt{2}}.\mathbf{true})$$

However notice that this does not work. Indeed

$$(had_{naive})\ \mathbf{false} \to^* ((\mathbf{false})\ (\frac{1}{\sqrt{2}}.\mathbf{false} - \frac{1}{\sqrt{2}}.\mathbf{true}))\ (\frac{1}{\sqrt{2}}.\mathbf{false} + \frac{1}{\sqrt{2}}.\mathbf{true})$$

$$\to^* \frac{1}{2}.((\mathbf{false})\ \mathbf{false})\ \mathbf{false} + \frac{1}{2}.((\mathbf{false})\ \mathbf{false})\ \mathbf{true} - \frac{1}{2}.((\mathbf{false})\ \mathbf{true})\ \mathbf{false} - \frac{1}{2}.((\mathbf{false})\ \mathbf{true})\ \mathbf{true}$$

$$\to^* \frac{1}{2}.\mathbf{false} + \frac{1}{2}.\mathbf{true} - \frac{1}{2}.\mathbf{false} - \frac{1}{2}.\mathbf{true} \to^* \mathbf{0}$$

where $\to^*$ represents zero or more reductions by the relation $\to$.

The problem is that the linearity of λ_{lin} makes not possible to use the classical if-then-else construction of λ-calculus, that is the **true** and **false** encoding. Instead, the linearity must be prevented in the cases when **true** and **false** are used as deciders. To this end, their arguments can be enclosed under a lambda to make it act actually as an argument in this call-by-base strategy, and then released by applying it to any base term when needed: for example, instead of having $(\lambda x.\lambda y.x)\ (\alpha.\mathbf{true} + \beta.\mathbf{false})$ it can be done $(\lambda x.\lambda y.(x)\ z)\ \lambda w.(\alpha.\mathbf{true} + \beta.\mathbf{false})$ which will avoid the linearity. Since the names of the variables z and w are not important (they are never used), the following notation is introduced: $[\mathbf{t}] = \lambda w.\mathbf{t}$, called the 'canon' of $\mathbf{t}$, and $\{\mathbf{t}\} = (\mathbf{t})\ z$, called the 'cocanon' of $\mathbf{t}$, where w and z are fresh variables.

So the Hadamard gate is encoded as:

$$had = \lambda x.\{((x)\ [\frac{1}{\sqrt{2}}.\mathbf{false} - \frac{1}{\sqrt{2}}.\mathbf{true}])\ [\frac{1}{\sqrt{2}}.\mathbf{false} + \frac{1}{\sqrt{2}}.\mathbf{true}]\}$$

Then $(had)\ \mathbf{false} \to^* \frac{1}{\sqrt{2}}.\mathbf{false} + \frac{1}{\sqrt{2}}.\mathbf{true}$ and $(had)\ \mathbf{true} \to^* \frac{1}{\sqrt{2}}.\mathbf{false} - \frac{1}{\sqrt{2}}.\mathbf{true}$, as expected.

Analogously, the R_θ gate can be encoded as $R_\theta = \lambda x.\{((x)\ [e^{i\theta}.\mathbf{true}])\ [\mathbf{false}]\}$. Notice that the canon in the **false** branch is also needed to match the outer cocanon.

Two-qubits encoding. Since λ_{lin} application is bilinear, the usual encoding for tuples can serve as an encoding for the bilinear operation $\otimes$:

$$\otimes = \lambda x.\lambda y.\lambda f.((f)\ x)\ y$$

To put gates together it can be used $\bigotimes = \lambda f.\lambda g.\lambda x.((\otimes)\ ((f)\ ((\mathbf{true})\ x)))\ ((g)\ ((\mathbf{false})\ x))$. To make the terms more readable it is used the infix notation: $((\otimes)\ \mathbf{t})\ \mathbf{r} = \mathbf{t} \otimes \mathbf{r}$ and $((\bigotimes)\ had)\ had = had \bigotimes had$.

With this definitions, the *cnot* gate can be encoded as follows:

$$cnot = \lambda x.((\mathbf{true})\ x) \otimes ((((\mathbf{true})\ x)\ ((not)\ ((\mathbf{false})\ x)))\ ((\mathbf{false})\ x))$$

This way to encode quantum computing in λ_{lin} suggest that it can be used for quantum computing, since any quantum algorithm can be expressed on it. However notice that fixing this encoding, not every term in the calculus will represent a quantum program. Just by restricting the calculus it would be allowed to consider it as a quantum λ-calculus. Hence our motivations to study the norm of lineal terms.

1.3 Plan of the thesis

Chapter 2: In this chapter the design choices of λ_{lin} are fully studied and compared with those of the algebraic λ-calculus (λ_{alg}) [Vaux, 2009], a fragment and extension of the realms of differential λ-calculus [Ehrhard and Regnier, 2003]. Both calculi are algebraic: they have an additive structure and a scalar multiplication, and their set of term is closed under linear combinations. However, they have been constructed using different approaches: λ_{alg} is a call-by-name language while λ_{lin} is call-by-base; the first considers algebraic equalities while the second considers rewriting rules.

We study how these different approaches relate to each another. To this end, we propose four canonical languages, each based on one of the options: the call-by-name versus call-by basis, equality versus algebraic rewriting. We show that these different languages simulate each other.

This chapter had led to the following papers: [Díaz-Caro, Perdrix, Tasson, and Valiron, 2010, 2011].

Chapter 3: Here we introduce *Scalar*, an extension of System F which allows to keep track of the scalars in the terms. If $\mathbf{t}$ and $\mathbf{u}$ have type T, then $\alpha.\mathbf{t}+\beta.\mathbf{u}$ has type $(\alpha+\beta).T$. The type system has some direct applications, such as the ability to determine when a given term will reduce to a barycentric ($\sum_i \alpha_i = 1$) distribution of terms. We show that this type system has both the subject reduction and the strong normalisation properties, which are the main technical results of this chapter. The strong normalisation entails a significant simplification of λ_{lin}, removing the need for the restrictions in the reduction rules, as discussed in Section 1.2.

This chapter had led to the following papers: [Arrighi and Díaz-Caro, 2011a,b].

Chapter 4: In this chapter λ^{add} is defined: the confluent, additive fragment of λ_{lin}, typed with the *Additive* type system, which includes sums of types as a reflection of those in the terms. After proving subject reduction for this system, we study the role of sums within the calculus by interpreting λ^{add} into System F with pairs. It is shown that this calculus can be interpreted as System F with an associative and commutative pair constructor, which is distributive under application. This translation leads to the strong normalisation proof for this system, which will set the base for proving this property in the system of Chapter 6.

This chapter had led to the following paper: [Díaz-Caro and Petit, 2010].

Chapter 5: In this chapter the first fully *Vectorial* type system is defined. We combine the approaches of Chapters 3 and 4 with the aim of statically describe the linear combinations of terms resulting from the reduction of terms. This gives rise to an original type theory where types, in the same way as terms, can be superposed into linear combinations. Some applications to quantum computing are shown and we discuss the strengths and weaknesses of it. In particular, we show that only a weak version of the subject reduction property can be proved in this Curry style version, suggesting that we ought to move to Church style in order to have the full property, which we do in the following chapters. We also provide an original proof of strong normalisation, which will serve as a base to prove this property for the system of Chapter 7.

This chapter had led to the following paper: [Arrighi, Díaz-Caro, and Valiron, 2011b].

Chapter 6: In this chapter

Before moving to a Church version of *Vectorial*, as suggested in Chapter 5, we explore an alternative, simpler path: an extension of *Additive* for the full calculus. This extension has the advantage of

having sums in the types but no scalars, which allows, as shown in Chapter 4, an encoding in System F with pairs. The system defined is called λ^{CA}, and is an explicitly typed version of λ_{lin}, which deals with the interaction between scalars and additions by approximating the scalars to natural numbers. This system is similar to *Vectorial*. The main differences are, first that this is in Church style, avoiding the problems mentioned in Chapter 5, and second that despite the fact that this provides strong normalisation, and so is a good alternative to have a simplified language, its types only provide an approximation of the vectors. Instead of any ring it takes the semi-ring of non-negative real numbers.

This chapter had led to the following paper: [Buiras, Díaz-Caro, and Jaskelioff, 2011].

Chapter 7: We define *Lineal*, an explicitly typed algebraic λ-calculus based on λ_{lin} and *Vectorial*, with the subject reduction and strong normalisation properties. This language allows arbitrary linear combination of terms. The type system is a static analysis tool to describe the vectorial shape of the normal form of the terms. It keeps track of the "amplitude of a term", *i.e.* if $\mathbf{t}$ and $\mathbf{r}$ are term of the same type T, $\alpha.\mathbf{t} + \beta.\mathbf{r}$ have type $(\alpha + \beta).T$. Also, it keep track of the "direction of a term", *i.e.* if $\mathbf{t}$ and $\mathbf{r}$ have types T and R respectively, $\alpha.\mathbf{t} + \beta.\mathbf{r}$ has type $\alpha.T + \beta.R$. This calculus is able to encode matrices and vectors, just as *Vectorial*, but with the advantage of having subject reduction: all the problems described in Chapter 5 are solved by using explicit types, and a subtyping system.

A paper based on this chapter is under preparation.

Chapter 8: In this chapter we briefly summarise the achievements and show some clues for future work.

Chapter 2

Call-by-name, call-by-base and the reduction/equality duality

Résumé du Chapitre

Nous examinons la relation entre le λ-calcul algébrique, un fragment du λ-calcul différentiel, et λ_{lin}. Les deux calculs sont algébriques : chacun est équipé d'une structure additive et d'une structure de multiplication par un scalaire, et leur ensemble de termes est clos par combinaisons linéaires. Toutefois, les deux langages ont été construits en utilisant des approches différentes : le premier est un langage en appel-par-nom tandis que le second est en appel-par-base; le premier considère des égalités algébriques alors que le second considère des règles de réécriture.

Nous étudions comment les différentes approches se rapportent l'une de l'autre. À cette fin, nous proposons quatre langages canoniques chacun basés sur un des choix possibles : l'appel par nom par rapport à l'appel par base, l'égalité algébrique versus réécriture. Nous montrons que les différentes langages se simulent entre eux. ∎

We analyse the different decision choices that make λ_{lin} (*cf.* Section 1.2) different from the *algebraic λ-calculus*, λ_{alg} [Vaux, 2009]. The latter is another algebraic extension to the λ-calculus introduced independently in the context of linear logic as a fragment of the differential λ-calculus [Ehrhard and Regnier, 2003]: the algebraic structure allows to gather in a non deterministic manner different terms, *i.e.* each term represent one possible execution.

This chapter is devoted to analysing how these different approaches relate one to the other. To this end, we propose four canonical languages based on each of the possible choices: call-by-name versus call-by-base, algebraic equality versus algebraic rewriting. We show that the various languages simulate one another.

Four languages with different behaviours. In both languages, functions which are linear combinations of terms are interpreted pointwise: $(\alpha.\mathbf{t} + \beta.\mathbf{r})\ x = \alpha.(\mathbf{t})\ x + \beta.(\mathbf{r})\ x$, where "." denotes the scalar multiplication. The two languages differ on the treatment of the arguments. In λ_{lin}, the evolution is call-by-base (*cf.* Section 1.2.1) and in order to deal with the algebraic structure, any function is considered

as a linear map: $(\mathbf{t})\ (\alpha.x + \beta.y) \to^* \alpha.(\mathbf{t})\ x + \beta.(\mathbf{t})\ y$, reflecting the fact that any quantum evolution is a linear map. In the opposite, λ_{alg} has a call-by-name evolution: $(\lambda x.\mathbf{t})\ \mathbf{r} \to \mathbf{t}[\mathbf{r}/x]$, without any restriction on $\mathbf{r}$. As a consequence, the evolutions are different as illustrated by the following example. In λ_{lin}, $(\lambda x.(x)\ x)\ (\alpha.y+\beta.z) \to^* \alpha.(y)\ y+\beta.(z)\ z$ while in λ_{alg}, $(\lambda x\,(x)\ x)\ (\alpha.y+\beta.z) \to (\alpha.y+\beta.z)\ (\alpha.y+\beta.z) = \alpha^2.(y)\ y + (\alpha \times \beta).(y)\ z + (\beta \times \alpha).(z)\ y + \beta^2.(z)\ z$.

Because they were designed for different purposes, another difference appears between the two languages: the way the algebraic part of the calculus is treated. In λ_{lin}, the algebraic structure is captured with a rewrite system, whereas in λ_{alg} terms are considered up to algebraic equivalence.

The two choices – call-by-base versus call-by-name, algebraic equality versus algebraic reduction – allow one to construct four possible calculi. We name them in Figure 2.1, where they are presented according to their evolution policy and the way they take care of the algebraic part of the language.

Besides, we slightly modify the operational semantics. The unique modification to λ_{alg} consists in avoiding reduction under λ, so that for any $\mathbf{t}$, $\lambda x.\mathbf{t}$ is in normal form. As a consequence, the λ-abstraction is not linear anymore: $\lambda x.(\alpha.\mathbf{t} + \beta.\mathbf{r}) \neq \alpha.\lambda x.\mathbf{t} + \beta.\lambda x.\mathbf{r}$. In λ_{lin}, we change the original restrictions, which where there for confluence reasons, to make it more coherent with a call-by-base evaluation. The restriction in λ_{alg} is a common restriction: reducing under λ could be considered as "optimising the program. Concerning λ_{lin}, waving the restrictions make sense when confluence can be ensured by other means, as will be the case in the following chapters.

Contribution of this chapter: relation between the four languages through simulation. Although these languages behave differently, we show in this chapter that they simulate each other. This result reveals strong connections between two distinct research areas and unifies the works done in λ_{lin} with those done in λ_{alg}, *e.g.* [Vaux, 2007, Ehrhard and Regnier, 2003, Ehrhard, 2003, 2005, Tasson, 2009, Pagani and Tranquilli, 2009, Ehrhard, 2010, Pagani and Rocca, 2010].

We show that call-by-name algebraic λ-calculi simulate call-by-base ones and *vice-versa* by extending the continuation passing style (CPS) [Plotkin, 1975] to the algebraic case. We also provide simulations between algebraic equality and algebraic reduction in both directions. The simulations proved are summed up in Figure 2.2. The solid arrows stand for theorems that do not require confluence in their hypothesis whereas the dashed arrows stand for theorems that do.

Consistency. Without restrictions on the set of terms, both algebraic reductions and algebraic equalities cause problems of consistency, albeit differently.

Taking up again the example of Section 1.2, let $Y_{\mathbf{b}} = (\lambda x.(\mathbf{b} + (x)\ x))\ \lambda x.(\mathbf{b} + (x)\ x)$. In a system with algebraic reduction, the term $Y_{\mathbf{b}} - Y_{\mathbf{b}}$ reduces to 0, but also reduces to $\mathbf{b} + Y_{\mathbf{b}} - Y_{\mathbf{b}}$ and hence to $\mathbf{b}$, breaking confluence. To solve this issue, several distinct techniques can be used to make an algebraic calculus confluent. The original technique in [Arrighi and Dowek, 2008], as presented in Section 1.2, were the restrictions (*) and (**) depicted in Figure 1.1. In the following chapters, type systems are set up which forbid diverging terms such as $Y_{\mathbf{b}}$, and so there is no need for these restrictions.

In a system with algebraic equalities, for any term $\mathbf{c}$, any term $\mathbf{b}$ reduces to $\mathbf{b}+Y_{\mathbf{c}-\mathbf{b}}-Y_{\mathbf{c}-\mathbf{b}}$, therefore to $\mathbf{c}$. In λ_{alg} a restriction to positive scalars, thus a semi-ring without additive inverse instead of a ring, is proposed to solve the problem. However such a solution does not work in a system with algebraic reduction (*cf.* Section 2.2).

In this chapter we do not make a choice *a priori*; instead we show that the simulations between the

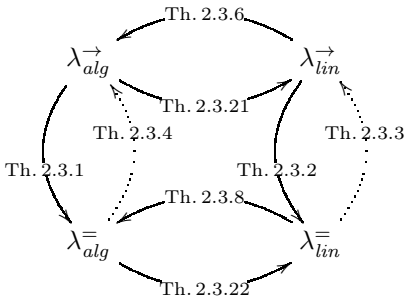
four calculi are correct, providing a general enough methodology to work in a large variety of restrictions on the language. Therefore, we do not force one specific method to make the calculi consistent, leaving the choice to the user.

Plan of the chapter. In Section 2.1, we define the set of terms and the rewrite systems we consider in this chapter. In Section 2.2, we discuss the confluence of the algebraic rewrite systems. Section 2.3 is concerned with the actual simulations. In Section 2.3.1 we consider the correspondence between algebraic reduction and algebraic equality whereas in Section 2.3.2 and 2.3.3 we consider the distinction call-by-name versus call-by-base. In Section 2.3.4, we show how the simulations can compose to obtain the correspondence between any two of the four languages. In Section 2.4 we conclude by providing some paths of open problems for future work.

	call-by-name	call-by-base
algebraic reduction	$\lambda^{\rightarrow}_{alg}$	$\lambda^{\rightarrow}_{lin}$
algebraic equality	$\lambda^{=}_{alg}$	$\lambda^{=}_{lin}$

cf. Definition 2.1.1

Figure 2.1: The four algebraic λ-calculi



$A \to B$ means "A is simulated by B"

Figure 2.2: Relations between the languages

2.1 Algebraic λ-calculi

The languages λ_{lin} and λ_{alg} share the same syntax, defined as follows:

$$\begin{array}{rcll} \mathbf{t},\mathbf{r},\mathbf{s} & ::= & \mathbf{b} \mid (\mathbf{t})\ \mathbf{r} \mid \alpha.\mathbf{t} \mid \mathbf{t}+\mathbf{r} & \text{(terms)}, \\ \mathbf{u},\mathbf{v},\mathbf{w} & ::= & \mathbf{0} \mid \mathbf{b} \mid \alpha.\mathbf{v} \mid \mathbf{v}+\mathbf{w} & \text{(values)}, \\ \mathbf{b} & ::= & x \mid \lambda x.\mathbf{t} & \text{(base terms)}. \end{array}$$

where α ranges over a ring $(\mathcal{S},+,\times)$, called *the ring of scalars*.

We provide a complete formalisation of the rewrite rules and show how they relate to each other. We summarise in Figure 2.3 all the rewrite rules that are being used. They are grouped with respect to their intuitive meaning[1].

The languages λ_{lin} and λ_{alg} have two distinct opinions on the nature of the addition and scalar multiplication of lambda-terms. In λ_{lin}, the terms are purely syntactic and have no other meaning than the one given by the rewrite system (with the exception of the associativity and commutativity). In particular, $\alpha.(\mathbf{t}+\mathbf{r})$ and $\alpha.\mathbf{t}+\alpha.\mathbf{r}$ do not represent the same term: the former rewrites to the latter but this is not reversible, indicating that it is preferred in a more canonical presentation. In λ_{alg}, the lambda-terms are considered as being part of a mathematical vector space; therefore addition is the actual addition

[1] In [Arrighi and Dowek, 2008], the rule $\alpha.(\beta.\mathbf{t}) \to (\alpha \times \beta).\mathbf{t}$ is included in the Elementary group (*cf.* Figure 1.1). Here it is placed in the factorisation group since it is required to close the critical pair $\alpha.\mathbf{t}+\alpha.\mathbf{t} \to (1+1).(\alpha.\mathbf{t})$ and $\alpha.\mathbf{t}+\alpha.\mathbf{t} \to (\alpha+\alpha).\mathbf{t}$.

Specific rules for λ_{alg}

Call-by-name (β_n)

$$(\lambda x.\mathbf{t})\ \mathbf{r} \to \mathbf{t}[\mathbf{r}/x]$$

Linearity of the application (A)

$$\begin{aligned}(\mathbf{t}+\mathbf{r})\ \mathbf{s} &\to (\mathbf{t})\ \mathbf{s}+(\mathbf{r})\ \mathbf{s}\\ (\alpha.\mathbf{t})\ \mathbf{r} &\to \alpha.(\mathbf{t})\ \mathbf{r}\\ (\mathbf{0})\ \mathbf{t} &\to \mathbf{0}\end{aligned}$$

Specific rules for λ_{lin}

Call-by-base (β_b)

$$(\lambda x.\mathbf{t})\ \mathbf{b} \to \mathbf{t}[\mathbf{b}/x]$$

Context rule ($\xi_{\lambda_{lin}}$)

$$\frac{\mathbf{t}\to\mathbf{t}'}{(\mathbf{v})\ \mathbf{t}\to(\mathbf{v})\ \mathbf{t}'}$$

Linearity of the application

Left linearity (A_l)

$$\begin{aligned}(\mathbf{t}+\mathbf{r})\ \mathbf{v} &\to (\mathbf{t})\ \mathbf{v}+(\mathbf{r})\ \mathbf{v}\\ (\alpha.\mathbf{t})\ \mathbf{v} &\to \alpha.(\mathbf{t})\ \mathbf{v}\\ (\mathbf{0})\ \mathbf{v} &\to \mathbf{0}\end{aligned}$$

Right linearity (A_r)

$$\begin{aligned}(\mathbf{b})\ (\mathbf{t}+\mathbf{r}) &\to (\mathbf{b})\ \mathbf{t}+(\mathbf{b})\ \mathbf{r}\\ (\mathbf{b})\ (\alpha.\mathbf{t}) &\to \alpha.(\mathbf{b})\ \mathbf{t}\\ (\mathbf{b})\ \mathbf{0} &\to \mathbf{0}\end{aligned}$$

Common rules

Ring rules ($L = Asso \cup Com \cup F \cup S$)

Associativity ($Asso$)

$$\begin{aligned}\mathbf{t}+(\mathbf{r}+\mathbf{s}) &\to (\mathbf{t}+\mathbf{r})+\mathbf{s}\\ (\mathbf{t}+\mathbf{r})+\mathbf{s} &\to \mathbf{t}+(\mathbf{r}+\mathbf{s})\end{aligned}$$

Commutativity (Com)

$$\mathbf{t}+\mathbf{r} \to \mathbf{r}+\mathbf{t}$$

Factorization (F)

$$\begin{aligned}\alpha.\mathbf{t}+\beta.\mathbf{t} &\to (\alpha+\beta).\mathbf{t}\\ \alpha.\mathbf{t}+\mathbf{t} &\to (\alpha+1).\mathbf{t}\\ \mathbf{t}+\mathbf{t} &\to (1+1).\mathbf{t}\\ \alpha.(\beta.\mathbf{t}) &\to (\alpha\times\beta).\mathbf{t}\end{aligned}$$

Simplification (S)

$$\begin{aligned}\alpha.(\mathbf{t}+\mathbf{r}) &\to \alpha.\mathbf{t}+\alpha.\mathbf{r}\\ 1.\mathbf{t} &\to \mathbf{t}\\ 0.\mathbf{t} &\to \mathbf{0}\\ \alpha.\mathbf{0} &\to \mathbf{0}\\ \mathbf{0}+\mathbf{t} &\to \mathbf{t}\end{aligned}$$

Context rules (ξ)

$$\frac{\mathbf{t}\to\mathbf{t}'}{(\mathbf{t})\ \mathbf{r}\to(\mathbf{t}')\ \mathbf{r}} \qquad \frac{\mathbf{t}\to\mathbf{t}'}{\mathbf{t}+\mathbf{r}\to\mathbf{t}'+\mathbf{r}} \qquad \frac{\mathbf{r}\to\mathbf{r}'}{\mathbf{t}+\mathbf{r}\to\mathbf{t}+\mathbf{r}'} \qquad \frac{\mathbf{t}\to\mathbf{t}'}{\alpha.\mathbf{t}\to\alpha.\mathbf{t}'}$$

Figure 2.3: Rewrite rules of $\lambda^{\to}_{lin}$ and $\lambda^{\to}_{alg}$

in the space and terms are considered up to the equality in the vector space. For example, $\alpha.(\mathbf{t}+\mathbf{r})$ and $\alpha.\mathbf{t}+\alpha.\mathbf{r}$ are two representations of the same term.

In this chapter, no assumptions are made and the distinction call-by-name/call-by-base and the distinction equality/reduction are separately considered. Therefore four languages are developed: a call-

by-base language $\lambda^{=}_{lin}$ with algebraic equality, a call-by-base language $\lambda^{\to}_{lin}$ with algebraic reduction, a call-by-name language $\lambda^{=}_{alg}$ with algebraic equality and a call-by-name language $\lambda^{\to}_{alg}$ with algebraic reduction.

These four languages are summarised in Figure 2.1 and formalised in Definition 2.1.1. Informally, we use the notation $\to_a$ for the algebraic reductions in λ_{alg}, $\to_\ell$ for the algebraic reductions in λ_{lin} and $\to_{\beta_n}$ and $\to_{\beta_b}$ to the β-reduction in λ_{alg} and λ_{lin} respectively. It is also formalised in the Definition 2.1.1.

Definition 2.1.1. *The following notations for the rewrite systems obtained by combining the rules described in Figure 2.3 are used:*

$$\begin{array}{llllllllll} \to_a & := & A \cup L \cup \xi & \to_\ell & := & A_l \cup A_r \cup L \cup \xi \cup \xi_{\lambda_{lin}} & \to_{\beta_b} & := & \beta_b \cup \xi \cup \xi_{\lambda_{lin}} \\ \to^{=}_a & := & (\to_a)_\leftrightarrow & \to^{=}_\ell & := & (\to_\ell)_\leftrightarrow & \to_{\beta_n} & := & \beta_n \cup \xi \end{array}$$

The four languages of Figure 2.1, and their associated rewrite systems, are defined as follows:

Language	*Corresponding Rewrite System*
$\lambda^{\to}_{lin}$	$\to_{\ell+\beta} := (\to_\ell) \cup (\to_{\beta_b})$
$\lambda^{=}_{lin}$	$\to^{=}_{\ell+\beta} := (\to^{=}_\ell) \cup (\to_{\beta_b})$
$\lambda^{\to}_{alg}$	$\to_{a+\beta} := (\to_a) \cup (\to_{\beta_n})$
$\lambda^{\to}_{alg}$	$\to^{=}_{a+\beta} := (\to^{=}_a) \cup (\to_{\beta_n})$

2.2 Discussion on consistency and confluence

2.2.1 Local confluence

In this section we show that the four languages $\lambda^{\to}_{lin}$, $\lambda^{\to}_{alg}$, $\lambda^{=}_{lin}$, and $\lambda^{=}_{alg}$ are locally confluent. We first concentrate on the algebraic rules. For each of these calculi, we use the reductions describing the algebraic structure: $\to_a$ and $\to_\ell$ correspond to an oriented rewriting description whereas $\to^{=}_a$ and $\to^{=}_\ell$ correspond to a description by equalities (since every rewrite rule can be reversed, *cf.* Definition 2.1.1).

Lemma 2.2.1. *The rewrite systems $\to_a$, $\to_\ell$, $\to^{=}_a$ and $\to^{=}_\ell$ are locally confluent.*

Proof. For $\to_\ell$ and $\to_a$, we give a semi-automatised proof in the theorem prover Coq [Coq Dev. Team, 2009]. The interested reader can find the proof in [Valiron, 2011a] which is sketched in Appendix A.9. Since for any rewrite system R, its symmetric closure $R_\leftrightarrow$ is trivially locally confluent, both $\to^{=}_a$ and $\to^{=}_\ell$ are locally confluent. □

The rewrites systems considered in this chapter are also locally confluent in the presence of the β-rewrite rules.

Lemma 2.2.2 (Local confluence)**.** *The four languages in Figure 2.1 are locally confluent.*

Proof (Sketch). The local confluence of the algebraic fragment is proven in Lemma 2.2.1. The beta-reduction is confluent using a straightforward extension of the confluence of lambda calculus. Finally, the beta-reduction and the algebraic fragments commute, making each rewrite system locally confluent. The full proof is developed in Appendix A.1. □

2.2.2 Simulations and the confluence issue

In this section, we show that the algebraic fragments are confluent modulo associativity and commutativity. Concerning the full languages, we show that they are either not confluent or trivially confluent (in the sense that any term is reducing to any other). As a consequence, we introduce a generic notion of language fragment to describe confluent and consistent sub-languages. In particular, fragments are used in simulations theorems in section 2.3 for abstractly representing confluent sub-languages.

The algebraic fragment. It is clear that neither $\to_a$ nor $\to_\ell$ is strongly normalising: with both systems one can go back and forth between $\mathbf{t}+\mathbf{r}$ and $\mathbf{r}+\mathbf{t}$. They are however strongly normalising *modulo associativity and commutativity* in the sense that any rewrite sequence consists eventually of terms that are equal modulo associativity and commutativity. On the contrary, the rewrite systems $\to_a^=$ and $\to_\ell^=$ are not.

In order to formalise this, AC denotes the system generated by $AC = Asso \cup Com$ and $\overline{R}$ the rewrite system obtained by taking off the rules $Asso$ and Com where R stands for $\to_a$ or $\to_\ell$. Hence, $\overline{\to_a}$ stands for the system generated by $A \cup S \cup F \cup \xi$ and $\overline{\to_\ell}$ for the system generated by $A_l \cup A_r \cup S \cup F \cup \xi \cup \xi_{\lambda_{lin}}$.

Definition 2.2.3. *Let R be either $\to_\ell$ or $\to_a$ and $\mathbf{t}_1\, R\, \mathbf{t}_2\, R \ldots$ be a reduction sequence (finite or not) characterised by the family of terms $\{\mathbf{t}_i\}_i$ and the family of rules $\{R_i\}_i$ used to go from $\mathbf{t}_i$ to $\mathbf{t}_{i+1}$, where R_i stands for a fixed rule in R. We say that the reduction is* AC-finite *if $\{R_i\}_i \cap \overline{R}$ is finite. The* AC-length *of the rewrite sequence is the cardinal of the set $\{R_i\}_i \cap \overline{R}$. The rewrite system R is* AC-strongly-normalising *(AC-SN) if for any term $\mathbf{t}$, there exists a number n such that the AC-length of any rewrite sequence starting at $\mathbf{t}$ is less than n, in other words, the rewrite system R is AC-SN if every rewrite sequence is AC-finite. A term M is* AC-normal *with respect to a rewrite system R if any rewrite sequence starting with M consists only of rules AC.*

Theorem 2.2.4. *The systems $\to_a$ and $\to_\ell$ are AC-SN.*

Proof. We use the technique described in [Arrighi and Dowek, 2008]. An auxiliary *measure* is defined on terms by $|(\mathbf{t})\ \mathbf{r}| = (3|\mathbf{t}|+2)(3|\mathbf{r}|+2)$, $|\alpha.\mathbf{t}| = 1+2|\mathbf{t}|$, $|\mathbf{t}+\mathbf{r}| = 2+|\mathbf{t}|+|\mathbf{r}|$, $|\mathbf{0}| = 0$, $|\lambda x.\mathbf{t}| = 1$ and $|x| = 1$. This measure is preserved by rules AC and strictly decreasing on the other algebraic rules. □

Local confluence plus strong normalisation implies confluence (*cf.* for example [TeReSe, 2003]).

Corollary 2.2.5. *The rewrite systems $\to_a$ and $\to_\ell$ are confluent, modulo AC.* □

The four calculi. Although we have proved that the four languages under consideration are locally confluent, neither $\lambda_{lin}^{\to}$ nor $\lambda_{alg}^{\to}$ is confluent: In each one, the term $Y_{\mathbf{b}} - Y_{\mathbf{b}}$ rewrites both to $\mathbf{0}$ and $\mathbf{b}$, where $Y_{\mathbf{b}} = (\lambda x.(\mathbf{b} + (x)\ x))\ \lambda x.(\mathbf{b} + (x)\ x)$.

Remark 2.2.6. *Regarding $\lambda_{lin}^{=}$ and $\lambda_{alg}^{=}$, without restriction both are trivially confluent since for all terms $\mathbf{t}$ and $\mathbf{r}$, $\mathbf{t}$ reduces to $\mathbf{r}$: $\mathbf{t} \to^= \mathbf{t}+Y_{\mathbf{r}-\mathbf{t}}-Y_{\mathbf{r}-\mathbf{t}} \to^* \mathbf{r}$. Hence, with the algebraic equality, both languages can simulate any rewrite system.*

For getting back consistency, it is of course possible to modify the rewrite systems as it has been done in [Arrighi and Dowek, 2008] but it would break the correspondence between call-by-base and call-by-name. In this chapter we propose instead to restrict the set of terms. There are two known methods:

[Vaux, 2009] considers positive scalars (a semi-ring without additive inverse) on a language with algebraic equality. The restriction on scalars is enough for getting uniqueness of normal forms. Although this solves the consistency problem for the languages with algebraic equality, it does not give confluence for the languages with algebraic reduction. Indeed, consider the critical pair $Y_{\mathbf{t}} + Y_{\mathbf{t}} \to_\ell 2.Y_{\mathbf{t}}$, $Y_{\mathbf{t}} + Y_{\mathbf{t}} \to_{\beta_b} Y_{\mathbf{t}} + \mathbf{t} + Y_{\mathbf{t}} \to_\ell 2.Y_{\mathbf{t}} + \mathbf{t}$. The term $2.Y_{\mathbf{t}}$ can only produce an even number of $\mathbf{t}$'s: we cannot close the pair.

In the following chapters we use type systems for retrieving strong normalisation. This could be adapted to this chapter's setting, for example with a simple type system. We can get strong normalisation for well-typed terms and, since both $\lambda^{\to}_{lin}$ and $\lambda^{\to}_{alg}$ satisfy local confluence, retrieve the existence and unicity of normal forms for these languages. As the languages $\lambda^{=}_{lin}$ and $\lambda^{=}_{alg}$ have the same equational theory, this guarantees that they are well-behaved with respect to the type system.

The simulation theorems that we develop in this chapter are correct in an untyped setting (and in fact trivially true when we simulate a language with algebraic reduction with a language with algebraic equality as remarked above) but also true in any typed setting, provided that it satisfies subject reduction. Thus, we do not restrict the calculi *a priori*: instead, we propose a notion of *language fragments* to parametrise the simulation results. The definition of fragment is general enough to capture many settings: various typed systems, but also restrictions to a given set of terms such as the set of AC-SN terms.

We define formally a fragment in the following way:

Definition 2.2.7. *A* fragment *S of $\lambda^{\to}_{lin}$ (resp. $\lambda^{\to}_{alg}$) is a language defined on a subset of terms closed under $\to_{\ell+\beta}$-reduction (resp. $\to_{a+\beta}$-reduction). The rewrite system of S is inherited from the one of $\lambda^{\to}_{lin}$ (resp. $\lambda^{\to}_{alg}$).*

The definition of a fragment in the presence of algebraic equalities should be treated carefully. Indeed, note that the algebraic equalities are defined as $\mathbf{t} \to^{=} \mathbf{r}$ if and only if $\mathbf{t} \to \mathbf{r}$ or $\mathbf{r} \to \mathbf{t}$. As a consequence, for any subset S of terms closed under $\to^{=}$-reduction, if $\mathbf{t}$ is in S then for any $\mathbf{r}$ (in S or not), $\mathbf{t}+\mathbf{r}-\mathbf{r} \in S$ since $\mathbf{t} \to^{=} \mathbf{t} + \mathbf{r} - \mathbf{r}$. We therefore need to define the algebraic equality with respect to the particular subset of terms under consideration.

Definition 2.2.8. *A fragment S of $\lambda^{=}_{lin}$ (resp. $\lambda^{=}_{alg}$) is a fragment of $\lambda^{\to}_{lin}$ (resp. $\lambda^{\to}_{alg}$) together with an algebraic equality defined as $\mathbf{t} \to^{=}_{\ell} \mathbf{r}$ (resp. $\mathbf{t} \to^{=}_{a} \mathbf{r}$) if and only if $\mathbf{t}, \mathbf{r} \in S$ and $\mathbf{r} \to_\ell \mathbf{t}$ or $\mathbf{t} \to_\ell \mathbf{r}$ (resp. $\mathbf{r} \to_a \mathbf{t}$ or $\mathbf{t} \to_a \mathbf{r}$). The β-reduction is not modified.*

Remark 2.2.9. *When referring to a fragment of $\lambda^{=}_{lin}$ (resp. $\lambda^{=}_{alg}$), we use the abuse of notation $\to^{=}_{\ell}$ (resp. $\to^{=}_{a}$) for the restricted rewrite system, when the fragment under consideration is clear.*

2.3 Simulations

This section is concerned with the mutual simulations of the four languages.

The first class of problems relates algebraic reduction with algebraic equality. If simulating a language with algebraic reduction with a language with algebraic equality is not specially difficult, going in the opposite direction is not possible in general. Indeed, if $\mathbf{0} =_\ell Y_{\mathbf{b}} - Y_{\mathbf{b}} \to_{\beta_b} Y_{\mathbf{b}} + \mathbf{b} - Y_{\mathbf{b}} =_\ell \mathbf{b}$ is possible in $\lambda^{=}_{lin}$, (where $Y_{\mathbf{b}} = (\lambda x.(\mathbf{b} + (x)\ x))\ \lambda x.(\mathbf{b} + (x)\ x)$) it is difficult to see how one could make $\mathbf{0}$ go to $\mathbf{b}$ in $\lambda^{\to}_{lin}$ without further hypotheses. In this section, we show that a fragment of a language with algebraic equality can be simulated by the corresponding fragment with algebraic reduction provided that the latter is *confluent* (Theorems 2.3.3 and 2.3.4).

The second class of problems is concerned with call-by-base and call-by-name. In this chapter, the simulations of call-by-name by call-by-base and its reverse are treated using continuation passing style (CPS), extending the simulation techniques described in [Fischer, 1972, Plotkin, 1975] to the algebraic case (Theorems 2.3.6, 2.3.8, 2.3.21 and 2.3.22).

The results are summarised in Figure 2.2. Solid arrows correspond to results where no particular hypothesis on the language is made. Dashed arrows correspond to results where confluence is required.

2.3.1 Algebraic reduction versus algebraic equality

As the relation $\to_{\ell+\beta}$ is contained in $\to^{=}_{\ell+\beta}$ and the relation $\to_{a+\beta}$ is contained in $\to^{=}_{a+\beta}$, the first simulation theorems are trivial.

Theorem 2.3.1. *For any term* $\mathbf{t}$ *if* $\mathbf{t}\to_{a+\beta}\mathbf{r}$*, then* $\mathbf{t}\to^{=}_{a+\beta}\mathbf{r}$. □

Theorem 2.3.2. *For any term* $\mathbf{t}$ *if* $\mathbf{t}\to_{\ell+\beta}\mathbf{r}$*, then* $\mathbf{t}\to^{=}_{\ell+\beta}\mathbf{r}$. □

The simulations going in the other direction are only valid in the presence of confluence. In the following two theorems, the algebraic equality is defined with respect to the considered fragment (*cf.* Definition 2.2.8).

Theorem 2.3.3. *For any term* $\mathbf{t}$ *in a confluent fragment of* $\lambda^{\to}_{lin}$*, if* $\mathbf{t}\to^{=}_{\ell+\beta}{}^{*}\mathbf{v}$*, then* $\mathbf{t}\to^{*}_{\ell+\beta}\mathbf{v}'$*, with* $\mathbf{v}\to^{=}_{\ell}{}^{*}\mathbf{v}'$.

Proof. First note that a value can only reduce to another value. This follows from direct inspection of the rewriting rules. We proceed by induction on the length of the reduction.

- If $\mathbf{t}\to^{=}_{\ell+\beta}{}^{*}\mathbf{t}$, then choose $\mathbf{v}'=\mathbf{t}$ and note that $\mathbf{t}\to^{*}_{\ell+\beta}\mathbf{t}$.
- Assume the result true for $\mathbf{t}\to^{=}_{\ell+\beta}{}^{*}\mathbf{v}$: there is a value $\mathbf{v}'$ such that $\mathbf{t}\to^{*}_{\ell+\beta}\mathbf{v}'$ and $\mathbf{v}\to^{=}_{\ell}{}^{*}\mathbf{v}'$. Let $\mathbf{r}\to^{=}_{\ell+\beta}\mathbf{t}$. Case distinction:
  - $\mathbf{r}\to_{\beta_b}\mathbf{t}$, then $\mathbf{r}\to_{\beta_b}\mathbf{t}\to^{*}_{\ell+\beta}\mathbf{v}'$ which implies $\mathbf{r}\to^{*}_{\ell+\beta}\mathbf{v}'$.
  - $\mathbf{r}\to^{=}_{\ell}\mathbf{t}$, then either $\mathbf{r}\to_{\ell}\mathbf{t}$, and then this case is analogous to the previous one, or $\mathbf{t}\to_{\ell}\mathbf{r}$. Due to the confluence of the subset, there exists a term $\mathbf{s}$ such that $\mathbf{r}\to^{*}_{\ell+\beta}\mathbf{s}$ and $\mathbf{v}'\to^{*}_{\ell}\mathbf{s}$, implying that $\mathbf{s}$ is a value, thus $\mathbf{v}'\to^{=}_{\ell}{}^{*}\mathbf{s}$. Then we have $\mathbf{v}'\to^{=}_{\ell}{}^{*}\mathbf{s}$ and $\mathbf{v}\to^{=}_{\ell}{}^{*}\mathbf{v}'$, so $\mathbf{s}\to^{=}_{\ell}{}^{*}\mathbf{v}$, closing the case.

□

Theorem 2.3.4. *For any term* $\mathbf{t}$ *in a confluent fragment of* $\lambda^{\to}_{alg}$*, if* $\mathbf{t}\to^{=}_{a+\beta}{}^{*}\mathbf{v}$*, then* $\mathbf{t}\to^{*}_{a+\beta}\mathbf{v}'$*, with* $\mathbf{v}\to^{=}_{a}{}^{*}\mathbf{v}'$.

Proof. Similar to the previous theorem. □

2.3.2 Call-by-name simulates call-by-base

The simulation of λ_{lin} with λ_{alg}. To prove the simulation of λ_{lin} with λ_{alg} we introduce an algebraic extension of the continuation passing style encoding used to prove that call-by-name simulates call-by-value in the regular λ-calculus [Plotkin, 1975].

Let $[\![\cdot]\!] : \lambda_{lin} \to \lambda_{alg}$ be the following encoding where f, g and h are fresh variables.

$$\begin{array}{rclcrcl} [\![x]\!] & = & \lambda f.\,(f)\ x, & & [\![\mathbf{0}]\!] & = & \mathbf{0}, \\ [\![\lambda x.\mathbf{t}]\!] & = & \lambda f.(f)\ \lambda x.[\![\mathbf{t}]\!], & & [\![(\mathbf{t})\ \mathbf{r}]\!] & = & \lambda f.([\![\mathbf{t}]\!])\ \lambda g.([\![\mathbf{r}]\!])\ \lambda h.((g)\ h)\ f, \\ [\![\alpha.\mathbf{t}]\!] & = & \lambda f.(\alpha.[\![\mathbf{t}]\!])\ f, & & [\![\mathbf{t}+\mathbf{r}]\!] & = & \lambda f.([\![\mathbf{t}]\!]+[\![\mathbf{r}]\!])\ f. \end{array}$$

Let Ψ be the encoding for values defined by $\Psi(x) = x$, $\Psi(\mathbf{0}) = \mathbf{0}$, $\Psi(\lambda x.\mathbf{t}) = \lambda x.[\![\mathbf{t}]\!]$, $\Psi(\alpha.\mathbf{v}) = \alpha.\Psi(\mathbf{v})$, $\Psi(\mathbf{v}+\mathbf{w}) = \Psi(\mathbf{v}) + \Psi(\mathbf{w})$. Note that this encoding is compatible with substitution:

Lemma 2.3.5. *$[\![\mathbf{t}[\mathbf{b}/x]]\!] = [\![\mathbf{t}]\!][\Psi(\mathbf{b})/x]$ with $\mathbf{b}$ a base term.*

Proof. Induction on $\mathbf{t}$. *cf.* Appendix A.2. □

Using this encoding, we can simulate $\lambda^{\to}_{lin}$ with $\lambda^{\to}_{alg}$, as formalised in the following theorem. The proof is developed in the second part of this section.

Theorem 2.3.6 (Simulation)**.** *For any term $\mathbf{t}$, if $\mathbf{t}\to^*_{\ell+\beta}\mathbf{v}$ where $\mathbf{v}$ is a value, then $([\![\mathbf{t}]\!])\ \lambda x.x\to^*_{a+\beta}\Psi(\mathbf{v})$.*

Example 2.3.7. *For any terms $\mathbf{t}$ and $\mathbf{r}$, let $\langle \mathbf{t}, \mathbf{r}\rangle := \lambda y.((y)\ \mathbf{t})\ \mathbf{r}$. Let copy be the term $\lambda x.\langle x, x\rangle$, and let $\mathbf{u} = \lambda x.\mathbf{r}_1$ and $\mathbf{v} = \lambda x.\mathbf{r}_2$ be two values. Then $(\mathsf{copy})\ (\mathbf{u}+\mathbf{v})\to^*_{\ell+\beta}\langle \mathbf{u}, \mathbf{u}\rangle + \langle \mathbf{v}, \mathbf{v}\rangle$ and $(\mathsf{copy})\ (\mathbf{u}+\mathbf{v})\to^*_{a+\beta}\langle \mathbf{u}+\mathbf{v}, \mathbf{u}+\mathbf{v}\rangle$. We consider the simulation $\lambda^{\to}_{lin}$ to $\lambda^{\to}_{alg}$.*

$$\begin{aligned} [\![(\mathsf{copy})\ (\mathbf{u}+\mathbf{v})]\!] &= \lambda f.([\![\mathsf{copy}]\!])\ \lambda g.([\![\mathbf{u}+\mathbf{v}]\!])\ \lambda h.((g)\ h)\ f \\ [\![\mathsf{copy}]\!] &= \lambda f.(f)\ \lambda x.[\![\langle x, x\rangle]\!] \\ [\![\langle \mathbf{t}, \mathbf{r}\rangle]\!] &= \lambda f.(f)\ \Psi(\langle \mathbf{t}, \mathbf{r}\rangle) \\ [\![\mathbf{u}+\mathbf{v}]\!] &= \lambda f.([\![\mathbf{u}]\!]+[\![\mathbf{v}]\!])\ f \\ [\![\mathbf{u}]\!] &= \lambda g.(g)\ \Psi(\mathbf{u}) \end{aligned}$$

We now rewrite $\mathbf{t} = ([\![(\mathsf{copy})\ (\mathbf{u}+\mathbf{v})]\!])\ \lambda z.z$ in $\lambda^{\to}_{alg}$.

$$\begin{array}{rll} \mathbf{t} \to_{a+\beta} & ([\![\mathsf{copy}]\!])\ \lambda g.([\![\mathbf{u}+\mathbf{v}]\!])\ \lambda h.((g)\ h)\ \lambda z.z & \\ = & (\lambda f.(f)\ \lambda x.[\![\langle x, x\rangle]\!])\ \lambda g.([\![\mathbf{u}+\mathbf{v}]\!])\ \lambda h.((g)\ h)\ \lambda z.z & \\ \to_{a+\beta} & (\lambda g.([\![\mathbf{u}+\mathbf{v}]\!])\ \lambda h.((g)\ h)\ \lambda z.z)\ \lambda x.[\![\langle x, x\rangle]\!] & \\ \to_{a+\beta} & ([\![\mathbf{u}+\mathbf{v}]\!])\ \lambda h.((\lambda x.[\![\langle x, x\rangle]\!])\ h)\ \lambda z.z & \\ \to_{a+\beta} & ([\![\mathbf{u}]\!]+[\![\mathbf{v}]\!])\ \lambda h.((\lambda x.[\![\langle x, x\rangle]\!])\ h)\ \lambda z.z & \\ \to_{a+\beta} & ([\![\mathbf{u}]\!])\ \lambda h.((\lambda x.[\![\langle x, x\rangle]\!])\ h)\ \lambda z.z + ([\![\mathbf{v}]\!])\ \lambda h.((\lambda x.[\![\langle x, x\rangle]\!])\ h)\ \lambda z.z & \\ \to^*_{a+\beta} & (\lambda h.((\lambda x.[\![\langle x, x\rangle]\!])\ h)\ \lambda z.z)\ \Psi(\mathbf{u}) + (\lambda h.((\lambda x.[\![\langle x, x\rangle]\!])\ h)\ \lambda z.z)\ \Psi(\mathbf{v}) & (*) \\ \to^*_{a+\beta} & ((\lambda x.[\![\langle x, x\rangle]\!])\ \Psi(\mathbf{u}))\ \lambda z.z + ((\lambda x.[\![\langle x, x\rangle]\!])\ \Psi(\mathbf{v}))\ \lambda z.z & \\ \to^*_{a+\beta} & ([\![\langle x, x\rangle]\!][\Psi(\mathbf{u})/x])\ \lambda z.z + ([\![\langle x, x\rangle]\!][\Psi(\mathbf{v})/x])\ \lambda z.z & \\ \textit{(Lemma 2.3.5)} \to^*_{a+\beta} & ([\![\langle \mathbf{u}, \mathbf{u}\rangle]\!])\ \lambda z.z + ([\![\langle \mathbf{v}, \mathbf{v}\rangle]\!])\ \lambda z.z & \\ \to^*_{a+\beta} & (\lambda z.z)\ \Psi(\langle \mathbf{u}, \mathbf{u}\rangle) + (\lambda z.z)\ \Psi(\langle \mathbf{v}, \mathbf{v}\rangle) & (**) \\ \to^*_{a+\beta} & \Psi(\langle \mathbf{u}, \mathbf{u}\rangle) + \Psi(\langle \mathbf{v}, \mathbf{v}\rangle) & \\ = & \Psi(\langle \mathbf{u}, \mathbf{u}\rangle + \langle \mathbf{v}, \mathbf{v}\rangle) & \end{array}$$

Similarly, one can relate fragments of $\lambda^{=}_{alg}$ to fragments of $\lambda^{=}_{lin}$ as follows.

Theorem 2.3.8 (Simulation)**.** *For any two fragments S_ℓ of $\lambda^{=}_{lin}$ and S_a of $\lambda^{=}_{alg}$ such that $\forall \mathbf{t} \in S_\ell$, $([\![\mathbf{t}]\!])\ \lambda x.x \in S_a$, and for any term $\mathbf{t}$ in S_ℓ, if $\mathbf{t}\to^{=}_{\ell+\beta}{}^*\mathbf{v}$ where $\mathbf{v}$ is a value, then $([\![\mathbf{t}]\!])\ \lambda x.x\to^{=}_{a+\beta}{}^*\Psi(\mathbf{v})$.*

Again, the proof is developed later in the section.

Remark 2.3.9. *As we already noted several times, without restricting the languages, Theorem 2.3.8 would be trivial. Any term reducing to any other one, the desired reduction would be of course valid without restriction. This theorem shows that if the calculi are restricted to fragments, the result is still true.*

Note that the fragments of algebraic λ-calculi usually considered in the literature satisfy the property that if $\mathbf{t}$ *is in a fragment of* $\lambda^{=}_{lin}$ *then* $(\llbracket\mathbf{t}\rrbracket)\ \lambda x.x$ *is in the corresponding fragment of* $\lambda^{=}_{alg}$*. This is for instance the case of the fragments obtained by considering non-negative scalars like in [Vaux, 2009], this is also true for the various typing systems developed in the following chapters.*

Once a term is encoded it can be reduced either by $\to^{*}_{a+\beta}$ or by $\to^{*}_{\ell+\beta}$ (respectively $\to^{=}_{a+\beta}{}^{*}$ or $\to^{=}_{\ell+\beta}{}^{*}$) without distinction, and still obtain the same result. We state this fact as a corollary:

Corollary 2.3.10 (Indifference)**.**

- *For any term* $\mathbf{t}$*, if* $\mathbf{t}\to^{*}_{\ell+\beta}\mathbf{v}$ *where* $\mathbf{v}$ *is a value, then* $(\llbracket\mathbf{t}\rrbracket)\ \lambda x.x\to^{*}_{\ell+\beta}\Psi(\mathbf{v})$*;*
- *For any fragment* S *of* $\lambda^{=}_{lin}$ *such that* $\forall\mathbf{t}\in S$*,* $(\llbracket\mathbf{t}\rrbracket)\ \lambda x.x\in S$*, and for any term* $\mathbf{t}$ *in* S*, if* $\mathbf{t}\to^{=}_{\ell+\beta}{}^{*}\mathbf{v}$ *where* $\mathbf{v}$ *is a value, then* $(\llbracket\mathbf{t}\rrbracket)\ \lambda x.x\to^{=}_{\ell+\beta}{}^{*}\Psi(\mathbf{v})$*.*

Proof. It suffices to check the proofs of Theorems 2.3.6 and 2.3.8 to verify that all the reductions $\to^{*}_{a+\beta}$ are done by rules common in both languages. □

Example 2.3.11. *Note that in Example 2.3.7 one could have as well rewrite with* $\to_{\ell+\beta}$ *which illustrates the indifference property (Corollary 2.3.10).*

Now we proceed to prove Theorems 2.3.6 and 2.3.8 by extending the proof in [Plotkin, 1975] to the algebraic case.

An administrative operation. We define a convenient infix operation (:) capturing the behaviour of translated terms. For example, if $\mathbf{b}$ is a base term, *i.e.* a variable or an abstraction, then its translation into $\lambda^{\to}_{alg}$ is $\llbracket\mathbf{b}\rrbracket = \lambda f.(f)\ \Psi(\mathbf{b})$. If we apply this translated term to a certain $\mathbf{b}'$, we obtain $(\lambda f.(f)\ \Psi(\mathbf{b}))\ \mathbf{b}'\to_{a+\beta}(\mathbf{b}')\ \Psi(\mathbf{b})$. We define $\mathbf{b}:\mathbf{b}' = (\mathbf{b}')\ \Psi(\mathbf{b})$ and get that $(\llbracket\mathbf{b}\rrbracket)\ \mathbf{b}'\to_{a+\beta}\mathbf{b}:\mathbf{b}'$. This fact will be generalised to $(\llbracket\mathbf{t}\rrbracket)\ \mathbf{b}\to_{a+\beta}\mathbf{t}:\mathbf{b}$ in Lemma 2.3.13.

Definition 2.3.12. *Let* $(:):\Lambda_{\lambda_{lin}}\times\Lambda_{\lambda_{alg}}\to\Lambda_{\lambda_{alg}}$ *be the infix binary operation defined by:*

$$\begin{array}{rcl} \mathbf{0}:\mathbf{b} & = & \mathbf{0} \\ \mathbf{b}':\mathbf{b} & = & (\mathbf{b})\ \Psi(\mathbf{b}') \\ \alpha.\mathbf{t}:\mathbf{b} & = & \alpha.(\mathbf{t}:\mathbf{b}) \\ \mathbf{t}+\mathbf{r}:\mathbf{b} & = & \mathbf{t}:\mathbf{b}+\mathbf{r}:\mathbf{b} \end{array} \qquad \begin{array}{rcl} (\mathbf{0})\ \mathbf{t}:\mathbf{b} & = & \mathbf{0} \\ (\mathbf{b}')\ \mathbf{t}:\mathbf{b} & = & \mathbf{t}:\lambda f.((\Psi(\mathbf{b}'))\ f)\ \mathbf{b} \\ (\alpha.\mathbf{t})\ \mathbf{r}:\mathbf{b} & = & \alpha.(\mathbf{t})\ \mathbf{r}:\mathbf{b} \\ (\mathbf{t}+\mathbf{r})\ \mathbf{s}:\mathbf{b} & = & ((\mathbf{t})\ \mathbf{s}+(\mathbf{t})\ \mathbf{s}):\mathbf{b} \\ ((\mathbf{t})\ \mathbf{r})\ \mathbf{s}:\mathbf{b} & = & (\mathbf{t})\ \mathbf{r}:\lambda g.(\llbracket\mathbf{s}\rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} \end{array}$$

Lemma 2.3.13. *If* $\mathbf{b}$ *is a base term, then for any* $\mathbf{t}$*,* $(\llbracket\mathbf{t}\rrbracket)\ \mathbf{b}\to^{*}_{a+\beta}\mathbf{t}:\mathbf{b}$*.*

Proof. Structural induction on $\mathbf{t}$. We give the case $\mathbf{t}=(\mathbf{t}')\ \mathbf{r}$, as an example (*cf.* Appendix A.3 for the full proof). First an intermediate result is needed: for any $\mathbf{t}$, $\mathbf{t}:\lambda g.(\llbracket\mathbf{r}\rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b}\to_{a+\beta}(\mathbf{t})\ \mathbf{r}:\mathbf{b}$. This can be proved by structural induction on $\mathbf{t}$.

Then $(\llbracket(\mathbf{t}')\ \mathbf{r}\rrbracket)\ \mathbf{b} = (\lambda f.(\llbracket\mathbf{t}'\rrbracket)\ \lambda g.(\llbracket\mathbf{r}\rrbracket)\ \lambda h.((g)\ h)\ f)\ \mathbf{b}\to_{a+\beta}(\llbracket\mathbf{t}'\rrbracket)\ \lambda g.(\llbracket\mathbf{r}\rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b}$. Note that $\lambda g.(\llbracket\mathbf{r}\rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b}$ is a base term, so by the induction hypothesis the above term reduces to $\mathbf{t}':\lambda g.(\llbracket\mathbf{r}\rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b}$ which by the previous intermediate result, $\to_{a+\beta}$-reduces to $(\mathbf{t}')\ \mathbf{r}:\mathbf{b}$. □

The following lemmas and its corollary state that the (:) operation preserves reduction.

Lemma 2.3.14. *If* $\mathbf{t}\to_\ell\mathbf{r}$ *then* $\forall\mathbf{b}$ *base term,* $\mathbf{t}:\mathbf{b}\to_a^*\mathbf{r}:\mathbf{b}$.

Proof. Induction on the possible rule applied from $\mathbf{t}\to_\ell\mathbf{r}$. We give one simple case as an example (*cf.* Appendix A.4 for the full proof). Let $\alpha.(\mathbf{t}+\mathbf{r})\to_\ell\alpha.\mathbf{t}+\alpha.\mathbf{r}$. Then $\alpha.(\mathbf{t}+\mathbf{r}):\mathbf{b}=\alpha.(\mathbf{t}:\mathbf{b}+\mathbf{r}:\mathbf{b})\to_a\alpha.(\mathbf{t}:\mathbf{b})+\alpha.(\mathbf{r}:\mathbf{b})=\alpha.\mathbf{t}+\alpha.\mathbf{r}:\mathbf{b}$. □

Lemma 2.3.15. *If* $\mathbf{t}\to_{\ell+\beta}\mathbf{r}$ *then* $\forall\mathbf{b}$ *base term,* $\mathbf{t}:\mathbf{b}\to_{a+\beta}^*\mathbf{r}:\mathbf{b}$.

Proof. If $\mathbf{t}\to_\ell\mathbf{r}$, then use Lemma 2.3.14. If $\mathbf{t}\to_{\beta_b}\mathbf{r}$, then we can prove it by induction on the possible rule applied (either β_b, $\xi_{\lambda_{lin}}$ or one of ξ). We give the case of the β_b-reduction as an example (*cf.* Appendix A.5 for the full proof): $(\lambda x.\mathbf{t})\ \mathbf{b}':\mathbf{b}=\mathbf{b}':\lambda f.((\Psi(\lambda x.\mathbf{t}))\ f)\ \mathbf{b}=(\lambda f.((\Psi(\lambda x.\mathbf{t}))\ f)\ \mathbf{b})\ \Psi(\mathbf{b}')$ and this β_n-reduces to $((\Psi(\lambda x.\mathbf{t}))\ \Psi(\mathbf{b}'))\ \mathbf{b}=((\lambda x.[\![\mathbf{t}]\!])\ \Psi(\mathbf{b}'))\ \mathbf{b}$ which also β_n-reduces to $[\![\mathbf{t}]\!][\Psi(\mathbf{b}')/x]\ \mathbf{b}$ which by Lemma 2.3.5 is equal to $[\![\mathbf{t}[\mathbf{b}'/x]]\!]\ \mathbf{b}$, which by Lemma 2.3.13, $\to_{a+\beta}^*$-reduces to $\mathbf{t}[\mathbf{b}'/x]:\mathbf{b}$. □

Corollary 2.3.16. *If* $\mathbf{t}\to_{\ell+\beta}^=\mathbf{r}$ *then* $\forall\mathbf{b}$ *base terms,* $\mathbf{t}:\mathbf{b}\to_{a+\beta}^{=\ *}\mathbf{r}:\mathbf{b}$.

Proof. By case distinction. If $\mathbf{t}\to_{\ell+\beta}\mathbf{r}$, then by Lemma 2.3.15, $\mathbf{t}:\mathbf{b}\to_{a+\beta}^*\mathbf{r}:\mathbf{b}$, which implies $\mathbf{t}:\mathbf{b}\to_{a+\beta}^{=\ *}\mathbf{r}:\mathbf{b}$. If $\mathbf{r}\to_\ell\mathbf{t}$, then by Lemma 2.3.14, $\mathbf{r}:\mathbf{b}\to_a^*\mathbf{t}:\mathbf{b}$, which also implies $\mathbf{t}:\mathbf{b}\to_{a+\beta}^{=\ *}\mathbf{r}:\mathbf{b}$. □

Finally, the (:) operation also captures the translation of values in the following way:

Lemma 2.3.17. *For any value* $\mathbf{v}$, $\mathbf{v}:\lambda x.x\to_{a+\beta}^*\Psi(\mathbf{v})$

Proof. We proceed by structural induction on $\mathbf{v}$.

- Let $\mathbf{v}$ be a base term. Then $\mathbf{v}:\lambda x.x=(\lambda x.x)\ \Psi(\mathbf{v})\to_{a+\beta}\Psi(\mathbf{v})$.
- Let $\mathbf{v}=\mathbf{v}_1+\mathbf{v}_2$. Then $\mathbf{v}:\lambda x.x=\mathbf{v}_1:\lambda x.x+\mathbf{v}_2:\lambda x.x$, which by the induction hypothesis, reduces to $\Psi(\mathbf{v}_1)+\Psi(\mathbf{v}_2)=\Psi(\mathbf{v})$.
- Let $\mathbf{v}=\alpha.\mathbf{v}'$. Then $\mathbf{v}:\lambda x.x=\alpha.(\mathbf{v}':\lambda x.x)$, which by the induction hypothesis, reduces to $\alpha.\Psi(\mathbf{v}')=\Psi(\mathbf{v})$.
- Let $\mathbf{v}=\mathbf{0}$. Then $\mathbf{v}:\lambda x.x=\mathbf{0}=\Psi(\mathbf{v})$.

□

Example 2.3.18. *We discuss Example 2.3.7 in the light of these results. The term* $(*)$ *is equal to the terms* $(\mathsf{copy})\ (\mathbf{u}+\mathbf{v}):\lambda z.z$ *and* $((\mathsf{copy})\ \mathbf{u}+(\mathsf{copy})\ \mathbf{v}):\lambda z.z$. *The term* $(**)$ *is equal to the term* $(\langle\mathbf{u},\mathbf{u}\rangle+\langle\mathbf{v},\mathbf{v}\rangle):\lambda z.z$ *which reduces to* $\Psi(\langle\mathbf{u},\mathbf{u}\rangle+\langle\mathbf{v},\mathbf{v}\rangle)$. *We do indeed have the rewrites requested by Lemmas 2.3.13, 2.3.15 and 2.3.17.*

Now the proofs of Theorems 2.3.6 and 2.3.8 go as follows.

Proof of Theorem 2.3.6. From Lemma 2.3.13, $([\![\mathbf{t}]\!])\ \lambda x.x\to_{a+\beta}^*\mathbf{t}:\lambda x.x$ and from Lemma 2.3.15, it $\to_{a+\beta}^*$-reduces to $\mathbf{v}:\lambda x.x$. From Lemma 2.3.17, $\mathbf{v}:\lambda x.x\to_{a+\beta}^*\Psi(\mathbf{v})$. □

Proof of Theorem 2.3.8. From Lemma 2.3.13, $([\![\mathbf{t}]\!])\ \lambda x.x\to_{a+\beta}^*\mathbf{t}:\lambda x.x$, which implies that $([\![\mathbf{t}]\!])\ \lambda x.x\to_{a+\beta}^{=\ *}$-reduces to $\mathbf{t}:\lambda x.x$. From Corollary 2.3.16, this latter term $\to_{a+\beta}^{=\ *}$-reduces to $\mathbf{v}:\lambda x.x$. From Lemma 2.3.17, $\mathbf{v}:\lambda x.x\to_{a+\beta}^*\Psi(\mathbf{v})$, which implies that $\mathbf{v}:\lambda x.x\to_{a+\beta}^{=\ *}\Psi(\mathbf{v})$. Note that since $([\![\mathbf{t}]\!])\ \lambda x.x\in S_a$, $\mathbf{t}:\lambda x.x$ is also in S_ℓ due to the closeness under $\to_a^=$ of S_a. The same applies to $\mathbf{t}:\lambda x.x$, thus also to $\mathbf{v}:\lambda x.x$ and finally to $\Psi(\mathbf{v})$. □

2.3.3 Call-by-base simulates call-by-name

The simulation of $\lambda^{\to}_{alg}$ with $\lambda^{\to}_{lin}$. To state that $\lambda^{\to}_{lin}$ simulates $\lambda^{\to}_{alg}$, we use an algebraic extension of the continuation passing style encoding following again [Plotkin, 1975].

Let $\{\!|\cdot|\!\} : \lambda^{\to}_{alg} \to \lambda^{\to}_{lin}$ be the following encoding where f, g and h are fresh variables.

$$\begin{array}{rclcrcl}
\{\!|x|\!\} & = & x, & & \{\!|\mathbf{0}|\!\} & = & \lambda f.(\mathbf{0})\ f,\\
\{\!|\lambda x.\mathbf{t}|\!\} & = & \lambda f.(f)\ \lambda x.\{\!|\mathbf{t}|\!\}, & & \{\!|(\mathbf{t})\ \mathbf{r}|\!\} & = & \lambda f.(\{\!|\mathbf{t}|\!\})\ \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ f,\\
\{\!|\alpha.\mathbf{t}|\!\} & = & \lambda f.(\alpha.\{\!|\mathbf{t}|\!\})\ f, & & \{\!|\mathbf{t}+\mathbf{r}|\!\} & = & \lambda f.(\{\!|\mathbf{t}|\!\}+\{\!|\mathbf{r}|\!\})\ f.
\end{array}$$

This encoding satisfies two useful properties (the first is a trivial result and the second follows by induction on $\mathbf{t}$ (*cf.* Appendix A.6)).

Lemma 2.3.19. *For all terms $\mathbf{t}$, the term $\{\!|\mathbf{t}|\!\}$ is a base term.* □

Lemma 2.3.20. $\{\!|\mathbf{t}[\mathbf{r}/x]|\!\} = \{\!|\mathbf{t}|\!\}[\{\!|\mathbf{r}|\!\}/x]$. □

Let Φ be the encoding for values defined by $\Phi(x) = (x)\ \lambda y.y$, $\Phi(\mathbf{0}) = \mathbf{0}$, $\Phi(\lambda x.\mathbf{t}) = \lambda x.\{\!|\mathbf{t}|\!\}$, $\Phi(\alpha.\mathbf{v}) = \alpha.\Phi(\mathbf{v})$, $\Phi(\mathbf{v}+\mathbf{w}) = \Phi(\mathbf{v}) + \Phi(\mathbf{w})$.

Simulation theorems, similar to Theorems 2.3.6 and 2.3.22, can be stated as follows.

Theorem 2.3.21 (Simulation)**.** *For any program $\mathbf{t}$ (*i.e. *closed term) if $\mathbf{t}\to^{*}_{a+\beta}\mathbf{v}$ where $\mathbf{v}$ is a value, then $(\{\!|\mathbf{t}|\!\})\ \lambda x.x\to^{*}_{\ell+\beta}\Phi(\mathbf{v})$.*

Theorem 2.3.22 (Simulation)**.** *For any two fragments S_a of $\lambda^{=}_{alg}$ and S_ℓ of $\lambda^{=}_{lin}$ such that $\forall \mathbf{t} \in S_a$, $(\{\!|\mathbf{t}|\!\})\ \lambda x.x \in S_\ell$, and for any program $\mathbf{t}$ in S_a, if $\mathbf{t}\to^{=\ *}_{a+\beta}\mathbf{v}$ where $\mathbf{v}$ is a value, then $(\{\!|\mathbf{t}|\!\})\ \lambda x.x\to^{=\ *}_{\ell+\beta}\Phi(\mathbf{v})$.*

A result similar to Corollary 2.3.10 can also be formulated. It is proven in a similar manner.

Corollary 2.3.23 (Indifference)**.**

- *For any program $\mathbf{t}$, if $\mathbf{t}\to^{*}_{a+\beta}\mathbf{v}$ where $\mathbf{v}$ is a value, then $(\{\!|\mathbf{t}|\!\})\ \lambda x.x\to^{*}_{a+\beta}\Phi(\mathbf{v})$;*
- *For any fragment S of $\lambda^{=}_{alg}$ such that $\forall \mathbf{t} \in S$, $(\{\!|\mathbf{t}|\!\})\ \lambda x.x \in S$, and for any program $\mathbf{t}$ in S, if $\mathbf{t}\to^{=\ *}_{a+\beta}\mathbf{v}$ where $\mathbf{v}$ is a value, then $(\{\!|\mathbf{t}|\!\})\ \lambda x.x\to^{=\ *}_{a+\beta}\Phi(\mathbf{v})$.* □

Before moving to the description of the proof of Theorems 2.3.21 and 2.3.22, let us consider an example.

Example 2.3.24. *We illustrate Theorem 2.3.21 using the term $(\mathsf{copy})\ (\mathbf{u}+\mathbf{v})$ of Example 2.3.7 which reduces to $\langle \mathbf{u}, \mathbf{u}\rangle + \langle \mathbf{v}, \mathbf{v}\rangle$ in $\lambda^{\to}_{lin}$ and to $\langle \mathbf{u}+\mathbf{v}, \mathbf{u}+\mathbf{v}\rangle$ in $\lambda^{\to}_{alg}$.*

$$\begin{array}{rl}
\{\!|(\mathsf{copy})\ (\mathbf{u}+\mathbf{v})|\!\} = & \lambda f.(\{\!|\mathsf{copy}|\!\})\ \lambda g.((g)\ \{\!|\mathbf{u}+\mathbf{v}|\!\})\ f\\
\{\!|\mathsf{copy}|\!\} = & \lambda f.(f)\ \lambda x.\{\!|\langle x, x\rangle|\!\}\\
\{\!|\langle \mathbf{t}, \mathbf{r}\rangle|\!\} = & \lambda f.(f)\ \Phi(\langle \mathbf{t}, \mathbf{r}\rangle)\\
\{\!|\mathbf{u}+\mathbf{v}|\!\} = & \lambda f.(\{\!|\mathbf{u}|\!\}+\{\!|\mathbf{v}|\!\})\ f\\
\{\!|\mathbf{u}|\!\} = & \lambda g.(g)\ \Phi(\mathbf{u})
\end{array}$$

We now rewrite $\mathbf{r} = (\{\!|(\mathsf{copy})\ (\mathbf{u}+\mathbf{v})|\!\})\ \lambda z.z$ *in* $\lambda^{\rightarrow}_{alg}$.

$$\begin{array}{rll}
\mathbf{r} & \rightarrow_{\ell+\beta} & (\{\!|\mathsf{copy}|\!\})\ \lambda g.((g)\ \{\!|\mathbf{u}+\mathbf{v}|\!\})\ \lambda z.z \\
& = & (\lambda f.(f)\ \lambda x.\{\!|\langle x,x\rangle|\!\})\ \lambda g.((g)\ \{\!|\mathbf{u}+\mathbf{v}|\!\})\ \lambda z.z \\
& \rightarrow_{\ell+\beta} & (\lambda g.((g)\ \{\!|\mathbf{u}+\mathbf{v}|\!\})\ \lambda z.z)\ \lambda x.\{\!|\langle x,x\rangle|\!\} \\
& \rightarrow_{\ell+\beta} & ((\lambda x.\{\!|\langle x,x\rangle|\!\})\ \{\!|\mathbf{u}+\mathbf{v}|\!\})\ \lambda z.z \qquad (***) \\
\textit{(Lemma 2.3.19)} & \rightarrow_{\ell+\beta} & (\{\!|\langle x,x\rangle|\!\}[\{\!|\mathbf{u}+\mathbf{v}|\!\}/x])\ \lambda z.z \\
\textit{(Lemma 2.3.20)} & = & (\{\!|\langle \mathbf{u}+\mathbf{v},\mathbf{u}+\mathbf{v}\rangle|\!\})\ \lambda z.z \\
& \rightarrow_{\ell+\beta} & (\lambda z.z)\ \Phi(\langle \mathbf{u}+\mathbf{v},\mathbf{u}+\mathbf{v}\rangle) \qquad (****) \\
& \rightarrow_{\ell+\beta} & \Phi(\langle \mathbf{u}+\mathbf{v},\mathbf{u}+\mathbf{v}\rangle)
\end{array}$$

Proof of the simulation theorems. In Section 2.3.2, the proofs of the simulations theorems were performed using an administrative operation ":" and three intermediate results, as follows. The term $\mathbf{b}$ is taken as a base term.

1. Prove that $(\llbracket\mathbf{t}\rrbracket)\ \mathbf{b} \rightarrow^{*}_{a+\beta} \mathbf{t} : \mathbf{b}$;
2. prove that if $\mathbf{t} \rightarrow_{\ell+\beta} \mathbf{r}$ then $\mathbf{t} : \mathbf{b} \rightarrow^{*}_{a+\beta} \mathbf{r} : \mathbf{b}$;
3. prove that if $\mathbf{v}$ is a value, $\mathbf{v} : \lambda x.x \rightarrow^{*}_{a+\beta} \Psi(\mathbf{v})$.

For the simulation theorems of the present section, we use a similar procedure.

An administrative operation. We keep the same notation for the administrative, infix operation defined for the purpose of the proof.

Definition 2.3.25. *Let* $(:) : \Lambda_{\lambda_{alg}} \times \Lambda_{\lambda_{lin}} \rightarrow \Lambda_{\lambda_{lin}}$ *be the infix binary operation defined by:*

$$\begin{array}{rcl}
\mathbf{0} : \mathbf{b} & = & \mathbf{0} \\
\mathbf{b}' : \mathbf{b} & = & (\mathbf{b})\ \Phi(\mathbf{b}') \\
\alpha.\mathbf{t} : \mathbf{b} & = & \alpha.(\mathbf{t} : \mathbf{b}) \\
\mathbf{t}+\mathbf{r} : \mathbf{b} & = & \mathbf{t} : \mathbf{b} + \mathbf{r} : \mathbf{b}
\end{array}
\qquad
\begin{array}{rcl}
(\mathbf{0})\ \mathbf{r} : \mathbf{b} & = & \mathbf{0} \\
(\mathbf{b}')\ \mathbf{r} : \mathbf{b} & = & ((\Phi(\mathbf{b}'))\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} \\
(\alpha.\mathbf{t})\ \mathbf{r} : \mathbf{b} & = & \alpha.(\mathbf{t})\ \mathbf{r} : \mathbf{b} \\
(\mathbf{t}+\mathbf{r})\ \mathbf{s} : \mathbf{b} & = & ((\mathbf{t})\ \mathbf{s} + (\mathbf{r})\ \mathbf{s}) : \mathbf{b} \\
((\mathbf{t})\ \mathbf{r})\ \mathbf{s} : \mathbf{b} & = & (\mathbf{t})\ \mathbf{r} : \lambda f.((f)\ \{\!|\mathbf{s}|\!\})\ \mathbf{b}
\end{array}$$

The three lemmas needed for the proof of the simulation theorems now read as follow.

Lemma 2.3.26. *If* $\mathbf{b}$ *is a base term, then for any closed term* $\mathbf{t}$, $(\{\!|\mathbf{t}|\!\})\ \mathbf{b} \rightarrow^{*}_{\ell+\beta} \mathbf{t} : \mathbf{b}$.

Proof. The proof is done by structural induction on $\mathbf{t}$. We follow the sketch of the proof of Lemma 2.3.13, and give the case $\mathbf{t} = (\mathbf{t}')\ \mathbf{r}$, as an example (*cf.* Appendix A.7 for the full proof). First we prove by induction on $\mathbf{t}$ that $\mathbf{t} : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} \rightarrow^{*}_{\ell+\beta} (\mathbf{t})\ \mathbf{r} : \mathbf{b}$. Then $(\{\!|(\mathbf{t}')\ \mathbf{r}|\!\})\ \mathbf{b} = (\lambda f.(\{\!|\mathbf{t}'|\!\})\ \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ f)\ \mathbf{b}$, $\rightarrow_{\ell+\beta}$-reduces to $(\{\!|\mathbf{t}'|\!\})\ \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b}$. Note that $\lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b}$ is a base term, so by the induction hypothesis the above term reduces to $\mathbf{t}' : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b}$ which by the previous intermediate result, $\rightarrow_{\ell+\beta}$-reduces to $(\mathbf{t}')\ \mathbf{r} : \mathbf{b}$. □

Lemma 2.3.27. *If* $\mathbf{t} \rightarrow_{a+\beta} \mathbf{r}$ *then* $\forall \mathbf{b}$ *base term,* $\mathbf{t} : \mathbf{b} \rightarrow^{*}_{\ell+\beta} \mathbf{r} : \mathbf{b}$.

Proof. Case by case on the rules of $\lambda^{\rightarrow}_{alg}$. We give the case of the β_n-reduction as an example (*cf.* Appendix A.8 for the full proof): $(\lambda x.\mathbf{t})\ \mathbf{r} : \mathbf{b} = ((\Phi(\lambda x.\mathbf{t}))\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} = ((\lambda x.\{\!|\mathbf{t}|\!\})\ \{\!|\mathbf{r}|\!\})\ \mathbf{b}$ which by

Lemma 2.3.19, $\to_{\ell+\beta}$-reduces to $(\{\!|\mathbf{t}|\!\}[\{\!|\mathbf{r}|\!\}/x])\ \mathbf{b}$. This, by Lemma 2.3.20, is equal to $(\{\!|\mathbf{t}[\mathbf{r}/x]|\!\})\ \mathbf{b}$ and this, by Lemma 2.3.26, $\to^*_{\ell+\beta}$-reduces to $\mathbf{t}[\mathbf{r}/x] : \mathbf{b}$.

Note that in the previous derivation, the reduction $(\lambda x.\{\!|\mathbf{t}|\!\})\ \{\!|\mathbf{r}|\!\} \to_{\ell+\beta} \{\!|\mathbf{t}|\!\}[\{\!|\mathbf{r}|\!\}/x]$ is valid since for any term $\mathbf{r}$, $\{\!|\mathbf{r}|\!\}$ is a base term. □

Lemma 2.3.28. *If* $\mathbf{v}$ *is a value and* $\mathbf{b}$ *is a base term,* $\mathbf{v} : \lambda x.x \to^*_{\ell+\beta} \Phi(\mathbf{v})$. □

Example 2.3.29. *We discuss Example 2.3.24 in the light of these results. The term* $(***)$ *is equal to the terms* $(\mathsf{copy})\ (\mathbf{u}+\mathbf{v}) : \lambda z.z$*. The term* $(****)$ *is equal to the term* $\langle \mathbf{u}+\mathbf{v}, \mathbf{u}+\mathbf{v}\rangle : \lambda z.z$ *which reduces to* $\Phi(\langle \mathbf{u}+\mathbf{v}, \mathbf{u}+\mathbf{v}\rangle)$*. Again, we have the rewrites requested by Lemmas 2.3.26, 2.3.27 and 2.3.28.*

We are now ready to prove the simulation theorems. As advertised, these proofs reflect the exact same structures of the proofs of Theorems 2.3.6 and 2.3.8.

Proof of Theorem 2.3.21. From Lemma 2.3.26, $(\{\!|\mathbf{t}|\!\})\ \lambda x.x \to^*_{\ell+\beta} \mathbf{t} : \lambda x.x$ and from Lemma 2.3.27 it $\to^*_{\ell+\beta}$-reduces to $\mathbf{v} : \lambda x.x$. From Lemma 2.3.28, $\mathbf{v} : \lambda x.x \to^*_{\ell+\beta} \Phi(\mathbf{v})$. □

Proof of Theorem 2.3.22. From Lemma 2.3.26, $(\{\!|\mathbf{t}|\!\})\ \lambda x.x \to^*_{\ell+\beta} \mathbf{t} : \lambda x.x$, which implies that $(\{\!|\mathbf{t}|\!\})\ \lambda x.x \to^{=}_{\ell+\beta}{}^*$-reduces to $\mathbf{t} : \lambda x.x$. A result equivalent to Corollary 2.3.16 can be shown as easily: if $\mathbf{t} \to^{=}_{a} \mathbf{r}$ then for all base terms $\mathbf{b}$,$\mathbf{t} : \mathbf{b} \to^{=}_{\ell}{}^* \mathbf{r} : \mathbf{b}$. This entails that $\mathbf{t} : \lambda x.x \to^{=}_{\ell+\beta}{}^*$-reduces to $\mathbf{v} : \lambda x.x$. From Lemma 2.3.28, $V : \lambda x.x \to^*_{\ell+\beta} \Phi(\mathbf{v})$, which implies that $\mathbf{v} : \lambda x.x \to^{=}_{\ell+\beta}{}^* \Phi(\mathbf{v})$. Note that since $(\{\!|\mathbf{t}|\!\})\ \lambda x.x \in S_\ell$, $\mathbf{t} : \lambda x.x$ is also in S_ℓ due to the closeness under $\to^{=}_{\ell}$ of S_ℓ. The same applies to $\mathbf{t} : \lambda x.x$, thus also to $\mathbf{v} : \lambda x.x$ and finally to $\Phi(\mathbf{v})$. □

2.3.4 The remaining simulations

In Figure 2.2, some arrows are missing. We are now showing that the already existing arrows "compose" well. The first two simulations are $\lambda^{\to}_{alg} \to \lambda^{=}_{lin}$ and $\lambda^{\to}_{lin} \to \lambda^{=}_{alg}$ and do not require confluence.

Theorem 2.3.30. *For any program* $\mathbf{t}$*, if* $\mathbf{t} \to^*_{\ell+\beta} \mathbf{v}$ *(respectively* $\mathbf{t} \to^*_{a+\beta} \mathbf{v}$*) where* $\mathbf{v}$ *is a value, then* $(\llbracket \mathbf{t} \rrbracket)\ \lambda x.x \to^{=}_{a+\beta}{}^* \Psi(\mathbf{v})$ *(resp.* $(\{\!|\mathbf{t}|\!\})\ \lambda x.x \to^{=}_{\ell+\beta}{}^* \Phi(\mathbf{v})$*).*

Proof. Given that $\mathbf{t} \to^*_{\ell+\beta} \mathbf{v}$, by Theorem 2.3.6, $(\llbracket \mathbf{t} \rrbracket)\ \lambda x.x \to^*_{a+\beta} \Psi(\mathbf{v})$, which by Theorem 2.3.1 implies $(\llbracket \mathbf{t} \rrbracket)\ \lambda x.x \to^{=}_{a+\beta}{}^* \Psi(\mathbf{v})$.

Analogously, given that $\mathbf{t} \to^*_{a+\beta} \mathbf{v}$, by Theorem 2.3.21, $(\{\!|\mathbf{t}|\!\})\ \lambda x.x \to^*_{\ell+\beta} \Phi(\mathbf{v})$, which by Theorem 2.3.2 implies $(\{\!|\mathbf{t}|\!\})\ \lambda x.x \to^{=}_{\ell+\beta}{}^* \Phi(\mathbf{v})$. □

The other two simulations are $\lambda^{=}_{alg} \to \lambda^{\to}_{lin}$ and $\lambda^{=}_{lin} \to \lambda^{\to}_{alg}$ and they do require confluence.

Theorem 2.3.31. *For any program* $\mathbf{t}$ *in a confluent fragment of* $\lambda^{=}_{lin}$ *(resp.* $\lambda^{=}_{alg}$*), if* $\mathbf{t} \to^{=}_{\ell+\beta}{}^* \mathbf{v}$ *(respectively* $\mathbf{t} \to^{=}_{a+\beta}{}^* \mathbf{v}$*) then* $(\llbracket \mathbf{t} \rrbracket)\ \lambda x.x \to^*_{a+\beta} \Psi(\mathbf{v}')$ *with* $\mathbf{v} \to^{=}_{a}{}^* \mathbf{v}'$ *(respectively* $(\{\!|\mathbf{t}|\!\})\ \lambda x.x \to^*_{\ell+\beta} \Phi(\mathbf{v}')$ *with* $\mathbf{v} \to^{=}_{\ell}{}^* \mathbf{v}'$*).*

Proof. Given that $\mathbf{t} \to^{=}_{\ell+\beta}{}^* \mathbf{v}$, by Theorem 2.3.6, $(\llbracket \mathbf{t} \rrbracket)\ \lambda x.x \to^{=}_{a+\beta}{}^* \Psi(\mathbf{v})$, which by Theorem 2.3.4, which requires confluence, implies $(\llbracket \mathbf{t} \rrbracket)\ \lambda x.x \to^*_{a+\beta} \Psi(\mathbf{v})$. The other result is similar using Theorem 2.3.21. Notice that that the theorems with arrows are trivially valid when we take a fragment. □

2.4 Conclusion and open questions

In this chapter we described four canonical algebraic lambda-calculi with vectorial structures, recapitulating the few existing means of writing such a language. We has shown how each language can simulate the other, by taking care of marking where confluence is used or not.

As already shown by Plotkin [Plotkin, 1975], if the simulation of call-by-value by call-by-name is sound, it fails to be complete for general (possibly non-terminating) programs. A known solution to this problem is developed by Sabry and Wadler [1997]. A recent work [Assaf and Perdrix, 2011] shown that the technique can be adapted to the algebraic case to retrieve completeness. The work by Sabry and Wadler [1997] develop a Galois connection between call-by-name and call-by-value. A direction for study is to build on this work to also get a Galois connection in the algebraic case.

Concerning semantics, the algebraic λ-calculus admits finiteness spaces as a model [Ehrhard, 2005, 2010]. What is the structure of the model of the linear algebraic λ-calculus induced by the continuation-passing style translation in finiteness spaces? The algebraic lambda-calculus can be equipped with a differential operator. What is the corresponding operator in call-by-base through the translation?

Chapter 3

A type system accounting for scalars

Résumé du Chapitre

Dans ce chapitre, nous introduisons Scalar; un système de types similaire au Système F pour le lambda-calcul linéaire algébrique. Nous démontrons que le système de types Scalar vérifie à la fois la propriété de préservation du type par réduction et la propriété de normalisation forte, ce qui constitue nos principaux résultats techniques de ce chapitre. La normalisation forte offre un simplification significative de λ_{lin} lui-même, en enlevant la nécessité de certaines restrictions dans ses règles de réduction. Mais le point le plus important et le plus original de ce système de type, est le fait qu'il garde la trace de « la quantité d'un type » présente dans un terme. A titre d'exemple de son utilisation, nous montrons qu'il peut servir comme une garantie que la forme normale d'un terme est barycentrique, c'est à dire que ses scalaires somment à un. ■

In this chapter we provide a fine-grained, System F-like type system for the linear-algebraic lambda-calculus, λ_{lin} (*cf.* Section 1.2). We show that this "scalar" type system enjoys both the subject-reduction property and the strong-normalisation property, our main technical results. The latter yields a significant simplification of the linear-algebraic lambda-calculus itself, by removing the need for most of the restrictions in its reduction rules. But the more important, original feature of this scalar type system is that it keeps track of 'the amount of a type' that is present in each term. As an example of its use, we shown that it can serve as a guarantee that the normal form of a term is barycentric, *i.e.* that its scalars are summing to one.

Plan of the chapter. Section 3.1 presents the *Scalar* type system with its grammar, equivalences and inference rules. Section 3.2 shows the subject reduction property giving consistency to the system. Section 3.3 shows the strong normalisation property for this system, allowing us to lift the above discussed restrictions in the reduction rules. In section 3.4 we formalise the type system $\mathcal{B}$ for barycentric calculi. Section 3.5 concludes.

3.1 The *Scalar* Type System

The grammar of types, *cf.* Figure 3.1, defines the set of types (notation: $\mathcal{T}$) and its syntactic subclass (notation: $\mathcal{U}$) of what we call *unit types*. Notice that the grammar for $\mathcal{U}$ does not allow for scalars except

to the right of an arrow. More generally, notice the novelty of having scalars *weighting* the amount of a type.

Type variables are denoted by X, Y, etc. and can only ever be substituted by a unit type. Contexts are denoted by Γ, Δ, etc. and are defined as sets $\{x\!:\!U, \ldots\}$, where x is a term variable appearing only once in the set, and $U \in \mathcal{U}$. We usually omit the brackets of the set. The substitution of X by V in U is defined as usual, and is written $U[V/X]$. We sometimes use the vectorial notation $U[\vec{V}/\vec{X}]$ for $U[V_1/X_1]\cdots[V_n/X_n]$ if $\vec{X} = X_1, \ldots, X_n$ and $\vec{V} = V_1, \ldots, V_n$. We also may abuse notation and say $\vec{X} \notin S$ meaning that none of the X_i from $\vec{X}$ are in the set S. We write $\Gamma[U/X]$ to the substitution of X by U in each type of Γ. Also we write $(\alpha.T)[U/X]$ for $\alpha.T[U/X]$. $FV(T)$ designates the set of free variables of the type T, defined in the usual manner, and $FV(\Gamma)$ is the union of the sets of free variables of each type in Γ. Scalars are denoted by $\alpha, \beta, \gamma \ldots$ and are members of the same commutative ring $(\mathcal{S}, +, \times)$ as those of terms.

We also define an equivalence relation upon types as follows:

Definition 3.1.1. *For any $\alpha, \beta \in \mathcal{S}$ and $T \in \mathcal{T}$. We define the* type equivalence $\equiv$ *to be the least congruence such that*

- $\alpha.\overline{0} \equiv \overline{0}$
- $0.T \equiv \overline{0}$
- $1.T \equiv T$
- $\alpha.(\beta.T) \equiv (\alpha \times \beta).T$
- $\forall X.\alpha.T \equiv \alpha.\forall X.T$

$$\begin{array}{lll} \textit{Types:} & T, R, S ::= & U \mid \forall X.T \mid \alpha.T \mid \overline{0} \\ \textit{Unit types:} & U, V, W ::= & X \mid U \to T \mid \forall X.U \end{array}$$

$$\frac{}{\Gamma, x\!:\!U \vdash x\!:\!U}\, ax \qquad \frac{\Gamma \vdash \mathbf{t}\!:\!T \qquad T \equiv S}{\Gamma \vdash \mathbf{t}\!:\!S} \equiv \qquad \frac{\Gamma \vdash \mathbf{t}\!:\!\alpha.(U \to T) \qquad \Gamma \vdash \mathbf{r}\!:\!\beta.U}{\Gamma \vdash (\mathbf{t})\ \mathbf{r}\!:\!(\alpha \times \beta).T} \to_E$$

$$\frac{\Gamma, x\!:\!U \vdash \mathbf{t}\!:\!T}{\Gamma \vdash \lambda x.\mathbf{t}\!:\!U \to T} \to_I \qquad \frac{\Gamma \vdash \mathbf{t}\!:\!\forall X.T}{\Gamma \vdash \mathbf{t}\!:\!T[U/X]} \forall_E \qquad \frac{\Gamma \vdash \mathbf{t}\!:\!T \qquad X \notin FV(\Gamma)}{\Gamma \vdash \mathbf{t}\!:\!\forall X.T} \forall_I$$

$$\frac{}{\Gamma \vdash \mathbf{0}\!:\!\overline{0}}\, ax_{\overline{0}} \qquad \frac{\Gamma \vdash \mathbf{t}\!:\!\alpha.T \qquad \Gamma \vdash \mathbf{r}\!:\!\beta.T}{\Gamma \vdash \mathbf{t} + \mathbf{r}\!:\!(\alpha + \beta).T} +_I \qquad \frac{\Gamma \vdash \mathbf{t}\!:\!T}{\Gamma \vdash \alpha.\mathbf{t}\!:\!\alpha.T}\, s_I$$

Figure 3.1: Types and typing rules of *Scalar*

The complete set of typing rules of *Scalar* is shown in Figure 3.1. Let us justify this type system. Splitting the grammar into general types and unit types is a necessary consequence of the fact that we want scalars in the types to reflect scalars in the terms (*e.g.* $\alpha.\lambda x.\mathbf{t}$ should have a type $\alpha.U$). Indeed if we did not have the restriction on the left side of an arrow being a unit type, *i.e.* $U \to T$, then we would have types like $(\alpha.X) \to X$, which *a priori* do not make sense, because abstractions receive only base terms as arguments. This could be fixed by adding the equivalence $(\alpha.A) \to B \equiv \alpha.(A \to B)$, making sure that α is non-zero. But still we would need to keep the $\to_E$ rule restricted to having a unit type on the left hand side of the arrow, otherwise we would break the required correspondence between scalars-in-types and scalars-in-terms, *e.g.* :

$$\frac{\vdash \alpha.\lambda x.x\!:\!(\alpha.T) \to T \qquad \vdash \mathbf{t}\!:\!\alpha.T}{\vdash (\alpha.\lambda x.x)\ \mathbf{t}\!:\!T} \qquad \text{but } (\alpha.\lambda x.x)\ \mathbf{t} \to^* \alpha.\mathbf{t} \text{ which should be of type } \alpha^2.T$$

Again, we want the scalars in the types to represent those in the terms, hence the rule s_I. Rule $+_I$ takes care of sums of terms, and term $\mathbf{0}$ gets the special type $\overline{0}$ by an axiom.

Finally, let us go back to the application. The standard rule $\to_E$ from System F needs to be consistent with the extra rules for application that we have on top of β-reduction in λ_{lin}; namely the *Application rules*:

1. $(\mathbf{t}+\mathbf{r})\ \mathbf{u} \to (\mathbf{t})\ \mathbf{u} + (\mathbf{r})\ \mathbf{u}$
2. $(\mathbf{u})\ (\mathbf{t}+\mathbf{r}) \to (\mathbf{u})\ \mathbf{t} + (\mathbf{u})\ \mathbf{r}$
3. $(\alpha.\mathbf{t})\ \mathbf{r} \to \alpha.(\mathbf{t})\ \mathbf{r}$
4. $(\mathbf{r})\ (\alpha.\mathbf{t}) \to \alpha.(\mathbf{r})\ \mathbf{t}$
5. $(\mathbf{0})\ \mathbf{t} \to \mathbf{0}$
6. $(\mathbf{t})\ \mathbf{0} \to \mathbf{0}$

Notice that the terms $\mathbf{t}$ and $\mathbf{r}$ in rules (1) and (2) must now have the same type (up to a scalar) according to rule $+_I$, so the type of $\mathbf{t}+\mathbf{r}$ is analogous to the type of $\alpha.\mathbf{t}$ in rules (3) and (4). Also, the type for $\mathbf{0}$ in rules (5) and (6) is the same as that of $0.\mathbf{t}$ if we take $\alpha = 0$ in rules (3) and (4). Thus we can focus our discussion on rules (3) and (4).

By rule (3), we must have:

$$\frac{\Gamma \vdash \mathbf{t} : \alpha.(U \to T) \qquad \Gamma \vdash \mathbf{r} : U}{\Gamma \vdash (\mathbf{t})\ \mathbf{r} : \alpha.T}$$

By rule (4), we must have:

$$\frac{\Gamma \vdash \mathbf{t} : U \to T \qquad \Gamma \vdash \mathbf{r} : \beta.U}{\Gamma \vdash (\mathbf{t})\ \mathbf{r} : \beta.T}$$

By combining these two we obtain the $\to_E$ rule presented in Figure 3.1.

Remark 3.1.2. *A good insight into the type system is that, due to equivalences, scalars occur only at top-level and the level of arrows. This fits very well with the idea that in λ_{lin} all constructs are linear, except for abstraction. With this in mind, the syntax of arrow types could have been restricted to $U \to \alpha.V$, or even written as $U \to_\alpha V$ instead. In other words, we could get rid of the type equivalences (*cf. *Definition 3.1.1) and represent each type equivalence class by just its canonical member. Such a design choice would spare us some lemmas (*cf. *Section 3.2), but comes at a price:*

- *Equivalences between $0.T$ and $0.R$ or $1.T$ and T would still need to be enforced through an equivalence relation or some inelegant case distinctions, at least if we want to maintain them;*
- *More generally our aim is to reflect some of the vectorial structure of the terms of λ_{lin} up at the level of types. In that sense the explicit type equivalences we have given provide a good indication that types have the desired structure.*

3.2 Subject reduction

The following theorem ensures that typing is preserved by reduction, making our type system consistent. Having such a property is part of the basic requirements for a type system.

Theorem 3.2.1 (Subject Reduction)**.** *For any terms $\mathbf{t}$, $\mathbf{t}'$, any context Γ and any type T, if $\mathbf{t} \to \mathbf{t}'$, then $\Gamma \vdash \mathbf{t} : T \Rightarrow \Gamma \vdash \mathbf{t}' : T$.*

The proof of this theorem is quite long and non-trivial. This is one of the main technical contributions of the chapter.

3.2.1 Preliminary lemmas

In order to prove this theorem, we need several auxiliary lemmas standing for general properties of our system. We have tried to provide an intuition of every lemma so as to make it easier to follow. Also, we divided them in four groups, reflecting the nature of their statement.

Lemmas about types

The lemmas in this section are statements about the properties of the types themselves, *i.e.* their equivalences.

It is not so hard to see that every type is equivalent to a scalar multiplied by a unit type (*i.e.* a type in $\mathcal{U}$). A type in $\mathcal{U}$ can of course always be multiplied by 1 (proof in Appendix B.1).

Lemma 3.2.2 (α unit)**.** *$\forall T \in \mathcal{T},\ \exists U \in \mathcal{U}, \alpha \in \mathcal{S}$ such that $T \equiv \alpha.U$.* □

This first lemma should not be misinterpreted however: this does not mean to say that any scalar appearing within a type can be factored out of the type. For example, even a simple unit type $X \to \alpha.X$ is not equivalent to $\alpha.(X \to X)$.

The following just says that when two types are equivalent, then the outer left scalars are the same:

Lemma 3.2.3 (Unit does not add scalars)**.** *$\forall U, U' \in \mathcal{U},\ \forall \alpha, \beta \in \mathcal{S}$, if $\alpha.U \equiv \beta.U'$ then $\alpha = \beta$ and, if $\alpha \neq 0$, then $U \equiv U'$.*

Proof. Following $\mathcal{U}$ grammar, neither U nor U' could contain scalars in this head form but only in the right side of a type $U \to T$. However, no equivalence rule lets it come out from the right of the arrow and get to the head-form, so if $\alpha.U \equiv \beta.U'$ that means $\alpha = \beta = 0$ or $U \equiv U'$ and $\alpha = \beta$. □

Several of the following lemmas will be proved by induction on the size of the derivation tree, so, we need to formally define what we mean by this size. In our definition we count the depth of the tree, but ignoring any application of an equivalence rule:

We define the *size of a derivation tree* inductively as follows

$$\texttt{size}\left(\dfrac{}{\dfrac{S'}{S}\equiv}\,R\right) = 0 \qquad \texttt{size}\left(\dfrac{\pi_1 \quad \pi_2}{\dfrac{S'}{S}\equiv}\,R'\right) = \max\{\texttt{size}(\pi_1), \texttt{size}(\pi_2)\} + 1$$

where π_1, π_2 are derivation trees, S is a sequent, R and R' are type inference rules. We denote by S_n a sequent that can be derived with a proof of size n.

Without actually making a subtyping theory, we define a pre-order ($\preceq$) between types following [Barendregt, 1992]:

Definition 3.2.4. *For any types T, R, S and any type variable X,*

1. *write $T \prec R$ if either $R \equiv \forall X.T$ or $T \equiv \forall X.S$ and $R \equiv S[U/X]$ for some $U \in \mathcal{U}$.*
2. *$\preceq$ is the reflexive (in terms of $\equiv$) and transitive closure of $\prec$.*

Notice that scalars do not interfere with the order, as stated by the following lemma.

Lemma 3.2.5 (Scalars keep order)**.** *For any types T, R and for any scalar α, if $T \preceq R$ then $\alpha.T \preceq \alpha.R$.*

Proof. Let $T \preceq R$, then assume $T \equiv R_1 \prec \cdots \prec R_n \equiv R$. Then $\forall i$ one has $R_i \prec R_{i+1}$. So, the possible cases are:

- $R_{i+1} \equiv \forall X.R_i$, then $\alpha.R_i \prec \forall X.\alpha.R_i \equiv \alpha.\forall X.R_i \equiv \alpha.R_{i+1}$.
- $R_i \equiv \forall X.S$ and $R_{i+1} \equiv S[U/X]$, then $\alpha.R_i \equiv \alpha.\forall X.S \equiv \forall X.\alpha.S \prec (\alpha.S)[U/X] \equiv \alpha.(S[U/X]) \equiv \alpha.R_{i+1}$.

□

The following lemma states that if two arrow types are ordered, then they are equivalent up to some substitution. (proof in Appendix B.2).

Lemma 3.2.6 (Arrows comparison)**.** *For any types $U, V \in \mathcal{U}$ and $T, R \in \mathcal{T}$, if $V \to R \preceq U \to T$, then $\exists \vec{W}, \vec{X}$ / $U \to T \equiv (V \to R)[\vec{W}/\vec{X}]$.* □

Classic lemmas

The lemmas in this section are the classic ones, which appear in most subject reduction proofs.

As a pruned version of a subtyping system, we can prove the subtyping rule:

Lemma 3.2.7 ($\preceq$-subsumption)**.** *For any term $\mathbf{t}$, for any types T, R and any context Γ such that $FV(T) \cap FV(\Gamma) = \emptyset$, if $T \preceq R$, then*

$$\frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \mathbf{t} : R}$$

where renaming of type variables may occur.

Proof. It suffices to show the property for $T \prec R$. Cases

- $R \equiv \forall X.T$. Cases.
  - $X \notin FV(\Gamma)$, by rule $\forall_I$, $\Gamma \vdash \mathbf{t} : \forall X.T$.
  - $X \in FV(\Gamma)$, then $X \notin FV(T)$, so take a fresh variable Y, which is not in $FV(\Gamma)$, and then by rule $\forall_I$, $\Gamma \vdash \mathbf{t} : \forall Y.T$. Notice that $\forall X.T = \forall Y.T$ since neither X nor Y appears in T.
- $T \equiv \forall X.S$ and $R \equiv S[U/X]$, then by rule $\forall_E$, $\Gamma \vdash \mathbf{t} : S[U/X]$.

□

Proving subject reduction means proving that each reduction rule preserves the type. The way to do this is to go in the opposite direction to the reduction rule, *i.e.* to study the reduct so as to understand where it may come from, thereby decomposing the redex in its basic constituents. Generation lemmas accomplish that purpose.

We will need five generation lemmas: the two classical ones, one for applications (Lemma 3.2.8) and one for abstractions (Lemma 3.2.9); and three new ones for the algebraic rules, one for products by scalars different to 0 (Lemma 3.2.10) other for product by 0 (Lemma 3.2.11) and one for sums (Lemma 3.2.12). Their proofs follows by induction on the typing derivation and can be found in Appendices B.3 to B.7.

Lemma 3.2.8 (Generation lemma (app))**.** *For any terms $\mathbf{t}, \mathbf{r}$, any type T, any scalar γ, any context Γ and any number $n \in \mathbb{N}$, if $S_n = \Gamma \vdash (\mathbf{t})\ \mathbf{r} : \gamma.T$, then $\exists \alpha, \beta \in \mathcal{S}$, $r, s \in \mathbb{N}$ with $\max(r, s) < n$, $U \in \mathcal{U}$ and $R \preceq T$ such that $S_r = \Gamma \vdash \mathbf{r} : \alpha.U$ and $S_s = \Gamma \vdash \mathbf{t} : \beta.U \to R$ with $\alpha \times \beta = \gamma$.* □

Lemma 3.2.9 (Generation lemma (abs)). *For any term* $\mathbf{t}$*, any type* T*, any context* Γ *and any number* $n \in \mathbb{N}$*, if* $S_n = \Gamma \vdash \lambda x.\mathbf{t} : T$ *then* $\exists U \in \mathcal{U}$*,* $R \in \mathcal{T}$ *and* $m < n$ *such that* $S_m = \Gamma, x : U \vdash \mathbf{t} : R$ *and* $U \to R \preceq T$*.* □

Lemma 3.2.10 (Generation lemma (sc)). *For any scalar* $\alpha \neq 0$*, any context* Γ*, any term* $\mathbf{t}$*, any type* T *and any number* $n \in \mathbb{N}$*, if* $S_n = \Gamma \vdash \alpha.\mathbf{t} : \alpha.T$*, then* $\exists m < n$ *such that* $S_m = \Gamma \vdash \mathbf{t} : T$*.* □

Lemma 3.2.11 (Generation lemma (sc-0)). *For any context* Γ*, any term* $\mathbf{t}$*, any type* T *and any number* $n \in \mathbb{N}$*, if* $S_n = \Gamma \vdash 0.\mathbf{t} : T$*, then* $\exists R$ *and* $m < n$ *such that* $S_m = \Gamma \vdash \mathbf{t} : R$*.* □

Lemma 3.2.12 (Generation lemma (sum)). *For any terms* $\mathbf{t}$*,* $\mathbf{r}$*, any scalar* α*, any type* $U \in \mathcal{U}$*, any context* T *and any number* $n \in \mathbb{N}$*, if* $S_n = \Gamma \vdash \mathbf{t} + \mathbf{r} : \alpha.U$*, then* $\exists \delta, \gamma \in \mathcal{S}$ *and* $r, s \in \mathbb{N}$ *with* $\max(r, s) < n$ *such that* $S_r = \Gamma \vdash \mathbf{t} : \delta.U$ *and* $S_s = \Gamma \vdash \mathbf{r} : \gamma.U$ *with* $\delta + \gamma = \alpha$*.* □

The following lemma is quite standard in proofs of subject reduction for System F-like systems, and can be found in [Barendregt, 1992, Krivine, 1990]. It ensures that when substituting type variables for types, or term variables for terms, in an adequate manner, then the type derived remains valid. Its proof is fully depicted in Appendix B.8.

Lemma 3.2.13 (Substitution). *For any term* $\mathbf{t}$*, any base terms* $\mathbf{b}$*, any types* $T \in \mathcal{T}$*,* $\vec{U} \in \mathcal{U}^n$ *and any context* Γ*,*

1. $\Gamma \vdash \mathbf{t} : T \Rightarrow \Gamma[\vec{U}/\vec{X}] \vdash \mathbf{t} : T[\vec{U}/\vec{X}]$.
2. $\{\ \Gamma, x : U \vdash \mathbf{t} : T$ *and* $\Gamma \vdash \mathbf{b} : U\ \} \Rightarrow \Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T$*.* □

The following corollary allows the arrow to be split without needing to consider the order relation:

Corollary 3.2.14 (of Lemma 3.2.9). *For any term* $\mathbf{t}$*, any types* $T \in \mathcal{T}$*,* $U \in \mathcal{U}$ *and any context* Γ*, if* $\Gamma \vdash \lambda x.\mathbf{t} : U \to T$*, then* $\Gamma, x : U \vdash \mathbf{t} : T$*.*

Proof. Let $\Gamma \vdash \lambda x.\mathbf{t} : U \to T$. By Lemma 3.2.9, $\exists V, R$ such that $V \to R \preceq U \to T$ and $\Gamma, x : V \vdash \mathbf{t} : R$, then by Lemma 3.2.6, $\exists \vec{W}, \vec{X}$ such that $U \to T \equiv (V \to R)[\vec{W}/\vec{X}]$ and so by Lemma 3.2.13, $\Gamma[\vec{W}/\vec{X}], x : V[\vec{W}/\vec{X}] \vdash \mathbf{t} : R[\vec{W}/\vec{X}]$, *i.e.* $\Gamma[\vec{W}/\vec{X}], x : U \vdash \mathbf{t} : T$.

Notice that if $\Gamma[\vec{W}/\vec{X}] \equiv \Gamma$, then we have finished. In the other case, $\vec{X}$ appears free on Γ. Since $V \to R \preceq U \to T$ and $\Gamma \vdash \lambda x.\mathbf{t} : V \to R$, according to Lemma 3.2.7, $U \to T$ can be obtained from $V \to R$ as a type for $\lambda x.\mathbf{t}$; then we would need to use the rule $\forall_I$; thus $\vec{X}$ cannot appear free in Γ, which constitutes a contradiction. So, $\Gamma, x : U \vdash \mathbf{t} : T$. □

Lemmas about the scalars

This section contains the lemmas which make statements about the relative behaviour of the scalars within terms and within types. For example, scalars appearing in the terms are reflected within the types also. This is formalised in the following lemma and proved by induction on the typing derivation in Appendix B.9.

Lemma 3.2.15 (Scaling unit). *For any term* $\mathbf{t}$*, scalar* α*, type* T *and context* Γ*, if* $\Gamma \vdash \alpha.\mathbf{t} : T$ *then there exists* $U \in \mathcal{U}$ *and* $\gamma \in \mathcal{S}$ *such that* $T \equiv \alpha.\gamma.U$*.* □

A base term can always be given a unit type (proof by induction on the typing derivation in Appendix B.10).

Lemma 3.2.16 (Base terms in unit)**.** *For any base term* $\mathbf{b}$*, context* Γ *and type* T*,* $\Gamma \vdash \mathbf{b} : T \Rightarrow \exists U \in \mathcal{U}$ *such that* $T \equiv U$*.* □

By $ax_{\overline{0}}$, it is easy to see that $\mathbf{0}$ has type $\overline{0}$, but also by using equivalences between types we have that $\forall X.\overline{0}$ is equivalent to $\overline{0}$ and any T such that $\overline{0} \preceq T$, will also be equivalent to $\overline{0}$. Then we can state the following lemma (proof by induction in Appendix B.11).

Lemma 3.2.17 (Type for $\mathbf{0}$)**.** *For any context* Γ *and type* T*,* $\Gamma \vdash \mathbf{0} : T \Rightarrow T \equiv \overline{0}$*.* □

The following theorem is an important one. It says that our *Scalar* type system is polymorphic only in the unit types but not in the general types in the sense that even if it is possible to derive two types for the same term, the outer left scalar (*i.e.* scalar in the head position) must remain the same. Its proof is not trivial, as it uses several of the previously defined lemmas.

Theorem 3.2.18 (Uniqueness of scalars)**.** *For any term* $\mathbf{t}$*, any context* Γ*, any scalars* α *and* β *and any unit types* U *and* V*, if* $\Gamma \vdash \mathbf{t} : \alpha.U$ *and* $\Gamma \vdash \mathbf{t} : \beta.V$*, then* $\alpha = \beta$*.*

Proof. Structural induction over $\mathbf{t}$.

1. $\mathbf{t} = \mathbf{0}$. Then by Lemma 3.2.17, $\alpha = \beta = 0$.
2. $\mathbf{t} = x$ or $\mathbf{t} = \lambda x.\mathbf{t}'$. Then by Lemma 3.2.16, $\alpha = \beta = 1$.
3. $\mathbf{t} = \gamma.\mathbf{t}'$. Then by Lemma 3.2.15, $\exists \sigma, \delta, U', V'$, such that $\alpha.U \equiv \gamma.\sigma.U'$ and $\beta.V \equiv \gamma.\delta.V'$. If $\gamma = 0$, then $\gamma \times \sigma = \gamma \times \delta = 0$ and then by Lemma 3.2.3, $\alpha = \beta = 0$. If $\gamma \neq 0$, then by Lemma 3.2.10, $\Gamma \vdash \mathbf{t}' : \sigma.U'$ and $\Gamma \vdash \mathbf{t}' : \delta.V'$, so by the induction hypothesis $\sigma = \delta$. Notice that, by Lemma 3.2.3, $\alpha = \gamma \times \sigma$ and $\beta = \gamma \times \delta$, so $\alpha = \gamma \times \sigma = \gamma \times \delta = \beta$.
4. $\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2$. Then by Lemma 3.2.12, $\exists \gamma_1, \gamma_2$ such that $\Gamma \vdash \mathbf{t}_1 : \gamma_1.U$ and $\Gamma \vdash \mathbf{t}_2 : \gamma_2.U$ with $\gamma_1 + \gamma_2 = \alpha$; and also by the same Lemma, $\exists \delta_1, \delta_2$ such that $\Gamma \vdash \mathbf{t}_1 : \delta_1.V$ and $\Gamma \vdash \mathbf{t}_2 : \delta_2.V$ with $\delta_1 + \delta_2 = \beta$. Then by the induction hypothesis $\gamma_1 = \delta_1$ and $\gamma_2 = \delta_2$, so $\alpha = \gamma_1 + \gamma_2 = \delta_1 + \delta_2 = \beta$.
5. $\mathbf{t} = (\mathbf{t}_1)\ \mathbf{t}_2$. Then by Lemma 3.2.8, $\exists \gamma_1, \gamma_2, W$ and $T \preceq U$ such that $\Gamma \vdash \mathbf{t}_1 : \gamma_1.W \to T$ and $\Gamma \vdash \mathbf{t}_2 : \gamma_2.W$ with $\gamma_1 \times \gamma_2 = \alpha$; and also by the same Lemma, $\exists \delta_1, \delta_2, W'$ and $R \preceq V$ such that $\Gamma \vdash \mathbf{t}_1 : \delta_1.W' \to R$ and $\Gamma \vdash \mathbf{t}_2 : \delta_2.W'$ with $\delta_1 \times \delta_2 = \beta$. Then by the induction hypothesis $\gamma_1 = \delta_1$ and $\gamma_2 = \delta_2$, so $\alpha = \gamma_1 \times \gamma_2 = \delta_1 \times \delta_2 = \beta$.

□

From this theorem, the uniqueness of $\overline{0}$ comes out, in the sense that no term can have type $\overline{0}$ and some other type T which is not equivalent to $\overline{0}$.

Corollary 3.2.19 (Uniqueness of $\overline{0}$)**.** *For any term* $\mathbf{t}$ *and any context* Γ*,* $\Gamma \vdash \mathbf{t} : \overline{0} \Rightarrow \forall T \not\equiv \overline{0}, \Gamma \nvdash \mathbf{t} : T$*.*

Proof. Assume $\Gamma \vdash \mathbf{t} : T$, then by Lemma 3.2.2, $T \equiv \alpha.U$. Since $\Gamma \vdash \mathbf{t} : \overline{0} \equiv 0.U$, then by Theorem 3.2.18, $\alpha = 0$. □

Since $\mathbf{0}$ has type $\overline{0}$ which is equivalent to $0.U$ for any U, $\mathbf{0}$ can still act as argument for an abstraction, or even be applied to another term. In either case the result will be a term of type $\overline{0}$:

Lemma 3.2.20 (Linearity of $\mathbf{0}$)**.** *For any term* $\mathbf{t}$*, any context* Γ *and any type* T*,*

1. $\Gamma \vdash (\mathbf{0})\ \mathbf{t} : T \Rightarrow T \equiv \overline{0}$. *2.* $\Gamma \vdash (\mathbf{t})\ \mathbf{0} : T \Rightarrow T \equiv \overline{0}$.

Proof.

1. Let $\Gamma \vdash (\mathbf{0})\ \mathbf{t} : T$. By Lemma 3.2.2, $T \equiv \gamma.U$. Moreover, by Lemma 3.2.8, $\exists \alpha, \beta, U'$ and $R \preceq U$ such that $\Gamma \vdash \mathbf{0} : \beta.U' \to R$ and $\Gamma \vdash \mathbf{t} : \alpha.U'$ with $\gamma = \alpha \times \beta$. Hence, by Corollary 3.2.19, $\beta.U' \to R \equiv \overline{0} \equiv 0.U$, so by Lemma 3.2.3, $\beta = 0$, then $\gamma = \alpha \times 0 = 0$, so $T \equiv \gamma.U \equiv \overline{0}$.
2. Analogous to 1.

□

Subject reduction cases

The following three lemmas are in fact cases of subject reduction, however, they will also be necessary as lemmas in subsequent proofs.

Lemma 3.2.21 (Product)**.** *For any term* $\mathbf{t}$*, any scalars* α *and* β *and any context* Γ*,* $\Gamma \vdash \alpha.(\beta.\mathbf{t}) : T \Rightarrow \Gamma \vdash (\alpha \times \beta).\mathbf{t} : T$.

Proof. By Lemma 3.2.15, $\exists U \in \mathcal{U}, \gamma \in \mathcal{S}$ such that $T \equiv \alpha.\gamma.U$. Cases:

$\alpha \neq 0$ and $\beta \neq 0$: By Lemma 3.2.10, $\Gamma \vdash \beta.\mathbf{t} : \gamma.U$. Moreover, by Lemma 3.2.15 again, $\exists U' \in \mathcal{U}, \gamma' \in \mathcal{S}$ such that $\gamma.U \equiv \beta.\gamma'.U'$. So, by Lemma 3.2.10, $\Gamma \vdash \mathbf{t} : \gamma'.U'$, from which, using rule s_I one can derive $\Gamma \vdash (\alpha \times \beta).\mathbf{t} : (\alpha \times \beta).\gamma'.U'$. Notice that $(\alpha \times \beta).\gamma'.U' \equiv \alpha.\beta.\gamma'.U' \equiv \alpha.\gamma.U \equiv T$.

$\alpha = 0$: By Lemma 3.2.11, $\exists R$ such that $\Gamma \vdash \beta.\mathbf{t} : R$. Then by Lemma 3.2.10 or 3.2.11, depending if $\beta = 0$ or not, $\exists S$ such that $\Gamma \vdash \mathbf{t} : S$, from which, using rule s_I, one can derive $\Gamma \vdash 0.\mathbf{t} : 0.S$. Notice that $0.S \equiv \overline{0} \equiv 0.\gamma.U \equiv T$. Also notice that $0.\mathbf{t} = (0 \times \beta).\mathbf{t}$.

□

Lemma 3.2.22 (Distributivity)**.** *For any terms* $\mathbf{t}$ *and* $\mathbf{r}$*, any scalar* α*, any context* Γ *and any type* T*,* $\Gamma \vdash \alpha.(\mathbf{t} + \mathbf{r}) : T \Rightarrow \Gamma \vdash \alpha.\mathbf{t} + \alpha.\mathbf{r} : T$.

Proof. Let $\Gamma \vdash \alpha.(\mathbf{t} + \mathbf{r}) : T$. By Lemma 3.2.15, $\exists \alpha, R$ such that $T \equiv \alpha.R$. Cases

$\alpha \neq 0$: By Lemma 3.2.10, $\Gamma \vdash \mathbf{t} + \mathbf{r} : R$. Since $R \equiv 1.R$, by Lemma 3.2.12, $\Gamma \vdash \mathbf{t} : \delta.R$ and $\Gamma \vdash \mathbf{r} : \gamma.R$ with $\delta + \gamma = 1$. Then by rules s_I and $\equiv$, $\Gamma \vdash \alpha.\mathbf{t} : (\alpha \times \delta).R$ and $\Gamma \vdash \alpha.\mathbf{r} : (\alpha \times \gamma).R$, from which using rule $+_I$ one can derive $\Gamma \vdash \alpha.\mathbf{t} + \alpha.\mathbf{r} : (\alpha \times \delta + \alpha \times \gamma).R$. Notice that $(\alpha \times \delta + \alpha \times \gamma).R = \alpha.R \equiv T$.

$\alpha = 0$: Then $T \equiv 0.R \equiv \overline{0}$ and by Lemma 3.2.11, $\exists S$ such that $\Gamma \vdash \mathbf{t} + \mathbf{r} : S$. In addition, by Lemma 3.2.2, $\exists \delta, V$ such that $S \equiv \delta.V$. Then by Lemma 3.2.12, $\exists \sigma, \varsigma$ such that $\Gamma \vdash \mathbf{t} : \sigma.V$ and $\Gamma \vdash \mathbf{r} : \varsigma.V$. By rules s_I and $\equiv$, $\Gamma \vdash 0.\mathbf{t} : 0.V$ and $\Gamma \vdash 0.\mathbf{r} : 0.V$, from which using rule $+_I$ one can derive $\Gamma \vdash 0.\mathbf{t} + 0.\mathbf{r} : 0.V$. Notice that $0.V \equiv \overline{0} \equiv T$.

□

Lemma 3.2.23 (Factorisation)**.** *For any term* $\mathbf{t}$*, scalars* α *and* β*, type* T *and context* Γ*,* $\Gamma \vdash \alpha.\mathbf{t} + \beta.\mathbf{t} : T \Rightarrow \Gamma \vdash (\alpha + \beta).\mathbf{t} : T$.

Proof. Let $\Gamma \vdash \alpha.\mathbf{t} + \beta.\mathbf{t} : T$. By Lemma 3.2.12, $\exists \delta, \gamma \in \mathcal{S}$ such that $\Gamma \vdash \alpha.\mathbf{t} : \delta.T$ and $\Gamma \vdash \beta.\mathbf{t} : \gamma.T$ with $\delta + \gamma = 1$. In addition, by Lemma 3.2.2, $\exists U \in \mathcal{U}$ and $\sigma \in \mathcal{S}$ such that $T \equiv \sigma.U$. Then $\Gamma \vdash \alpha.\mathbf{t} : \delta.\sigma.U$ and $\Gamma \vdash \beta.\mathbf{t} : \gamma.\sigma.U$. Then by Lemma 3.2.15, $\exists \phi, \varphi \in \mathcal{S}$ and $U', U'' \in \mathcal{U}$ such that $\delta.\sigma.U \equiv \alpha.\phi.U'$ and $\gamma.\sigma.U \equiv \beta.\varphi.U''$. So, by Lemma 3.2.3, $\delta \times \sigma = \alpha \times \phi$ and $\gamma \times \sigma = \beta \times \varphi$. Cases

$\sigma = 0$**:** Then $T \equiv \overline{0}$, and so one can derive $\Gamma \vdash \alpha.\mathbf{t} : \alpha.\overline{0}$. Thus by Lemma 3.2.10, $\Gamma \vdash \mathbf{t} : \overline{0}$. Using rules s_I and $\equiv$, one can derive $\Gamma \vdash (\alpha + \beta).\mathbf{t} : \overline{0}$.

$\sigma \neq 0$**,** $\delta = 0$**:** Then since $\delta + \gamma = 1$, $\gamma = 1$, and so $\Gamma \vdash \beta.\mathbf{t} : T \equiv \beta.\varphi.U''$, then by Lemma 3.2.10, $\Gamma \vdash \mathbf{t} : \varphi.U''$. Using rules s_I and $\equiv$, one can derive $\Gamma \vdash (\alpha + \beta).\mathbf{t} : ((\alpha + \beta) \times \varphi).U''$. Since $\delta = 0$, the possible cases are:

$\alpha = 0$**:** Then $((\alpha + \beta) \times \varphi).U'' \equiv (\beta \times \varphi).U'' \equiv \sigma.U \equiv T$.

$\alpha \neq 0$**:** Then, since $\Gamma \vdash \alpha.\mathbf{t} : \overline{0} \equiv \alpha.\overline{0}$, by Lemma 3.2.10, $\Gamma \vdash \mathbf{t} : \overline{0}$. In addition, as $\Gamma \vdash \beta.\mathbf{t} : \beta.\varphi.U''$, by Lemma 3.2.10, $\Gamma \vdash \mathbf{t} : \varphi.U''$, then by Corollary 3.2.19, $\varphi.U'' \equiv \overline{0}$, so $\varphi = 0$, and then $\gamma = 0$, so $\delta = 1$, which is a contradiction.

$\sigma \neq 0$**,** $\gamma = 0$**:** Analogous to previous case.

$\alpha, \beta, \phi, \varphi \neq 0$**:** Then by Lemma 3.2.3, $U \equiv U' \equiv U''$. Then $\Gamma \vdash \alpha.\mathbf{t} : \alpha.\phi.U$ and $\Gamma \vdash \beta.\mathbf{t} : \beta.\varphi.U$. Hence by Lemma 3.2.10, $\Gamma \vdash \mathbf{t} : \phi.U$ and $\Gamma \vdash \mathbf{t} : \varphi.U$. Then by Theorem 3.2.18, $\phi = \varphi$ and then by rule s_I, one can derive $\Gamma \vdash (\alpha + \beta).\mathbf{t} : (\alpha + \beta).\phi.U$. Notice that $(\alpha + \beta).\phi.U \equiv ((\alpha + \beta) \times \phi).U = (\alpha \times \phi + \beta \times \varphi).U = (\delta \times \sigma + \gamma \times \sigma).U = ((\delta + \gamma) \times \sigma).U = (1 \times \sigma).U = \sigma.U \equiv T$.

□

3.2.2 Subject reduction proof

Now we are able to prove the subject reduction property (Theorem 3.2.1).

Proof. We proceed by checking that every reduction rule preserves the type. We give two cases as example, the full proof can be find in Appendix B.12.

rule $(\mathbf{t} + \mathbf{r})\ \mathbf{u} \to (\mathbf{t})\ \mathbf{u} + (\mathbf{r})\ \mathbf{u}$**:** Let $\Gamma \vdash (\mathbf{t} + \mathbf{r})\ \mathbf{u} : T \equiv 1.T$. Then, by Lemma 3.2.8, $\exists \alpha, \beta, U$ and $T' \preceq T$ such that $\Gamma \vdash \mathbf{u} : \alpha.U$ and $\Gamma \vdash \mathbf{t} + \mathbf{r} : \beta.U \to T' \equiv 1.\beta.U \to T'$ with $\alpha \times \beta = 1$. Then by Lemma 3.2.12, $\exists \delta$ and γ such that $\Gamma \vdash \mathbf{t} : \delta.\beta.U \to T' \equiv (\delta \times \beta).U \to T'$ and $\Gamma \vdash \mathbf{r} : \gamma.\beta.U \to T' \equiv (\gamma \times \beta).U \to T'$ with $\delta + \gamma = 1$. Then by rule $\to_E$, $\Gamma \vdash (\mathbf{t})\ \mathbf{u} : (\delta \times \beta \times \alpha).T'$ and $\Gamma \vdash (\mathbf{r})\ \mathbf{u} : (\gamma \times \beta \times \alpha).T'$. Notice that $(\delta \times \beta \times \alpha).T' = (\delta \times 1).T' = \delta.T'$ and $(\gamma \times \beta \times \alpha).T' = (\gamma \times 1).T' = \gamma.T'$. Then by Lemmas 3.2.5 and 3.2.7 $\Gamma \vdash (\mathbf{t})\ \mathbf{u} : \delta.T$ and $\Gamma \vdash (\mathbf{r})\ \mathbf{u} : \gamma.T$, from which, using rule $+_I$, one can derive $\Gamma \vdash (\mathbf{t})\ \mathbf{u} + (\mathbf{r})\ \mathbf{u} : (\delta + \gamma).T \equiv T$.

rule $(\lambda x.\mathbf{t})\ \mathbf{b} \to \mathbf{t}[\mathbf{b}/x]$**:** Let $\Gamma \vdash (\lambda x.\mathbf{t})\ \mathbf{b} : T$. By rule $\equiv$, $\Gamma \vdash (\lambda x.\mathbf{t})\ \mathbf{b} : 1.T$, so by Lemma 3.2.8, $\exists \alpha, \beta, U, T' \preceq T$ such that $\Gamma \vdash \lambda x.\mathbf{t} : \beta.U \to T'$ and $\Gamma \vdash \mathbf{b} : \alpha.U$ with $\alpha \times \beta = 1$. Since $\mathbf{b}$ is a base term, by Lemma 3.2.16, $\alpha = 1$ and so $\beta = 1$. Then by Corollary 3.2.14, $\Gamma, x : U \vdash \mathbf{t} : T'$. Thus, by Lemma 3.2.13, $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T'$, from which, by Lemma 3.2.7, one obtains $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T$.

□

3.3 Strong normalisation, simplified reduction rules and confluence

The *Scalar* type system will now be proved to have the strong normalisation property, *i.e.* every typable term is strongly normalising, it cannot reduce forever. In order to show this we first set up another

type system, which simply 'forgets' the scalars. Hence this simpler type system is just a straightforward extension of System F to λ_{lin}, which we call $\lambda 2^{la}$ (definition 3.3.1). In the literature surrounding not λ_{lin} but its cousin, the algebraic λ-calculus, one finds such a System F in [Ehrhard, 2010], which extends the simply typed algebraic λ-calculus of [Vaux, 2007, 2009] – our $\lambda 2^{la}$ is very similar. Secondly we prove strong normalisation for it (Theorem 3.3.8). Thirdly we show that every term which has a type in *Scalar* has a type in $\lambda 2^{la}$ (Lemma 3.3.10), which entails strong normalisation in *Scalar* (Theorem 3.3.11).

This strong normalisation proof constitutes the second main technical contribution of the chapter. The confluence of a simplified version of λ_{lin} is proven as a corollary.

In this section we use the following notation: $\mathbb{T}(\lambda 2^{la})$ is the set of types of $\lambda 2^{la}$. Λ is the set of terms of λ_{lin}. $\Gamma \Vdash \mathbf{t} : T$ says that it is possible to derive the type $T \in \mathbb{T}(\lambda 2^{la})$ for the term $\mathbf{t} \in \Lambda$ in the context Γ under the typing rules of $\lambda 2^{la}$. We just use $\vdash$ for *Scalar*. In addition, we use $Name^{\triangleleft}$ to distinguish the names of the typing rules in $\lambda 2^{la}$ from those of *Scalar*.

Definition 3.3.1. *The type grammar of $\lambda 2^{la}$ is the following:*

$$A, B, C = X \mid A \to B \mid \forall X.A$$

The typing rules of $\lambda 2^{la}$ are those of System F plus the following rules:

$$\frac{}{\Gamma \Vdash \mathbf{0} : A}\, ax_0^{\triangleleft} \qquad \frac{\Gamma \Vdash \mathbf{t} : A \qquad \Gamma \Vdash \mathbf{r} : A}{\Gamma \Vdash \mathbf{t} + \mathbf{r} : A}\, +_I^{\triangleleft} \qquad \frac{\Gamma \Vdash \mathbf{t} : A}{\Gamma \Vdash \alpha.\mathbf{t} : A}\, s_I^{\triangleleft}$$

In order to prove strong normalisation we extend the proof for $\lambda 2$. The standard method was invented by Tait [Tait, 1967] for simply typed λ-calculus and generalised to System F by Girard [Girard, 1972]. Our presentation follows [Barendregt, 1992, Section 4.3]. The following definitions are taken from this reference – with slight modifications to handle the extra $\lambda 2^{la}$ rules.

The strong normalisation property entails that every term is strongly normalising, so first we define the set of strongly normalising terms.

Definition 3.3.2. $SN = \{\mathbf{t} \in \Lambda \mid \mathbf{t}$ *is strongly normalising*$\}$.

The notion of closure is often captured by the notion of saturated set. We use the notation $\vec{\mathbf{t}} = \mathbf{t}_1, \ldots, \mathbf{t}_n$ with $n \geq 0$. Also $(\mathbf{r})\ \vec{\mathbf{t}} = (((\mathbf{r})\ \mathbf{t}_1) \ldots)\ \mathbf{t}_n$ where if $n = 0$ it is just $\mathbf{r}$.

Definition 3.3.3.

1. *A subset $X \subseteq SN$ is called* saturated *if*

   (a) $\mathbf{0} \in X$;

   (b) $\forall x, \vec{\mathbf{t}} \in SN$, $(x)\ \vec{\mathbf{t}} \in X$;

   (c) $(\mathbf{t}[\mathbf{b}/x])\ \vec{\mathbf{r}} \in X \Rightarrow ((\lambda x.\mathbf{t})\ \mathbf{b})\ \vec{\mathbf{r}} \in X$;

   (d) $(\forall i \in I, (\mathbf{t}_i)\ \vec{\mathbf{r}} \in X) \Rightarrow (\sum_{i \in I} \mathbf{t}_i)\ \vec{\mathbf{r}} \in X$;

   (e) $(\forall i \in I, ((\mathbf{u})\ \mathbf{t}_i)\ \vec{\mathbf{r}} \in X) \Rightarrow ((\mathbf{u})\ \sum_{i \in I} \mathbf{t}_i)\ \vec{\mathbf{r}} \in X$;

   (f) $\forall \alpha \in \mathcal{S}$, $\mathbf{t} \in X \Leftrightarrow \alpha.\mathbf{t} \in X$;

   (g) $\alpha.((((\mathbf{t}_1)\ \mathbf{t}_2)\ \ldots)\ \mathbf{t}_n) \in X \Leftrightarrow (((((\mathbf{t}_1)\ \mathbf{t}_2) \ldots)\ \alpha.\mathbf{t}_k) \ldots)\ \mathbf{t}_n \in X\ (1 \leq k \leq n)$;

   (h) $\forall \vec{\mathbf{t}} \in SN$, $(\mathbf{0})\ \vec{\mathbf{t}} \in X$;

(i) $\forall \mathbf{t}, \vec{\mathbf{u}} \in SN$, $((\mathbf{t})\ \mathbf{0})\ \vec{\mathbf{u}} \in X$.

2. $SAT = \{X \subseteq \Lambda \mid X \text{ is saturated}\}$

The basic idea is to prove that types correspond to saturated sets. In order to achieve this, we define a valuation from types to SAT (in fact, from type variables to SAT and then, we define a set in SAT by using such a valuation).

Definition 3.3.4.

1. *A* valuation in SAT *is a map* $\xi : \mathbb{V} \to SAT$, *where* $\mathbb{V}$ *is the set of type variables.*
2. *For any* $A, B \subseteq \Lambda$, *we define* $A \to B = \{\mathbf{t} \in \Lambda \mid \forall \mathbf{r} \in A, (\mathbf{t})\ \mathbf{r} \in B\}$.
3. *Given a valuation* ξ *in* SAT, *we define for every* $T \in \mathbb{T}(\lambda 2^{la})$ *a set* $[\![T]\!]_\xi \subseteq \Lambda$ *as follows:*

$$\begin{aligned} [\![X]\!]_\xi &= \xi(X), \text{ where } X \in \mathbb{V} \\ [\![A \to B]\!]_\xi &= [\![A]\!]_\xi \to [\![B]\!]_\xi \\ [\![\forall X.A]\!]_\xi &= \bigcap_{Y \in SAT} [\![A]\!]_{\xi(X:=Y)} \end{aligned}$$

Lemma 3.3.5.

1. $SN \in SAT$,
2. $A, B \in SAT \Rightarrow A \to B \in SAT$,
3. *Let* $\{A_i\}_{i \in I}$ *be a collection of members of* SAT, $\bigcap_{i \in I} A_i \in SAT$,
4. *Given a valuation* ξ *in* SAT *and a* A *in* $\mathbb{T}(\lambda 2^{la})$, *then* $[\![A]\!]_\xi \in SAT$.

Proof. cf. Appendix B.13. □

Just like in definition 3.3.4, we define another valuation, this time from term variables to base terms. We use it to check what happens when we change every free variable of a term for any other base term. The basic idea is the following: we define $\rho, \xi \vDash \mathbf{t} : A$ to be the property of changing every free term variable in $\mathbf{t}$ for another term with the help of the valuation ρ (a base term, since term variables only run over base terms) and still having the resulting term in the set $[\![A]\!]_\xi$. So, we define $\Gamma \vDash \mathbf{t} : A$ to be the same property, when the property holds for every pair in Γ and for every valuations ρ and ξ.

This is formalised in the following definition (definition 3.3.6) and with this definition, we prove that if a term has a type in a valid context, then the property above holds (Theorem 3.3.7), which will yield the strong normalisation theorem (Theorem 3.3.8) via the concept of *saturated* set (because *saturated* sets are subsets of SN).

Definition 3.3.6.

- *A* valuation in Λ *is a map* $\rho : V \to \Lambda_b$, *where* V *is the set of term variables and* $\Lambda_b = \{\mathbf{b} \in \Lambda \mid \mathbf{b} \text{ is a base term}\}$.
- *Let* ρ *be a valuation in* Λ. *Then* $[\![\mathbf{t}]\!]_\rho = \mathbf{t}[x_1 := \rho(x_1), \ldots, x_n := \rho(x_n)]$, *where* $\vec{x} = x_1, \ldots, x_n$ *is the set of free variables in* $\mathbf{t}$.

- *Let ρ be a valuation in Λ and ξ a valuation in SAT. Then*
  - ρ, ξ satisfies $\mathbf{t} : A$, *notation* $\rho, \xi \vDash \mathbf{t} : A \Leftrightarrow [\![\mathbf{t}]\!]_\rho \in [\![A]\!]_\xi$.
  - $\rho, \xi \vDash \Gamma \Leftrightarrow \rho, \xi \vDash x : A$ *for all* $x : A$ *in* Γ
  - $\Gamma \vDash \mathbf{t} : A \Leftrightarrow \forall \rho, \xi \ [\rho, \xi \vDash \Gamma \Rightarrow \rho, \xi \vDash \mathbf{t} : A]$.

Theorem 3.3.7 (Soundness)**.** $\Gamma \Vdash \mathbf{t} : A \Rightarrow \Gamma \vDash \mathbf{t} : A$.

Proof. We proceed by induction on the derivation of $\Gamma \Vdash \mathbf{t} : T$. We show one case as example. The full proof is in Appendix B.14.

$$\frac{\Gamma \Vdash \mathbf{t} : A \to B \qquad \Gamma \Vdash \mathbf{r} : A}{\Gamma \Vdash (\mathbf{t})\ \mathbf{r} : B} \to_E^{\triangleleft}$$

By the induction hypothesis, $\Gamma \vDash \mathbf{t} : A \to B$ and $\Gamma \vDash \mathbf{r} : A$. Assume $\rho, \xi \vDash \Gamma$ in order to show $\rho, \xi \vDash (\mathbf{t})\ \mathbf{r} : B$. Then $\rho, \xi \vDash \mathbf{t} : A \to B$, *i.e.* $[\![\mathbf{t}]\!]_\rho \in [\![A \to B]\!]_\xi = [\![A]\!]_\xi \to [\![B]\!]_\xi$ and $[\![\mathbf{r}]\!]_\rho \in [\![A]\!]_\xi$. Then $[\![(\mathbf{t})\ \mathbf{r}]\!]_\rho = [\![\mathbf{t}]\!]_\rho\ [\![\mathbf{r}]\!]_\rho \in [\![B]\!]_\xi$, so $\rho, \xi \vDash (\mathbf{t})\ \mathbf{r} : B$.

□

Theorem 3.3.8 (Strong normalisation for $\lambda 2^{la}$)**.** $\Gamma \Vdash \mathbf{t} : A \Rightarrow \mathbf{t}$ *is strongly normalising.*

Proof. Let $\Gamma \Vdash \mathbf{t} : A$. Then by Theorem 3.3.7, $\Gamma \vDash \mathbf{t} : A$. Define $\rho_0(x) = x$ for all x and let ξ be a valuation in SAT. Then $\rho_0, \xi \vDash \Gamma$ (*i.e.* for all $(x : B) \in \Gamma$, $\rho_0, \xi \vDash x : B$ since $x \in [\![B]\!]_\xi$ holds because $[\![B]\!]_\xi$ is saturated). Therefore $\rho_0, \xi \vDash \mathbf{t} : A$, hence $\mathbf{t} = [\![\mathbf{t}]\!]_{\rho_0} \in [\![A]\!]_\xi \subseteq SN$. □

It is possible to map every type from *Scalar* to a type in $\lambda 2^{la}$ as follows.

Definition 3.3.9. *Let* $(\cdot)^\natural : \mathcal{T} \to \mathbb{T}(\lambda 2^{la})$ *be the following mapping:*

- $X^\natural = X$
- $(\forall X.T)^\natural = \forall X.T^\natural$
- $(U \to T)^\natural = U^\natural \to T^\natural$
- $(\alpha.T)^\natural = T^\natural$
- $\overline{0}^\natural = A$

with $A \in \mathbb{T}(\lambda 2^{la})$.

We also use the following abuse of notation $\Gamma^\natural = \{(x : T^\natural) \mid (x : T) \in \Gamma\}$.

The following lemma ensures that if it is possible to give a type to a term in *Scalar* then it is possible to give to the term the mapped type in $\lambda 2^{la}$.

Lemma 3.3.10 (Correspondence with $\lambda 2^{la}$)**.** $\Gamma \vdash \mathbf{t} : T \Rightarrow \Gamma^\natural \Vdash \mathbf{t} : T^\natural$.

Proof. We proceed by induction on the derivation of $\Gamma \vdash \mathbf{t} : T$. We show one case as example. The full proof is in Appendix B.15.

$$\frac{\Gamma \vdash \mathbf{t} : \alpha.T \qquad \Gamma \vdash \mathbf{r} : \beta.T}{\Gamma \vdash \mathbf{t} + \mathbf{r} : (\alpha + \beta).T} +_I$$

By the induction hypothesis $\Gamma^\natural \Vdash \mathbf{t} : T^\natural$ and $\Gamma^\natural \Vdash \mathbf{r} : T^\natural$, so by rule $+_I^{\triangleleft}$, $\Gamma^\natural \Vdash \mathbf{t} + \mathbf{r} : T^\natural = ((\alpha + \beta).T)^\natural$.

□

Strong normalisation arises as a consequence of strong normalisation for $\lambda 2^{la}$ and the above lemma.

Theorem 3.3.11 (Strong normalisation)**.** $\Gamma \vdash \mathbf{t} : T \Rightarrow \mathbf{t}$ *is strongly normalising.*

Proof. By Lemma 3.3.10, $\Gamma^\natural \Vdash \mathbf{t} : T^\natural$, then by Theorem 3.3.8, $\mathbf{t}$ is strong normalising. □

Taking up again the example of Section 1.2, diverging terms like $\mathbf{Y}$ are simply not allowed in this typed setting, since Theorem 3.3.11 will ensure that all the typable terms have a normal form. So we do not have infinities, and hence the intuitive reasons for having restrictions $(*)$ on the *Factorisation rules* of the Linear-algebraic calculus (*cf.* the reduction rules in Section 1.2) have now vanished. If we drop them, the example becomes as follows.

Example 3.3.12. *Consider some arbitrary typable, and hence normalising term* $\mathbf{t}$*. Then* $\alpha.\mathbf{t} - \alpha.\mathbf{t}$ *can be reduced by a factorisation rule into* $(\alpha - \alpha).\mathbf{t}$*. This reduces in one step to* $\mathbf{0}$*, without the need to reduce* $\mathbf{t}$*.*

It turns out that, in general, for typable terms we can indeed drop the restrictions (*) and (**) without breaking the confluence of λ_{lin}. These restrictions were there only due to the impossibility of checking for the normalisation property in the untyped setting. In fact the confluence becomes a corollary of the strong normalisation theorem.

Corollary 3.3.13 (Confluence)**.** *Let* $\mathbf{t}$ *be a term of* λ_{lin}*, as it appears in Figure 1.1, but without restrictions (*) and (**). If* $\mathbf{t}$ *is typable in Scalar, and* $\mathbf{t} \to^* \mathbf{u}$ *and* $\mathbf{t} \to^* \mathbf{r}$*, then there exists a term* $\mathbf{t}'$ *such that* $\mathbf{u} \to^* \mathbf{t}'$ *and* $\mathbf{r} \to^* \mathbf{t}'$*.*

Proof. First let us introduce some notation. Let $\to_\beta$ be a β-reduction and $\to_a$ any reduction from Figure 1.1 but the β-reduction, without restrictions (*) and (**).

The proof follows in several steps.

1. First we prove local confluence for the algebraic fragment, *i.e.* if $\mathbf{t} \to_a \mathbf{u}$ and $\mathbf{t} \to_a \mathbf{r}$, then there exists a term $\mathbf{t}'$ such that $\mathbf{u} \to_a^* \mathbf{t}'$ and $\mathbf{r} \to_a^* \mathbf{t}'$.

2. Then we prove local confluence for the β-reduction, *i.e.* if $\mathbf{t} \to_\beta \mathbf{u}$ and $\mathbf{t} \to_\beta \mathbf{r}$, then there exists a term $\mathbf{t}'$ such that $\mathbf{u} \to_\beta^* \mathbf{t}'$ and $\mathbf{r} \to_\beta^* \mathbf{t}'$.

3. Finally, we prove that algebraic rules and β-reduction commutes, *i.e.* if $\mathbf{t} \to_a \mathbf{u}$ and $\mathbf{t} \to_\beta \mathbf{r}$, then there exists a term $\mathbf{t}'$ such that $\mathbf{u} \to^* \mathbf{t}'$ and $\mathbf{r} \to^* \mathbf{t}'$, where $\to^*$ is a reduction sequence of zero or more steps involving any rules.

This proves the local confluence of the system. Local confluence plus strong normalisation implies confluence (*cf.* for example [TeReSe, 2003]).

Let proceed with the proofs.

1. Valiron did a semi-automatised proof in the interactive theorem prover Coq [Coq Dev. Team, 2009]. The interested reader can find the proof in [Valiron, 2011b].

2. The confluence of the β-reduction is a trivial extension of the confluence of lambda-calculus.

3. This proof goes by structural induction. *cf.* Appendix B.16.

□

Notice that the proofs of subject reduction (Theorem 3.2.1) and strong normalisation (Theorem 3.3.11) have been done in the general case, without consider restrictions (*) and (**), so they are still valid for the simplified calculus.

Having dropped restrictions (*) and (**) is an important simplification of the linear-algebraic λ-calculus, which becomes really just an oriented version of the axioms of vector spaces [Arrighi and Dowek, 2004] together with a linear extension of the β-reduction (*i.e.* restriction (***) remains of course, which makes this calculus to be call-by-base and all functions remain linear in their arguments, in the sense of linear-algebra).

3.4 Barycentric λ-calculus

By slightly modifying our system, the *Scalar* type system may be used in order to specialise λ_{lin} into a higher-order barycentric calculus. In order to illustrate this point, let us consider the following type judgement, which can be obtained from *scalar*:

$$f ::= \lambda x.(((x)\ (\frac{1}{2}.(\mathbf{true} + \mathbf{false})))\ (\frac{1}{4}.\mathbf{true} + \frac{3}{4}.\mathbf{false})) : \mathbb{B} \to \mathbb{B};$$

where $\mathbb{B}$ stands for $\forall X.X \to X \to X$. Notice that the type $\mathbb{B}$ has **true**, **false**, and linear combinations of them with scalars summing to one, as members. In this example the type system provides a guarantee that the function is barycentric, and that if it receives a barycentric argument, it preserves this property. For example, if we apply such a function to $\frac{1}{2}.(\mathbf{true} + \mathbf{false})$ we obtain:

$$(f)\ (\frac{1}{2}.(\mathbf{true} + \mathbf{false})) \longrightarrow^* \frac{3}{8}.\mathbf{true} + \frac{5}{8}.\mathbf{false}.$$

Although this seems feasible, we will not develop a full-blown barycentric higher-order λ-calculus and associated properties in this thesis. We will just show that the *scalar* type system accomplishes part of the job by checking for the barycentric property, *i.e.* checking that the normal form of a term has amplitudes summing to one. A barycentric λ-calculus fragment of the algebraic λ-calculus has already been studied for its own sake [Tasson, 2009], however in this work the calculus was endowed with a simple type system, not one that would recognise barycentric terms amongst other terms.

To this end let us define a type system with the rules and grammar of *Scalar*, but where the valid types are the classic ones (*i.e.* types exempt of any scalar, which we have referred to as $\mathbb{T}(\lambda 2^{la})$ in Definition 3.3.1), whilst all the other types are just intermediate types:

Definition 3.4.1. *We define the type system $\mathcal{B}$ for the barycentric calculus to be the Scalar type system with the following restrictions:*

- $\mathcal{S} = \mathbb{R}$,
- *Contexts are sets of tuples $(x : A)$, with $A \in \mathbb{T}(\lambda 2^{la})$,*
- *Type variables run over $\mathbb{T}(\lambda 2^{la})$ instead of unit types,* i.e. *the rule $\forall_E$ accepts only $A \in \mathbb{T}(\lambda 2^{la})$,*
- *The final sequent have to be* well-formed *in the following sense: $\forall A \in \mathbb{T}(\lambda 2^{la})$, any derivable sequent $\Gamma \vdash \mathbf{t} : A$ is well-formed, even if the derivation has scalars appearing at intermediate stages.*

In order to show that this type system does the job, let us define the *weight function* which checks for the barycentric property:

Definition 3.4.2. *Let $\omega : \Lambda \to \mathbb{R}$ be a function defined inductively by:*

$$\omega(\mathbf{0}) = 0; \quad \omega(\mathbf{b}) = 1; \quad \omega(\mathbf{t}_1 + \mathbf{t}_2) = \omega(\mathbf{t}_1) + \omega(\mathbf{t}_2); \quad \omega((\mathbf{t}_1)\ \mathbf{t}_2) = \omega(\mathbf{t}_1) \times \omega(\mathbf{t}_2); \quad \omega(\alpha.\mathbf{t}) = \alpha \times \omega(\mathbf{t})$$

where $\mathbf{b}$ is a base term.

We can enunciate the following theorem which shows that every term with a well-formed typing in the type system $\mathcal{B}$ reduces to a term with weight 1:

Theorem 3.4.3 (Normal-form of terms in $\mathcal{B}$ have weight 1)**.** *Let $\Gamma \vdash \mathbf{t} : A$ be well-formed, then $\omega(\mathbf{t}\downarrow) = 1$.*

Proof. Instead, we prove the more general case: $\Gamma \vdash \mathbf{t} : \alpha.A \Rightarrow \omega(\mathbf{t}\downarrow) = \alpha$, by structural induction on $\mathbf{t}\downarrow$. We take $\Gamma \vdash \mathbf{t}\downarrow : \alpha.A$, which is true by Theorem 3.2.1. We show one case as example. The full proof is in Appendix B.17.

Take the case $\mathbf{t} \downarrow= \gamma.\mathbf{t}'$. Then $\omega(\mathbf{t} \downarrow) = \gamma.\omega(\mathbf{t}')$. By Lemma 3.2.15, $\exists U \in \mathcal{U}, \delta \in \mathcal{S}$ such that $\alpha.A \equiv \gamma.\delta.U$. Notice that $\mathbb{T}(\lambda 2^{la}) \subseteq \mathcal{U}$, so by Lemma 3.2.3, $\alpha = \gamma \times \delta$. We consider two cases:

$\alpha = 0$: Then either $\gamma = 0$, and so $\omega(\gamma.\mathbf{t}') = 0 \times \omega(\mathbf{t}') = 0$, or $\gamma \neq 0$ but $\delta = 0$, and so by Lemma 3.2.10, $\Gamma \vdash \mathbf{t}' : 0.U \equiv 0.A$, so by the induction hypothesis $\omega(\mathbf{t}') = 0$, and then $\omega(\gamma.\mathbf{t}') = \gamma \times 0 = 0$.

$\alpha \neq 0$: Then $A \equiv U$, so by Lemma 3.2.10, $\Gamma \vdash \mathbf{t}' : \delta.A$. Then by the induction hypothesis $\omega(\mathbf{t}') = \delta$. Notice that $\omega(\mathbf{t}\downarrow) = \gamma \times \omega(\mathbf{t}') = \gamma \times \delta = \alpha$.

□

Remark 3.4.4.

- *By Proposition 1.2.2, closed normal terms have form*

$$\sum_{i=1}^{n} \alpha_i.\lambda x.\mathbf{t}_i + \sum_{j=1}^{m} \lambda x.\mathbf{u}_j$$

  Thus the Theorem 3.4.3 entails that $\sum_{i=1}^{n} \alpha_i + m = 1$. *Hence the type system $\mathcal{B}$, an easy variation of the Scalar type system, checks for the barycentric property,* i.e. *it checks that a given term will reduce to a barycentric distribution of terms.*

- *It is easy to prove that*

$$z : A, w : A \vdash ((2.\lambda x.\lambda y.\frac{1}{4}.x + \frac{1}{4}.y)\ z)\ w : A.$$

  But notice that $\omega(((2.\lambda x.\lambda y.\frac{1}{4}.x+\frac{1}{4}.y)\ z)\ w) = 2$, *even when* $((2.\lambda x.\lambda y.\frac{1}{4}.x+\frac{1}{4}.y)\ z)\ w \to^* \frac{1}{2}.z+\frac{1}{2}.w$, *whose weight is equal to one. So,* a priori *this ω function cannot tell us that this term will yield a barycentric term. However the fact that has type A in $\mathbb{T}(\lambda 2^{la})$, according to the Theorem 3.4.3, anticipates this result.*

- *One might think that unit types $\mathcal{U}$ are just as good as barycentric types $\mathcal{B}$ for the sake of obtaining Theorem 3.4.3, via a combination of Subject-reduction and the uniqueness of scalars property (*cf. *Theorem 3.2.18). This is only morally true, here is a counter-example:*

$$x : U \to 2.U,\ y : U \vdash (x)\ \frac{1}{2}.y : U \qquad \textit{but } \omega((x)\ \frac{1}{2}.y) = \frac{1}{2}$$

  But the more significant difference between $\mathcal{U}$ and $\mathcal{B}$ is one of composability: the application of a term of unit type to another is not necessarily of unit type; whereas barycentric types, on the other hand, are preserved under application. Thus terms in $\mathcal{B}$ are not only barycentric; they can also be viewed as barycentric-preserving functions.

3.5 Conclusion and open questions

In summary, we have defined a System F-like type system for an extension of λ_{lin}, a λ-calculus which allows making arbitrary linear combinations of λ-calculus terms $\alpha.\mathbf{t} + \beta.\mathbf{u}$. This *Scalar* type system is fine-grained in that it keeps track of the 'amount of a type', *i.e.* the type of terms contain a scalar which is the sum of the amplitudes of the terms which contribute to the type.
Our main technical contributions were:

- A proof of the subject reduction property of this *Scalar* type system (Theorem 3.2.1). This came out after having proven a set of lemmas related to the equivalence relation intrinsic to the types, and another set of lemmas explaining how the scalars within the types are related to the scalars within the terms. Once all of the important properties were known, we were able to use them to decompose and recompose any term before and after applying a reduction rule, so as to show that every reduction rule preserves the types.
- A proof of the strong normalisation property of this *Scalar* type system (Theorem 3.3.11). The technique used to prove the strong normalisation property was by proving that such property would hold for a simpler system, and then to show the correspondence between the two systems.
- A proof that under strong normalisation, most of the conditions upon the λ_{lin} reduction rules can be lifted (*e.g.* allowing the factorisation not only of closed normal terms but of any term) without jeopardising confluence, thereby simplifying the λ_{lin} language.
- A proof that the *Scalar* type system can be used to check that a term has the barycentric property, *i.e.* that the amplitudes of its normal form are summing to one.

Arguably a denotational semantics approach might have led to less syntactic proofs of the properties of the type system, sustained by the guiding intuition about an underlying mathematical space. On the other hand, the complexity of the proofs in this chapter is largely due to the large number of rules (16 rules plus associativity and commutativity of $+$). Moreover the issue of models of (Linear-)Algebraic λ-calculus is a challenging, active topic of current research. We know of the categorical model of simply typed λ_{lin} [Valiron, 2010], and the finiteness space model of simply typed Algebraic λ-calculus [Ehrhard, 2005, Tasson, 2009]. Moreover, even if both calculi simulate each other (*cf.* Chapter 2), it is not clear whether the translation applies at the level of models. Hence known models are intricate and tend not to cover the set of terms under consideration in this chapter. Notice also that since the models of untyped λ-calculus are uncountable, the models of (Linear-)Algebraic λ-calculus are likely to be vector space of uncountable dimension. These are fascinating, open questions.

Chapter 4

Introducing sums of types

Résumé du Chapitre

Nous définissons le fragment additif de λ_{lin}. Nous définissons également un système de types qui inclut les sommes de types comme un reflet de celles présentes dans les termes. Après avoir prouvé la propriété de préservation du type par réduction, nous étudions le rôle des sommes dans le calcul via l'interprétation de notre système dans le Système F avec des paires. Nous montrons que ce calcul peut être interprété comme le Système F avec un constructeur de paires associatives et commutatives, et distributives par rapport aux applications. La normalisation forte de notre système dérive de cette interpretation. ■

A variant of the algebraic λ-calculi consists on adding sums to the pure λ-calculus. This amounts to an algebraic λ-calculus where coefficients are natural number. The only difference is that some reductions may happen 'faster' with coefficients: *e.g.* if $\mathbf{t} \to \mathbf{t}'$, then $2.\mathbf{t} \to 2.\mathbf{t}'$, but $\mathbf{t}+\mathbf{t} \to \mathbf{t}'+\mathbf{t} \to \mathbf{t}'+\mathbf{t}'$. Notice that this contrasts with the purely non-deterministic setting [de'Liguoro and Piperno, 1995] and its 'choice' operator, where $1+1$ is equal to 1.

Most presentations of calculi associated with differential linear logic, *e.g.* [Ehrhard and Regnier, 2003, Pagani and Rocca, 2010, Pagani and Tranquilli, 2009], are carried out in the Boolean setting or without coefficients (or with the possibility of introducing coefficients only mentioned as an aside). Other examples, with slight differences, are Bouldol's parallel λ-calculus [Boudol, 1994], and other probabilistic extensions of calculi [Bournez and Hoyrup, 2003, Di Pierro, Hankin, and Wiklicky, 2005, Herescu and Palamidessi, 2000].

Therefore the λ-calculus with sums, but without scalars, becomes the most basic object of study. Hence it was natural to try to understand the behaviour of sums in the context of λ_{lin}. To this end we have isolated a fragment of λ_{lin} with sums only (without scalars). We call it λ^{add}. We have endowed it with an intuitive type system, *Additive*, and proved its subject reduction.

The first attempt was to interpret *Additive* in a fragment of the multiplicative exponential linear logic, however it appeared that the target fragment was not linear, in the sense of linear logic (exponentials appeared all over the types in the translation). In fact the linearity of λ_{lin} that allows distributing application over sums, does not seem to be the same linearity found in linear logic.

Consequently, we adapted the translation in order to interpret *Additive* in System F with pairs. The key idea is that the distribution of application over sums is performed during the translation, and then

β-reduction is made within the usual λ-calculus with pairs. The translation says that whenever a term has a type in *Additive*, its translation has a type in System F with pairs. It is also true that it is possible to translate back the translated terms, and obtain the same we started with. We relate the reductions in *Additive* with those in System F with pairs in order to use the strong normalisation of System F to ensure the same property in *Additive*. The confluence of the typed λ^{add} is a corollary of the local confluence and the strong normalisation.

Plan of the chapter. Section 4.1 defines λ^{add}, the additive fragment of λ_{lin} and its type system *Additive*. Section 4.2 proves the subject reduction property in this setting. Section 4.3 is the core of this chapter. First, Section 4.3.1 introduces Add_{struct}, a simplification of *Additive*, which does not consider AC-equivalences nor neutrality of 0 within types. It is shown to be nevertheless equivalent to *Additive*. Then, in Section 4.3.2 the System F with pairs is presented and in Sections 4.3.3 and 4.3.4, an interpretation of *Additive* in it is set up, using Add_{struct} as an intermediate step. Section 4.3.5 proves the strong normalisation and confluence of our system by taking advantage of its relation with System F with pairs. Section 4.4 concludes.

4.1 The *Additive* Type System for λ^{add}

The λ^{add} calculus, *cf.* Figure 4.1, is a purely additive fragment of λ_{lin}, *cf.* Figure 1.1. The scalars have been removed from the grammar and the rewriting rules involving have been removed also. In addition to β-reduction, there remain five rules, specifying the behaviour of $+$: application distributes over it, and $\mathbf{0}$ is absorbing with respect to application and neutral with respect of sums.

Terms:	$\mathbf{t}, \mathbf{r}, \mathbf{u} ::= \mathbf{b} \mid (\mathbf{t})\ \mathbf{r} \mid \mathbf{0} \mid \mathbf{t} + \mathbf{r}$
Basis terms:	$\mathbf{b} ::= x \mid \lambda x.\mathbf{t}$

Distributivity rules:	*Zero rules:*	*β-reduction:*
$(\mathbf{u} + \mathbf{t})\ \mathbf{r} \to (\mathbf{u})\ \mathbf{r} + (\mathbf{t})\ \mathbf{r}$	$(\mathbf{0})\ \mathbf{t} \to \mathbf{0}$	$(\lambda x.\mathbf{t})\ \mathbf{b} \to \mathbf{t}[\mathbf{b}/x]$
$(\mathbf{r})\ (\mathbf{u} + \mathbf{t}) \to (\mathbf{r})\ \mathbf{u} + (\mathbf{r})\ \mathbf{t}$	$(\mathbf{t})\ \mathbf{0} \to \mathbf{0}$	
	$\mathbf{t} + \mathbf{0} \to \mathbf{t}$	

Figure 4.1: Syntax and reduction rules of λ^{add}

The grammar of types, *cf.* Figure 4.2, is analogous to the grammar of *Scalar cf.* Figure 3.1. We slightly simplify this scheme and do not allow $\forall X.T$ for general T but only for unit types. Notice that in *Scalar* it is morally the case, since there is the equivalence $\forall X.\alpha.T \equiv \alpha.\forall X.T$. The analogous equivalence in *Additive* would be $\forall X.(T + R) \equiv \forall X.T + \forall X.R$ which would be unusual, since it would allow each $\forall X$ to be replaced by different types. Instead we choose to restrict the universal quantifier to unit types.

We define an equivalence relation $\equiv$ on types as follows

Definition 4.1.1. *For any types T, R and S, we define the* type equivalence $\equiv$ *to be the least congruence such that:*

- $T + R \equiv R + T$
- $T + (R + S) \equiv (T + R) + S$
- $T + \overline{0} \equiv T$

Within this equivalence, it makes sense to use the following notation:

$$\sum_{i=1}^{0} T = \overline{0} \quad ; \quad \sum_{i=1}^{\alpha} T_i = \sum_{i=1}^{\alpha-1} T_i + T_\alpha \quad \text{if } \alpha \geq 1$$

Notice that it could be also possible to use the notation $\sum_{i=1}^{\alpha} T = \alpha.T$, when all the types T are the same, however it may induce some confusion with *Scalar*, so we chose to avoid such a notation in this Chapter.

Remark 4.1.2. *Notice that every type is equivalent to a sum of unit types.*

Types: $T, R, S ::= U \mid T + R \mid \overline{0}$

Unit types: $U, V, W ::= X \mid U \to T \mid \forall X.U$

$$\frac{}{\Gamma, x:U \vdash x:U}\, ax \qquad \frac{}{\Gamma \vdash \mathbf{0}:\overline{0}}\, ax_{\overline{0}} \qquad \frac{\Gamma \vdash \mathbf{t}:\sum_{i=1}^{\alpha}(U \to T_i) \qquad \Gamma \vdash \mathbf{r}:\sum_{j=1}^{\beta} U}{\Gamma \vdash (\mathbf{t})\ \mathbf{r}:\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i}\, \to_E$$

$$\frac{\Gamma, x:U \vdash \mathbf{t}:T}{\Gamma \vdash \lambda x.\mathbf{t}:U \to T}\, \to_I \qquad \frac{\Gamma \vdash \mathbf{t}:\forall X.U}{\Gamma \vdash \mathbf{t}:U[V/X]}\, \forall_E$$

$$\frac{\Gamma \vdash \mathbf{t}:T \qquad \Gamma \vdash \mathbf{r}:R}{\Gamma \vdash \mathbf{t}+\mathbf{r}:T+R}\, +_I \qquad \frac{\Gamma \vdash \mathbf{t}:T \qquad T \equiv R}{\Gamma \vdash \mathbf{t}:R}\, \equiv \qquad \frac{\Gamma \vdash \mathbf{t}:U \qquad X \notin FV(\Gamma)}{\Gamma \vdash \mathbf{t}:\forall X.U}\, \forall_I$$

Figure 4.2: Types and typing rules of *Additive*

Typing rules are also given in Figure 4.2. Rules for the universal quantifier, axiom and introduction of arrow are the usual ones. Any sum of typable terms can be typed using rule $+_I$. Notice that there is no elimination rule for $+$. An arrow elimination in this setting may seen complex: a direct analogous to the arrow elimination from *Scalar*, *cf.* Figure 3.1, would be of the form

$$\frac{\Gamma \vdash \mathbf{t}:\sum_{i=1}^{\alpha} U \to T \qquad \Gamma \vdash \mathbf{r}:\sum_{j=1}^{\beta} U}{\Gamma \vdash (\mathbf{t})\ \mathbf{r}:\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T}$$

where the scalars α and β in the $\to_E$ rule of *Scalar* have been replaced by $\sum_{i=1}^{\alpha}$ and $\sum_{j=1}^{\beta}$. However, notice that this restricts the calculus, since all the $U \to T$ that are summed have to be the same, and the same happens with the U, so the term $(\mathbf{t}_1 + \mathbf{t}_2)\ (\mathbf{r}_1 + \mathbf{r}_2)$ would be forced to have $\mathbf{t}_1$ and $\mathbf{t}_2$ with the same type, and also $\mathbf{r}_1$ and $\mathbf{r}_2$ with the same type.

One way to gradually relax this restriction is to allow having different T's, obtaining to the rule

$$\frac{\Gamma \vdash \mathbf{t}:\sum_{i=1}^{\alpha} U \to T_i \qquad \Gamma \vdash \mathbf{r}:\sum_{j=1}^{\beta} U}{\Gamma \vdash (\mathbf{t})\ \mathbf{r}:\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i}\, \to_E$$

Continuing the example, this allows $\mathbf{t}_1$ and $\mathbf{t}_2$ to have different types, provided that they are arrows starting with the same type U.

Example 4.1.3. *Let* $\Gamma \vdash \mathbf{b}_1 : U$, $\Gamma \vdash \mathbf{b}_2 : U$, $\Gamma \vdash \lambda x.\mathbf{t} : U \to T$ *and* $\Gamma \vdash \lambda y.\mathbf{r} : U \to R$. *Then*

$$\frac{\Gamma \vdash \lambda x.\mathbf{t} + \lambda y.\mathbf{r} : (U \to T) + (U \to R) \quad \Gamma \vdash \mathbf{b}_1 + \mathbf{b}_2 : U + U}{\Gamma \vdash (\lambda x.\mathbf{t} + \lambda y.\mathbf{r})\ (\mathbf{b}_1 + \mathbf{b}_2) : T + T + R + R} \to_E$$

Notice that $(\lambda x.\mathbf{t} + \lambda y.\mathbf{r})\ (\mathbf{b}_1 + \mathbf{b}_2) \to^* \underbrace{(\lambda x.\mathbf{t})\ \mathbf{b}_1}_{T} + \underbrace{(\lambda x.\mathbf{t})\ \mathbf{b}_2}_{T} + \underbrace{(\lambda y.\mathbf{r})\ \mathbf{b}_1}_{R} + \underbrace{(\lambda y.\mathbf{r})\ \mathbf{b}_2}_{R}$

On the contrary, allowing different U's is not so straightforward. On account of the distributivity rules it is required that all the arrows in the first addend start by a type which has to be the type of all the addends in the second term. For example, if the given term is $(\mathbf{t} + \mathbf{r})\ (\mathbf{b}_1 + \mathbf{b}_2)$, the terms $\mathbf{t}$ and $\mathbf{r}$ have to be able to receive the both $\mathbf{b}_1$ and $\mathbf{b}_2$ as arguments. This could be done by taking advantage of the polymorphism, however the arrow-elimination rule would become much more complex since it will have to do both, an arrow-elimination and a forall-elimination at the same time. Instead, we delay this (needed) choice to the *Vectorial* type system in Chapter 5, and here we just conserve this restricted version, which is enough for the aims of the present Chapter.

4.2 Subject reduction

The *Additive* type system is consistent, in the sense that typing is preserved by reduction. In this section, we prove the *subject reduction* property. We adapt the proof of Theorem 3.2.1, the subject reduction of *Scalar*.

Theorem 4.2.1 (Subject Reduction)**.** *For any terms* $\mathbf{t}, \mathbf{t}'$, *any context* Γ *and any type* T, *if* $\mathbf{t} \to \mathbf{t}'$ *then* $\Gamma \vdash \mathbf{t} : T \Rightarrow \Gamma \vdash \mathbf{t}' : T$.

Analogously to what has been done in Section 3.2, this result requires some definitions and lemmas. We just enunciate them, all the omitted proofs can be found in Appendix C.

Definition 4.2.2 (Relation $\preceq$ on types)**.** *For any unit types* U_1, U_2, V *and any type variable* X,

- *write* $U_1 \prec U_2$ *if either* $U_2 \equiv \forall X.U_1$, *or* $U_1 \equiv \forall X.V$ *and* $U_2 \equiv V[W/X]$ *for some type* W.
- $\preceq$ *is the reflexive (in terms of the type equivalence) and transitive closure of* $\prec$.

Lemma 4.2.3 (Arrows comparison)**.** *For any types* T, R *and any unit types* U, V, *if* $V \to R \preceq U \to T$, *then there exist* $\vec{W}, \vec{X}$ *such that* $U \to T \equiv (V \to R)[\vec{W}/\vec{X}]$. □

As a pruned version of a subtyping system, we can prove the subsumption rule:

Lemma 4.2.4 ($\preceq$-subsumption)**.** *For any context* Γ, *any term* $\mathbf{t}$ *and any unit types* U, V *such that* $U \preceq V$ *and no free type variable in* U *occurs in* Γ, *if* $\Gamma \vdash \mathbf{t} : U$ *then* $\Gamma \vdash \mathbf{t} : V$. □

Proving subject reduction means proving that each reduction rule preserves the type. Thus three generation lemmas are required: two classical ones, for applications (Lemma 4.2.5) and for abstractions (Lemma 4.2.6 and its Corollary 4.2.9) and a new one for the sums (Lemma 4.2.7).

Lemma 4.2.5 (Generation lemma (app))**.** *For any* $\Gamma, T, \mathbf{t}$ *and* $\mathbf{u}$*, if* $\Gamma \vdash (\mathbf{t})\ \mathbf{u} : T$ *is derivable, then there are* $\alpha, \beta \in \mathbb{N}_0$*, and some types* U *(in* $\mathcal{U}$*),* $T_1, \ldots, T_\alpha$ *such that* $\Gamma \vdash \mathbf{t} : \sum_{i=1}^{\alpha}(U \to T_i)$ *and* $\Gamma \vdash \mathbf{u} : \sum_{j=1}^{\beta} U$ *with* $\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i \preceq T$ *and each derivation of the typing judgement for* $\mathbf{t}$ *and* $\mathbf{u}$ *smaller than the one for* $(\mathbf{t})\ \mathbf{u}$*.* □

Lemma 4.2.6 (Generation lemma (abs))**.** *For any* Γ*,* $\mathbf{t}$ *and* T*, if* $\Gamma \vdash \lambda x.\mathbf{t} : T$ *is derivable, then there exist a unit type* U *and a type* R *such that* $\Gamma, x : U \vdash \mathbf{t} : R$ *and* $U \to R \preceq T$*.* □

Lemma 4.2.7 (Generation lemma (sum))**.** *For any* Γ*,* T*,* $\mathbf{r}$ *and* $\mathbf{u}$*, if* $\Gamma \vdash \mathbf{r} + \mathbf{u} : T$*, then there are some types* R, S *such that* $\Gamma \vdash \mathbf{r} : R$ *and* $\Gamma \vdash \mathbf{u} : S$ *with* $R + S \equiv T$ *and the typing derivations for* $\mathbf{r}$ *and* $\mathbf{u}$ *smaller than the one for* $\mathbf{r} + \mathbf{u}$*.* □

The standard substitution lemma can also be stated in this context.

Lemma 4.2.8 (Substitution)**.** *For any* Γ*,* T*,* U*,* $\mathbf{b}$ *and* $\mathbf{t}$*,*

1. $\Gamma \vdash \mathbf{t} : T \Rightarrow \Gamma[\vec{U}/\vec{X}] \vdash \mathbf{t} : T[\vec{U}/\vec{X}]$.
2. $\{\ \Gamma, x : U \vdash \mathbf{t} : T$ *and* $\Gamma \vdash \mathbf{b} : U\ \} \Rightarrow \Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T$. □

Corollary 4.2.9 (of Lemma 4.2.6)**.** *For any* Γ*,* U*,* T *and* $\mathbf{t}$*, if* $\Gamma \vdash \lambda x.\mathbf{t} : U \to T$ *then* $\Gamma, x : U \vdash \mathbf{t} : T$*.*

Proof. Analogous to the proof of Corollary 3.2.14: Let $\Gamma \vdash \lambda x.\mathbf{t} : U \to T$. By Lemma 4.2.6, $\exists V, R$ such that $V \to R \preceq U \to T$ and $\Gamma, x : V \vdash \mathbf{t} : R$, then by Lemma 4.2.3, $\exists \vec{W}, \vec{X}$ such that $U \to T \equiv (V \to R)[\vec{W}/\vec{X}]$ and so by Lemma 4.2.8, $\Gamma[\vec{W}/\vec{X}], x : V[\vec{W}/\vec{X}] \vdash \mathbf{t} : R[\vec{W}/\vec{X}]$, *i.e.* $\Gamma[\vec{W}/\vec{X}], x : U \vdash \mathbf{t} : T$.

Notice that if $\Gamma[\vec{W}/\vec{X}] \equiv \Gamma$, then we have finished. In the other case, $\vec{X}$ appears free in Γ. Since $V \to R \preceq U \to T$ and $\Gamma \vdash \lambda x.\mathbf{t} : V \to R$, according to Lemma 4.2.4, $U \to T$ can be obtained from $V \to R$ as a type for $\lambda x.\mathbf{t}$; then we would need to use the rule $\forall_I$; thus $\vec{X}$ cannot appear free in Γ, which constitutes a contradiction. So, $\Gamma, x : U \vdash \mathbf{t} : T$. □

Using $ax_{\overline{0}}$ it is easy to see that $\mathbf{0}$ has type $\overline{0}$, and there is no way to change its type except by rule $\equiv$ (notice that $\forall$-rules only apply on unit types).

Lemma 4.2.10 (Typing $\mathbf{0}$)**.** *For any* Γ*, if* $\Gamma \vdash \mathbf{0} : T$ *then* $T \equiv \overline{0}$*.*

Proof. Trivial: there is only one way to derive a type for $\mathbf{0}$, and it is using the rule $ax_{\overline{0}}$, so the only possible type for $\mathbf{0}$ is $\overline{0}$. □

A base term can always be given a unit type.

Lemma 4.2.11 (Base terms in Unit)**.** *Let* $\mathbf{b}$ *be a base term, that is a variable or an abstraction. Then for any* Γ *and* T*,* $\Gamma \vdash \mathbf{b} : T$ *implies* $T \equiv U \in \mathcal{U}$*.*

Proof. Notice that $\forall$-rules only produce unit types. So if $\mathbf{b}$ is a variable, it has either a type given in the context, which must be unit, or it gets its type through $\forall$-rules. If $\mathbf{b}$ is an abstraction, it has either a type given by $\to_I$ rule, which is unit, or it gets its type through $\forall$-rules. □

The proof of subject reduction is done by induction on the derivation of $\mathbf{t} \to \mathbf{t}'$, similarly to the proof for *Scalar* given in Chapter 3. We omit this proof here and place it directly in Appendix C.7.

4.3 Logical Interpretation

In this section, we interpret the *Additive* type system into System F with pairs (System F_P). Superposition will be interpreted as pairs. Since this product is neither associative nor commutative in System F_P, we consider at first that the sum operator in *Additive* also does not have these properties. This involves a slightly modified type system, that we call Add_{struct}. We then prove that this modified type system is equivalent to the original one (Proposition 4.3.7), and then translate every term of Add_{struct} into a term of System F_P. Finally, we show that this translation is correct with respect to typing (Theorem 4.3.19) and reduction (Theorem 4.3.21), from where we can prove the strong normalisation property for *Additive* (Corollary 4.3.22).

4.3.1 Structured additive type system

The system Add_{struct} is defined with the same grammar of types as *Additive*, and the same rules ax, $ax_{\overline{0}}$, $\to_I$, $+_I$, $\forall_I$ and $\forall_E$. There is no type equivalence, and thereby no commutativity nor associativity for sums, (also $\overline{0}$ is not neutral for sums). Hence rule $\to_E$, has to be precised. To specify what an n-ary sum is, we introduce a structure of trees for types.

Definition 4.3.1. *A* tree *is given by the following grammar:*

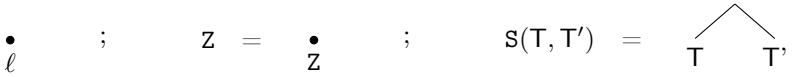
$$\mathsf{T}, \mathsf{T}' := \ell \mid \mathtt{Z} \mid \mathtt{S}(\mathsf{T}, \mathsf{T}')$$

Intuitively, a label ℓ represents a unit type, $\mathtt{Z}$ stands for $\overline{0}$, and $\mathtt{S}$ for the sum. A leaf *is given by a finite word over $\{\mathtt{l}, \mathtt{r}\}$ (that describes the path from the root to the leaf, with* `left` *or* `right` *steps). We call* labelling *a function from leaves to unit types. Given a tree T and a labelling s, the type $\mathsf{T}[s]$ is defined inductively by:*

$$\begin{aligned} \ell[s] &= s(\varepsilon) \qquad (\varepsilon \text{ denotes the empty word}) \\ \mathtt{Z}[s] &= \overline{0} \\ \mathtt{S}(\mathsf{T}, \mathsf{T}')[s] &= \mathsf{T}[w \mapsto s(\mathtt{l}w)] + \mathsf{T}'[w \mapsto s(\mathtt{r}w)] \end{aligned}$$

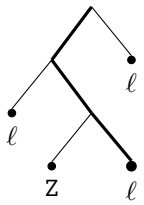
where $w \mapsto s(\mathtt{l}w)$ is the function that maps the word w to the result of s evaluated in $\mathtt{l}w$.
This definition is naturally extended to functions s from leaves to types.

Graphical representation. We may use the usual graphical representation of trees:

ℓ =  ; $\mathtt{Z}$ = ; $\mathtt{S}(\mathsf{T}, \mathsf{T}')$ = 

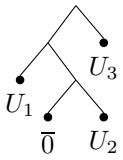
Example 4.3.2. *Within this representation, the tree on the right is*

$$\mathsf{T} = \mathtt{S}(\ \mathtt{S}(\ell,\ \mathtt{S}(\mathtt{Z},\ell)\ ),\ell\ ).$$



The third leaf (at the end of the bold path) corresponds to $\mathtt{lrr}$*. Writing* s *to the labelling* $\{\mathtt{ll} \mapsto U_1, \mathtt{lrr} \mapsto U_2, \mathtt{r} \mapsto U_3\}$*, we get*

$$\mathsf{T}[s] = (U_1 + (\overline{0} + U_2)) + U_3.$$

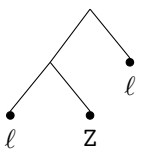
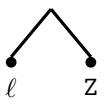
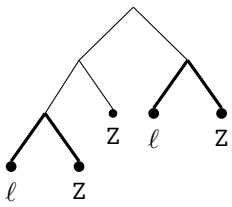


By extending the graphical notation, this type might be represented by the labelled tree on the left.

Conversely, for any type T, there exists a tree T_T and a unique labelling s_T such that $T = \mathsf{T}_T[s_T]$. Tree composition is defined as follows:

$$\begin{aligned}
\ell \circ \mathsf{T} &= \mathsf{T} \\
\mathtt{Z} \circ \mathsf{T} &= \mathtt{Z} \\
\mathsf{T} \circ \mathtt{Z} &= \mathtt{Z} \\
\mathtt{S}(\mathsf{T}_1, \mathsf{T}_2) \circ \mathsf{T} &= \mathtt{S}(\mathsf{T}_1 \circ \mathsf{T}, \mathsf{T}_2 \circ \mathsf{T}) \qquad \text{(if } \mathsf{T} \neq \mathtt{Z}\text{)}
\end{aligned}$$

Intuitively, composing $\mathsf{T}$with T' consists of "branching" T' to each leaf of T, and when this leaf is $\overline{0}$, it absorbs the tree.

Example 4.3.3. *Let* $\mathsf{T}=$  *and* $\mathsf{T}'=$  . *Then* $\mathsf{T} \circ \mathsf{T}'=$ 

An immediate induction (*cf.* Appendix C.8) ensures that

Lemma 4.3.4. $\mathsf{T}[w \mapsto \mathsf{T}'[s]] \equiv \mathsf{T} \circ \mathsf{T}'[wv \mapsto s(v)]$ *where* w *denotes a* ℓ*-leaf of* T*, and* v *a* ℓ*-leaf of* T'*.* □

Now we can give the rule for the arrow elimination in Add_{struct}:

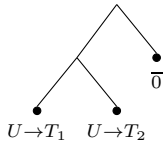
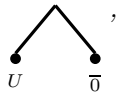
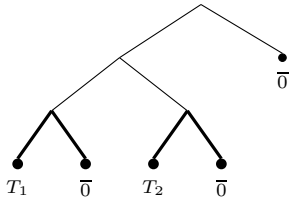
$$\frac{\Gamma \vdash \mathbf{t} : \mathsf{T}[w \mapsto (U \to T_w)] \qquad \Gamma \vdash \mathbf{u} : \mathsf{T}'[v \mapsto U]}{\Gamma \vdash (\mathbf{t})\ \mathbf{u} : \mathsf{T} \circ \mathsf{T}'[wv \mapsto T_w]} \to_{E'}$$

Where wv is a word whose prefix w represents a leaf of T.

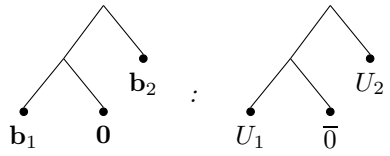
Example 4.3.5. *The following derivation is correct:*

$$\frac{\Gamma \vdash \mathbf{t} : \big((U \to T_1) + (U \to T_2)\big) + \overline{0} \qquad \Gamma \vdash \mathbf{u} : U + \overline{0}}{\Gamma \vdash (\mathbf{t})\ \mathbf{u} : \big((T_1 + \overline{0}) + (T_2 + \overline{0})\big) + \overline{0}} \to_{E'}$$

Graphically, we can represent this rule as follows:

if $\mathbf{t}$ *has type*  *and* $\mathbf{u}$ *has type*  , *then* $(\mathbf{t})\ \mathbf{u}$ *has type* 

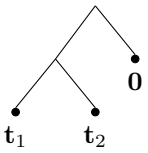
Remark 4.3.6. *The following informal remark can be made in anticipation of definition 4.3.8: we could also use labellings that map leaves to base terms, and represent any (closed normal) term by a labelled tree. Then we could type a term* $\mathsf{T}[w_i \mapsto \mathbf{b}_i]$ *with* $\mathsf{T}[w_i \mapsto U_i]$ *if every* $\mathbf{b}_i$ *has type* U_i*. For example,* $\vdash (\mathbf{b}_1 + \mathbf{0}) + \mathbf{b}_2 : (U_1 + \overline{0}) + U_2$ *would be represented as*



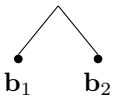
Conversely, if a term has type $\mathsf{T}[w_i \mapsto U_i]$*, it intuitively means that it will reduce on a term* $\mathsf{T}[w_i \mapsto \mathbf{b}_i]$*, with every* $\mathbf{b}_i$ *of type* U_i*. Concerning application, we actually have: if* $\mathbf{t} \to^* \mathsf{T}[w_i \mapsto \mathbf{t}_i]$ *and* $\mathbf{u} \to^* \mathsf{T}[v_j \mapsto \mathbf{u}_j]$*, then* $(\mathbf{t})\ \mathbf{u} \to^* \mathsf{T} \circ \mathsf{T}'[w_i v_j \mapsto (\mathbf{t}_i)\ \mathbf{u}_j]$*. As instance, using the same terms as in Example 4.3.5, we can see that* $\mathbf{t} \to^* (\mathbf{t}_1 + \mathbf{t}_2) + \mathbf{0}$ *(with* $\mathbf{t}_i$ *of type* $U \to T_i$*) and* $\mathbf{u} \to^* \mathbf{b}_1 + \mathbf{b}_2$ *(with* $\mathbf{b}_j$ *of type* U*) imply*

$$\begin{aligned}(\mathbf{t})\ \mathbf{u} &\to^* \big((\mathbf{t}_1)\ (\mathbf{b}_1 + \mathbf{b}_2)\ +\ (\mathbf{t}_2)\ (\mathbf{b}_1 + \mathbf{b}_2)\big)\ +\ (\mathbf{0})\ (\mathbf{b}_1 + \mathbf{b}_2)\\ &\to^* \big(((\mathbf{t}_1)\ \mathbf{b}_1 + (\mathbf{t}_1)\ \mathbf{b}_2)\ +\ ((\mathbf{t}_2)\ \mathbf{b}_1 + (\mathbf{t}_2)\ \mathbf{b}_2)\big)\ +\ \mathbf{0}\end{aligned}$$

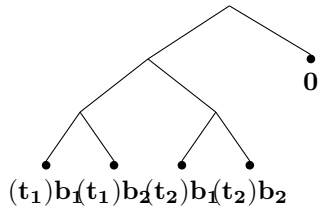
with $(\mathbf{t}_i)\ \mathbf{u}_j$ *of type* T_i*. In terms of trees, they remain the same as in Example 4.3.5:*
if $\mathbf{t}$ *reduces on*



and $\mathbf{u}$ *on*



, *then* $(\mathbf{t})\ \mathbf{u}$ *reduces on*



The *Additive* type system can be seen to be the same as the Add_{struct} type system up to associativity and commutativity and neutrality of $\overline{0}$ in the sense of the Proposition 4.3.7.

Proposition 4.3.7 (*Additive* equivalent to Add_{struct})**.**

1. *If* $\Gamma \vdash \mathbf{t} : T$ *is derivable in Additive, then there is a type* $T' \equiv T$ *such that* $\Gamma \vdash \mathbf{t} : T'$ *is derivable in* Add_{struct}*.*

2. *If* $\Gamma \vdash \mathbf{t} : T$ *is derivable in* Add_{struct}*, then it is also derivable in Additive.*

Proof.

1. We proceed by induction on the depth of the derivation of $\Gamma \vdash \mathbf{t} : T$ in *Additive*.

   - If the last rule that is applied is ax, $ax_{\overline{0}}$, $\to_I$, $+_I$, $\forall_I$ or $\forall_E$, then we can derive the same judgement in Add_{struct} using the induction hypothesis and the rule with the same name.
   - If the last rule that is used is $\equiv$, then we conclude directly using the induction hypothesis and transitivity of $\equiv$.
   - Assume the typing derivation ends with $\dfrac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{\alpha}(U \to T_i) \quad \Gamma \vdash \mathbf{u} : \sum_{j=1}^{\beta} U}{\Gamma \vdash (\mathbf{t})\ \mathbf{u} : \sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i} \to_E$.
     By induction hypothesis, there exist $T \equiv \sum_{i=1}^{\alpha}(U \to T_i)$ and $R \equiv \sum_{j=1}^{\beta} U$ such that $\Gamma \vdash \mathbf{t} : T$ and $\Gamma \vdash \mathbf{u} : R$ are both derivable in Add_{struct}. So we have $T = \mathsf{T}[w_i \mapsto (U \to T_i)]$ (where each w_i is the path to some ℓ-leaf of T) and $R = \mathsf{T}'[v \mapsto U]$ for some trees T, T'. Then we can derive in Add_{struct}
     $$\frac{\Gamma \vdash \mathbf{t} : \mathsf{T}[w_i \mapsto (U \to T_i)] \qquad \Gamma \vdash \mathbf{u} : \mathsf{T}'[v \mapsto U]}{\Gamma \vdash (\mathbf{t})\ \mathbf{u} : \mathsf{T} \circ \mathsf{T}'[w_i v \mapsto T_i]} \to_{E'}$$

By Lemma 4.3.4, $\mathsf{T}\circ\mathsf{T}'[w_i v \mapsto T_i] \equiv \mathsf{T}[w_i \mapsto \mathsf{T}'[v \mapsto T_i]]$. Then $\mathsf{T}\circ\mathsf{T}'[w_i v \mapsto T_i] \equiv \sum_{i=1}^{\alpha} \mathsf{T}'[v \mapsto T_i] \equiv \sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i$.

2. This side is immediate since every derivation rule of Add_{struct} can be seen as a derivation rule of *Additive*, except $\rightarrow_{E'}$ that is equivalent to the rules $\rightarrow_E$ and $\equiv$.

□

4.3.2 System F with pairs

We recall the definition of System F_P [Di Cosmo, 1995] (Figure 4.3). It will be used to interpret Add_{struct}. This system satisfies both the subject reduction and the strong normalisation properties [Di Cosmo, 1995]. We write Λ_F for the set of terms of System F_P, and we use the notation $\pi_{i_1\ldots i_n}(t)$ for $\pi_{i_1}(\pi_{i_2}(\ldots\pi_{i_n}(t)))$.

Terms : $t, u := \quad x \mid \lambda x.t \mid tu \mid \star \mid \langle t, u\rangle \mid \pi_1(t) \mid \pi_2(t)$

Types : $A, B := \quad X \mid A \Rightarrow B \mid \forall X.A \mid \mathbf{1} \mid A\times B$

Reduction rules : $(\lambda x.t)u \rightarrow t[u/x] \quad ; \quad \pi_i(\langle t_1, t_2\rangle) \rightarrow t_i$

η **– equivalence** : $\lambda x.tx \equiv_\eta t \quad if\ x\notin FV(t) \quad ; \quad \langle \pi_1(p), \pi_2(p)\rangle \equiv_\eta p$

Typing rules :

$$\frac{}{\Delta, x : A \vdash_F x : A}\,Ax \qquad \frac{}{\Delta \vdash_F \star : \mathbf{1}}\,\mathbf{1} \qquad \frac{\Delta, x : A \vdash_F t : B}{\Delta \vdash_F \lambda x.t : A \Rightarrow B}\Rightarrow I \qquad \frac{\Delta \vdash_F t : A \Rightarrow B \qquad \Delta \vdash_F u : A}{\Delta \vdash_F tu : B}\Rightarrow E$$

$$\frac{\Delta \vdash_F t : A \qquad \Delta \vdash_F u : B}{\Delta \vdash_F \langle t, u\rangle : A\times B}\times I \qquad \frac{\Delta \vdash_F t : A\times B}{\Delta \vdash_F \pi_1(t) : A}\times E_\ell \qquad \frac{\Delta \vdash_F t : A\times B}{\Delta \vdash_F \pi_2(t) : B}\times E_r$$

$$\frac{\Delta \vdash_F t : A \qquad X \notin FV(\Delta)}{\Delta \vdash_F t : \forall X.A}\forall I \qquad \frac{\Delta \vdash_F t : \forall X.A}{\Delta \vdash_F t : A[B/X]}\forall E$$

Figure 4.3: System F with pairs

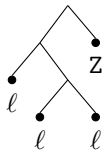
System F_P and trees. We have seen in the previous subsection that a tree can be labelled with unit types to form a type of Add_{struct}. It can also be labelled by terms or types of System F_P to form a new term in Λ_F or a new type of System F_P.

Definition 4.3.8. *We call* term-labelling *a function from leaves to Λ_F. Given τ, a term-labelling, and T, a tree, $\mathsf{T}[\tau]$ is the term of System F_P defined inductively by:*

$$\begin{aligned}\ell[\tau] &= \tau(\varepsilon)\\ \mathtt{Z}[\tau] &= \star\\ \mathtt{S}(\mathsf{T},\mathsf{T}')[\tau] &= \langle\, \mathsf{T}[w \mapsto \tau(\mathtt{l}w)]\,,\, \mathsf{T}'[w \mapsto \tau(\mathtt{r}w)]\,\rangle\end{aligned}$$

If $t = \mathsf{T}[\tau]$ and w is a ℓ-leaf of T, then $\tau(w)$ is a subterm of t that can be obtained by reducing $\pi_{\overline{w}}(t)$, where $\overline{w}$ is the mirror of word w where $\mathtt{l}$ is replaced by 1 and $\mathtt{r}$ by 2 (*i.e.* $\overline{\varepsilon} = \varepsilon$; $\overline{\mathtt{l}w} = \overline{w}1$; $\overline{\mathtt{r}w} = \overline{w}2$).

Example 4.3.9. *Let* $t = \langle\langle u_1, \langle u_2, u_3\rangle\rangle, \star\rangle$. *Then* $t = \mathsf{T}[\mathtt{ll} \mapsto u_1, \mathtt{lrl} \mapsto u_2, \mathtt{lrr} \mapsto u_3]$ *(where* T *is the tree on the right) and* u_3 *reduces from* $\pi_{221}(t)$.



Definition 4.3.10. *We call* F-labelling *a function from leaves to types of System* F_P. *Given* ϕ, *a F-labelling, and* T, *a tree, the type* $\mathsf{T}[\phi]$ *of System* F_P *is defined as expected:*

$$\begin{aligned} \ell[\phi] &= \phi(\varepsilon) \\ \mathtt{Z}[\phi] &= \mathbf{1} \\ \mathtt{S}(\mathsf{T}, \mathsf{T}')[\phi] &= \mathsf{T}[w \mapsto \phi(\mathtt{l}w)] \times \mathsf{T}'[w \mapsto \phi(\mathtt{r}w)] \end{aligned}$$

There is a trivial relation between the term-labelling of a tree, and its F-labelling, that we give in the following lemma.

Lemma 4.3.11. *Let* T *be a tree.*

1. *If* $\Gamma \vdash_F t_w : A_w$ *for each of its* ℓ*-leaves* w, *then* $\Gamma \vdash_F \mathsf{T}[w \mapsto t_w] : \mathsf{T}[w \mapsto A_w]$
2. *If* $\Gamma \vdash_F t : \mathsf{T}[w \mapsto A_w]$, *then for each* ℓ*-leaf* w *of* T, $\Gamma \vdash_F \pi_{\overline{w}}(t) : A_w$.

Proof. We prove both items by induction over T. Item 1 is immediate, we detail Item 2.

- If $\mathsf{T} = \mathtt{Z}$, it has no ℓ-leaf. If $\mathsf{T} = \ell$, its only ℓ-leaf is ε, and $\mathsf{T}[\varepsilon \mapsto A_\varepsilon] = A_\varepsilon$, and $\pi_{\overline{\varepsilon}}(t) = t$.
- If $\mathsf{T} = \mathtt{S}(\mathsf{T}_1, \mathsf{T}_2)$, then each ℓ-leaf of T is either $\mathtt{l}w'$ with w' a ℓ-leaf of T_1, or $\mathtt{r}w'$ with w' a ℓ-leaf of T_2. Moreover, $\mathsf{T}[w \mapsto A_w] = \mathsf{T}_1[w' \mapsto A_{\mathtt{l}w}] \times \mathsf{T}_2[w' \mapsto A_{\mathtt{r}w}]$, and by the induction hypothesis $\Gamma \vdash_F t' : \mathsf{T}_1[w' \mapsto A_{\mathtt{l}w'}]$ implies $\Gamma \vdash_F \pi_{\overline{w'}}(t) : A_{\mathtt{l}w'}$ (and same for T_2). If $w = \mathtt{l}w'$ is a ℓ-leaf of T, then $\pi_{\overline{w}}(t) = \pi_{\overline{w'}1}(t) = \pi_{\overline{w'}}(\pi_1(t))$. So $\Gamma \vdash_F t : \mathsf{T}[w \mapsto A_w]$ implies $\Gamma \vdash_F \pi_1(t) : \mathsf{T}_1[w' \mapsto A_{\mathtt{l}w'}]$, that implies $\Gamma \vdash_F \pi_{\overline{w}}(t) : A_{\mathtt{l}w'}$. Symmetrically, if $w = \mathtt{r}w'$ is a ℓ-leaf of T, we can also show that $\Gamma \vdash_F t : \mathsf{T}[w \mapsto A_w]$ implies $\Gamma \vdash_F \pi_{\overline{w}}(t) : A_{\mathtt{r}w'}$.

□

4.3.3 Translation from Add_{struct} to System F_P

The translation we propose here from Add_{struct} to System F_P consists of two steps:

- Every type T is interpreted by a formula $|T|$.
- Every term $\mathbf{t}$, typable by a typing derivation $\mathcal{D}$, is interpreted by a term $[\mathbf{t}]_{\mathcal{D}} \in \Lambda_F$ (which formally depends on $\mathcal{D}$).

Notice that only typable terms are translated into System F_P. Indeed, the translation of an application $(\mathbf{t})\ \mathbf{u}$ depends on the sum structure of $\mathbf{t}$ and $\mathbf{u}$, that is indicated by their types.

Notation. We may use $\mathcal{D}$ to denote typing derivations, however we may also abuse of this notation and identify a derivation only by its last sequent, so $\mathcal{D} = \Gamma \vdash \mathbf{t} : T$ is just a notation for a typing derivation named $\mathcal{D}$ which ends with the sequent $\Gamma \vdash \mathbf{t} : T$.

Interpretation of types. Types are translated in the intuitive way, given that sums are interpreted with Cartesian products:

$$|X| = X \qquad |\overline{0}| = \mathbf{1} \qquad |U \to T| = |U| \Rightarrow |T| \qquad |T + R| = |T| \times |R| \qquad |\forall X.U| = \forall X.|U|$$

We naturally extend this translation to contexts: $|\{x_1 : U_1, \ldots, x_k : U_k\}| = \{x_1 : |U_1| \ldots, x_k : |U_k|\}$.

It can be immediately checked that the tree structure of a type is preserved by translation, as expressed in the following lemma (the proof proceeds by induction on T, *cf.* Appendix C.9).

Lemma 4.3.12. *If $T = \mathsf{T}[w \mapsto U_w]$ is a type of Add_{struct}, then $|T| = \mathsf{T}[w \mapsto |U_w|]$.* □

Interpretation of typable terms. The interpretation of a typable term is defined in Figure 4.4, following the last step of its typing derivation.

ax	If $\mathcal{D} = \dfrac{}{\Gamma, x : T \vdash x : T}$, then $[x]_{\mathcal{D}} = x$	$ax_{\overline{0}}$	If $\mathcal{D} = \dfrac{}{\Gamma \vdash \mathbf{0} : \overline{0}}$, then $[\mathbf{0}]_{\mathcal{D}} = \star$
$+_I$	If $\mathcal{D} = \dfrac{\vdots \quad \mathcal{D}_1 \qquad \vdots \quad \mathcal{D}_2}{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{r} : R \;\; / \;\; \Gamma \vdash \mathbf{t} + \mathbf{r} : T + R}$, then $[\mathbf{t} + \mathbf{r}]_{\mathcal{D}} = \langle [\mathbf{t}]_{\mathcal{D}_1}, [\mathbf{r}]_{\mathcal{D}_2} \rangle$, where $\mathcal{D}_1$ and $\mathcal{D}_2$ denote the derivations in the premises.	$\to_I$	If $\mathcal{D} = \dfrac{\vdots \quad \mathcal{D}'}{\Gamma, x : U \vdash \mathbf{t} : T \;\; / \;\; \Gamma \vdash \lambda x.\mathbf{t} : U \to T}$, then $[\lambda x.\mathbf{t}]_{\mathcal{D}} = \lambda x.[\mathbf{t}]_{\mathcal{D}'}$, where $\mathcal{D}'$ denotes $\dfrac{\vdots}{\Gamma, x : U \vdash \mathbf{t} : T}$
$\to_{E'}$	If $\mathcal{D} = \dfrac{\vdots \quad \mathcal{D}_1 \qquad \vdots \quad \mathcal{D}_2}{\Gamma \vdash \mathbf{t} : \mathsf{T}[w \mapsto (U \to T_w)] \qquad \Gamma \vdash \mathbf{u} : \mathsf{T}'[v \mapsto U] \;\; / \;\; \Gamma \vdash (\mathbf{t})\ \mathbf{u} : \mathsf{T} \circ \mathsf{T}'[wv \mapsto T_w]}$, then $[(\mathbf{t})\ \mathbf{u}]_{\mathcal{D}} = \mathsf{T} \circ \mathsf{T}'[wv \mapsto \pi_{\overline{w}}([\mathbf{t}]_{\mathcal{D}_1})\pi_{\overline{v}}([\mathbf{u}]_{\mathcal{D}_2})]$, where $\mathcal{D}_1$ and $\mathcal{D}_2$ denote the derivations in the premises.		
$\forall_I$	If $\mathcal{D} = \dfrac{\vdots \quad \mathcal{D}'}{\Gamma \vdash \mathbf{t} : U \qquad X \notin FV(\Gamma) \;\; / \;\; \Gamma \vdash \mathbf{t} : \forall X.U}$, then $[\mathbf{t}]_{\mathcal{D}} = [\mathbf{t}]_{\mathcal{D}'}$, where $\mathcal{D}' = \dfrac{\vdots}{\Gamma \vdash \mathbf{t} : U}$.	$\forall_E$	If $\mathcal{D} = \dfrac{\vdots \quad \mathcal{D}'}{\Gamma \vdash \mathbf{t} : \forall X.U \;\; / \;\; \Gamma \vdash \mathbf{t} : U[V/X]}$, then $[\mathbf{t}]_{\mathcal{D}} = [\mathbf{t}]_{\mathcal{D}'}$, where $\mathcal{D}' = \dfrac{\vdots}{\Gamma \vdash \mathbf{t} : \forall X.U}$.

Figure 4.4: Translation from Add_{struct} to System F_P

This interpretation is in fact a direct translation of sums with pairs at each step of derivation, except for the arrow elimination that is translated by a more complex construction (Remark 4.3.6 may help to understand this).

Example 4.3.13. *Let*

$$\mathcal{D} = \dfrac{\Gamma \vdash \mathbf{t} : \big((U \to T_1) + (U \to T_2)\big) + \overline{0} \qquad \Gamma \vdash \mathbf{u} : U + (\overline{0} + U)}{\Gamma \vdash (\mathbf{t})\ \mathbf{u} : \big((T_1 + (\overline{0} + T_1)) + (T_2 + (\overline{0} + T_2))\big) + \overline{0}} \to_{E'}$$

with $t = [\mathbf{t}]_{\mathcal{D}_1}$ *and* $u = [\mathbf{u}]_{\mathcal{D}_2}$. *Write* $t_1 = \pi_{11}(t)$, $t_2 = \pi_{21}(t)$, $u_1 = \pi_1(u)$, $u_2 = \pi_{12}(u)$ *and* $u_3 = \pi_1(u)$. *Then*

$$[(\mathbf{t})\ \mathbf{u}]_{\mathcal{D}} = \langle\ \langle\, \langle t_1 u_1, \langle \star, t_1 u_3 \rangle \rangle\, ,\, \langle t_2 u_1, \langle \star, t_2 u_3 \rangle \rangle\ \rangle\, , \star \rangle$$

Intuitively, this translation of application consists of first distributing the application over sums and collapsing zeros, before translating subterms: $[\big((\mathbf{t}_1 + \mathbf{t}_2) + \mathbf{0}\big)\ \big(\mathbf{u}_1 + (\mathbf{0} + \mathbf{u}_3)\big)]_{\mathcal{D}}$ *is the translation of*

$$\Big(\big(\ (\mathbf{t_1})\ \mathbf{u_1} + (\mathbf{0} + (\mathbf{t_1})\ \mathbf{u_3})\ \big) + \big(\ (\mathbf{t_2})\ \mathbf{u_1} + (\mathbf{0} + (\mathbf{t_2})\ \mathbf{u_3})\ \big)\Big)\ +\ \mathbf{0}$$

Theorem 4.3.14 (Encoding of Add_{struct} in System F_P)**.** *If* $\Gamma \vdash \mathbf{t} : T$ *is derivable in* Add_{struct} *with derivation* $\mathcal{D}$, *then* $|\Gamma| \vdash_F [\mathbf{t}]_{\mathcal{D}} : |T|$ *is derivable in System* F_P.

Proof. We proceed by induction on the depth of the derivation $\mathcal{D}$. If it consists of rule ax or $ax_{\overline{0}}$, we use rule Ax or $\mathbf{1}$ respectively in System F_P. If the last rule of $\mathcal{D}$ is $+_I$ or $\to_I$ we can conclude by induction. If the last rule is $\forall_I$, we just need to note that $X \notin FV(\Gamma)$ implies $X \notin FV(|\Gamma|)$. If it is the rule $\forall_E$, we just have to note that $|U[V/X]| = |U|[|V|/X]$ to conclude with induction hypothesis. The only interesting case is when $\mathcal{D}$ ends with rule $\to_{E'}$:

$$\mathcal{D} = \dfrac{\Gamma \vdash \mathbf{t} : \mathsf{T}[w \mapsto (U \to T_w)] \qquad \Gamma \vdash \mathbf{u} : \mathsf{T}'[v \mapsto U]}{\Gamma \vdash (\mathbf{t})\ \mathbf{u} : \mathsf{T} \circ \mathsf{T}'[wv \mapsto T_w]} \to_{E'}$$

By induction hypothesis, $|\Gamma| \vdash_F [\mathbf{t}]_{\mathcal{D}_1} : |\mathsf{T}[w \mapsto (U \to T_w)]|$ and $|\Gamma| \vdash_F [\mathbf{u}]_{\mathcal{D}_2} : |\mathsf{T}'[v \mapsto U]|$. By Lemma 4.3.12, it means that $|\Gamma| \vdash_F [\mathbf{t}]_{\mathcal{D}_1} : \mathsf{T}[w \mapsto |U| \Rightarrow |T_w|]$ and $|\Gamma| \vdash_F [\mathbf{u}]_{\mathcal{D}_2} : \mathsf{T}'[v \mapsto |U|]$. By Lemma 4.3.11.2, for every ℓ-leaf w of T, and every ℓ-leaf v of T', we can derive

$$\dfrac{|\Gamma| \vdash_F \pi_{\overline{w}}([\mathbf{t}]_{\mathcal{D}_1}) : |U| \Rightarrow |T_w| \qquad |\Gamma| \vdash_F \pi_{\overline{v}}([\mathbf{u}]_{\mathcal{D}_2}) : |U|}{|\Gamma| \vdash_F \pi_{\overline{w}}([\mathbf{t}]_{\mathcal{D}_1})\pi_{\overline{v}}([\mathbf{u}]_{\mathcal{D}_2}) : |T_w|} \Rightarrow_E$$

Since $[(\mathbf{t})\ \mathbf{u}]_{\mathcal{D}} = \mathsf{T} \circ \mathsf{T}'[wv \mapsto \pi_{\overline{w}}([\mathbf{t}]_{\mathcal{D}_1})\ \pi_{\overline{v}}([\mathbf{u}]_{\mathcal{D}_2})]$, by Lemma 4.3.11.1 we can conclude $|\Gamma| \vdash_F [(\mathbf{t})\ \mathbf{u}]_{\mathcal{D}} : \mathsf{T} \circ \mathsf{T}'[wv \mapsto |T_w|]$, and then conclude using Lemma 4.3.12 again. □

Notice that this theorem is not enough: a trivial translation that sends every type in Add_{struct} to $\mathbf{1}$ and every term to $\star$ will also allow to prove such a theorem. To show that our translation is meaningful and not trivial, we can provide a partial encoding from System F_P to a sequent of Add_{struct}. Then we prove that one can go from Add_{struct} to System F_P and back, and obtain the same original sequent.

Definition 4.3.15 (Partial translation from System F_P to Add_{struct})**.** *Let us define the following type grammar:*

$$T, R, S := X \mid T \to R \mid \forall X.T \mid T + R \mid \overline{0}$$

The types of Add_{struct} *are included in such a grammar in the following sense: every type in* Add_{struct} *can be produced by this grammar, whereas the inverse is not true.*

The translation from System F_P *to* Add_{struct} *will have this grammar as target, and then we will ensure to use only those types that are also types of* Add_{struct} *(notice that, for example,* $(T + R) \to S$ *is in this grammar but it not a valid type in* Add_{struct}*). This translation is done in the intuitive way, given that Cartesian products are interpreted with sums:*

$$(\!|X|\!) = X \qquad (\!|\mathbf{1}|\!) = \overline{0} \qquad (\!|A \Rightarrow B|\!) = (\!|A|\!) \to (\!|B|\!) \qquad (\!|A \times B|\!) = (\!|A|\!) + (\!|B|\!) \qquad (\!|\forall X.A|\!) = \forall X.(\!|A|\!)$$

We naturally extend this translation to contexts: $|\{x_1 : A_1, \dots, x_k : A_k\}| = \{x_1 : (\!|A_1|\!) \dots, x_k : (\!|A_k|\!)\}$.

The terms are translated accordingly, however notice that this is a partial translation: no every term in System F_P *has an image through it. Let* t *and* u *not of the form* $\pi_i(t')$ *for any* t'*. Then the partial translation is unambiguously defined by*

$$\langle\!|x|\!\rangle = x \ ; \ \langle\!|\lambda x.t|\!\rangle = \lambda x.\langle\!|t|\!\rangle \ ; \ \langle\!|tu|\!\rangle = (\langle\!|t|\!\rangle)\ \langle\!|u|\!\rangle \ ; \ \langle\!|\star|\!\rangle = \mathbf{0} \ ; \ \langle\!|\langle t, u\rangle|\!\rangle = \langle\!|t|\!\rangle + \langle\!|u|\!\rangle$$

$$\langle\!|\mathsf{T}[wv \mapsto \pi_{\overline{w}}(t)\pi_{\overline{v}}(u)]|\!\rangle = (\langle\!|t|\!\rangle)\ \langle\!|u|\!\rangle \qquad \textit{with } \mathsf{T} \neq \mathsf{Z} \textit{ and } \mathsf{T} \neq \ell$$

Using this partial translation we can prove that the encoding from Add_{struct} into System F_P is not trivial in the following way.

Theorem 4.3.16. *If* $\Gamma \vdash \mathbf{t} : T$ *is derivable in* Add_{struct} *with derivation* $\mathcal{D}$*, then* $(\!|\Gamma|\!) \vdash \langle\!|[\mathbf{t}]_{\mathcal{D}}|\!\rangle : (\!|T|\!)$ *is syntactically the same sequent.*

Proof. A straightforward structural induction shows that for any type T in Add_{struct}, $T = (\!|T|\!)$. This also extends to contexts, since for any variable x and derivation $\mathcal{D}$, $\langle\!|[x]_{\mathcal{D}}|\!\rangle = \langle\!|x|\!\rangle = x$. So if $\mathbf{t} = \langle\!|[\mathbf{t}]_{\mathcal{D}}|\!\rangle$, one has $(\!|\Gamma|\!) \vdash \langle\!|[\mathbf{t}]_{\mathcal{D}}|\!\rangle : (\!|T|\!)$.

We need to prove that for any term $\mathbf{t}$ and derivation $\mathcal{D} = \Gamma \vdash \mathbf{t} : T$, $\mathbf{t} = \langle\!|(\!|\mathbf{t}|\!)|\!\rangle$. We proceed by induction on the depth of the derivation $\mathcal{D}$. We may abuse of the notation of $\mathcal{D}$ by taking only its last sequent.

1. $\mathcal{D} = \dfrac{}{\Gamma, x : U \vdash x : U}\, ax$ Then $\langle\!|[x]_{\mathcal{D}}|\!\rangle = \langle\!|x|\!\rangle = x$.

2. $\mathcal{D} = \dfrac{}{\Gamma \vdash \mathbf{0} : \overline{0}}\, ax_{\overline{0}}$ Then $\langle\!|[\mathbf{0}]_{\mathcal{D}}|\!\rangle = \langle\!|\star|\!\rangle = \mathbf{0}$.

3. $\mathcal{D} = \dfrac{\mathcal{D}_1 \qquad \mathcal{D}_2}{\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R}\, +_I$ where $\mathcal{D}_1 = \Gamma \vdash \mathbf{t} : T$, $\mathcal{D}_2 = \Gamma \vdash \mathbf{r} : R$
   By the induction hypothesis one has $\mathbf{t} = \langle\!|[\mathbf{t}]_{\mathcal{D}_1}|\!\rangle$ and $\mathbf{r} = \langle\!|[\mathbf{r}]_{\mathcal{D}_2}|\!\rangle$. Since $[\mathbf{t}]_{\mathcal{D}_1}$ has a translation by $\langle\!|\cdot|\!\rangle$, it is not of the form $\pi_i(t')$ for any t'. The same happens with $[\mathbf{r}]_{\mathcal{D}_2}$. Then $\langle\!|[\mathbf{t} + \mathbf{r}]_{\mathcal{D}}|\!\rangle = \langle\!|\langle[\mathbf{t}]_{\mathcal{D}_1}, [\mathbf{r}]_{\mathcal{D}_2}\rangle|\!\rangle = \langle\!|[\mathbf{t}]_{\mathcal{D}_1}|\!\rangle + \langle\!|[\mathbf{r}]_{\mathcal{D}_2}|\!\rangle = \mathbf{t} + \mathbf{r}$.

4. $\mathcal{D} = \dfrac{\mathcal{D}'}{\Gamma \vdash \lambda x.\mathbf{t} : U \to T}\, \to_I$ where $\mathcal{D}' = \Gamma, x : U \vdash \mathbf{t} : T$
   By the induction hypothesis $\mathbf{t} = \langle\!|[\mathbf{t}]_{\mathcal{D}'}|\!\rangle$, then $\lambda x.\mathbf{t} = \lambda x.\langle\!|[\mathbf{t}]_{\mathcal{D}'}|\!\rangle = \langle\!|\lambda x.[\mathbf{t}]_{\mathcal{D}'}|\!\rangle = \langle\!|[\lambda x.\mathbf{t}]_{\mathcal{D}}|\!\rangle$.

5. $\mathcal{D} = \dfrac{\mathcal{D}'}{\Gamma \vdash \mathbf{t} : \forall X.U}\, \forall_I$ where $\mathcal{D}' = \Gamma \vdash \mathbf{t} : U$
   By the induction hypothesis $\mathbf{t} = \langle\!|[\mathbf{t}]_{\mathcal{D}'}|\!\rangle = \langle\!|[\mathbf{t}]_{\mathcal{D}}|\!\rangle$.

6. $\mathcal{D} = \dfrac{\mathcal{D}'}{\Gamma \vdash \mathbf{t} : U[V/X]}\, \forall_E$ where $\mathcal{D}' = \Gamma \vdash \mathbf{t} : \forall X.U$
   By the induction hypothesis $\mathbf{t} = \langle\!|[\mathbf{t}]_{\mathcal{D}'}|\!\rangle = \langle\!|[\mathbf{t}]_{\mathcal{D}}|\!\rangle$.

7. $\mathcal{D} = \dfrac{\mathcal{D}_1 \qquad \mathcal{D}_2}{\Gamma \vdash (\mathbf{t})\ \mathbf{u} : \mathsf{T} \circ \mathsf{T}'[wv \mapsto T_w]}\, \to_{E'}$ where $\mathcal{D}_1 = \Gamma \vdash \mathbf{t} : \mathsf{T}[w \mapsto (U \to T_w)]$, $\mathcal{D}_2 = \Gamma \vdash \mathbf{u} : \mathsf{T}'[v \mapsto U]$
   By the induction hypothesis $\mathbf{t} = \langle\!|[\mathbf{t}]_{\mathcal{D}_1}|\!\rangle$ and $\mathbf{u} = \langle\!|[\mathbf{u}]_{\mathcal{D}_2}|\!\rangle$, then $(\mathbf{t})\ \mathbf{u} = (\langle\!|[\mathbf{t}]_{\mathcal{D}_1}|\!\rangle)\ \langle\!|[\mathbf{u}]_{\mathcal{D}_2}|\!\rangle = \langle\!|\mathsf{T} \circ \mathsf{T}'[wv \mapsto \pi_{\overline{w}}([\mathbf{t}]_{\mathcal{D}_1})\pi_{\overline{v}}([\mathbf{u}]_{\mathcal{D}_2})]|\!\rangle = \langle\!|[(\mathbf{t})\ \mathbf{u}]_{\mathcal{D}}|\!\rangle$.

□

4.3.4 Type equivalence

In this section we explain in which sense the type equivalence of *Additive* is provable in System F_P.

Definition 4.3.17 (equivalence in System F_P)**.** *In System F_P, we say that two formulae A and B are* equivalent *(notation $A \leftrightarrow B$) if there are two terms $\varepsilon_{A,B}$ and $\varepsilon_{B,A}$ such that $\vdash_F \varepsilon_{A,B} : A \Rightarrow B$, $\vdash_F \varepsilon_{B,A} : B \Rightarrow A$ and for any term t such that $\vdash_F t : A$, $\varepsilon_{B,A}(\varepsilon_{A,B}(t)) \to^* t$, analogously, if $\vdash_F t : B$, then $\varepsilon_{A,B}(\varepsilon_{B,A}(t)) \to^* t$.*
We define such terms for some specific types:

$$
\begin{array}{rrcl}
 & \varepsilon_{A,A} & = & \lambda x.x \\
 & \varepsilon_{A,A\times\mathbf{1}} & = & \lambda x.\langle x, \star\rangle \\
 & \varepsilon_{A\times\mathbf{1},A} & = & \lambda x.\pi_1(x) \\
 & \varepsilon_{A\times B,B\times A} & = & \lambda x.\langle \pi_2(x), \pi_1(x)\rangle \\
 & \varepsilon_{A\times(B\times C),(A\times B)\times C} & = & \lambda x.\langle\langle \pi_1(x), \pi_1(\pi_2(x))\rangle, \pi_2(\pi_2(x))\rangle \\
 & \varepsilon_{(A\times B)\times C,A\times(B\times C)} & = & \lambda x.\langle \pi_1(\pi_1(x)), \langle \pi_2(\pi_1(x)), \pi_2(x)\rangle\rangle \\
\text{if } B_1 \leftrightarrow B_2, & \varepsilon_{A\Rightarrow B_1,A\Rightarrow B_2} & = & \lambda f.\lambda x.\ \varepsilon_{B_1,B_2}(fx) \\
\text{if } B_1 \leftrightarrow B_2, & \varepsilon_{A\times B_1,A\times B_2} & = & \lambda p.\langle \pi_1(p), \varepsilon_{B_1,B_2}(\pi_2(p))\rangle
\end{array}
$$

Lemma 4.3.18. *For any types T, T', if $T \equiv T'$ then $|T| \leftrightarrow |T'|$*

Proof. Direct from the terms given in definition 4.3.17. □

This was the last step to prove the logical correctness of *Additive* type system.

Theorem 4.3.19. *If $\Gamma \vdash \mathbf{t} : T$ is derivable in Additive, then there is a type $T' \equiv T$ and a derivation $\mathcal{D}$ in Add_{struct} such that $|\Gamma| \vdash_F \varepsilon_{|T'|,|T|}[\mathbf{t}]_{\mathcal{D}} : |T|$.*

Proof. Assume $\Gamma \vdash \mathbf{t} : T$ is derivable in *Additive*. By Lemma 4.3.7, there is a type T' equivalent to T such that $\Gamma \vdash \mathbf{t} : T'$ is derivable in Add_{struct}. Calling $\mathcal{D}$ this derivation, this ensures that $|\Gamma| \vdash_F [t]_{\mathcal{D}} : |T'|$ (Theorem 4.3.14). Moreover, since $T \equiv T'$, by Lemma 4.3.18, $|T| \leftrightarrow |T'|$, so $|\Gamma| \vdash_F \varepsilon_{|T'||T|} : |T'| \Rightarrow |T|$. Then using rule $\Rightarrow_E$ one can conclude $|\Gamma| \vdash_F \varepsilon_{|T'||T|}[t]_{\mathcal{D}} : |T|$. □

Remark 4.3.20. *As noticed in the proof of the previous Theorem, if $\Gamma \vdash \mathbf{t} : T$, then there exists a type $T' \equiv T$ such that $\Gamma \vdash \mathbf{t} : T'$ is derivable in Add_{struct}. Then by Theorem 4.3.16, $(\!|\Gamma|\!) \vdash \langle\!|[\mathbf{t}]_{\mathcal{D}}|\!\rangle : (\!|T'|\!)$. Notice that $(\!|T'|\!) = T' \equiv T$, so by rule $\equiv$, $(\!|\Gamma|\!) \vdash \langle\!|[\mathbf{t}]_{\mathcal{D}}|\!\rangle : T$, which makes this translation non-trivial.*

4.3.5 Interpretation of reduction, strong normalisation and confluence

To some extent, the translation from Add_{struct} to System F_P is also correct with respect to reduction, as expressed in Theorem 4.3.21 (the proof is given in Appendix C.10).

Theorem 4.3.21. *Let $\Gamma \vdash \mathbf{t} : T$ be derivable (by $\mathcal{D}$) in Add_{struct}, and $\mathbf{t} \to \mathbf{r}$. If the reduction is not due to rule $\mathbf{t} + \mathbf{0} \to \mathbf{t}$, then there is $\mathcal{D}$' deriving $\Gamma \vdash \mathbf{r} : T$, and $[\mathbf{t}]_{\mathcal{D}} \to^* [\mathbf{r}]_{\mathcal{D}'}$.* □

Notice that the type equivalences, *i.e.* associativity and commutativity of types, have their analogous in the term equivalences, namely the associativity and commutativity properties of $+$. However, the equivalence $T + 0 \equiv T$ has its analogous with a reduction rule, $\mathbf{t} + \mathbf{0} \to \mathbf{t}$. Since Add_{struct} has no equivalences, this reduction rule is not correct in the translation. However, if $\Gamma \vdash \mathbf{t} + \mathbf{0} : T + \overline{0}$ is derivable by $\mathcal{D}$ in Add_{struct}, then there is some $\mathcal{D}' = \Gamma \vdash \mathbf{t} : T$ such that

$$\varepsilon_{|T+\overline{0}|,|T|}[\mathbf{t} + \mathbf{0}]_{\mathcal{D}} \to^* [\mathbf{t}]_{\mathcal{D}'}.$$

In the remaining of this section we write $\mathbf{t} \to_A \mathbf{r}$ to a one-step algebraic reduction, that is a reduction by any rule except the β-reduction. We write $\mathbf{t} \to_\beta \mathbf{r}$ to a one-step β-reduction.

Corollary 4.3.22 (Strong normalisation)**.** *If $\Gamma \vdash \mathbf{t} : T$ is derivable in Additive, then $\mathbf{t}$ is strongly normalising.*

Proof. First notice that if $\Gamma \vdash \mathbf{t} : T$ is derivable in *Additive*, then by Proposition 4.3.7, $\exists T' \equiv T$ such that $\Gamma \vdash \mathbf{t} : T'$ is derivable in Add_{struct}. Let us call this derivation $\mathcal{D}$. Then by Theorem 4.3.14, $|\Gamma| \vdash_F [\mathbf{t}]_{\mathcal{D}} : |T'|$ is derivable in System F_P.

Assume $\mathbf{t}$ is not strongly normalising, say $\mathbf{t} \to \mathbf{t}_1 \to \mathbf{t}_2 \to \cdots$. For a first approximation, consider that none of these reductions happens by rule $\mathbf{t} + \mathbf{0} \to \mathbf{t}$. Then by Theorem 4.3.21 there exists derivations $\mathcal{D}_1, \mathcal{D}_1, \ldots$, such that $[\mathbf{t}]_{\mathcal{D}} \to^* [\mathbf{t}_1]_{\mathcal{D}_1} \to^* [\mathbf{t}_2]_{\mathcal{D}_1} \to^* \cdots$. However, due to the strong normalisation of F_p, there exists a natural number n such that, $\forall i \geq n$, $[\mathbf{t}_i]_{\mathcal{D}_i} = [\mathbf{t}_{i+1}]_{\mathcal{D}_{i+1}}$.

We proceed in two steps:

1. We prove that, if $\mathbf{t}_i \to \mathbf{t}_{i+1}$ and $[\mathbf{t}_i]_{\mathcal{D}_i} = [\mathbf{t}_{i+1}]_{\mathcal{D}_{i+1}}$, then the reduction is an algebraic rule, (*i.e.* not a beta-reduction).

2. Then we show that algebraic rules are strictly decreasing with respect to the following measure [Arrighi and Dowek, 2008], which is always positive: $|\mathbf{0}| = 0$, $|x| = 1$, $|\lambda x.\mathbf{t}| = |\mathbf{t}|$, $|\mathbf{t} + \mathbf{r}| = 2 + |\mathbf{t}| + |\mathbf{r}|$, $|(\mathbf{t})\ \mathbf{r}| = (3|\mathbf{t}| + 2)(3|\mathbf{r}| + 2)$.

By item 1 only algebraic rules happen, which are strictly decreasing in the positive measure of item 2. Since $\mathbf{t} + \mathbf{0} \to \mathbf{t}$ is also strictly decreasing, then $\mathbf{t}$ has to be strongly normalising. The proofs of items 1 and 2 can be found in Appendix C.11. Item 1 proceeds by structural induction on $\mathbf{t}_i$ to show that if $\mathbf{t}_i \to_\beta \mathbf{t}_{i+1}$, then $[\mathbf{t}_i]_{\mathcal{D}_i} \neq [\mathbf{t}_{i+1}]_{\mathcal{D}_{i+1}}$. Item 2 follows from a case analysis on the possible reductions. □

The strong normalisation can be used to prove the confluence of the typed system.

Corollary 4.3.23 (Confluence)**.** *The typed language λ^{add} is confluent.*

Proof. The proof proceeds as follows:

1. First we prove the local confluence of the algebraic system. This has been done by modifying (simplifying) the Coq proof given to prove the local confluence of the algebraic fragment of λ_{lin} in Corollary 3.3.13. The modified Coq files, including the proper version of the library, can be downloaded from `http://membres-liglab.imag.fr/diazcaro/Additive.tar.bz2`.

2. Then prove the (local) commutation between algebraic rules and β-reduction. This proofs follows exactly as in the proof given in Corollary 3.3.13, taking only the valid cases.

3. The two previous steps plus the confluence of the β-reduction given in Corollary 3.3.13, give us the local confluence of λ^{add}.

Local confluence plus strong normalisation implies confluence [TeReSe, 2003]. □

4.4 Conclusions and open questions

We have treated the problem of interpreting (the role of) additions within algebraic calculi. Our main concept has been to make explicit the associativity and commutativity properties of sums and the neutral property of zero by defining a structured type system with sums. Moreover we have shown how to simulate sums by pairs by distributing the application while doing the translation, and collapsing all the zero applications.

In an earlier version of this research we attempted to translate *Additive* directly into the multiplicative exponential fragment of linear logic; however the choice of the multiplicative fragment, instead of the additive one, seemed arbitrary. That is to say, the target fragment did not deal with *linearity*, in the sense of linear logic. Thereby, the translation of λ^{add} in linear logic we got, can be decomposed into a first translation from λ^{add} into System F with pairs, as we have presented in this chapter, and then the standard translation from System F to exponential linear logic [Girard, 1987, Chapter 5]. This suggest that the linearity of λ^{add} is not directly related to the one of linear logic.

There still lacks an interpretation of the fragment with scalars of λ_{lin}. A first attempt was presented in Chapter 3; however there is no translation into a well-known theory. One interesting idea is to set up a type system for the whole calculus which reflects the scalars by sums in the types by taking its floor (considering only positive real numbers as scalars). This idea is presented latter in this thesis, *cf.* Chapter 6.

Chapter 5

A vectorial type system

Résumé du Chapitre

Ce chapitre fusionne les approches des chapitres 3 et 4, avec pour but précis celui de la caractérisation des vecteurs dans l'espace vectoriel des termes. Nous fournissons une version faible de la preuve de préservation de type et de la normalisation forte du langage typé. Nous montrons que, dans ce cadre du typage vectoriel à la Curry, la préservation du type n'est pas complète, ce qui justifie de passer à un typage à la Church dans les chapitres suivants. ■

The goal of this Chapter is to precisely explicit what is a vector in the space of terms of λ_{lin}. We want a characterisation of terms independent from the term reduction, highlighting the vectorial structure of terms. To that end, we propose a static analysis tool in the form of a type system. In Chapter 3 we presented *Scalar*. If α is a scalar and $\Gamma \vdash \mathbf{t} : T$ is a sequent, $\alpha.\mathbf{t}$ is of type $\alpha.T$. The developed language actually provides a static analysis tool for certain applications such as *barycentric* computation. It however fails to address a more general issue: let $\mathbb{B} = \forall X.X \to X \to X$, $\mathbf{true} = \lambda x.\lambda y.x$ and $\mathbf{false} = \lambda x.\lambda y.y$. Without sums but with negative numbers, the term $\mathbf{true} - \mathbf{false}$ is typed with $0.\mathbb{B}$, a type which fails to exhibits the fact that we have a superposition of terms: **true** and **false** are two base terms, *i.e.* they should be considered orthogonal since they both belongs to the same basis, and they are different. So better types for **true** and **false**, in the sense that they can be distinguished from each other, would be $\mathbb{T} = \forall X.\forall Y.X \to Y \to X$ and $\mathbb{F} = \forall X.\forall Y.X \to Y \to Y$ respectively. However in this case, $\mathbf{true} - \mathbf{false}$ would no have a type in *Scalar*. Chapter 4 is concerned with the addition of sums to a regular type system. The language considered, λ^{add} takes into account only the additive fragment of λ_{lin}, it leaves scalars out of the picture. In this case, if $\Gamma \vdash \mathbf{s} : S$ and $\Gamma \vdash \mathbf{t} : T$ are two valid sequents, $\mathbf{s} + \mathbf{t}$ is of type $S + T$. In addition it has another flaw: as already noticed in Section 4.1, $(\mathbf{t})\ (\mathbf{u}_1 + \mathbf{u}_2)$ requires $\mathbf{u}_1$ and $\mathbf{u}_2$ to have the same type.

This Chapter builds on these two approaches for the precise goal of characterising vectors in complex vector spaces. Because of the possible negative or complex coefficients, this requires to keep track of the 'direction' as well as the 'amplitude' of a term. We propose a type system with both sums and scalars, reflecting the vectorial structure of λ_{lin}. Interestingly enough, combining the two separate features of the type systems presented in Chapter 3 and 4 raises subtle novel issues. In the end we achieve a type system which is such that if $\mathbf{t}$ has type $\sum_i \alpha_i.U_i$, then it must reduce to a $\mathbf{t}'$ of the form $\sum_i \alpha_i.\mathbf{b}_i$, where the $\mathbf{b}_i$'s are basis terms.

We provide a proof of strong normalisation, entailing the confluence of the calculus, and a weak subject-reduction property. We show that in this vectorial setting in Curry style, this property is not complete, justifying moving to Church style in the following chapters.

Plan of the chapter. In Section 5.1 we show λ_{lin} again and how quantum computing and other vectorial computation can be encoded on it. In Section 5.2, we expose the type system and the problem arising from the possibility of having linear combinations of types. Section 5.3 is devoted to subject reduction. We first say why the usual result is not valid, then we provide a solution and a candidate subject reduction theorem; the rest of the section is concerned with the proof of the result. In Section 5.4, we prove confluence and strong normalisation for this setting. Finally we close the paper with some examples in Section 5.5 and conclusions in Section 5.6.

5.1 Non-restricted λ_{lin}

Figure 5.1 shows the language λ_{lin} first presented in Figure 1.1 with the following changes: restrictions (*) and (**) has been lifted, since it has been shown in Chapter 3 that if one considers a typed language enforcing strong normalisation, one can wave them and consider a more canonical set of rewrite rules. Working with a type system enforcing strong normalisation (as shown in Section 5.4), we follow this approach. The second change is in rule $\mathbf{t} + \mathbf{0} \to \mathbf{t}$, which is placed in the Factorisation group of rules.

Terms: $\mathbf{t}, \mathbf{r}, \mathbf{u} ::= \mathbf{b} \mid (\mathbf{t})\ \mathbf{r} \mid \mathbf{0} \mid \alpha.\mathbf{t} \mid \mathbf{t} + \mathbf{r}$

Base terms: $\mathbf{b} ::= x \mid \lambda x.\mathbf{t}$

Elementary rules:
$0.\mathbf{t} \to \mathbf{0}$,
$1.\mathbf{t} \to \mathbf{t}$,
$\alpha.\mathbf{0} \to \mathbf{0}$,
$\alpha.(\beta.\mathbf{t}) \to (\alpha \times \beta).\mathbf{t}$,
$\alpha.(\mathbf{t} + \mathbf{r}) \to \alpha.\mathbf{t} + \alpha.\mathbf{r}$.

Factorisation rules:
$\alpha.\mathbf{t} + \beta.\mathbf{t} \to (\alpha + \beta).\mathbf{t}$,
$\alpha.\mathbf{t} + \mathbf{t} \to (\alpha + 1).\mathbf{t}$,
$\mathbf{t} + \mathbf{t} \to (1 + 1).\mathbf{t}$,
$\mathbf{t} + \mathbf{0} \to \mathbf{t}$.

Beta reduction:
$(\lambda x.\mathbf{t})\ \mathbf{b} \to \mathbf{t}[\mathbf{b}/x]$.

Application rules:
$(\mathbf{t} + \mathbf{r})\ \mathbf{u} \to (\mathbf{t})\ \mathbf{u} + (\mathbf{r})\ \mathbf{u}$,
$(\mathbf{u})\ (\mathbf{t} + \mathbf{r}) \to (\mathbf{u})\ \mathbf{t} + (\mathbf{u})\ \mathbf{r}$,
$(\alpha.\mathbf{t})\ \mathbf{r} \to \alpha.(\mathbf{t})\ \mathbf{r}$,
$(\mathbf{r})\ (\alpha.\mathbf{t}) \to \alpha.(\mathbf{r})\ \mathbf{t}$,
$(\mathbf{0})\ \mathbf{t} \to \mathbf{0}$,
$(\mathbf{t})\ \mathbf{0} \to \mathbf{0}$.

Contextual rules: If $\mathbf{t} \to \mathbf{r}$, then for any term $\mathbf{u}$, scalar α and variable x,
$(\mathbf{t})\ \mathbf{u} \to (\mathbf{r})\ \mathbf{u}$, $(\mathbf{u})\ \mathbf{t} \to (\mathbf{u})\ \mathbf{r}$, $\mathbf{t} + \mathbf{u} \to \mathbf{r} + \mathbf{u}$, $\mathbf{u} + \mathbf{t} \to \mathbf{u} + \mathbf{r}$, $\alpha.\mathbf{t} \to \alpha.\mathbf{r}$ and $\lambda x.\mathbf{t} \to \lambda x.\mathbf{r}$.

Figure 5.1: Syntax and reduction rules of λ_{lin}, without restrictions

Encoding booleans

We claimed in Section 1.2.2 that this language was a candidate language for quantum computation. In this paragraph we backup again this claim, and show how quantum gates and matrices can be encoded. We slightly modify the encoding presented in Section 1.2.2 to better fit the current setting.

First, both in λ_{lin} and in quantum computation one can interpret the notion of booleans. In the former we can consider the usual booleans $\lambda x.\lambda y.x$ and $\lambda x.\lambda y.y$ whereas in the latter we consider the regular quantum bits $|0\rangle$ and $|1\rangle$.

In λ_{lin}, a representation of *if* $\mathbf{r}$ *then* $\mathbf{u}$ *else* $\mathbf{t}$ needs to take into account the special relation between sums and applications. We cannot directly encode this test as the usual $((\mathbf{r})\ \mathbf{u})\ \mathbf{t}$. Indeed, if $\mathbf{r}$, $\mathbf{u}$ and $\mathbf{t}$ were respectively the terms $\mathbf{true}$, $\mathbf{u}_1 + \mathbf{u}_2$ and $\mathbf{t}_1 + \mathbf{t}_2$, the term $((\mathbf{r})\ \mathbf{u})\ \mathbf{t}$ would reduce to $((\mathbf{true})\ \mathbf{u}_1)\ \mathbf{t}_1 + ((\mathbf{true})\ \mathbf{u}_1)\ \mathbf{t}_2 + ((\mathbf{true})\ \mathbf{u}_2)\ \mathbf{t}_1 + ((\mathbf{true})\ \mathbf{u}_2)\ \mathbf{t}_2$, then to $2.\mathbf{u}_1 + 2.\mathbf{u}_2$ instead of $\mathbf{u}_1 + \mathbf{u}_2$. We need to "freeze" the computations in each branch of the test so that the sum does not distribute over the application. For that purpose we use the well-known notion of *thunks*: we encode the test as $\{((\mathbf{r})\ [\mathbf{u}])\ [\mathbf{t}]\}$, where $[-]$ is the term $\lambda f.-$ with f a fresh, unused term variable and where $\{-\}$ is the term $(-)\lambda x.x$. The former "freezes" the computation while the latter "releases" it. Then the term *if* $\mathbf{true}$ *then* $(\mathbf{u}_1 + \mathbf{u}_2)$ *else* $(\mathbf{t}_1 + \mathbf{t}_2)$ reduces to the term $\mathbf{u}_1 + \mathbf{u}_2$ as one could expect. Note that this test is linear, in the sense that the term *if* $(\alpha.\mathbf{true} + \beta.\mathbf{false})$ *then* $\mathbf{u}$ *else* $\mathbf{t}$ reduces to $\alpha.\mathbf{u} + \beta.\mathbf{t}$.

This has a striking similarity with the *quantum test* that can be found in Section 1.2.2 but also in [van Tonder, 2004, Altenkirch and Grattage, 2005, Arrighi and Dowek, 2008]. For example, consider the Hadamard gate *had* sending $|0\rangle$ to $\frac{\sqrt{2}}{2}(|0\rangle + |1\rangle)$ and $|1\rangle$ to $\frac{\sqrt{2}}{2}(|0\rangle - |1\rangle)$ (*cf.* Section 1.1). If x is a quantum bit, the value $(had)x$ can be represented as the quantum test *if* x *then* $\frac{\sqrt{2}}{2}(|0\rangle + |1\rangle)$ *else* $\frac{\sqrt{2}}{2}(|0\rangle - |1\rangle)$. As hinted in Section 1.2.2, one can simulate this operation in λ_{lin} using the test construction we just described: $\{(had)\ x\} = \{((x)\ [\frac{\sqrt{2}}{2}.\mathbf{true} + \frac{\sqrt{2}}{2}.\mathbf{false}])\ [\frac{\sqrt{2}}{2}.\mathbf{true} - \frac{\sqrt{2}}{2}.\mathbf{false}]\}$. Note that the thunks are necessary: the term $((x)\ (\frac{\sqrt{2}}{2}.\mathbf{true} + \frac{\sqrt{2}}{2}.\mathbf{false}))\ (\frac{\sqrt{2}}{2}.\mathbf{true} - \frac{\sqrt{2}}{2}.\mathbf{false})$ would reduce to the term $\frac{1}{2}(((x)\ \mathbf{true})\ \mathbf{true} + ((x)\ \mathbf{true})\ \mathbf{false} + ((x)\ \mathbf{false})\ \mathbf{true} + ((x)\ \mathbf{false})\ \mathbf{false})$, which is fundamentally different from the term *had* we are trying to emulate.

Of course, with this procedure we can "encode" any matrix. If the space is of some general dimension n, instead of the basis elements $\mathbf{true}$ and $\mathbf{false}$ we can choose the terms $\lambda x_1.\cdots.\lambda x_n.x_i$'s for $i = 1$ to n to encode the basis of the space.

5.2 The *Vectorial* Type System

Let us justify each constructor in the proposed type system. Since we are considering a lambda-calculus, we need at least an arrow type $A \to B$. The terms $\mathbf{true}$ and $\mathbf{false}$ can therefore be typed in the usual way with $\mathbb{B} = X \to (X \to X)$, for a fixed type X. Since the sum $\frac{\sqrt{2}}{2}.\mathbf{true} + \frac{\sqrt{2}}{2}.\mathbf{false}$ is a superposition of terms of type $\mathbb{B}$, one could decide to also type it with the type $\mathbb{B}$; in general, a linear combination of terms of type A would be of type A. But then the terms $\lambda x.(1.x)$ and $\lambda x.(2.x)$ would both be of the same type $A \to A$, failing to address the fact that the former respect the norm whereas the latter does not.

To address this problem, we incorporate the notion of scalars in the type system: if A is a valid type, the construction $\alpha.A$ is also a valid type and if the terms $\mathbf{s}$ and $\mathbf{t}$ are of type A, the term $\alpha.\mathbf{s} + \beta.\mathbf{t}$ is of type $(\alpha + \beta).A$. This was achieved in Chapter 3 and it allows us to distinguish between the two functions $\lambda x.(1.x)$ and $\lambda x.(2.x)$: the former is of type $A \to A$ whereas the latter is of type $A \to (2.A)$.

Let us now consider the term $\frac{\sqrt{2}}{2}.(\mathbf{true} - \mathbf{false})$. Using the above extension to the type system, this term should be of type $0.\mathbb{B}$, which is odd in the light of the use we want to make of it. Indeed, applying the Hadamard gate to this term produces the term $\mathbf{false}$ of type $\mathbb{B}$: the "amplitude" of the type (the sum of the squares of the absolute values of the scalars) jumps from 0 to 1.

This time, the problem comes from the fact that the type system does not keep track of the "direction" of a term. We therefore propose to go one step further, and to allow sums in types, as presented in Chapter 4. Provided that $\mathbb{T} = X \to (Y \to X)$ and $\mathbb{F} = X \to (Y \to Y)$ (with Y another fixed type), we can type the term $\frac{\sqrt{2}}{2}.(\mathbf{true} - \mathbf{false})$ with $\frac{\sqrt{2}}{2}.(\mathbb{T} - \mathbb{F})$, which has "amplitude" 1, in the same way that the type of **false** has "amplitude" 1.

This type system is also able to type the term $had = \lambda x.((x)\ [\frac{\sqrt{2}}{2}.\mathbf{true} + \frac{\sqrt{2}}{2}.\mathbf{false}])\ [\frac{\sqrt{2}}{2}.\mathbf{true} - \frac{\sqrt{2}}{2}.\mathbf{false}]$, with $((\mathbf{I} \to \frac{\sqrt{2}}{2}.(\mathbb{T} + \mathbb{F})) \to (\mathbf{I} \to \frac{\sqrt{2}}{2}.(\mathbb{T} - \mathbb{F})) \to T) \to T$ provided that $\mathbf{I}$ is an identity type of the form $Z \to Z$, and for T and Z any fixed types.

Let us try to type the term $\{(had)\ \mathbf{true}\}$. This is possible provided that the fixed type T is equal to $\mathbf{I} \to \frac{\sqrt{2}}{2}.(\mathbb{T} + \mathbb{F})$. If we now want to type the term $\{(had)\ \mathbf{false}\}$, the fixed type T needs to be equal to $\mathbf{I} \to \frac{\sqrt{2}}{2}.(\mathbb{T} - \mathbb{F})$: we cannot type the term $\{(had)\ (\frac{\sqrt{2}}{2}.\mathbf{true} + \frac{\sqrt{2}}{2}.\mathbf{false})\}$ since there is no possibility to conciliate the two constraints on T.

To solve this last problem, we introduce the forall construction in the type system. The term *had* can now be typed with $\forall T.((\mathbf{I} \to \frac{\sqrt{2}}{2}.(\mathbb{T} + \mathbb{F})) \to (\mathbf{I} \to \frac{\sqrt{2}}{2}.(\mathbb{T} - \mathbb{F})) \to T) \to T$ and the types $\mathbb{T}$ and $\mathbb{F}$ are updated to be respectively $\forall XY.X \to (Y \to X)$ and $\forall XY.X \to (Y \to Y)$. The terms $\{(had)\ \mathbf{true}\}$ and $\{(had)\ \mathbf{false}\}$ can both be well-typed with respective types $\frac{\sqrt{2}}{2}.(\mathbb{T} + \mathbb{F})$ and $\frac{\sqrt{2}}{2}.(\mathbb{T} - \mathbb{F})$, as expected.

Let us try to type the term $\mathbf{0}$. Analogously to what was done for terms, a natural possibility is to add a special type $\overline{0}$ to type it. This is a reasonable solution that has been used in Chapters 3 and 4. In this naive interpretation, we would have $0.T$ equivalent to $\overline{0}$ and $\overline{0}$ would be the unit for the addition on types.

However, consider the following example. Let $\lambda x.x$ be of type $U \to U$ and let $\mathbf{r}$ be of type R. The term $\lambda x.x + \mathbf{r} - \mathbf{r}$ is of type $(U \to U) + 0.R$, that is, $(U \to U)$. Now choose $\mathbf{b}$ of type U: we are allowed to say that $(\lambda x.x + \mathbf{r} - \mathbf{r})\ \mathbf{b}$ is of type U. This term reduces to $\mathbf{b} + (\mathbf{r})\ \mathbf{b} - (\mathbf{r})\ \mathbf{b}$. If the type system is reasonable enough, we should at least be able to type $(\mathbf{r})\ \mathbf{b}$. However, since there is no constraints on the type R, this is difficult to enforce.

The problem comes from the fact that along the typing of $\mathbf{r} - \mathbf{r}$, the type of $\mathbf{r}$ is lost in the equivalence $\overline{0} \equiv 0.R$. The only solution is to distinguish $\overline{0}$ from $0.R$. We can also remove $\overline{0}$ altogether, and this is the choice we make for *Vectorial*: without type $\overline{0}$, we do not equate $T + 0.R$ and T.

The term $\mathbf{0}$ can be typed with any type $0.T$, so long as T is inhabited (*i.e.* $\mathbf{0}$ can come from a reduction of $\mathbf{r} - \mathbf{r}$ for some term $\mathbf{r}$ of type T).

5.2.1 Types

The grammar of types is defined in Figure 5.2. As in the previous chapters, types come in two flavours: *unit types* and general types, that is, linear combinations of types. Unit types include all types of System F and intuitively they are used to type basis terms. The arrow type admits only a unit type in its domain. Same as before, this is due to the fact that the argument of a λ-abstraction can only be substituted by a basis term. For the same reason, type variables, denoted by X, Y, etc. can only be substituted by unit types.

We also define an equivalence relation upon types as follows:

Definition 5.2.1. *For any scalar α, β and types T, R, S, we define the* type equivalence $\equiv$ *to be the least*

congruence such that

$$\begin{array}{rclrclrcl} 1.T & \equiv & T, & \alpha.T+\alpha.R & \equiv & \alpha.(T+R), & T+R & \equiv & R+T, \\ \alpha.(\beta.T) & \equiv & (\alpha\times\beta).T, & \alpha.T+\beta.T & \equiv & (\alpha+\beta).T, & T+(R+S) & \equiv & (T+R)+S. \end{array}$$

This makes the types into a weak module over the scalars: they almost form a module apart from the fact that there is no neutral element for the addition. Note that although we do not have any special type $\overline{0}$ (as discussed at the beginning of the section), we do have $0.T$; however $0.T$ is not the neutral element of the addition on types.

We may use the summation ($\sum$) notation without ambiguity, due to the associativity and commutativity equivalences of $+$.

$$\begin{array}{lll} \textit{Types:} & T,R,S ::= & U \mid \alpha.T \mid T+R \\ \textit{Unit types:} & U,V,W ::= & X \mid U\to T \mid \forall X.U \end{array}$$

$$\frac{}{\Gamma,x:U\vdash x:U}\,ax \qquad \frac{\Gamma\vdash \mathbf{t}:T}{\Gamma\vdash \mathbf{0}:0.T}\,0_I \qquad \frac{\Gamma,x:U\vdash \mathbf{t}:T}{\Gamma\vdash \lambda x.\mathbf{t}:U\to T}\to_I$$

$$\frac{\Gamma\vdash \mathbf{t}:\sum_{i=1}^{n}\alpha_i.\forall\vec{X}.(U\to T_i) \qquad \Gamma\vdash \mathbf{r}:\sum_{j=1}^{m}\beta_j.V_j \qquad \forall V_j,\exists\vec{W}_j, U[\vec{W}_j/\vec{X}]=V_j}{\Gamma\vdash (\mathbf{t})\ \mathbf{r}:\sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_i\times\beta_j.T_i[\vec{W}_j/\vec{X}]}\to_E$$

$$\frac{\Gamma\vdash \mathbf{t}:\sum_{i=1}^{n}\alpha_i.U_i \quad X\notin FV(\Gamma)}{\Gamma\vdash \mathbf{t}:\sum_{i=1}^{n}\alpha_i.\forall X.U_i}\,\forall_I \qquad \frac{\Gamma\vdash \mathbf{t}:\sum_{i=1}^{n}\alpha_i.\forall X.U_i}{\Gamma\vdash \mathbf{t}:\sum_{i=1}^{n}\alpha_i.U_i[V/X]}\,\forall_E$$

$$\frac{\Gamma\vdash \mathbf{t}:T}{\Gamma\vdash \alpha.\mathbf{t}:\alpha.T}\,s_I \qquad \frac{\Gamma\vdash \mathbf{t}:T \qquad \Gamma\vdash \mathbf{r}:R}{\Gamma\vdash \mathbf{t}+\mathbf{r}:T+R}\,+_I \qquad \frac{\Gamma\vdash \mathbf{t}:T \qquad T\equiv R}{\Gamma\vdash \mathbf{t}:R}\,\equiv$$

Figure 5.2: Types and typing rules of *Vectorial*

The following lemmas give some properties of the equivalence relation. Types are linear combinations of unit types (Lemma 5.2.2). Finally, the equivalence is well-behaved with respect to type constructs (Lemma 5.2.3).

Lemma 5.2.2 (Types characterisation)**.** *For any type T, there exist $n\in\mathbb{N}$, $\alpha_1,\dots,\alpha_n\in\mathcal{S}$ and unit types $U_1,\dots,U_n$ such that $T\equiv\sum_{i=1}^{n}\alpha_i.U_i$.*

Proof. Structural induction over T. If T is a unit type, take $\alpha=n=1$ and so $T\equiv\sum_{i=1}^{1}1.U=1.U$. If $T=\alpha.T'$, then by the induction hypothesis $T'\equiv\sum_{i=1}^{n}\alpha_i.U_i$, so $T=\alpha.T'\equiv\alpha.\sum_{i=1}^{n}\alpha_i.U_i\equiv\sum_{i=1}^{n}(\alpha\times\alpha_i).U_i$. If $T=R+S$, then by the induction hypothesis $R\equiv\sum_{i=1}^{n}\alpha_i.U_i$ and $S\equiv\sum_{j=1}^{m}\beta_j.V_j$, so $T=R+S\equiv\sum_{i=1}^{n}\alpha_i.U_i+\sum_{j=1}^{m}\beta_j.V_j$. □

Lemma 5.2.3 (Equivalence forall-introduction)**.**

1. $\sum_{i=1}^{n} \alpha_i.U_i \equiv \sum_{j=1}^{m} \beta_j.V_j \Leftrightarrow \sum_{i=1}^{n} \alpha_i.\forall X.U_i \equiv \sum_{j=1}^{m} \beta_j.\forall X.V_j$.
2. $\sum_{i=1}^{n} \alpha_i.\forall X.U_i \equiv \sum_{j=1}^{m} \beta_j.V_j \Rightarrow \forall V_j, \exists W_j, V_j \equiv \forall X.W_j$.
3. $T \equiv R \Rightarrow T[U/X] \equiv R[U/X]$.

Proof. Straightforward case analysis over the equivalence rules. □

5.2.2 Typing Rules

The typing rules are described in Figure 5.2. The axiom (ax) and the arrow introduction rule $(\to_I)$ are the usual ones. The rule to type the term $\mathbf{0}$ (0_I) takes into account the discussion at the beginning of Section 5.2. This rule also ensures that the type of $\mathbf{0}$ is inhabited, discarding problematic types like $0.\forall X.X$. Any sum of typed terms can be typed using rule $+_I$. Similarly, any scaled typed term can be typed with s_I. These two rules are inherited from *Additive* and *Scalar* respectively. The rule $\equiv$ ensures that equivalent types can be used to type the same terms. Finally, the particular form of the arrow-elimination rule $(\to_E)$ is due to the rewrite rules in the Application rules that distribute sums and scalars over application. This has been already discussed in Section 4.1. The need and use of this complicated arrow elimination can be illustrated by three examples.

Example 5.2.4. *The rule $\to_E$ is easier to read for trivial linear combinations. It states that provided that $\Gamma \vdash \mathbf{r} : \forall X.U \to R$ and $\Gamma \vdash \mathbf{t} : V$, if there exists some type W such that $V = U[W/X]$, then since the sequent $\Gamma \vdash \mathbf{r} : V \to R[W/X]$ is valid, we also have $\Gamma \vdash (\mathbf{r})\ \mathbf{t} : R[W/X]$.*

Example 5.2.5. *Consider the terms $\mathbf{b}_1$ and $\mathbf{b}_2$, of respective types U_1 and U_2. The term $\mathbf{b}_1 + \mathbf{b}_2$ is of type $U_1 + U_2$. We would reasonably expect the term $(\lambda x.x)\ (\mathbf{b}_1 + \mathbf{b}_2)$ to be also of type $U_1 + U_2$. This is the case thanks to rule $\to_E$. Indeed, type the term $\lambda x.x$ with the type $\forall X.X \to X$ and we can now apply the rule.*

Example 5.2.6. *A slightly more developed example is the projection of a pair of elements. It is possible to encode in System F the notion of pairs and projections: $\langle \mathbf{b}_1, \mathbf{b}_2 \rangle = \lambda x.((x)\ \mathbf{b}_1)\ \mathbf{b}_2$, $\langle \mathbf{b}'_1, \mathbf{b}'_2 \rangle = \lambda x.((x)\ \mathbf{b}'_1)\ \mathbf{b}'_2$, $\pi_1 = \lambda x.(x)\ (\lambda y.\lambda z.y)$ and $\pi_2 = \lambda x.(x)\ (\lambda y.\lambda z.z)$. Provided that $\mathbf{b}_1$, $\mathbf{b}'_1$, $\mathbf{b}_2$ and $\mathbf{b}'_2$ have respective types U,U', V and V', the type of $\langle \mathbf{b}_1, \mathbf{b}_2 \rangle$ is $\forall X.(U \to V \to X) \to X$ and the type of $\langle \mathbf{b}'_1, \mathbf{b}'_2 \rangle$ is $\forall X.(U' \to V' \to X) \to X$. The term π_1 and π_2 can be typed respectively with $\forall XYZ.((X \to Y \to X) \to Z) \to Z$ and $\forall XYZ.((X \to Y \to Y) \to Z) \to Z$.*

The term $(\pi_1 + \pi_2)\ (\langle \mathbf{b}_1, \mathbf{b}_2 \rangle + \langle \mathbf{b}'_1, \mathbf{b}'_2 \rangle)$ is then typable of type $U + U' + V + V'$, thanks to rule $\to_E$. Note that this is consistent with the rewrite system, since it reduces to $\mathbf{b}_1 + \mathbf{b}_2 + \mathbf{b}'_1 + \mathbf{b}'_2$.

5.3 Subject Reduction

Since the terms of λ_{lin} are not explicitly typed, we are bound to have sequents such as $\Gamma \vdash \mathbf{t} : T_1$ and $\Gamma \vdash \mathbf{t} : T_2$ for the same term $\mathbf{t}$. Using rules $+_I$ and s_I we get the valid typing judgement $\Gamma \vdash \alpha.\mathbf{t} + \beta.\mathbf{t} : \alpha.T_1 + \beta.T_2$. Given that $\alpha.\mathbf{t} + \beta.\mathbf{t}$ reduces to $(\alpha+\beta).\mathbf{t}$, a regular subject reduction would ask for the valid sequent $\Gamma \vdash (\alpha + \beta).\mathbf{t} : \alpha.T_1 + \beta.T_2$. Since in general we do not have $\alpha.T_1 + \beta.T_2 \equiv (\alpha + \beta).T_1 \equiv (\alpha + \beta).T_2$, we need to find a way around this.

A first natural solution could be by using the notion of principal types. However, since our type system can be seen as an extension of System F, the usual examples for the absence of principal types apply to our settings: we cannot rely on that.

A second potentially natural solution could be to ask for the sequent $\Gamma \vdash (\alpha+\beta).\mathbf{t} : \alpha.T_1 + \beta.T_2$ to be valid. If we force this typing rule into the system, it seems to solve the problem but then the type of a term becomes pretty much arbitrary: with typing context Γ, the term $(\alpha+\beta).\mathbf{t}$ could be typed with any combination $\gamma.T_1 + \delta.T_2$, when $\alpha+\beta = \gamma+\delta$.

The approach we favour in this Chapter is by using a notion of order on types. The order, denoted with $\sqsubseteq$, will be chosen so that the factorisation rules make the types of terms smaller according to the order. We will ask in particular that $(\alpha+\beta).T_1 \sqsubseteq \alpha.T_1 + \beta.T_2$ and $(\alpha+\beta).T_2 \sqsubseteq \alpha.T_1 + \beta.T_2$ whenever T_1 and T_2 are types for the same term.

In particular this approach solves a pitfall coming the rule $\mathbf{t}+\mathbf{0} \to \mathbf{t}$. Indeed, although $x : X \vdash x+\mathbf{0} : X + 0.T$ is well-typed for any inhabited T, the sequent $x : X \vdash x : X + 0.T$ is not valid in general. Hence the ordering is extended to state that $X \sqsubseteq X + 0.T$.

5.3.1 An Ordering Relation on Types

We start with another relation $\prec$ inspired from [Barendregt, 1992], which has also been defined for *Scalar* and *Additive* (Definitions 3.2.4 and 4.2.2). This relation can be deduced from the rules $\forall_I$ and $\forall_E$ as follows:

Definition 5.3.1 (Relation $\preceq$ on types)**.** *Write $T \prec R$ if either $T \equiv \sum_{i=1}^n \alpha_i.U_i$ and $R \equiv \sum_{i=1}^n \alpha_i.\forall X.U_i$ or $T \equiv \sum_{i=1}^n \alpha_i.\forall X.U_i$ and $R \equiv \sum_{i=1}^n \alpha_i.U_i[V/X]$. We denote the reflexive (with respect to $\equiv$) and transitive closure of $\prec$ with $\preceq$.*

The relation $\preceq$ admits a subsumption lemma (proof in Appendix D.1).

Lemma 5.3.2 ($\preceq$-subsumption)**.** *For any context Γ, any term $\mathbf{t}$ and any types T, R such that $T \preceq R$ and no free type variable in T occurs in Γ. Then $\Gamma \vdash \mathbf{t} : T$ implies $\Gamma \vdash \mathbf{t} : R$.* □

We can now define the ordering relation $\sqsubseteq$ on types discussed above.

Definition 5.3.3 (Relation $\sqsubseteq$ on types)**.** *Let $\sqsubseteq$ be the smallest reflexive transitive relation satisfying the rules:*

- $(\alpha+\beta).T \sqsubseteq \alpha.T + \beta.T'$*, if* $\exists \Gamma, \mathbf{t}$ *such that* $\Gamma \vdash \alpha.\mathbf{t} : \alpha.T$ *and* $\Gamma \vdash \beta.\mathbf{t} : \beta.T'$*.*
- $T \sqsubseteq T + 0.R$ *for any type R.*
- *If $T \preceq R$, then $T \sqsubseteq R$.*
- *If $T \sqsubseteq R$ and $U \sqsubseteq V$, then $T + S \sqsubseteq R + S$, $\alpha.T \sqsubseteq \alpha.R$, $U \to T \sqsubseteq U \to R$ and $\forall X.U \sqsubseteq \forall X.V$.*

Note that the fact that $\Gamma \vdash \mathbf{t} : T$ and $\Gamma \vdash \mathbf{t} : T'$ does not imply that $\beta.T \sqsubseteq \beta.T'$. Indeed, although $\beta.T \sqsubseteq 0.T + \beta.T'$, we do not have $0.T + \beta.T' \equiv \beta.T'$. Note also that this ordering is not a subtyping relation. Indeed, although $\vdash (\alpha+\beta).\lambda x.\lambda y.x : (\alpha+\beta).\forall X.X \to (X \to X)$ is valid and $(\alpha+\beta).\forall X.X \to (X \to X) \sqsubseteq \alpha.\forall X.X \to (X \to X) + \beta.\forall XY.X \to (Y \to Y)$, the sequent $\vdash (\alpha+\beta).\lambda x.\lambda y.x : \alpha.\forall X.X \to (X \to X) + \beta.\forall XY.X \to (Y \to Y)$ is not valid.

5.3.2 Weak Subject Reduction

Let R be any reduction rule from Figure 5.1. We denote $\to_R$ a one-step reduction by rule R. A weak version of the subject reduction theorem can be stated as follows.

Theorem 5.3.4 (Weak subject reduction)**.** *For any terms* $\mathbf{t}$, $\mathbf{t}'$, *any context* Γ *and any type* T, *if* $\mathbf{t} \to_R \mathbf{t}'$ *and* $\Gamma \vdash \mathbf{t} : T$, *then*

- *If* $R \notin$ *Factorisation rules, then* $\Gamma \vdash \mathbf{t}' : T$.
- *If* $R \in$ *Factorisation rules, then* $\exists S \sqsubseteq T$ *such that* $\Gamma \vdash \mathbf{t}' : S$ *and* $\Gamma \vdash \mathbf{t} : S$.

How weak is this *weak* subject reduction? First, note that the usual subject reduction result holds for most of the rules. Second, Theorem 5.3.4 ensures that a term $\mathbf{t}$ of a given type, when reduced, can be typed with a type that is also valid for the term $\mathbf{t}$. Third, we can characterise the order relation as follows.

Lemma 5.3.5 (Order characterisation)**.** *For any type* R, *unit types* $V_1, \ldots, V_m$ *and scalars* $\beta_1, \ldots, \beta_m$, *if* $R \sqsubseteq \sum_{j=1}^{m} \beta_j.V_j$, *then there exist a scalar* δ, *a natural number* k, *a set* $N \subseteq \{1, \ldots, m\}$ *and a unit type* $W \preceq V_k$ *such that* $R \equiv \delta.W + \sum_{j \in N} \beta_j.V_j$ *and* $\sum_{j=1}^{m} \beta_j = \delta + \sum_{j \in N} \beta_j$. □

How informative is the type judgement? The following three lemmas express formal relations between the types and their terms.

Lemma 5.3.6 (Scalars)**.** *For any context* Γ, *term* $\mathbf{t}$, *type* T *and scalar* α, *if* $\Gamma \vdash \alpha.\mathbf{t} : T$, *then there exists a type* R *such that* $T \equiv \alpha.R$ *and if* $\alpha \neq 0$, $\Gamma \vdash \mathbf{t} : R$. *Moreover, if* $\Gamma \vdash \alpha.\mathbf{t} : \alpha.T$, *then* $\Gamma \vdash \mathbf{t} : T$. □

Lemma 5.3.6 is a precursor of the generation lemma for scalars (Lemma 5.3.15). However it is more specific since it assumes a specific type and therefore more accurate in the sense that it gives a specific type for the inverted rule which is not possible in the actual generation lemma (its proof is in Appendix D.3).

Lemma 5.3.6 excludes the case of scaling by 0. It is covered by the following (proof in Appendix D.4).

Lemma 5.3.7 (Zeros)**.** *For any context* Γ, *term* $\mathbf{t}$, *unit types* $U_1, \ldots, U_n$ *and scalars* $\alpha_1, \ldots, \alpha_n$, *if* $\Gamma \vdash 0.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.U_i$, *then* $\forall i,\ \alpha_i = 0$ *and there are scalars* $\delta_1, \ldots, \delta_n$ *such that* $\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \delta_i.U_i$. □

The same invariant that has started in *Scalar* and continued in *Additive* is probable in *Vectorial*: a basis term can always be given a unit type (proof in Appendix D.5).

Lemma 5.3.8 (Basis terms)**.** *For any context* Γ, *type* T *and basis term* $\mathbf{b}$, *if* $\Gamma \vdash \mathbf{b} : T$ *then there exists a unit type* U *such that* $T \equiv U$. □

In the remainder of this section we provide a few definitions and lemmas that are required in order to prove Theorem 5.3.4.

In the same way that we can change a type in a sequent by an equivalent one using rule $\equiv$, we can prove that this can also be done in the context.

Lemma 5.3.9 (Context equivalence)**.** *For any term* $\mathbf{t}$, *any context* $\Gamma = (x_i : U_i)_i$ *and any type* T, *if* $\Gamma \vdash \mathbf{t} : T$ *and* $\Gamma' = (x_i : V_i)_i$ *where* $U_i \equiv V_i$, *then* $\Gamma' \vdash \mathbf{t} : T$.

Proof. We use the notation $(\Gamma, x : U)' = \Gamma', x : U'$ with $U \equiv U'$. If we have $\Gamma, x : U \vdash x : U$ as a consequence of an ax rule, we can use ax followed by $\equiv$ rules to prove $\Gamma', x : U' \vdash x : U$. If we have $\Gamma \vdash \lambda x.\mathbf{t} : U \to T$ as a

consequence of an $\to_I$ rule, by the induction hypothesis $\Gamma', x:U' \vdash \mathbf{t}:T$, so by rule $\to_I$, $\Gamma' \vdash \lambda x.\mathbf{t}:U' \to T$. Since $U' \to T \equiv U \to T$, we end with rule $\equiv$. All the other cases are straightforward using the induction hypothesis. □

The following lemma ensures that by substituting type variables for types or term variables for terms in an adequate manner, the derived type is still valid. Its proof is in Appendix D.6.

Lemma 5.3.10 (Substitution lemma)**.** *For any term* $\mathbf{t}$*, basis term* $\mathbf{b}$*, term variable* x*, context* Γ*, types* T*,* U*,* $\vec{W}$ *and type variables* $\vec{X}$*,*

1. *if* $\Gamma \vdash \mathbf{t}:T$*, then* $\Gamma[U/X] \vdash \mathbf{t}:T[U/X]$*;*
2. *if* $\Gamma, x:U \vdash \mathbf{t}:T$*,* $\Gamma \vdash \mathbf{b}:U[\vec{W}/\vec{X}]$ *and* $\vec{X} \notin FV(\Gamma)$*, then* $\Gamma \vdash \mathbf{t}[\mathbf{b}/x]:T[\vec{W}/\vec{X}]$. □

Three generation lemmas are required: two classical ones, for applications (Lemma 5.3.11) and for abstractions (Lemma 5.3.12, and Corollary 5.3.14,) and one for linear combinations: sums, scalars and zero (Lemma 5.3.15). The missing proofs are in Appendices D.7 to D.11.

Lemma 5.3.11 (Generation lemma (app))**.** *For any terms* $\mathbf{t}$*,* $\mathbf{r}$*, any context* Γ *and any type* T*, if* $\Gamma \vdash (\mathbf{t})\ \mathbf{r}:T$*, then there exist natural numbers* n, m*, unit types* $U, V_1, \dots, V_m$*, types* $T_1, \dots, T_n$ *and scalars* $\alpha_1, \dots, \alpha_n$ *and* $\beta_1, \dots, \beta_m$*, such that* $\Gamma \vdash \mathbf{t}: \sum_{i=1}^n \alpha_i.\forall\vec{X}.(U \to T_i)$*,* $\Gamma \vdash \mathbf{r}: \sum_{j=1}^m \beta_j.V_j$*, where for all* V_j*, there exists* $\vec{W}_j$ *such that* $U[\vec{W}_j/\vec{X}] = V_j$ *and* $\sum_{i=1}^n \sum_{j=1}^m \alpha_i \times \beta_j.T_i[\vec{W}_j/\vec{X}] \preceq T$. □

Lemma 5.3.12 (Generation lemma (abs))**.** *For any term variable* x*, term* $\mathbf{t}$*, context* Γ *and type* T*, if* $\Gamma \vdash \lambda x.\mathbf{t}:R$*, there exist types* U *and* T *such that* $U \to T \preceq R$ *and* $\Gamma, x:U \vdash \mathbf{t}:T$. □

The following lemma is needed for the proof of Corollary 5.3.14.

Lemma 5.3.13 (Arrows comparison)**.** *For any types* T, R *and any unit types* U, V*, if* $V \to R \preceq U \to T$*, then there exist* $\vec{W}, \vec{X}$ *such that* $U \to T \equiv (V \to R)[\vec{W}/\vec{X}]$. □

Corollary 5.3.14 (of Lemma 5.3.12)**.** *For any context* Γ*, term variable* x*, term* $\mathbf{t}$*, type variables* $\vec{X}$ *and types* U *and* T*, if* $\Gamma \vdash \lambda x.\mathbf{t}:\forall\vec{X}.(U \to T)$ *then the typing judgement* $\Gamma, x:U \vdash \mathbf{t}:T$ *is valid.* □

Proof. By Lemma 5.3.12, $\exists V, R, V \to R \preceq \forall\vec{X}.(U \to T)$ and $\Gamma, x:V \vdash \mathbf{t}:R$. Note that $V \to R \preceq \forall\vec{X}.(U \to T) \preceq U \to T$, so by Lemma 5.3.13, $\exists\vec{W}, \vec{Y}$ such that $U \to T \equiv (V \to R)[\vec{W}/\vec{Y}] \equiv V[\vec{W}/\vec{Y}] \to R[\vec{W}/\vec{Y}]$ so $U \equiv V[\vec{W}/\vec{Y}]$ and $T \equiv R[\vec{W}/\vec{Y}]$. Also by Lemma 5.3.10, $\Gamma[\vec{W}/\vec{Y}], x:V[\vec{W}/\vec{Y}] \vdash \mathbf{t}:R[\vec{W}/\vec{Y}]$. By Lemma 5.3.9 and rule $\equiv$, $\Gamma[\vec{W}/\vec{Y}], x:U \vdash \mathbf{t}:T$. If $\Gamma[\vec{W}/\vec{Y}] \equiv \Gamma$, then we are finished. In the other case, $\vec{Y}$ appears free in Γ. Since $V \to R \preceq U \to T$ and $\Gamma \vdash \lambda x.\mathbf{t}:V \to R$, according to Lemma 5.3.2, $U \to T$ can be obtained from $V \to R$ as a type for $\lambda x.\mathbf{t}$; then we would need to use the rule $\forall_I$; thus $\vec{Y}$ cannot appear free in Γ, which constitutes a contradiction. So, $\Gamma, x:U \vdash \mathbf{t}:T$. □

Lemma 5.3.15 (Generation lemma (linear combinations))**.** *For any context* Γ*, scalar* α*, terms* $\mathbf{t}$ *and* $\mathbf{r}$ *and types* S *and* T*:*

1. *if* $\Gamma \vdash \mathbf{t} + \mathbf{r}:S$ *then there exist types* R *and* R' *such that* $\Gamma \vdash \mathbf{t}:R$*,* $\Gamma \vdash \mathbf{r}:R'$ *and* $R + R' \preceq S$*;*
2. *if* $\Gamma \vdash \alpha.\mathbf{t}:T$*, then there exists a type* R *such that* $\alpha.R \preceq T$ *and* $\Gamma \vdash \alpha.\mathbf{t}:\alpha.R$*;*
3. *if* $\Gamma \vdash \mathbf{0}:T$*, then there exists a type* R *such that* $T \equiv 0.R$. □

5.3.3 Proof of Theorem 5.3.4

We are now ready to prove Theorem 5.3.4.

Proof. Let $\mathbf{t} \to_R \mathbf{t}'$ and $\Gamma \vdash \mathbf{t} : T$. We proceed by induction. We only give two interesting cases (the full proof can be found in Appendix D.12).

$R = \alpha.\mathbf{t} + \beta.\mathbf{t} \to (\alpha+\beta).\mathbf{t}$. Let $\Gamma \vdash \alpha.\mathbf{t} + \beta.\mathbf{t} : T$. Then by Lemma 5.3.15, $\exists R, S$ such that $\Gamma \vdash \alpha.\mathbf{t} : R$ and $\Gamma \vdash \beta.\mathbf{t} : S$ with $R + S \preceq T$. Then by Lemma 5.3.15, $\exists R', \alpha.R' \preceq R$ and $\Gamma \vdash \alpha.\mathbf{t} : \alpha.R'$, also $\exists S', \beta.S' \preceq S$ and $\Gamma \vdash \beta.\mathbf{t} : \beta.S'$.

- If $\alpha \neq 0$ (or analogously $\beta \neq 0$), then by Lemma 5.3.6, $\Gamma \vdash \mathbf{t} : R'$ and so by s_I we conclude $\Gamma \vdash (\alpha+\beta).\mathbf{t} : (\alpha+\beta).R'$. Notice that $(\alpha+\beta).R' \sqsubseteq \alpha.R' + \beta.S' \sqsubseteq R + S \sqsubseteq T$. Also using rules $+_I$ and $\equiv$ we conclude $\Gamma \vdash \alpha.\mathbf{t} + \beta.\mathbf{t} : (\alpha+\beta).R'$.
- If $\alpha = \beta = 0$, then notice that $\Gamma \vdash 0.\mathbf{t} : 0.R'$ and $0.R' \sqsubseteq 0.R' + 0.S' \sqsubseteq R + S \sqsubseteq T$. Also, using rules $+_I$ and $\equiv$, we conclude $\Gamma \vdash 0.\mathbf{t} + 0.\mathbf{t} : 0.R'$.

$R = (\lambda x.\mathbf{t})\ \mathbf{b} \to \mathbf{t}[\mathbf{b}/x]$. Let $\Gamma \vdash (\lambda x.\mathbf{t})\ \mathbf{b} : T$. Then by Lemma 5.3.11, there exist numbers n, m, scalars $\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_m$, a unit type U, and general types $T_1, \ldots, T_n$ such that the it can be derived: $\Gamma \vdash \lambda x.\mathbf{t} : \sum_{i=1}^n \alpha_i.\forall \vec{X}.(U \to T_i)$ and $\Gamma \vdash \mathbf{b} : \sum_{j=1}^m \beta_j.V_j$ with $\sum_{i=1}^n \sum_{j=1}^m \alpha_i \times \beta_j.T_i[\vec{W_j}/\vec{X}] \preceq T$, where $\forall V_j$, $\vec{W_j}$ is such that $U[\vec{W_j}/\vec{X}] \equiv V_j$.

By Lemma 5.3.8, $\sum_{i=1}^n \alpha_i.\forall \vec{X}.(U \to T_i) \equiv \forall \vec{X}.(U \to T_i)$ and $\forall i, k,\ T_i = T_k$, analogously $\sum_{j=1}^m \beta_j.V_j$ is equivalent to V_j where $\forall j, h,\ V_j = V_h$. So $\sum_{i=1}^n \alpha_i = 1$ and $\sum_{j=1}^m \beta_i = 1$. Then by rule $\equiv$, $\Gamma \vdash \lambda x.\mathbf{t} : \forall \vec{X}.(U \to T_i)$, and $\Gamma \vdash \mathbf{b} : V_i$.

Thus, by Corollary 5.3.14, $\Gamma, x : U \vdash \mathbf{t} : T_i$. Notice that $V_i \equiv U[\vec{W_i}/\vec{X}]$, then, by Lemma 5.3.10, we have $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T_i[\vec{W_i}/\vec{X}]$. Since $T_i[\vec{W_j}/\vec{X}] \equiv (1\times 1).T_i[\vec{W_j}/\vec{X}] = (\sum_{i=1}^n \alpha_i) \times (\sum_{j=1}^m \beta_j).T_i[\vec{W_j}/\vec{X}]$ which is equal to $(\sum_{i=1}^n \sum_{j=1}^m \alpha_i \times \beta_j).T_i[\vec{W_j}/\vec{X}]$, and as all the T_i are equivalents between them, this type is equivalent to $\sum_{i=1}^n \sum_{j=1}^m \alpha_i \times \beta_j.T_i[\vec{W_j}/\vec{X}] \preceq T$. By Lemma 5.3.2, $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T$.

□

5.4 Confluence and Strong Normalisation

The language has the usual properties for a typed lambda-calculus: the reduction is locally confluent, as has been proved in the proof of Corollary 3.3.13, and the type system enforces strong normalisation. From these two results, we infer the confluence of the rewrite system.

For proving strong normalisation of well-typed terms, we use reducibility candidates, a well-known method described for example in [Girard, Lafont, and Taylor, 1989, Chapter 14] The technique is adapted to linear combinations of terms.

A *neutral term* is a term that is not a lambda-abstraction and that is not in normal form (*i.e.* it does reduce). The set of *closed neutral terms* is denoted with $\mathcal{N}$. We write Λ_0 for the set of closed terms and SN for the set of closed, strongly normalising terms. If $\mathbf{t}$ is any term, $\mathrm{Red}(\mathbf{t})$ is the set of all terms $\mathbf{t}'$ such that $\mathbf{t} \to \mathbf{t}'$. It is naturally extended to sets of terms.

We say that a set S of closed terms is a reducibility candidate, denoted with $S \in \mathcal{RC}$ if the following conditions are verified:

RC$_1$ Strong normalisation: $S \subseteq SN$.

RC$_2$ Stability under reduction: $\mathbf{t} \in S$ implies $\mathrm{Red}(\mathbf{t}) \subseteq S$.

RC$_3$ Stability under neutral expansion: if $\mathbf{t} \in \mathcal{N}$ and $\mathrm{Red}(\mathbf{t}) \subseteq S$ then $\mathbf{t} \in S$.

RC$_4$ The common inhabit: $\mathbf{0} \in S$.

We define the following operations on reducibility candidates. Let $\mathcal{A}$ and $\mathcal{B}$ be in $\mathcal{RC}$.

- $\mathcal{A} \to \mathcal{B}$ is the closure of $\{\mathbf{t} \in \Lambda_0 \,|\, \forall \mathbf{b} \in \mathcal{A}, (\mathbf{t})\,\mathbf{b} \in \mathcal{B}\}$ under **RC**$_3$ and **RC**$_4$, where $\mathbf{b}$ is a base term.
- If $\{\mathcal{A}_i\}_i$ is a family of reducibility candidates, $\sum_i \mathcal{A}_i$ is the closure of $\{\sum_i \alpha_i.\mathbf{t}_i \,|\, \mathbf{t}_i \in \mathcal{A}_i\}$ under **RC**$_2$ and **RC**$_3$. If there is only one A in the sum, we write $1.A$ instead.

These operation also define reducibility candidates, as stated by the following lemma (*cf.* Appendix D.13 for its proof).

Lemma 5.4.1. *If $\mathcal{A}$, $\mathcal{B}$ and all the $\mathcal{A}_i$'s are in $\mathcal{RC}$, then so are $\mathcal{A} \to \mathcal{B}$, $\sum_i \mathcal{A}_i$ and $\cap_i \mathcal{A}_i$.* □

Reducibility model

A *single type valuation* is a partial function from type variables to reducibility candidates, that we define as a sequence of comma-separated mappings, with $\emptyset$ denoting the empty valuation:

$$\rho := \emptyset \,|\, \rho, X \mapsto \mathcal{A}.$$

Type variables are interpreted using pairs of single type valuations, that we simply call *valuations*, with common domain:

$$\rho = (\rho_+, \rho_-) \qquad \text{with} \qquad |\rho_+| = |\rho_-|$$

Given a valuation $\rho = (\rho_+, \rho_-)$, the *complementary valuation* $\bar{\rho}$ is the pair (ρ_-, ρ_+). We write $(X_+, X_-) \mapsto (A_+, A_-)$ for the valuation $(X_+ \mapsto A_+, X_- \mapsto A_-)$. A valuation is called *valid* if for all X, $\rho_-(X) \subseteq \rho_+(X)$.

To define the interpretation of a type T, we use the following result: the decomposition of a type can be made in a sum of unit types such that they appears only once in the decomposition (proof in Appendix D.14).

Corollary 5.4.2 (of Lemma 5.2.2)**.** *Any type T has a unique canonical decomposition $T \equiv \sum_{i=1}^{n} \alpha_i.U_i$ such that $\forall j, k$, $U_j \not\equiv U_k$.*

The interpretation $[\![T]\!]_\rho$ of a type T in a valuation $\rho = (\rho_+, \rho_-)$ defined for each free type variable of T is given by:

$$\begin{aligned}
[\![X]\!]_\rho &= \rho_+(X), \\
[\![U \to T]\!]_\rho &= [\![U]\!]_{\bar{\rho}} \to [\![T]\!]_\rho, \\
[\![\forall X.U]\!]_\rho &= \cap_{\mathcal{A} \subseteq \mathcal{B} \in \mathcal{RC}} [\![U]\!]_{\rho,(X_+,X_-)\mapsto(\mathcal{A},\mathcal{B})}, \\
\text{if } T \equiv \sum_i \alpha_i.U_i \text{ is the canonical decomposition of } T \text{ then } [\![T]\!]_\rho &= \sum_i [\![U_i]\!]_\rho.
\end{aligned}$$

From Lemma 5.4.1, the interpretation of any type is a reducibility candidate.

Reducibility candidates deal with closed terms, whereas proving the adequacy lemma by induction requires the use of open terms with some assumptions on their free variables, that will be guaranteed by a context. Therefore we use *substitutions* σ to close terms:

$$\sigma := \emptyset \mid (x \mapsto \mathbf{b}; \sigma), \qquad \text{so} \qquad \mathbf{t}_\emptyset = \mathbf{t} \qquad \text{and} \qquad \mathbf{t}_{x\mapsto \mathbf{b};\sigma} = \mathbf{t}[\mathbf{b}/x]_\sigma.$$

Given a context Γ, we say that a substitution σ *satisfies* Γ for the valuation ρ (notation: $\sigma \in [\![\Gamma]\!]_\rho$) when $(x : U) \in \Gamma$ implies $x_\sigma \in [\![U]\!]_{\bar{\rho}}$ (Note the change in polarity). Let $T \equiv \sum_{i=1}^n \alpha_i.U_i$, such that $\forall i, j,\ U_i \not\equiv U_j$, which always exists by Corollary 5.4.2.

A typing judgement $\Gamma \vdash \mathbf{t} : T$, is said to be *valid* (notation $\Gamma \models \mathbf{t} : T$) if for every valuation ρ, and set of valuations $\{\rho_i\}_n$, where ρ_i acts on $FV(U_i) \setminus FV(\Gamma)$, and for every substitution $\sigma \in [\![\Gamma]\!]_\rho$, we have $\mathbf{t}_\sigma \in \sum_{i=1}^n \alpha_i.[\![U_i]\!]_{\rho,\rho_i}$.

Lemma 5.4.3 (Adequacy Lemma)**.** *Every derivable typing judgement is valid: for every valid sequent* $\Gamma \vdash \mathbf{t} : T$*, we have* $\Gamma \models \mathbf{t} : T$*.*

Proof. The proof is made by induction on the derivation of judgement $\Gamma \vdash \mathbf{t} : T$. We look at the last typing rule that is used, and show in each case that $\Gamma \models \mathbf{t} : T$, *i.e.* if $T \equiv \sum_{i=1}^n \alpha_i.U_i$ in the sense of Corollary 5.4.2, then $\mathbf{t}_\sigma \in \bigoplus_{i=1}^n \alpha_i.[\![U_i]\!]_{\rho,\rho_i}$ for every valuation ρ, set of valuations $\{\rho_i\}_n$, and substitution $\sigma \in [\![\Gamma]\!]_\rho$ (*i.e.* substitution σ such that $(x : V) \in \Gamma$ implies $x_\sigma \in [\![V]\!]_{\bar{\rho}}$). The full proof is depicted in Appendix D.15□

Theorem 5.4.4 (Strong normalisation)**.** *Assume that* $\Gamma \vdash \mathbf{t} : T$ *is a valid sequent, then* $\mathbf{t}$ *is strongly normalising.*

Proof. If Γ is the list $(x_i : U_i)_i$, the sequent $\vdash \lambda x_1 \dots x_n.\mathbf{t} : U_1 \to (\cdots \to (U_n \to T) \cdots)$ is valid. Using Lemma 5.4.3, we deduce that for any valuation ρ and any substitution $\sigma \in [\![\emptyset]\!]_\rho$, we have $\sigma(\mathbf{t}) \in [\![T]\!]_\rho$. By construction, σ does nothing on $\mathbf{t}$: $\sigma(\mathbf{t}) = \mathbf{t}$. Since $[\![T]\!]_\rho$ is a reducibility candidate, $\lambda x_1 \dots x_n.\mathbf{t}$ is strongly normalising. Now suppose that $\mathbf{t}$ were not strongly normalising. There would be an infinite rewrite sequence of terms $(\mathbf{t}_i)_i$ starting with $\mathbf{t}$. But then $(\lambda\vec{x}.\mathbf{t}_i)_i$ would then be an infinite rewrite sequence of terms starting with a strongly normalising term: contradiction. Therefore, $\mathbf{t}$ is strongly normalising. □

The confluence of the system is a corollary of this result.

Corollary 5.4.5 (Confluence)**.** *If* $\Gamma \vdash \mathbf{s} : S$ *is a valid typing judgement and if* $\mathbf{s} \to^* \mathbf{r}$ *and* $\mathbf{s} \to^* \mathbf{t}$*, then there exists* $\mathbf{s}'$ *such that* $\mathbf{r} \to^* \mathbf{s}'$ *and* $\mathbf{t} \to^* \mathbf{s}'$*.*

Proof. A rewrite system that is both locally confluent and strongly normalising is confluent [TeReSe, 2003]. □

5.5 Expressing Matrices and Vectors

In this section we come back to the motivating example introducing the type system and we show how λ_{lin} handles the Hadamard gate, and how to encode matrices and vectors.

With an empty typing context, the booleans

$$\mathbf{true} = \lambda x.\lambda y.x \qquad \text{and} \qquad \mathbf{false} = \lambda x.\lambda y.y$$

can be respectively typed with the types

$$\mathbb{T} = \forall XY.Y \to (Y \to X) \qquad \text{and} \qquad \mathbb{F} = \forall XY.X \to (Y \to Y).$$

The superposition has the following type

$$\vdash \alpha.\mathbf{true} + \beta.\mathbf{false} : \alpha.\mathbb{T} + \beta.\mathbb{F}.$$

Note that it can also be typed with $(\alpha + \beta).\forall X.X \to X \to X$.

With an empty typing context, the linear map $\mathbf{U}$ sending $\mathbf{true}$ to $a.\mathbf{true} + b.\mathbf{false}$ and $\mathbf{false}$ to $c.\mathbf{true} + d.\mathbf{false}$ is written as

$$\mathbf{U} = \lambda x.((x)[a.\mathbf{true} + b.\mathbf{false}])[c.\mathbf{true} + d.\mathbf{false}].$$

The following sequent is valid:

$$\vdash \mathbf{U} : \forall X.((I \to (a.\mathbb{T} + b.\mathbb{F})) \to (I \to (c.\mathbb{T} + d.\mathbb{F})) \to X) \to X.$$

This is consistent with the discussion in the introduction: the Hadamard gate is the case $a = b = c = \frac{\sqrt{2}}{2}$ and $d = -\frac{\sqrt{2}}{2}$. One can check that with an empty typing context, $\{(\mathbf{U})\ \mathbf{true}\}$ is well typed of type $a.\mathbb{T} + b.\mathbb{F}$, as expected since it reduces to $a.\mathbf{true} + b.\mathbf{false}$.

The term $\{(had)\ \frac{\sqrt{2}}{2}.(\mathbf{true} + \mathbf{false})\}$ is well-typed of type $\mathbb{T} + 0.\mathbb{F}$. Since the term reduces to $\mathbf{true}$, this is still consistent with the subject reduction: we indeed have $\mathbb{T} \sqsubseteq \mathbb{T} + 0.\mathbb{F}$.

5.6 Conclusions and open questions

In this Chapter we defined a typed, algebraic λ-calculus satisfying a weak subject reduction. The language allows making arbitrary linear combinations of λ-terms $\alpha.\mathbf{t} + \beta.\mathbf{u}$. Its *Vectorial* type system is a fine-grained analysis tool describing the "vectorial" properties of typed terms: first, it keeps track of the 'amplitude of a term', *i.e.* if $\mathbf{t}$ and $\mathbf{u}$ both have the same type U, then $\alpha.\mathbf{t} + \beta.\mathbf{u}$ has type $(\alpha + \beta).U$. Then it keeps track of the 'direction of a term', *i.e.* if $\mathbf{t}$ and $\mathbf{u}$ have types U and V respectively, then $\alpha.\mathbf{t} + \beta.\mathbf{u}$ has type $\alpha.U + \beta.V$. This type system is expressive enough to be able to type the encoding of matrices and vectors.

The resulting type system has the property that if $\Gamma \vdash \mathbf{t} : \sum_i \alpha_i.U_i$ then there exists $\mathbf{t}'$ such that $\mathbf{t} \to^* \mathbf{t}'$ and $\mathbf{t}' = \sum_i \alpha_i.\mathbf{b}_i$, where each $\mathbf{b}_i$ is a basis term of type U_i. Such a $\mathbf{t}'$ is obtained by normalising $\mathbf{t}$ under all rules but the factorisation rules. Within such a $\mathbf{t}'$ there may be subterms of the form $\alpha_1.\mathbf{b} + \alpha_2.\mathbf{b}$ of type $\alpha_1.V_1 + \alpha_2.V_2$, which are redexes for the factorisation rules. Under our type system, the reduct $(\alpha_1 + \alpha_2).\mathbf{b}$ can be given both the types $(\alpha_1 + \alpha_2).V_1$ and $(\alpha_1 + \alpha_2).V_2$.

The next step is to consider a Church style *Vectorial*, which would solve the factorisation problem and give back a full subject reduction property: $\mathbf{t} + \mathbf{t} \to 2.\mathbf{t}$ only of both $\mathbf{t}$ are the same, which in Church style means that they have the same type.

Chapter 6

Extending sums of types to the complete calculus via lower bounds

Résumé du Chapitre

Avant de passer à un système de type Vectorial à la Church, comme nous avons suggéré dans le dernier chapitre, nous explorons un chemin alternatif et plus classique : une extension d'Additive au calcul complet. Cette extension a l'avantage d'avoir des sommes, mais pas de scalaire, dans les types, permettant comme démontré au Chapitre 4, d'être encodé dans le Système F avec paires. En conséquence, la question qui se pose ici est comment caractériser les scalaires dans les types, en n'utilisant que des sommes. Nous proposons une modification de λ_{lin}, où les valeurs scalaires sont uniquement des nombres réels non-négatifs, le type consiste alors à prendre la partie entière des scalaires afin de les représenter au travers des sommes. Cela permet de considérer λ^{add} comme une interprétation abstraite de λ_{lin}, et donc aussi du Système F avec paires. Cette extension est une approche alternative à Vectorial, mais qui ne donne qu'une information approximative sur les superpositions positives. ∎

Before moving to a Church-style *Vectorial* type system, as suggested in the last chapter, we will explore an alternative and more classic path: an extension of *Additive* (*cf.* Chapter 4) to the full calculus. This have the advantage of having only sums –and not scalars– in the types, which as has been shown, is possible to encode in System F with pairs (System F_P). Extending *Additive* to the full calculus gives rise to the question of how to characterise the scalars in types by using only sums. We propose a modification of λ_{lin} where the scalars are only positive real numbers, and then the types will approximate the scalars by taking their floor in order to represent them through sums. This allows us to consider λ^{add} as an abstract interpretation of λ_{lin} and hence also System F_P. This extension is an alternative path to *Vectorial* which gives us some approximate information of positive superpositions, and provides strong normalisation to the calculus, and hence its confluence, in a simpler way than *Vectorial*.

Since this is just an study of feasibility, we will prefer a $\to_E$ rule *à la Additive*, which although simpler than the one from *Vectorial*, is incomplete since it can only type applications where the arguments are sums of terms with the same type (up to the scalar). On the contrary, we work in Church-style, since it has been shown in in the previous chapter that factorisation rules with sums fail to have a complete subject reduction.

Plan of the chapter. In section 6.1 the typed version of λ_{lin}, called λ^{CA}, is presented. Section 6.2 is devoted to proving that the system features some basic properties, namely a subject reduction result and the strong normalisation property, which entails the confluence of the calculus. Section 6.4 shows an abstract interpretation of λ^{CA} into λ^{add}. Finally, section 6.5 concludes.

6.1 The calculus λ^{CA}

We introduce the calculus λ^{CA}, which extends explicit System F [Reynolds, 1974] with linear combinations of λ-terms. Figure 6.1 shows the abstract syntax of types and terms of the calculus, where the terms are based on those of λ_{lin}. Scalars, written $\alpha, \beta, \gamma, \ldots$, range over non-negative real numbers. Remark that this is a semi-ring, and no a ring as in the previous systems. In fact, it can be shown that a semi-ring is enough for most of the systems presented in this thesis.

Notice that there are no scalars at the type level, however, we introduce the following notation: for any natural number $n \geq 0$, we write $n.T$ for the type $T + T + \cdots + T$ (n times). Also, $0.T = \overline{0}$. We may also use the summation symbol $\sum_{i=1}^{n} T_i$, with the convention that $\sum_{i=1}^{0} T = \overline{0}$.

The reduction rules, based on λ_{lin}, are also given in Figure 6.1. As usual, all reductions are performed modulo associativity and commutativity of the operator $+$. It is essentially the rewrite system of λ_{lin}, with the extra type-application rule.

Types:

$$T, R, S ::= U \mid T + R \mid \overline{0}$$
$$U, V, W ::= X \mid U \to T \mid \forall X.U$$

Terms:

$$\mathbf{t}, \mathbf{r}, \mathbf{u} ::= \mathbf{b} \mid (\mathbf{t})\ \mathbf{r} \mid \mathbf{t}@U \mid \mathbf{0} \mid \alpha.\mathbf{t} \mid \mathbf{t} + \mathbf{r} \mid \Lambda X.\mathbf{t}$$
$$\mathbf{b} ::= x : U \mid \lambda x{:}U.\mathbf{t} \mid \Lambda X.\mathbf{b}$$

Elementary rules:

$0.\mathbf{t} \to \mathbf{0}$,
$1.\mathbf{t} \to \mathbf{t}$,
$\alpha.\mathbf{0} \to \mathbf{0}$,
$\alpha.(\beta.\mathbf{t}) \to (\alpha \times \beta).\mathbf{t}$,
$\alpha.(\mathbf{t} + \mathbf{r}) \to \alpha.\mathbf{t} + \alpha.\mathbf{r}$,
$\mathbf{t} + \mathbf{0} \to \mathbf{t}$.

Factorisation rules:

$\alpha.\mathbf{t} + \beta.\mathbf{t} \to (\alpha + \beta).\mathbf{t}$,
$\alpha.\mathbf{t} + \mathbf{t} \to (\alpha + 1).\mathbf{t}$,
$\mathbf{t} + \mathbf{t} \to (1 + 1).\mathbf{t}$.

Beta reductions:

$(\lambda x{:}U.\mathbf{t})\ \mathbf{b} \to \mathbf{t}[\mathbf{b}/x]$,
$(\Lambda X.\mathbf{t})@U \to \mathbf{t}[U/X]$.

Application rules:

$(\mathbf{t} + \mathbf{r})\ \mathbf{u} \to (\mathbf{t})\ \mathbf{u} + (\mathbf{r})\ \mathbf{u}$,
$(\mathbf{u})\ (\mathbf{t} + \mathbf{r}) \to (\mathbf{u})\ \mathbf{t} + (\mathbf{u})\ \mathbf{r}$,
$(\alpha.\mathbf{t})\ \mathbf{r} \to \alpha.(\mathbf{t})\ \mathbf{r}$,
$(\mathbf{r})\ (\alpha.\mathbf{t}) \to \alpha.(\mathbf{r})\ \mathbf{t}$,
$(\mathbf{0})\ \mathbf{t} \to \mathbf{0}$,
$(\mathbf{t})\ \mathbf{0} \to \mathbf{0}$,

Contextual rules: If $\mathbf{t} \to \mathbf{r}$, then for any term $\mathbf{u}$, scalar α, variable x, unit type U and type variable X,

$(\mathbf{t})\ \mathbf{u} \to (\mathbf{r})\ \mathbf{u}$,
$(\mathbf{u})\ \mathbf{t} \to (\mathbf{u})\ \mathbf{r}$,
$\lambda x{:}U.\mathbf{t} \to \lambda x{:}U.\mathbf{r}$
$\mathbf{t} + \mathbf{u} \to \mathbf{r} + \mathbf{u}$,
$\mathbf{u} + \mathbf{t} \to \mathbf{u} + \mathbf{r}$,
$\Lambda X.\mathbf{t} \to \Lambda X.\mathbf{r}$
$\alpha.\mathbf{t} \to \alpha.\mathbf{r}$ and
$\lambda x.\mathbf{t} \to \lambda x.\mathbf{r}$

Figure 6.1: Syntax and reduction rules of λ^{CA}

Notice that the grammar of types is the same as the one of *Additive*, presented in Chapter 4 (*cf.* Figure 4.2). Also, the same equivalences apply, namely $T+\overline{0} \equiv T$, $T+R \equiv R+T$ and $T+(R+S) \equiv (T+R)+S$ (*cf.* Definition 4.1.1).

The premise of the type system, as has been announced in the introduction, is that if $\mathbf{t}$ has type T, then $\alpha.\mathbf{t}$ has type $\lfloor\alpha\rfloor.T$. For example, $\frac{5}{2}.\mathbf{t}$ would have type $T + T$. The intuitive interpretation is that the type $2.T$ gives a lower bound of the "amount" of $\mathbf{t} : T$ in the term.

The typing rules are given in Figure 6.2. It is very similar to *Additive* (*cf.* Figure 4.2), updated to explicit typed terms, and with the extra rule mentioned in the above paragraph (s_I).

Remark 6.1.1. *Recall the discussion about the type* $\overline{0}$ *for the Vectorial type system (*cf. *Section 5.2): if* $T + \overline{0} \equiv T$*, then the term* $(\lambda x.\mathbf{t} + \mathbf{r} - \mathbf{r})\ \mathbf{b}$ *may have a type, but its reduct* $(\lambda x.\mathbf{t})\ \mathbf{b} + (\mathbf{r})\ \mathbf{b} - (\mathbf{r})\ \mathbf{b}$ *may have not. Notice that this issue comes from the negative numbers, however, something similar could happen in our setting:* $(\lambda x\colon U.\mathbf{t} + \frac{1}{2}.\mathbf{r})\ \mathbf{b}$ *may have a type as soon as* $\mathbf{b}$ *has type* U*, since the type of* $\frac{1}{2}.\mathbf{r}$ *is* $\lfloor \frac{1}{2} \rfloor.R = \overline{0}$*. However in this case* $(\frac{1}{2}.\mathbf{r})\ \mathbf{b}$ *will also have a type: precisely the type* $\overline{0}$*.*

$$\frac{}{\Gamma, x\colon U \vdash x\colon U}\, ax \qquad \frac{}{\Gamma \vdash \mathbf{0}\colon \overline{0}}\, ax_{\overline{0}} \qquad \frac{\Gamma \vdash \mathbf{t}\colon \sum_{i=1}^{n}(U \to T_i) \qquad \Gamma \vdash \mathbf{r}\colon m.U}{\Gamma \vdash (\mathbf{t})\ \mathbf{r}\colon \sum_{i=1}^{n} m.T_i}\, \to_E$$

$$\frac{\Gamma, x\colon U \vdash \mathbf{t}\colon T}{\Gamma \vdash \lambda x\colon U.\mathbf{t}\colon U \to T}\, \to_I \qquad \frac{\Gamma \vdash \mathbf{t}\colon \forall X.U}{\Gamma \vdash \mathbf{t}@V\colon U[V/X]}\, \forall_E$$

$$\frac{\Gamma \vdash \mathbf{t}\colon T \qquad \Gamma \vdash \mathbf{r}\colon R}{\Gamma \vdash \mathbf{t} + \mathbf{r}\colon T + R}\, +_I \qquad \frac{\Gamma \vdash \mathbf{t}\colon T \qquad T \equiv R}{\Gamma \vdash \mathbf{t}\colon R}\, \equiv \qquad \frac{\Gamma \vdash \mathbf{t}\colon U \qquad X \notin FV(\Gamma)}{\Gamma \vdash \Lambda X.\mathbf{t}\colon \forall X.U}\, \forall_I \qquad \frac{\Gamma \vdash \mathbf{t}\colon T}{\Gamma \vdash \alpha.\mathbf{t}\colon \lfloor \alpha \rfloor.T}\, s_I$$

Figure 6.2: Typing rules of λ^{CA}

Going back to the discussion of Section 4.1, allowing the arguments to have different types in rule $\to_E$, would lead to a more interesting calculus, but with a more complex rule. It has been done in the *Vectorial* type system (*cf.* Chapter 5), however since the intention of this Chapter is to relate the system we are presenting, with *Additive*, we will keep it simpler and relax this restriction.

The main novelty of the calculus is its treatment of scalars (rule sI). In order to avoid having scalars at the type-level, when typing $\alpha.\mathbf{t}$ we take the floor of the term-level scalar α and assign the type $\lfloor \alpha \rfloor.T$ to the term, which is a sum of T's. Hence the type $n.T$ represents a lower bound of the "amount" of atoms $\mathbf{t} : T$ within the term.

6.2 Subject Reduction with lower-bound

In λ^{CA} the types are imprecise about the "amount" of each type in a term. For example, let $\Gamma \vdash \mathbf{t} : T$ and consider the term $\mathbf{s} = \frac{1}{2}.\mathbf{t} + \frac{3}{2}.\mathbf{t}$. It can be checked that $\Gamma \vdash \mathbf{s}\colon T$. However, $\mathbf{s} \to^* 2.\mathbf{t}$, and $\Gamma \vdash 2.\mathbf{t}\colon 2.T$. In this example a term with type T reduces to a term with type $T + T$, proving that strict subject reduction does not hold for λ^{CA}. Nevertheless, we prove a similar property: as reduction progresses, types are either preserved or *strengthened*, *i.e.* they become more precise according to the relation between types defined as follows.

Definition 6.2.1. *Let* $\preccurlyeq$ *be the smallest reflexive (in terms of* $\equiv$*) and transitive relation satisfying the rules:*

- *If* $\alpha \leq \beta$*, then* $\alpha.T \preccurlyeq \beta.T$*.*
- *If* $T_1 \preccurlyeq T_2$ *and* $S_1 \preccurlyeq S_2$*, then* $T_1 + S_1 \preccurlyeq T_2 + S_2$*.*
- *If* $U_2 \preccurlyeq U_1$ *and* $T_1 \preccurlyeq T_2$*, then* $U_1 \to T_1 \preccurlyeq U_2 \to T_2$*.*

- *If* $T \preccurlyeq R$*, then* $\forall X.T \preccurlyeq \forall X.R$*.*

This entails that the derived type for a term is a lower-bound (with respect to $\preccurlyeq$) for the actual type of the reduced term. We can state the property in other terms: subject reduction is granted in the sense that the types involved in a term describe some of the types involved in its normal form. However the "amount" of those types will be just a lower-bound. It is formalised in Theorem 6.2.2.

Theorem 6.2.2 (Subject Reduction up to $\preccurlyeq$)**.** *For any terms* $\mathbf{t}$ *and* $\mathbf{t}'$*, context* Γ *and type* T*, if* $\mathbf{t} \to \mathbf{t}'$ *and* $\Gamma \vdash \mathbf{t} : T$ *then there exists some type* R *such that* $\Gamma \vdash \mathbf{t}' : R$ *and* $T \preccurlyeq R$*.*

Intuitively, $T \preccurlyeq R$ (R is at least as precise as T) means that there are more addends of the same type in R than in T, *e.g.* $A \preccurlyeq A + A$ for a fixed type A. Notice that the order relation is not the trivial one: although $T \preccurlyeq T + R$ for any R (because $T \equiv T + 0.R \preccurlyeq T + 1.R \equiv T + R$), the type T will stay or get increased, but never decrease.

The remain of this section is devoted to the proof of Theorem 6.2.2. Let us give some preliminary lemmas.

Lemma 6.2.3 (Uniqueness of type)**.** *Let* $\Gamma \vdash \mathbf{t} : T$ *and* $\Gamma \vdash \mathbf{t} : R$*, then* $T \equiv R$*.*

Proof. Let $\Gamma \vdash \mathbf{t} : T$. Induction on the depth of the derivation. For each item there are two ways to obtain its type: the trivial one, and by rule $\equiv$, which proves the lemma. □

Lemma 6.2.4 (Generation lemmas)**.** *Let* T *be a type and* Γ *a typing context.*

1. *For arbitrary terms* $\mathbf{u}$ *and* $\mathbf{v}$*, if* $\Gamma \vdash (\mathbf{u})\ \mathbf{r} : T$*, then there exist natural numbers* n *and* m*, a unit type* U*, and general types* $T_1, \ldots, T_n$*, such that* $\Gamma \vdash \mathbf{u} : \sum_{i=1}^{n}(U \to T_i)$ *and* $\Gamma \vdash \mathbf{r} : m.U$ *with* $\sum_{i=1}^{n} m.T_i \equiv T$*.*

2. *For any term* $\mathbf{t}$ *and unit type* U*, if* $\Gamma \vdash \lambda x : U.\mathbf{r} : T$*, then there exists a type* R *such that* $\Gamma, x : U \vdash \mathbf{r} : R$ *and* $U \to R \equiv T$*.*

3. *For any terms* $\mathbf{u}$ *and* $\mathbf{v}$*, if* $\Gamma \vdash \mathbf{u} + \mathbf{r} : T$*, then there exist types* R *and* S *such that* $\Gamma \vdash \mathbf{u} : R$ *and* $\Gamma \vdash \mathbf{r} : S$*, with* $R + S \equiv T$*.*

4. *For any term* $\mathbf{u}$ *and non-negative real number* α*, if* $\Gamma \vdash \alpha.\mathbf{u} : T$*, then there exists a type* R *such that* $\Gamma \vdash \mathbf{u} : R$ *and* $\lfloor \alpha \rfloor.R \equiv T$*.*

5. *For any term* $\mathbf{t}$*, if* $\Gamma \vdash \Lambda X.\mathbf{r} : T$*, then there exists a type* R *such that* $\Gamma \vdash \mathbf{r} : R$ *and* $\forall X.R \equiv T$ *with* $X \notin FV(\Gamma)$*.*

6. *For any term* $\mathbf{t}$ *and unit type* U*, if* $\Gamma \vdash \mathbf{r}@U : T$*, then there exists a type* V *such that* $\Gamma \vdash \mathbf{r} : \forall X.V$ *and* $V[U/X] \equiv T$*.*

Proof. All the proofs are by induction on the length of the derivation. There are only two cases for each item: the trivial case and having $\equiv$ as the final rule, which is also trivial using the induction hypothesis and rule $\equiv$. □

Corollary 6.2.5. *For any context* Γ*, types* U *and* T *and term* $\mathbf{t}$*, if* $\Gamma \vdash \lambda x : U.\mathbf{t} : U \to T$ *then* $\Gamma, x : U \vdash \mathbf{t} : T$*.*

Proof. By Lemma 6.2.4, item 2, there exists R such that $\Gamma, x:U \vdash \mathbf{t}:R$ and $U \to R \equiv U \to T$, which implies $T \equiv R$, hence we finish by rule $\equiv$. □

Lemma 6.2.6 (Substitution lemma)**.** *For any term* $\mathbf{t}$*, base term* $\mathbf{b}$*, context* Γ *and types* U *and* T*,*

1. *If* $\Gamma \vdash \mathbf{t}:T$*, then* $\Gamma[U/X] \vdash \mathbf{t}[U/X]:T[U/X]$.
2. *If* $\Gamma, x:U \vdash \mathbf{t}:T$ *and* $\Gamma \vdash \mathbf{b}:U$*, then* $\Gamma \vdash \mathbf{t}[\mathbf{b}/x]:T$.

Proof. Similarly as the proof of Lemma 4.2.8 (*cf.* Appendix C.6): for each item there is one more case, corresponding to rule s_I. Let $\Gamma \vdash \alpha.\mathbf{t}:\lfloor\alpha\rfloor.T$ as a consequence of $\Gamma \vdash \mathbf{t}:T$ and rule s_I, then:

1. by the induction hypothesis, $\Gamma[U/X] \vdash \mathbf{t}[U/X]:T[U/X]$. Therefore, using rule s_I, we have $\Gamma[U/X] \vdash \alpha.\mathbf{t}[U/X]:\lfloor\alpha\rfloor.T[U/X]$. Notice that $\alpha.\mathbf{t}[U/X] = (\alpha.\mathbf{t})[U/X]$ and $\lfloor\alpha\rfloor.T[U/X] = (\lfloor\alpha\rfloor.T)[U/X]$.
2. by the induction hypothesis $\Gamma \vdash \mathbf{u}[\mathbf{b}/x]:T$. Therefore, using rule s_I, we have $\Gamma \vdash \alpha.\mathbf{u}[\mathbf{b}/x]:\lfloor\alpha\rfloor.T$. Notice that $\alpha.\mathbf{u}[\mathbf{b}/x] = (\alpha.\mathbf{u})[\mathbf{b}/x]$.

□

Apart from the classic lemmas given above, we need some extra lemmas. Notice that some of them are direct reduction cases, which are needed in other cases.

Remark 6.2.7. *Notice that* $\Gamma \vdash \mathbf{0}:T$ *implies* $T \equiv \overline{0}$*, since the only rules typing* $\mathbf{0}$ *are* $ax_{\overline{0}}$ *and* $\equiv$*.*

We can extend the previous remark to applications.

Lemma 6.2.8 (Linearity of 0)**.** *For any context* Γ*, type* T *and term* $\mathbf{t}$*, if* $\Gamma \vdash (\mathbf{0})\ \mathbf{t}:T$ *or* $\Gamma \vdash (\mathbf{t})\ \mathbf{0}:T$*, then* $T \equiv \overline{0}$.

Proof. Let $\Gamma \vdash (\mathbf{0})\ \mathbf{t}:T$. By Lemma 6.2.4(1), there exist natural numbers n and m and types $U, T_1, \ldots, T_n$ such that $\Gamma \vdash \mathbf{0}:\sum_{i=1}^{n} U \to T_i$, where $\sum_{i=1}^{n} m.T_i \equiv T$. Then by Remark 6.2.7, we have $n = 0$ and so $T \equiv \sum_{i=1}^{0} m.T_i \equiv \overline{0}$.

Analogously, let $\Gamma \vdash (\mathbf{t})\ \mathbf{0}$, using the same Lemma 6.2.4(1), there exist natural numbers n, m and types $U, T_1, \ldots, T_n$ such that $\Gamma \vdash \mathbf{0}:m.U$ and $\sum_{i=1}^{n} m.T_i \equiv T$. Then, by Remark 6.2.7, we have $m = 0$ and so $T \equiv \sum_{i=1}^{n} 0.T_i \equiv \overline{0}$. □

Lemma 6.2.9 (Product)**.** *If* $\Gamma \vdash \alpha.(\beta.\mathbf{u}):T$ *then* $\Gamma \vdash (\alpha \times \beta).\mathbf{u}:R$ *with* $T \preccurlyeq R$.

Proof. Let $\Gamma \vdash \alpha.(\beta.\mathbf{u}):T$. By Lemma 6.2.4(4), there exists R_1 such that $\Gamma \vdash \beta.\mathbf{u}:R_1$ where $\lfloor\alpha\rfloor.R_1 \equiv T$. Then again by Lemma 6.2.4(4), there exists R_2 such that $\Gamma \vdash \mathbf{u}:R_2$ where $\lfloor\beta\rfloor.R_2 \equiv R_1$. So, using rule s_I, one can derive $\Gamma \vdash (\alpha \times \beta).\mathbf{u}:\lfloor\alpha \times \beta\rfloor.R_2$. Since $\alpha, \beta \geq 0$, $(\lfloor\alpha\rfloor \times \lfloor\beta\rfloor).R_2 \preccurlyeq \lfloor\alpha \times \beta\rfloor.R_2$, so $T \equiv \lfloor\alpha\rfloor.R_1 \equiv (\lfloor\alpha\rfloor \times \lfloor\beta\rfloor).R_2 \preccurlyeq \lfloor\alpha \times \beta\rfloor.R_2$. □

Lemma 6.2.10 (Distributivity)**.** *If* $\Gamma \vdash \alpha.(\mathbf{u} + \mathbf{r}):T$ *then* $\Gamma \vdash \alpha.\mathbf{u} + \alpha.\mathbf{r}:T$.

Proof. Let $\Gamma \vdash \alpha.(\mathbf{u} + \mathbf{r}):T$. By Lemma 6.2.4(4), there exists R such that $\Gamma \vdash \mathbf{u} + \mathbf{r}:R$ where $\lfloor\alpha\rfloor.R \equiv T$. Then by Lemma 6.2.4(3), there exist S_1 and S_2 such that $\Gamma \vdash \mathbf{u}:S_1$ and $\Gamma \vdash \mathbf{r}:S_2$, with $S_1 + S_2 \equiv R$. So, using rules s_I and $+_I$, we can derive $\Gamma \vdash \alpha.\mathbf{u} + \alpha.\mathbf{r}:\lfloor\alpha\rfloor.S_1 + \lfloor\alpha\rfloor.S_2$. Notice that $\lfloor\alpha\rfloor.S_1 + \lfloor\alpha\rfloor.S_2 \equiv \lfloor\alpha\rfloor.(S_1 + S_2) \equiv \lfloor\alpha\rfloor.R \equiv T$. □

Lemma 6.2.11 (Factorisation)**.** *If* $\Gamma \vdash \alpha.\mathbf{t} + \beta.\mathbf{t}:T$ *then* $\Gamma \vdash (\alpha + \beta).\mathbf{t}:R$ *and* $T \preccurlyeq R$.

Proof. Let $\Gamma \vdash \alpha.\mathbf{t} + \beta.\mathbf{t} : T$. By Lemma 6.2.4(3), there exist T_1 and T_2 such that $\Gamma \vdash \alpha.\mathbf{t} : T_1$ and $\Gamma \vdash \beta.\mathbf{t} : T_2$, where $T_1 + T_2 \equiv T$. Then by Lemma 6.2.4(4), there exist R_1 and R_2 such that $\Gamma \vdash \mathbf{t} : R_1$ and $\Gamma \vdash \mathbf{t} : R_2$, with $\lfloor\alpha\rfloor.R_1 \equiv T_1$ and $\lfloor\beta\rfloor.R_2 \equiv T_2$. Using s_I on the former, we obtain $\Gamma \vdash (\alpha + \beta).\mathbf{t} : \lfloor\alpha + \beta\rfloor.R_1$. Since R_1 and R_2 are both types for $\mathbf{t}$, we have $R_1 \equiv R_2$, so $T \equiv T_1 + T_2 \equiv \lfloor\alpha\rfloor.R_1 + \lfloor\beta\rfloor.R_2 \equiv \lfloor\alpha\rfloor.R_1 + \lfloor\beta\rfloor.R_1 \equiv (\lfloor\alpha\rfloor + \lfloor\beta\rfloor).R_1 \preccurlyeq (\lfloor\alpha + \beta\rfloor).R_1$. □

Lemma 6.2.12 (Base terms in unit)**.** *For any base term* $\mathbf{b}$*, context* Γ *and type* T*, if* $\Gamma \vdash \mathbf{b} : T$*, then there exists a unit type* U *such that* $\Gamma \vdash \mathbf{b} : U$*.*

Proof. If $\mathbf{b}$ is a variable, it must have a type given by its context, which must be a unit type. If $\mathbf{b}$ is an abstraction, it must have a type given by the $\to_I$ rule or the $\forall_I$ rule. In either case, these are unit types. □

With these lemmas, we can prove the Theorem 6.2.2. Some of the lemmas given are direct cases of subject reduction. The remaining cases can be found in Appendix E.1.

6.3 Confluence and Strong Normalisation

In order to prove the confluence of the language, we first prove strong normalisation, which in a locally confluent setting entails confluence. The strong normalisation proof goes as follows: first we show how to translate the terms of λ^{CA} into terms of λ_{lin}, *i.e.* terms without type annotations. Then we show that typability in λ^{CA} implies typability in *Vectorial*, and so the strong normalisation of the translated term. This allows us to prove the strong normalisation of λ^{CA} terms.

We use the following notation: $\Gamma \vdash_v \mathbf{t} : T$ means that the type T can be derived for the term $\mathbf{t}$ in the context Γ using the *Vectorial* type system. Reductions in *Vectorial* are denoted by $\to_v$, and $\to_v^=$ denotes its reflexivity closure: *i.e.* $\mathbf{t} \to_v^= \mathbf{r}$ means that either $\mathbf{t} \to_v \mathbf{r}$ or $\mathbf{t} = \mathbf{r}$. In the following, we call *type beta rule* to rule $(\Lambda X.\mathbf{t})@U \to \mathbf{t}[U/X]$.

Let $|\cdot|$ be the following translation from terms in λ^{CA} to terms in λ_{lin}:

$$\begin{array}{rclrclrclrcl} |x| & = & x & |\Lambda X.\mathbf{t}| & = & |\mathbf{t}| & |\mathbf{t}@U| & = & |\mathbf{t}| & |\alpha.\mathbf{t}| & = & \alpha.|\mathbf{t}| \\ |\lambda x : U.\mathbf{t}| & = & \lambda x.|\mathbf{t}| & |(\mathbf{t})\ \mathbf{r}| & = & (|\mathbf{t}|)\ |\mathbf{r}| & |\mathbf{0}| & = & \mathbf{0} & |\mathbf{t} + \mathbf{r}| & = & |\mathbf{t}| + |\mathbf{r}| \end{array}$$

Lemma 6.3.1. *If* $\mathbf{t} \to \mathbf{r}$*, then* $|\mathbf{t}| \to_v^= |\mathbf{r}|$*, where the equality only happens if* $\mathbf{t} \to \mathbf{r}$ *by a type beta rule.*

Proof. Notice that the translation only removes the type notations. So for each reduction rule in λ^{CA}, we can use the analogous rule in *Vectorial* in the translated term and obtain the translation of the reduct. The only case when this is no possible is in the type beta rule: notice that $(\Lambda X.\mathbf{t})@U \to \mathbf{t}[U/X]$, and $|(\Lambda X.\mathbf{t})@U| = |\mathbf{t}| = |\mathbf{t}[U/X]|$. □

Lemma 6.3.2. *If* $\Gamma \vdash \mathbf{t} : T$*, then* $\exists \Delta, R$ *such that* $\Delta \vdash_v |\mathbf{t}| : R$*.*

Proof. First we define a translation from a subset of types in *Vectorial* to types in λ^{CA}. Consider the subset of types of *Vectorial*, where scalars range over non-negative real numbers. Then the translation has this subset as domain:

$$|X| = X \qquad |U \to T| = |U| \to |T| \qquad |\forall X.U| = \forall X.|U| \qquad |\alpha.T| = \lfloor\alpha\rfloor.|T| \qquad |T + R| = |T| + |R|$$

where $n.T = \sum_{i=1}^{n} T$ with the convention that $0.T = \overline{0}$. We also extend this definition to contexts: $|\Gamma| = \{x : |U| \mid U \in \Gamma\}$.

Then we prove by induction on the length of the type derivation of $\Gamma \vdash \mathbf{t} : T$ that $|\Gamma| \vdash_v |\mathbf{t}| : R$ with $|R| \equiv T$. The full proof can be found in Appendix E.2. □

Lemma 6.3.3. *There is no infinite sequence reduction consisting only of type beta rules.*

Proof. Consider the following function from terms to natural numbers:

$$\sigma(x : U) = 1 \qquad \sigma(\Lambda X.\mathbf{t}) = 1 + \sigma(\mathbf{t}) \qquad \sigma(\mathbf{t}@U) = \sigma(\mathbf{t}) \qquad \sigma(\alpha.\mathbf{t}) = \sigma(\mathbf{t})$$
$$\sigma(\lambda x : U.\mathbf{t}) = \sigma(\mathbf{t}) \qquad \sigma((\mathbf{t})\ \mathbf{r}) = \sigma(\mathbf{t})\ \sigma(\mathbf{r}) \qquad \sigma(\mathbf{0}) = 1 \qquad \sigma(\mathbf{t} + \mathbf{r}) = \sigma(\mathbf{t}) + \sigma(\mathbf{r})$$

Then we prove by structural induction on $\mathbf{t}$ that $\sigma((\Lambda X.\mathbf{t})@U) > \sigma(\mathbf{t}[U/X])$. Since it is a positive strictly decreasing function on the type beta reduction, the sequence cannot be infinite. The full induction can be found in Appendix E.3. □

Theorem 6.3.4 (Strong Normalisation)**.** *If $\Gamma \vdash \mathbf{t} : T$ in λ^{CA}, then $\mathbf{t}$ is strongly normalising.*

Proof. Let $\Gamma \vdash \mathbf{t} : T$, then by Lemma 6.3.2, exists Δ and R such that $\Delta \vdash_v |\mathbf{t}| : R$. Thus by Theorem 5.4.4, $|\mathbf{t}|$ is strongly normalising. Assume that $\mathbf{t}$ is not strongly normalising, then $\mathbf{t} \to \mathbf{t}_1 \to \mathbf{t}_2 \to \cdots$. By Lemma 6.3.1, $|\mathbf{t}| \to_v^= |\mathbf{t}_1| \to_v^= |\mathbf{t}_2| \to_v^= \cdots$, since $|\mathbf{t}|$ is strongly normalising, there exists n such that $\forall i > n$, $|\mathbf{t}_i| = |\mathbf{t}_{i+1}|$, and by Lemma 6.3.1, it happens only when $\mathbf{t}_i \to \mathbf{t}_{i+1}$ by the type beta rule. Then $\mathbf{t}$ must be strongly normalising, since by Lemma 6.3.3, there cannot be an infinite sequence of type beta-rules. □

Confluence

Now confluence follows as a corollary of the strong normalisation theorem.

Corollary 6.3.5 (Confluence)**.** *The typed language λ^{CA} is confluent: for any term $\mathbf{t}$, if $\mathbf{t} \to^* \mathbf{r}$ and $\mathbf{t} \to^* \mathbf{u}$, then there exists a term $\mathbf{t}'$ such that $\mathbf{r} \to^* \mathbf{t}'$ and $\mathbf{u} \to^* \mathbf{t}'$.*

Proof. The proof of the local confluence of the system, *i.e.* the property saying that $\mathbf{t} \to \mathbf{r}$ and $\mathbf{t} \to \mathbf{u}$ imply that there exists a term $\mathbf{t}'$ such that $\mathbf{r} \to^* \mathbf{t}'$ and $\mathbf{u} \to^* \mathbf{t}'$, is an extension of the one presented for the untyped calculus in Chapter 5, where the set of algebraic rules (*i.e.* all rules but the beta reductions) have been proved to be locally confluent using the proof assistant Coq. Then, a straightforward induction entails the (local) commutation between the algebraic rules and the β-reductions. Finally, the confluence of the β-reductions is a trivial extension of the proof for λ-calculus. Local confluence plus strong normalisation (*cf.* Theorem 6.3.4) imply confluence [TeReSe, 2003]. □

6.4 Abstract Interpretation

The type system of λ^{CA} approximates the more precise types that are obtained under reduction. The approximation suggests that λ^{add} could be seen as an abstract interpretation of λ^{CA}: its terms can approximate the terms of λ^{CA}. Scalars can be approximated to their floor, and hence be represented by sums, just as the types in λ^{CA} do. This intuition is formalised in this section, using λ^{add} and the *Additive* type system presented in Chapter 4. This calculus is a typed version of the additive fragment of the λ_{lin} (*cf.* Section 1.2), which in turn is the untyped version of λ^{CA}.

The λ^{add} calculus is shown in Figure 6.4. The types and equivalences coincide with those from λ^{CA}. We write the types explicitly in the terms to match the presentation of λ^{CA}. We use $\vdash_A$ to distinguish the judgements in λ^{CA} ($\vdash$) from the judgements in λ^{add}. Also, we write the reductions in λ^{add} as $\to_A$.

$$\begin{array}{lll} \textit{Terms:} & \mathbf{t},\mathbf{r},\mathbf{s} ::= & \mathbf{b} \mid (\mathbf{t})\ \mathbf{r} \mid \mathbf{t}@U \mid \mathbf{0} \mid \mathbf{t}+\mathbf{r} \\ \textit{Basis terms:} & \mathbf{b} ::= & x:U \mid \lambda x:U.\mathbf{t} \mid \Lambda X.\mathbf{t} \end{array}$$

Distributivity rules:	*Zero rules:*	*β-reduction:*
$(\mathbf{u}+\mathbf{t})\ \mathbf{r} \to_A (\mathbf{u})\ \mathbf{r} + (\mathbf{t})\ \mathbf{r}$	$(\mathbf{0})\ \mathbf{t} \to_A \mathbf{0}$	$(\lambda x.\mathbf{t})\ \mathbf{b} \to_A \mathbf{t}[\mathbf{b}/x]$
$(\mathbf{r})\ (\mathbf{u}+\mathbf{t}) \to_A (\mathbf{r})\ \mathbf{u} + (\mathbf{r})\ \mathbf{t}$	$(\mathbf{t})\ \mathbf{0} \to_A \mathbf{0}$	$(\Lambda X.\mathbf{t})@U \to_A \mathbf{t}[U/X]$
	$\mathbf{t}+\mathbf{0} \to_A \mathbf{t}$	

$$\frac{}{\Gamma, x:U \vdash_A x:U}\, ax \qquad \frac{}{\Gamma \vdash_A \mathbf{0}:\overline{0}}\, ax_{\overline{0}} \qquad \frac{\Gamma \vdash_A \mathbf{t}:\sum_{i=1}^{n}(U \to T_i) \qquad \Gamma \vdash_A \mathbf{r}:m.U}{\Gamma \vdash_A (\mathbf{t})\ \mathbf{r}:\sum_{i=1}^{n} m.T_i}\, \to_E$$

$$\frac{\Gamma, x:U \vdash_A \mathbf{t}:T}{\Gamma \vdash_A \lambda x:U.\mathbf{t}:U \to T}\, \to_I \qquad \frac{\Gamma \vdash_A \mathbf{t}:\forall X.U}{\Gamma \vdash_A \mathbf{t}@V:U[V/X]}\, \forall_E$$

$$\frac{\Gamma \vdash_A \mathbf{t}:T \qquad \Gamma \vdash_A \mathbf{r}:R}{\Gamma \vdash_A \mathbf{t}+\mathbf{r}:T+R}\, +_I \qquad \frac{\Gamma \vdash_A \mathbf{t}:T \qquad T \equiv R}{\Gamma \vdash_A \mathbf{t}:R}\, \equiv \qquad \frac{\Gamma \vdash_A \mathbf{t}:U \qquad X \notin FV(\Gamma)}{\Gamma \vdash_A \Lambda X.\mathbf{t}:\forall X.U}\, \forall_I$$

Figure 6.3: The λ^{add} calculus with the *Additive* type system, in Church-style

We define $\downarrow_A$ to be the function that takes a term and returns its normal form in λ^{add}. The normal form always exists and is unique due to strong normalisation and confluence of the calculus (Corollaries 4.3.22 and 4.3.23 respectively). We write $\downarrow$ to the analogous function for λ^{CA}. We will not prove strong normalisation for it, however we conjecture that it is the case. We use post-fix notation for these functions, so $\downarrow(\mathbf{t})$ is written $\mathbf{t}\downarrow$.

Let T_c be the set of terms in the calculus c. Consider the following *abstraction* function $\sigma : T_{\lambda^{CA}} \to T_{\lambda^{\text{add}}}$ from terms in λ^{CA} to terms in λ^{add}:

$$\begin{array}{rclcrcl} \sigma(x:U) & = & x:U & & \sigma(\mathbf{t}@U) & = & \sigma(\mathbf{t})@U \\ \sigma(\lambda x:U.\mathbf{t}) & = & \lambda x:U.\sigma(\mathbf{t}) & & \sigma(\mathbf{0}) & = & \mathbf{0} \\ \sigma(\Lambda X.\mathbf{t}) & = & \Lambda X.\sigma(\mathbf{t}) & & \sigma(\alpha.\mathbf{t}) & = & \sum_{i=1}^{\lfloor\alpha\rfloor} \sigma(\mathbf{t}) \\ \sigma((\mathbf{t})\ \mathbf{t}') & = & (\sigma(\mathbf{t}))\ \sigma(\mathbf{t}') & & \sigma(\mathbf{t}+\mathbf{t}') & = & \sigma(\mathbf{t}) + \sigma(\mathbf{t}') \end{array}$$

where for any term $\mathbf{t}$, $\sum_{i=1}^{0} \mathbf{t} = \mathbf{0}$.

We can also define a *concretisation* function $\gamma : T_{\lambda^{\text{add}}} \to T_{\lambda^{CA}}$, which is the obvious embedding of terms: $\gamma(\mathbf{t}) = \mathbf{t}$.

Let $\sqsubseteq \subseteq T_{\lambda^{\text{add}}} \times T_{\lambda^{\text{add}}}$ be the least relation satisfying:

$$\begin{array}{rcl} & & \alpha \le \beta \ \Rightarrow\ \sum_{i=1}^{\alpha} \mathbf{t} \sqsubseteq \sum_{i=1}^{\beta} \mathbf{t} \\ \mathbf{t} \sqsubseteq \mathbf{t}' \ \Rightarrow\ \lambda x:U.\mathbf{t} \sqsubseteq \lambda x:U.\mathbf{t}' & & \mathbf{t} \sqsubseteq \mathbf{t}' \ \wedge\ \mathbf{r} \sqsubseteq \mathbf{r}' \ \Rightarrow\ (\mathbf{t})\ \mathbf{r} \sqsubseteq (\mathbf{t}')\ \mathbf{r}' \\ \mathbf{t} \sqsubseteq \mathbf{t}' \ \Rightarrow\ \Lambda X.\mathbf{t} \sqsubseteq \Lambda X.\mathbf{t}' & & \mathbf{t} \sqsubseteq \mathbf{t}' \ \wedge\ \mathbf{r} \sqsubseteq \mathbf{r}' \ \Rightarrow\ \mathbf{t}+\mathbf{r} \sqsubseteq \mathbf{t}'+\mathbf{r}' \\ \mathbf{t} \sqsubseteq \mathbf{t}' \ \Rightarrow\ \mathbf{t}@U \sqsubseteq \mathbf{t}'@U & & \mathbf{t} \sqsubseteq \mathbf{r} \ \wedge\ \mathbf{r} \sqsubseteq \mathbf{s} \ \Rightarrow\ \mathbf{t} \sqsubseteq \mathbf{s} \end{array}$$

and let $\lesssim$ be the relation defined by $\mathbf{t}_1 \lesssim \mathbf{t}_2 \Leftrightarrow \mathbf{t}_1\downarrow_A \sqsubseteq \mathbf{t}_2\downarrow_A$.

The relation $\sqsubseteq$ is a partial order. Also, $\lesssim$ is a partial order if we quotient terms by the relation $\sim$, defined by $\mathbf{t} \sim \mathbf{r}$ if and only if $\mathbf{t}\downarrow = \mathbf{r}\downarrow$. We formalise this in the following lemma (its proof is in Appendix E.4).

Lemma 6.4.1 (Poset)**.**

1. $\sqsubseteq$ *is a partial order relation*
2. $\lesssim$ *is a partial order relation in* $T_{\lambda^{add}}/\sim$. □

The following theorem states that the terms in λ^{CA} can be seen as a refinement of those in λ^{add}, *i.e.* we can consider λ^{add} as an abstract interpretation of λ^{CA}. It follows by a non trivial structural induction on $\mathbf{t} \in T_{\lambda^{CA}}$, *cf.* Appendix E.5.

Theorem 6.4.2 (Abstract interpretation)**.** *The function* $\downarrow$ *is a valid concretisation of* $\downarrow_A$*:* $\forall \mathbf{t} \in T_{\lambda^{CA}}$, $\sigma(\mathbf{t})\downarrow_A \lesssim \sigma(\mathbf{t}\downarrow)$. □

The following lemma states that the abstraction preserves the typing (proof in Appendix E.6).

Lemma 6.4.3 (Typing preservation)**.** *For arbitrary context* Γ*, term* $\mathbf{t}$ *and type* T*, if* $\Gamma \vdash \mathbf{t} : T$ *then* $\Gamma \vdash_A \sigma(\mathbf{t}) : T$. □

Taking λ^{add} as an abstract interpretation of λ^{CA} entails the extension of the interpretation of λ^{add} into System F with pairs, F_p (*cf.* Section 4.3) as an abstract interpretation of λ^{CA}, as depicted in Figure 6.4.

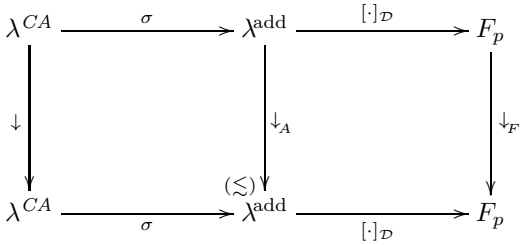


Figure 6.4: Abstract interpretation of λ^{CA} into System F_P

6.5 Conclusions and open questions

The first typed version of λ_{lin} was presented in Chapter 3. This type system, an extension of System F, deals with scalars in the terms by reflecting them in the types. However, this introduces an undesirable restriction in the calculus: it can only sum terms of the same type, up to the scalar weighing it. With λ^{CA}, which is also based on System F, we have shown how to design a typed algebraic calculus that combines polymorphism and sums of different types, thereby lifting this restriction. Sums of types can be encoded as pairs, as shown in Chapter 4, thus these constructions are quite standard.

λ^{CA} is a confluent, typed, strongly normalising, algebraic λ-calculus, based on λ_{lin}, which has an algebraic rewrite system without restrictions, in contrast with λ_{alg}, which is presented as an equational theory instead of a rewrite system; and λ_{lin}, where restrictions are introduced to make the calculus to make it confluent.

In this chapter, scalars are approximated by natural numbers. This approximation yields a subject reduction property which is exact about the types involved in a term, but only approximate in their "amount" or "weight". In addition, the approximation is a lower bound: if a term has a type that is a sum of some amount of different types, then after reducing it these amounts can be incremented but never decremented.

One of the original motivations for this system was to extend the translation of λ_{lin} to a standard calculus. The translation of the additive fragment of λ_{lin} (λ^{add}) into System F with pairs (F_p) was first presented in Chapter 4. We have shown that terms in λ^{add} can be seen as an abstract interpretation of terms in λ^{CA}, and then that F_p can also be used as an abstract interpretation of terms in λ^{CA} by using the translation from λ^{add} into F_p.

In our calculus, we have chosen to take the floor of the scalars to approximate types. However, this decision is arbitrary, and we could have chosen to approximate types using the ceiling instead. Therefore, an obvious extension of this system is to take both floor and ceiling of scalars to produce type intervals, thus obtaining more accurate approximations.

It is still an open question how to obtain a similar result for a calculus where scalars belong to an arbitrary ring.

Chapter 7

Lineal: a vectorial type system in Church style

Résumé du Chapitre

Nous définissons un lambda-calcul algébrique explicitement typé, fondé sur λ_{lin} et Vectorial, qui a la propriété de préservation du type et est fortement normalisable. Le langage permet de faire des combinaisons linéaires arbitraires de termes. Le système de type est un outil d'analyse décrivant les propriétés vectorielles des termes : il garde la trace de l'« amplitude d'un terme », c-à-d que si $\mathbf{t}$ *et* $\mathbf{u}$ *sont tous deux du même type* U *alors* $\alpha.\mathbf{t} + \beta.\mathbf{u}$ *est de type* $(\alpha + \beta).U$*. En outre, il garde la trace de la « direction d'un terme », c-à-d que si* $\mathbf{t}$ *et* $\mathbf{u}$ *sont de type* U *et* V *respectivement alors* $\alpha.\mathbf{t} + \beta.\mathbf{u}$ *est de type* $\alpha.U + \beta.V$*. Ce calcul est capable d'encoder des matrices et des vecteurs comme Vectorial, mais il a l'avantage d'avoir la propriété de préservation du type : les problèmes décrits dans le Chapitre 5 sont résolus ici à l'aide de types explicites dans les termes, ainsi qu'un système de sous-typage.* ∎

This chapter presents *Lineal*, a type system that solves the drawbacks of the previous type systems presented in this thesis. Recall that a straightforward extension of System F (*cf.* Definition 3.3.1) would have two problems. On the one hand, it is too restrictive because of rule $+^{\triangleleft}_I$, which allows adding only terms with the exact same type. And, on the other hand, it does not provide any information about the scalars, as it follows from rule $s^{\triangleleft}_I$; *i.e.* any typed term can be weighted and preserves the same type.

The *Scalar* type system presented in Chapter 3 is a novel approach to solve the second problem: by allowing scalars to weight types, the scalars in the terms are tracked in the types. However, it still does not solve the restriction of adding only terms of the same type (up to the scalar).

The *Additive* type system in Chapter 4 solves this problem, but it comes at the price of reintroducing a restriction: it types only the additive fragment of λ_{lin}. It has however the advantage of being more classic, in the sense that it can be interpreted in System F via a parallel between additions and pairs (*cf.* Section 4.3).

Finally, the *Vectorial* type system in Chapter 5 solves the previous two problems by combining the two approaches: types can be weighted and added, revealing the vectorial shape of the term. It has been shown however that the factorisation reduction rules prevent this system from preserving the types

during the reduction: just a weak version of the subject reduction property can be proven (*cf.* Section 5.3). We have suggested that a Church style presentation would solve this problem. This Chapter develops precisely this idea: a Church style *Vectorial* type system for the $\lambda_{lin}$with both strong normalisation and subject reduction. We call this explicitly typed calculus *Lineal*.

In Chapter 6 we had presented an alternative approach: λ^{CA}, a typed calculus where terms can be weighted and summed up without restrictions. It does not have scalars in the types, only sums (which have been proved to be interpretable with pairs). It has been developed in Church style, so it does not have the factorisation rules problem. Despite the fact that this is a simpler approach, it does only serve a purpose: that of providing the system with strong normalisation, and this without restrictions on the calculus other than taking scalars over positive real numbers. Notwithstanding, the information about scalars given by the type system itself is only a lower bound approximation. In addition, it adds a new restriction: scalars can only range over non-negative real numbers, in comparison to λ_{lin}, where any commutative ring is allowed.

Plan of the Chapter. Section 7.1 presents the typed calculus and discusses some design choices. The next two sections, 7.2 and 7.3, are devoted to the correctness of this calculus by showing subject reduction and strong normalisation, respectively. In Section 7.4, we show how to encode the Hadamard gate in this setting. Finally, Section 7.5 concludes.

7.1 The calculus *Lineal*

We introduce the calculus *Lineal*, an explicitly typed algebraic λ-calculus based on λ_{lin} and *Vectorial*. First, we will need a specific notation for arrays. Notice that what was written as $\forall\overrightarrow{X}$ in *Vectorial* will not be enough in this calculus. In order to eliminate $\forall$'s we now need to know how many X's are in $\overrightarrow{X}$. Hence we introduce the following vectorial notation.

The vectorial notation. As usual we write $\sum_{i=1}^{n} T_i$ for $T_1 + \cdots + T_n$. Also, since the terms are explicitly typed, we may write $\mathbf{b}^V$ to for term $\mathbf{b}$ when it has type V. Finally, we write $\mathbf{t}@U_1$ as a shorthand for $\mathbf{t}@(\sum_{i=1}^{1} U_i)$.

The *vectorial notation* $\langle\cdot\rangle_n$ is defined as follows:

$$\langle\forall X\rangle_k.\mathbf{t} = \forall X_1.\ldots.\forall X_k.\mathbf{t} \qquad (\mathbf{t})\ \langle\mathbf{b}\rangle_n = (\cdots(\mathbf{t})\ \mathbf{b}_1)\cdots\ \mathbf{b}_k$$

$$\langle\lambda x : U\rangle_k.\mathbf{t} = \lambda x_1 : U_1.\ldots.\lambda x_k : U_k.\mathbf{t} \qquad (\mathbf{t})\ \langle\mathbf{b}^V\rangle_k = (\cdots(\mathbf{t})\ \mathbf{b}_1^{V_1})\cdots\ \mathbf{b}_k^{V_k}$$

$$\langle\Lambda X\rangle_k.\mathbf{t} = \Lambda X_1.\ldots.\Lambda X_k.\mathbf{t} \qquad \mathbf{t}\langle[U_j/X]\rangle_k = \mathbf{t}[U_{j_1}/X_1]\cdots[U_{j_k}/X_k]$$

$$\mathbf{t}@\langle\sum_{i=1}^{n} U_i\rangle_k = (\cdots(\mathbf{t})@(\sum_{i=1}^{n_1} U_{i_1}))\cdots@(\sum_{i=1}^{n_k} U_{i_k}) \qquad \langle U\rangle_k \to T = U_1 \to \cdots \to U_k \to T$$

$$\langle\Gamma \vdash \mathbf{t} : T\rangle_k = \Gamma_1 \vdash \mathbf{t}_1 : T_1 \quad \cdots \quad \Gamma_k \vdash \mathbf{t}_k : T_k$$

Where

- The index k may be 0, in which case the term/type can be removed, *e.g.* $\langle\forall X\rangle_0.T@\langle[U/X]\rangle_0 = T$.
- We use $\langle\sum_{i=1}^{m+\delta} U_i\rangle_k$ for $\sum_{i=1}^{m+\delta_1} U_{i_1}, \ldots, \sum_{i=1}^{m+\delta_k} U_{i_k}$.
- We use it also to enumerate elements: $\langle e\rangle_n = e_1, \ldots, e_n$, where e may be a term, a type, a scalar, or any other element.

Figure 7.1 shows the abstract syntax of types and terms and their reduction rules. It includes all the rules from λ^{CA} (the only explicitly typed system presented so far), plus three new rules grouped in two categories: the *Type-linearity rules* and the *Type-distributivity rule*. In order to understand why they are needed, notice the differences in the syntax between what is presented here and λ^{CA} (*cf.* Figure 6.1). The terms are the same, except for the type application, where it can be read $\mathbf{t}@(\sum_{i=1}^{n} U_i)$ instead of $\mathbf{t}@U$. This sum of unit types acts as a choice operator: in $\mathbf{t}@U$ the type U is the argument for a lambda abstraction of types, in $\mathbf{t}@(\sum_{i=1}^{n} U_i)$ we are saying that one of the U_i will be the argument, without saying *a priori* which one.

This change comes from the fact that the type system is based on *Vectorial*, where the arrow elimination $\to_E$ (*cf.* Figure 5.2) is more versatile than the one in λ^{CA}: the arguments passed to an abstraction can be a sum of any terms, as soon as the abstraction is able to take any of these terms by means of polymorphism. As a consequence of this extension, the rule becomes a mix between an arrow elimination and a forall elimination. However, a forall elimination in a system with explicit types needs a type application. Then, roughly speaking, the sum of U_i's represents all the types that will be applied to the type application in order to specialise it to each of its arguments.

Having said this, let us analyse one by one the reduction rules concerning types. The beta-reduction is standard: when the given type is just one unit type, and the term is an abstraction of type, a type substitution occurs. The type-distributivity rule, despite its complicated aspect, is simply taking into account the above discussion: if the term is a type application, it has to take into account the arguments in order to choose the correct type, and hence we must have a condition saying that the arguments can be transformed into the types expected by the abstractions. The type of the arguments has been written as superscripts V_i, since they are known because they are provided by the terms. Notice the vectorial notation, which is used to allow more than one type-applications.

Example 7.1.1. *The type-distributivity rule: let U_1, U_2, U_3, such that $U_i[W_{1_1}/X_1][W_{3_2}/X_2] = V_i$, then*

$$(((((\Lambda X_1.\Lambda X_2.\lambda x_1 : U_1.\lambda x_2 : U_2.\lambda x_3 : U_3.\mathbf{t})@(W_{1_1} + W_{2_1}))@(W_{1_2} + W_{2_2} + W_{3_2}))\ \mathbf{b}_1^{V_1})\ \mathbf{b}_2^{V_2})\ \mathbf{b}_3^{V_3}$$

reduces to

$$(((\lambda x_1 : V_1.\lambda x_2 : V_2.\lambda x_3 : V_3.\mathbf{t}[W_{1_1}/X_1][W_{3_2}/X_2])\ \mathbf{b}_1)\ \mathbf{b}_2)\ \mathbf{b}_3$$

which, using the vectorial notation, can be written as

$$((\langle\Lambda X\rangle_2.\langle\lambda x : U\rangle_3.\mathbf{t})@\langle\sum_{i=1}^{m} W_i\rangle_2)\ \langle\mathbf{b}^V\rangle_3 \to (\langle\lambda x : V\rangle_3.\mathbf{t}\langle[W_j/X]\rangle_2)\ \langle\mathbf{b}\rangle_3$$

where $m_1 = 2$, $m_2 = 3$, $j_1 = 1$ and $j_2 = 3$.

The type-linearity rules are needed to allow for the type application to be linear with respect to sums and scalars. Notice however that they were not needed in λ^{CA}. This change comes from the difference between the forall typing rules in *Vectorial* and in λ^{CA}.

As usual, we define type equivalences, which are the same as in *Vectorial*, with one extra equivalence for the type application.

$$\begin{array}{rclcrclcrcl} 1.T & \equiv & T, & & \alpha.T + \alpha.R & \equiv & \alpha.(T+R), & & T+R & \equiv & R+T, \\ \alpha.(\beta.T) & \equiv & (\alpha\times\beta).T, & & \alpha.T + \beta.T & \equiv & (\alpha+\beta).T, & & T+(R+S) & \equiv & (T+R)+S, \\ & & & & (\forall X.U)@V & \equiv & U[V/X], & & & & \end{array}$$

Similarly to *Vectorial*, every type is a linear combination of unit types.

Types:

$$T, R, S ::= U \mid \alpha.T \mid T + R$$

$$U, V, W ::= X \mid U \to T \mid \forall X.U \mid U@(\sum_{i=1}^{n} V_i)$$

Terms:

$$\mathbf{t}, \mathbf{r}, \mathbf{u} ::= \mathbf{b} \mid (\mathbf{t})\ \mathbf{r} \mid \mathbf{t}@(\sum_{i=1}^{n} U_i) \mid \mathbf{0} \mid \alpha.\mathbf{t} \mid \mathbf{t} + \mathbf{r} \mid \Lambda X.\mathbf{t}$$

$$\mathbf{b} ::= x : U \mid \lambda x{:}U.\mathbf{t} \mid \Lambda X.\mathbf{b}$$

Contexts

Contexts are defined as sets of pairs $x : C$, where C is taken from the following grammar

$$C ::= X \mid C \to T \mid \forall X.C$$

Reduction rules

Elementary rules:

$0.\mathbf{t} \to \mathbf{0}$,
$1.\mathbf{t} \to \mathbf{t}$,
$\alpha.\mathbf{0} \to \mathbf{0}$,
$\alpha.(\beta.\mathbf{t}) \to (\alpha \times \beta).\mathbf{t}$,
$\alpha.(\mathbf{t}+\mathbf{r}) \to \alpha.\mathbf{t}+\alpha.\mathbf{r}$,
$\mathbf{t} + \mathbf{0} \to \mathbf{t}$.

Factorisation rules:

$\alpha.\mathbf{t} + \beta.\mathbf{t} \to (\alpha + \beta).\mathbf{t}$,
$\alpha.\mathbf{t} + \mathbf{t} \to (\alpha + 1).\mathbf{t}$,
$\mathbf{t} + \mathbf{t} \to (1 + 1).\mathbf{t}$.

Beta reductions:

$(\lambda x : U.\mathbf{t})\ \mathbf{b} \to \mathbf{t}[\mathbf{b}/x]$,
$(\Lambda X.\mathbf{t})@U \to \mathbf{t}[U/X]$.

Application rules:

$(\mathbf{t} + \mathbf{r})\ \mathbf{u} \to (\mathbf{t})\ \mathbf{u} + (\mathbf{r})\ \mathbf{u}$,
$(\mathbf{u})\ (\mathbf{t} + \mathbf{r}) \to (\mathbf{u})\ \mathbf{t} + (\mathbf{u})\ \mathbf{r}$,
$(\alpha.\mathbf{t})\ \mathbf{r} \to \alpha.(\mathbf{t})\ \mathbf{r}$,
$(\mathbf{r})\ (\alpha.\mathbf{t}) \to \alpha.(\mathbf{r})\ \mathbf{t}$,
$(\mathbf{0})\ \mathbf{t} \to \mathbf{0}$,
$(\mathbf{t})\ \mathbf{0} \to \mathbf{0}$.

Type-linearity rules:

$(\alpha.\mathbf{t})@(\sum_{i=1}^{n} U_i) \to \alpha.\mathbf{t}@(\sum_{i=1}^{n} U_i)$,
$(\mathbf{t}+\mathbf{r})@(\sum_{i=1}^{n} U_i) \to \mathbf{t}@(\sum_{i=1}^{n} U_i)+\mathbf{r}@(\sum_{i=1}^{n} U_i)$.

Type-distributivity rule:

If $\forall h = 1, \ldots, n$, $U_h\langle[W_j/X]\rangle_k = V_h$, then
$((\langle\Lambda X\rangle_k.\langle\lambda x : U\rangle_n.\mathbf{t})@\langle\sum_{i=1}^{m} W_i\rangle_k)\ \langle\mathbf{b}^V\rangle_n \to (\langle\lambda x : V\rangle_n.\mathbf{t}\langle[W_j/X]\rangle_k)\ \langle\mathbf{b}\rangle_n$

Contextual rules: If $\mathbf{t} \to \mathbf{r}$, then for any term $\mathbf{u}$, scalar α and variable x,

$(\mathbf{t})\ \mathbf{u} \to (\mathbf{r})\ \mathbf{u}$, $\mathbf{t} + \mathbf{u} \to \mathbf{r} + \mathbf{u}$, $\alpha.\mathbf{t} \to \alpha.\mathbf{r}$,
$(\mathbf{u})\ \mathbf{t} \to (\mathbf{u})\ \mathbf{r}$, $\mathbf{u} + \mathbf{t} \to \mathbf{u} + \mathbf{r}$, $\lambda x.\mathbf{t} \to \lambda x.\mathbf{r}$,
$\Lambda X.\mathbf{t} \to \Lambda X.\mathbf{r}$, $\mathbf{t}@(\sum_i U_i) \to \mathbf{r}@(\sum_i U_i)$.

Figure 7.1: Syntax and reduction rules of *Lineal*

Lemma 7.1.2 (Type characterisation)**.** *For any type T, $\exists\langle U\rangle_n$, $\langle\alpha\rangle_n$ such that $T \equiv \sum_{i=1}^{n} \alpha_i.U_i$.*

Proof. Structural induction on T. If T is a unit type, then take $n = 1$ and $\alpha_i = 1$. If T is $\alpha.T'$, then by the induction hypothesis $T' \equiv \sum_{i=1}^{n} \beta_i.U_i$, so $\alpha.T' \equiv \alpha.\sum_{i=1}^{n} \beta_i.U_i \equiv \sum_{i=1}^{n}(\alpha \times \beta_i).U_i$. If $T = R + S$, then it follows directly by the induction hypothesis. □

Figure 7.2 presents a subtyping relation and the typing rules. The subtyping relation takes care of the problem with 0's in the types (*cf.* discussion in Section 5.2): $T \preceq T + 0.R$ for any R (*cf.* rule *Ax*). The rest of the rules are the reflexivity, transitivity and contextual closure of the relation. Notice that, roughly speaking, the subtyping relation only adds types weighted with 0, but it does not change the structure of the type, in the sense of the following lemma.

Lemma 7.1.3 (Shapes comparison)**.**

If $\sum_{i=1}^{n} \alpha_i.U_i \preceq \sum_{j=1}^{m} \beta_j.\langle \forall X \rangle_k.V_j$*, then* $\forall i, \exists U_i' \;/\; U_i \equiv \langle \forall X \rangle_k.U_i'$.

If $\sum_{i=1}^{n} \alpha_i.U_i \preceq \sum_{j=1}^{m} \beta_j.V_j @ \langle \sum_{o=1}^{p} W_k \rangle_k$*, then* $\forall i, \exists U_i' \;/\; U_i \equiv U_i' @ \langle \sum_{o=1}^{p} W_k \rangle_k$.

If $\sum_{i=1}^{n} \alpha_i.U_i \preceq \sum_{j=1}^{m} \beta_j.V \to T_j$*, then* $\forall i, \exists T_i' \;/\; U_i \equiv V \to T_i'$.

Proof. Notice that in all the equivalences, the types on the left hand side of the equivalence appear also on the right hand side. This is the case also in the rule Ax, hence it is always the case that all the U_i are in the type on the right hand side. Since all the types on the right hand side have the same form, then U_i has to have this form too. □

All the typing rules are the explicitly typed analogues of those in *Vectorial*, with only one change: the $\forall_E$ is replaced by a $@_I$, which is more general since it allows introducing a sum of unit types for replacement. The forall elimination rule is a special case, when $m = 1$, which follows from the extra equivalence rule mentioned above.

$$\dfrac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i}{\Gamma \vdash \mathbf{t}@U : \sum_{i=1}^{n} \alpha_i.U_i[V/X]}\,\forall_E \qquad = \qquad \dfrac{\dfrac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i}{\Gamma \vdash \mathbf{t}@U : \sum_{i=1}^{n} \alpha_i.(\forall X.U_i)@V}\,@_I}{\Gamma \vdash \mathbf{t}@U : \sum_{i=1}^{n} \alpha_i.U_i[V/X]}\,\preceq$$

In spite of being written differently, the arrow elimination ($\to_E$) is in fact completely analogous to the one in *Vectorial*. We just changed the notation $\forall \overrightarrow{X}$ for the vectorial notation, and added the type application @ to instantiate the X's.

Example 7.1.4. *We show how to type the term from Example 7.1.1 (we use superscripts to denote several application of the same rule):*

$$\dfrac{\dfrac{\dfrac{\dfrac{x_1 : U_1, x_2 : U_2, x_3 : U_3 \vdash \mathbf{t} : T}{\vdash \langle \lambda x : U \rangle_3.\mathbf{t} : \langle U \rangle_3 \to T}\,{\to_I}^3}{\vdash \langle \Lambda X \rangle_2.\langle \lambda x : U \rangle_3.\mathbf{t} : \langle \forall X \rangle_2.\langle U \rangle_3 \to T}\,{\forall_I}^2}{(\langle \Lambda X \rangle_2.\langle \lambda x : U \rangle_3.\mathbf{t})@\langle \sum_{i=1}^{m} W_i \rangle_2 : (\langle \forall X \rangle_2.\langle U \rangle_3 \to T)@\langle \sum_{i=1}^{m} W_i \rangle_2}\,{@_I}^2 \qquad \vdash \mathbf{b}_1 : V_1}{\vdash ((\langle \Lambda X \rangle_2.\langle \lambda x : U \rangle_3.\mathbf{t})@\langle \sum_{i=1}^{m} W_i \rangle_2)\ \mathbf{b}_1 : V_2 \to V_3 \to T\langle [W_j/X] \rangle_2 \qquad \vdash \mathbf{b}_2 : V_2 \quad \vdash \mathbf{b}_3 : V_3}\,{\to_E}^2$$

$$\vdash ((\langle \Lambda X \rangle_2.\langle \lambda x : U \rangle_3.\mathbf{t})@\langle \sum_{i=1}^{m} W_i \rangle_2)\ \langle \mathbf{b} \rangle_3 : T\langle [W_j/X] \rangle_2$$

Notice that $((\langle \Lambda X \rangle_2.\langle \lambda x : U \rangle_3.\mathbf{t})@\langle \sum_{i=1}^{m} W_i \rangle_2)\ \langle \mathbf{b}^V \rangle_3 \to^* \mathbf{t}\langle [W_j/X] \rangle_2 \langle [\mathbf{b}/x] \rangle_3$ *And since* $x_1 : U_1, x_2 : U_2, x_3 : U_3 \vdash \mathbf{t} : T$*, by the Substitution Lemma (*cf. *Lemma 7.2.5 in Section 7.2),*

$$x_1 : V_1, x_2 : V_2, x_3 : V_3 \vdash \mathbf{t}\langle [W_j/X] \rangle_2 : T\langle [W_j/X] \rangle_2$$

Finally, using the Substitution Lemma again (three times), we get

$$\vdash \mathbf{t}\langle [W_j/X] \rangle_2 \langle [\mathbf{b}/x] \rangle_3 : T\langle [W_j/X] \rangle_2$$

Subtyping relation

$$\frac{}{T \preceq T + 0.R}\,Ax \qquad \frac{T \equiv R}{T \preceq R}\,Re \qquad \frac{T \preceq R \quad R \preceq S}{T \preceq S}\,Tr \qquad \frac{T \preceq R \quad V \preceq U}{U \to T \preceq V \to R}\,Cx_{\to}$$

$$\frac{T \preceq R}{T + S \preceq R + S}\,Cx_{+} \qquad \frac{U \preceq V}{\forall X.U \preceq \forall X.V}\,Cx_{\forall} \qquad \frac{U \preceq V}{U@(\sum_{i=1}^{n} W_i) \preceq V@(\sum_{i=1}^{n} W_i)}\,Cx_{@} \qquad \frac{T \preceq R}{\alpha.T \preceq \alpha.R}\,Cx_{s}$$

Typing rules

$$\frac{}{\Gamma, x\colon U \vdash x\colon U}\,ax \qquad \frac{\Gamma \vdash \mathbf{t}\colon T}{\Gamma \vdash \mathbf{0}\colon 0.T}\,0_I \qquad \frac{\Gamma, x\colon U \vdash \mathbf{t}\colon T}{\Gamma \vdash \lambda x\colon U.\mathbf{t}\colon U \to T}\,{\to_I}$$

$$\frac{\Gamma \vdash \mathbf{t}\colon \sum_{i=1}^{n} \alpha_i.(\langle \forall X \rangle_k.(U \to T_i))@\langle \sum_{j=1}^{m+\delta} W_j \rangle_k \qquad \Gamma \vdash \mathbf{r}\colon \sum_{j=1}^{m} \beta_j.V_j \qquad \begin{array}{l}\forall V_j, \exists j_1, \ldots, j_k \,/\\ U\langle [W_j/X] \rangle_k = V_j\end{array}}{\Gamma \vdash (\mathbf{t})\ \mathbf{r}\colon \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \times \beta_j.T_i\langle [W_j/X] \rangle_k}\,{\to_E}$$

$$\frac{\Gamma \vdash \mathbf{t}\colon \sum_{i=1}^{n} \alpha_i.U_i \quad X \notin FV(\Gamma)}{\Gamma \vdash \Lambda X.\mathbf{t}\colon \sum_{i=1}^{n} \alpha_i.\forall X.U_i}\,\forall_I \qquad \frac{\Gamma \vdash \mathbf{t}\colon \sum_{i=1}^{n} \alpha_i.\forall X.U_i}{\Gamma \vdash \mathbf{t}@(\sum_{j=1}^{m} V_j)\colon \sum_{i=1}^{n} \alpha_i.(\forall X.U_i)@(\sum_{j=1}^{m} V_j)}\,@_I$$

$$\frac{\Gamma \vdash \mathbf{t}\colon T}{\Gamma \vdash \alpha.\mathbf{t}\colon \alpha.T}\,s_I \qquad \frac{\Gamma \vdash \mathbf{t}\colon T \qquad \Gamma \vdash \mathbf{r}\colon R}{\Gamma \vdash \mathbf{t} + \mathbf{r}\colon T + R}\,{+_I} \qquad \frac{\Gamma \vdash \mathbf{t}\colon T \qquad T \preceq R}{\Gamma \vdash \mathbf{t}\colon R}\,\preceq$$

Figure 7.2: Typing rules of *Lineal*

7.2 Subject reduction

The Church style presentation plus the subtyping relation provides this calculus with an exact subject reduction theorem, in contrast with what happened in *Vectorial* (*cf.* Section 5.3) and λ^{CA} (*cf.* Section 6.2).

Theorem 7.2.1 (Subject reduction)**.** *For any terms* $\mathbf{t}$, $\mathbf{t}'$, *any context* Γ *and any type* T , *if* $\mathbf{t} \to \mathbf{t}'$ *then* $\Gamma \vdash \mathbf{t}\colon T \Rightarrow \Gamma \vdash \mathbf{t}'\colon T$.

In order to prove this Theorem, we need some previous results. The Generation Lemmas, which are straightforward in a Church style setting, can be stated as follows:

Lemma 7.2.2 (Generation lemmas)**.** *Let* T *be a type,* U *a unit type,* α *a scalar,* Γ *a typing context and* $\mathbf{t}$ *and* $\mathbf{r}$ *terms.*

1. *If* $\Gamma \vdash x\colon T$, *then there exists a type* U *and context* Δ *such that* $\Gamma = \Delta \cup \{x : U\}$, *and* $U \preceq T$.
2. *If* $\Gamma \vdash \mathbf{0}\colon T$, *then there exists a type* R *and a term* $\mathbf{t}$ *such that* $\Gamma \vdash \mathbf{t}\colon R$ *and* $0.R \preceq T$.

3. *If* $\Gamma \vdash (\mathbf{t})\ \mathbf{r}:T$, *then* $\exists n,\ m,\ k,\ \langle\delta\rangle_k,\ \langle X\rangle_k,\ \langle\alpha\rangle_n,\ \langle\beta\rangle_m,\ U,\ \langle T\rangle_n,\ \langle V\rangle_m,\ \langle\langle W\rangle_{m+\delta}\rangle_m$, *where* $\forall V_j$, $\exists j_1,\dots,j_k\ /\ U\langle[W_j/X]\rangle_k = V_j$, *and such that* $\Gamma \vdash \mathbf{t}:\sum_{i=1}^{n}\alpha_i.(\langle\forall X\rangle_k.(U \to T_i))@\langle\sum_{j=1}^{m+\delta}W_j\rangle_k$, $\Gamma \vdash \mathbf{r}:\sum_{j=1}^{m}\beta_j.V_j$ *and* $\sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_i\times\beta_j.T_i\langle[W_j/X]\rangle_k \preceq T$.

4. *If* $\Gamma \vdash \lambda x:U.\mathbf{t}:T$, *then there exists a type* R *such that* $\Gamma, x:U \vdash \mathbf{t}:R$ *and* $U \to R \preceq T$.

5. *If* $\Gamma \vdash \mathbf{t}+\mathbf{r}:T$, *then there exist types* R *and* S *such that* $\Gamma \vdash \mathbf{t}:R$, $\Gamma \vdash \mathbf{r}:S$ *and* $R+S \preceq T$.

6. *If* $\Gamma \vdash \alpha.\mathbf{t}:T$, *then there exists a type* R *such that* $\Gamma \vdash \mathbf{t}:R$ *and* $\alpha.R \preceq T$.

7. *If* $\Gamma \vdash \Lambda X.\mathbf{t}:T$, *then* $X \notin FV(\Gamma)$ *and there exists types* $\langle U\rangle_n$ *and scalars* $\langle\alpha\rangle_n$ *such that* $\Gamma \vdash \mathbf{t}:\sum_{i=1}^{n}\alpha_i.U_i$ *and* $\sum_{i=1}^{n}\alpha_i.\forall X.U \preceq T$.

8. *If* $\Gamma \vdash \mathbf{t}@(\sum_{j=1}^{m}V_j):T$, *then there exists types* $\langle U\rangle_n$, *a variable* X *and scalars* $\langle\alpha\rangle_n$ *such that* $\Gamma \vdash \mathbf{t}:\sum_{i=1}^{n}\alpha_i.\forall X.U_i$ *and* $\sum_{i=1}^{n}\alpha_i.(\forall X.U_i)@(\sum_{j=1}^{m}V_j) \preceq T$.

Proof. All the proofs follow by induction on the length of the derivation. There are only two cases for each item: the trivial case and having $\preceq$ as the final rule, which is also trivial using the induction hypothesis and rule $\preceq$. □

As usual, base terms have unit types, or any type bigger than a unit type, *i.e.* a unit type with extra 0's. (proof in Appendix F.1).

Corollary 7.2.3 (Base terms)**.** *If* $\Gamma \vdash \mathbf{b}:T$, *then exists* V *such that* $\Gamma \vdash \mathbf{b}:V$ *and* $V \preceq T$. □

Since the types are explicitly written in the terms, there are two ways of obtaining a type for a term: by construction, which gives the type written in the term, and by subtyping. This fact allows us to consider one type as principal (namely the one written in the term).

Lemma 7.2.4 (Principal types)**.** *Let* $\Gamma \vdash \mathbf{t}:T$ *and* $\Gamma \vdash \mathbf{t}:R$, *then* $\exists S$ *such that* $\Gamma \vdash \mathbf{t}:S$, $S \preceq T$ *and* $S \preceq R$.

Proof. Induction on the depth of the derivation of $\Gamma \vdash \mathbf{t}:T$. For each item there are two ways to obtain its type: the trivial one, and by rule $\preceq$, which proves the lemma. □

The substitution lemma is analogous to that in all the previous systems.

Lemma 7.2.5 (Substitution Lemma)**.** *For any term* $\mathbf{t}$, *base term* $\mathbf{b}$, *context* Γ *and types* U *and* T,

1. *If* $\Gamma \vdash \mathbf{t}:T$, *then* $\Gamma[U/X] \vdash \mathbf{t}[U/X]:T[U/X]$.

2. *If* $\Gamma, x:U \vdash \mathbf{t}:T$ *and* $\Gamma \vdash \mathbf{b}:U$, *then* $\Gamma \vdash \mathbf{t}[\mathbf{b}/x]:T$.

Proof. We reuse the proof of Lemma 5.3.10 (*cf.* Appendix D.6) with minimum changes. □

Finally, using the above lemmas we can prove Subject reduction. The proof follows from a rule by rule analysis and is rather long and technical, so we give it to Appendix F.2.

7.3 Strong normalisation

We prove this result by translating typed terms to typed terms in *Vectorial*. To avoid any ambiguity we use $\to_v$ to refer to reductions in *Vectorial* and $\vdash_v$ for its type derivations. We also write $\to_v^=$ for the reflexivity closure of $\to_v$, *i.e.* $\mathbf{t} \to_v^= \mathbf{r}$, if either $\mathbf{t} \to_v \mathbf{r}$ or $\mathbf{t} = \mathbf{r}$.

Let $\|\cdot\|$ be the following translation from terms in *Lineal* to terms in *Vectorial*:

$$\begin{array}{rcl} \|x : U\| & = & x \\ \|\lambda x : U.\mathbf{t}\| & = & \lambda x.\|\mathbf{t}\| \\ \|\Lambda X.\mathbf{t}\| & = & \|\mathbf{t}\| \\ \|(\mathbf{t})\ \mathbf{r}\| & = & (\|\mathbf{t}\|)\ \|\mathbf{r}\| \end{array} \qquad \begin{array}{rcl} \|\mathbf{t}@(\sum_{i=1}^n U_i)\| & = & \|\mathbf{t}\| \\ \|\mathbf{0}\| & = & \mathbf{0} \\ \|\alpha.\mathbf{t}\| & = & \alpha.\|\mathbf{t}\| \\ \|\mathbf{t}+\mathbf{r}\| & = & \|\mathbf{t}\| + \|\mathbf{r}\| \end{array}$$

We also define a translation from types in *Lineal* to types in *Vectorial* as follows:

$$\begin{array}{rcl} \|X\| & = & X \\ \|U \to T\| & = & \|U\| \to \|T\| \\ \|\forall X.U\| & = & \forall X.\|U\| \\ \|(\forall X.U)@V\| & = & \|U[V/X]\| \end{array} \qquad \begin{array}{rcll} \|U@(\sum_{i=1}^n V_i)\| & = & \|U\| & \text{If } U \not\equiv \forall X.W \text{ or } n > 1 \\ \|\alpha.T\| & = & \alpha.\|T\| & \\ \|T+R\| & = & \|T\| + \|R\| & \end{array}$$

Finally, we extend this definition to contexts as $\|\Gamma\| = \{x : \|U\| \mid x : U \in \Gamma\}$.

Notice that the translation of $U@V$ when $U \neq \forall X.W$ is not defined, however this type has not inhabits since it cannot be introduced by the context (it is not in the definition of contexts), nor by a typing rule.

This translation is stable under equivalence (Lemma 7.3.1), under substitution (Lemma 7.3.2), and neutral with respect to type substitution in terms (Lemma 7.3.3):

Lemma 7.3.1. *If $T \equiv R$, then $\|T\| \equiv \|R\|$.*

Proof. Case by case analysis. *cf.* Appendix F.3. □

Lemma 7.3.2.

1. $\|\mathbf{t}[\mathbf{b}/x]\| = \|\mathbf{t}\|[\|\mathbf{b}\|/x]$.
2. $\|T[W/X]\| \equiv \|T\|[\|W\|/X]$, *if* $T \neq U@V$ *with* $U \neq \forall X.W$.

Proof. Structural induction over $\mathbf{t}$ for the first case and over T for the second. *cf.* Appendix F.4. □

Lemma 7.3.3. $\|\mathbf{t}[U/X]\| = \|\mathbf{t}\|$.

Proof. Trivial since the translation removes all the types from the terms. □

The following lemma says that this translation preserves the reducibility, and the reductions are done in the same amount of steps before and after the translation, except in one case where both terms translate to the same. The proof is done by rule by rule analysis and can be found in Appendix F.5.

Lemma 7.3.4 (Reducibility preservation)**.** *If $\mathbf{t} \to \mathbf{r}$, then $\|\mathbf{t}\| \to_v^= \|\mathbf{r}\|$. Moreover, if the reduction $\mathbf{t} \to \mathbf{r}$ is not the type application beta-reduction, the type-distributivity rule nor the type linearity rules, then $\|\mathbf{t}\| \to_v \|\mathbf{r}\|$.* □

Another important lemma is the typability preservation: the translation of a typed term has a type which entails that the translation will be strongly normalising. The proof is done by induction on the last rule applied to derive the type. *cf.* Appendix F.6

Lemma 7.3.5 (Typability preservation)**.** *If* $\Gamma \vdash \mathbf{t} : T$*, then* $\exists R \preceq T$ *such that* $\|\Gamma\| \vdash_v \|\mathbf{t}\| : \|R\|$. □

Then, using the two previous lemmas we can state the strong normalisation of *Lineal*.

Theorem 7.3.6 (Strong normalisation)**.** *If* $\Gamma \vdash \mathbf{t} : T$ *is derivable in Lineal, then* $\mathbf{t}$ *is strongly normalising.*

Proof. Let $\Gamma \vdash \mathbf{t} : T$, then by Lemma 7.3.5, $\exists R \preceq T$ such that $\|\Gamma\| \vdash_v \|\mathbf{t}\| : \|R\|$, and so by Theorem 5.4.4, $\|\mathbf{t}\|$ is strongly normalising. Assume $\mathbf{t}$ is not strongly normalising, say $\mathbf{t} \to \mathbf{t}_1 \to \mathbf{t}_2 \cdots$. Then by Lemma 7.3.4, $\|\mathbf{t}\| \to_v^= \|\mathbf{t}_1\| \to_v^= \|\mathbf{t}_2\| \to_v^= \cdots$. Since $\|\mathbf{t}\|$ is strongly normalising, there exists n such that $\forall i \geq n$, $\|\mathbf{t}_i\| = \|\mathbf{t}_{i+1}\|$. By Lemma 7.3.4 it means that $\forall i \geq n$, the reduction $\mathbf{t}_i \to \mathbf{t}_{i+1}$ can be only one of the type application beta-reduction, the type-distributivity rule or the type linearity rules. We define a positive measure on terms and show that these rules are strictly decreasing with respect to the measure, so $\mathbf{t}$ has to be strongly normalising.

Consider the following *measure*:

$$\begin{array}{rclrcl} |x : U| = |\mathbf{0}| = |\lambda x : U.\mathbf{t}| & = & 1 & |\mathbf{t} + \mathbf{r}| & = & 2 + |\mathbf{t}| + |\mathbf{r}| \\ |(\mathbf{t})\ \mathbf{r}| & = & |\mathbf{t}| + |\mathbf{r}| & |\Lambda X.\mathbf{t}| & = & |\mathbf{t}| \\ |\alpha.\mathbf{t}| & = & 1 + |\mathbf{t}| & |\mathbf{t}@(\sum_{i=1}^{n} U_i)| & = & 1 + 2|\mathbf{t}| \end{array}$$

We proceed by checking case by case to show that the mentioned rules are strictly decreasing on this measure (*cf.* Appendix F.7). □

7.4 Example: the Hadamard gate

The Hadamard gate (*cf.* Section 1.1 for its formal definition), can be encoded in this setting as follows. First, we name some specific terms and types which will allow to express the Hadamard gate in a more friendly way.

The identity type will be written as $I = \forall X.X \to X$ and the term $\mathbf{I} = \Lambda X.\lambda x : X.x$. We use the idea of cannon and co-cannon first presented in Section 1.2.2: $[\mathbf{t}] = \lambda z : I.\mathbf{t}$, with $z : I$ a fresh variable, and $\{\mathbf{t}\} = (\mathbf{t})\ \mathbf{I}$. So $\{[\mathbf{t}]\} \to \mathbf{t}$. Analogously, we write a cannon for types as $[T] = I \to T$, so $\Gamma \vdash \mathbf{t} : T$ implies $\Gamma \vdash [\mathbf{t}] : [T]$.

We also give the following encoding for booleans. First, we name the following types: $\mathbb{T} = \forall X.\forall Y.X \to Y \to X$ and $\mathbb{F} = \forall X.\forall Y.X \to Y \to Y$. Then, using the standard encoding for booleans, $\mathbf{true} = \Lambda X.\Lambda Y.\lambda x : X.\lambda y : Y.x$ and $\mathbf{false} = \Lambda X.\Lambda Y.\lambda x : X.\lambda y : Y.y$, we get $\vdash \mathbf{true} : \mathbb{T}$ and $\vdash \mathbf{false} : \mathbb{F}$. Also, we can encode combinations of them, *e.g.* $|+\rangle = \frac{1}{\sqrt{2}}.(\mathbf{true} + \mathbf{false})$, and $|-\rangle = \frac{1}{\sqrt{2}}.(\mathbf{true} - \mathbf{false})$; so $\vdash |+\rangle : \boxplus$ and $\vdash |+\rangle : \boxminus$; where $\boxplus = \frac{1}{\sqrt{2}}.(\mathbb{T} + \mathbb{F})$ and $\boxminus = \frac{1}{\sqrt{2}}.(\mathbb{T} - \mathbb{F})$.

Finally, we encode the Hadamard gate, and call it $\mathbf{H}$, as follows:

$$had = \lambda x : [\boxplus] \to [\boxminus] \to Z.(((x)\ [|+\rangle])\ [|-\rangle])$$

$$\mathbf{H} = \Lambda Z.had$$

To simplify notation, we define two more terms: $\mathsf{T} = \lambda x : [\boxplus].\lambda y : [\boxminus].x$ and $\mathsf{F} = \lambda x : [\boxplus].\lambda y : [\boxminus].y$. So, $(\mathbf{true}@[\boxplus])@[\boxminus] \to \mathsf{T}$, $(\mathbf{false}@[\boxplus])@[\boxminus] \to \mathsf{F}$, $\vdash \mathsf{F} : [\boxplus] \to [\boxminus] \to [\boxminus]$ and $\vdash \mathsf{T} : [\boxplus] \to [\boxminus] \to [\boxplus]$.

Now, the application of the Hadamard gate to the qubit $|+\rangle$ is encoded as follows:

$$
\begin{array}{ll}
& \{(\mathbf{H})@([\boxplus] + [\boxminus])\ (|+\rangle@[\boxplus])@[\boxminus]\} \\
\to^* & \{\frac{1}{\sqrt{2}}.((\mathbf{H})@([\boxplus] + [\boxminus])\ (\mathbf{true}@[\boxplus])@[\boxminus] + (\mathbf{H})@([\boxplus] + [\boxminus])\ (\mathbf{false}@[\boxplus])@[\boxminus])\} \\
\to^* & \{\frac{1}{\sqrt{2}}.((\mathbf{H})@([\boxplus] + [\boxminus])\ \mathsf{T} + (\mathbf{H})@([\boxplus] + [\boxminus])\ \mathsf{F})\} \\
\to^* & \{\frac{1}{\sqrt{2}}.((\lambda x : [\boxplus] \to [\boxminus] \to [\boxplus].(((x)\ [|+\rangle])\ [|-\rangle])\ \mathsf{T} + (\lambda x : [\boxplus] \to [\boxminus] \to [\boxminus].(((x)\ [|+\rangle])\ [|-\rangle])\ \mathsf{F})\} \\
\to^* & \{\frac{1}{\sqrt{2}}.((((\mathsf{T})\ [|+\rangle])\ [|-\rangle]) + (((\mathsf{F})\ [|+\rangle])\ [|-\rangle]))\} \\
\to^* & \{\frac{1}{\sqrt{2}}.([|+\rangle] + [|-\rangle])\} \\
\to^* & \frac{1}{\sqrt{2}}.(|+\rangle + |-\rangle) \\
\to^* & \mathbf{true}
\end{array}
$$

We show that this term has type $\mathbb{T}$, as expected. Let $\Gamma = \{x : [\boxplus] \to [\boxminus] \to Z\}$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash x : [\boxplus] \to [\boxminus] \to Z}\,ax \qquad \Gamma \vdash [|+\rangle] : [\boxplus]}{\Gamma \vdash (x)\ [|+\rangle] : [\boxminus] \to Z}\,{\to_E} \qquad \Gamma \vdash [|-\rangle] : [\boxminus]}{\Gamma \vdash ((x)\ [|+\rangle])\ [|-\rangle] : Z}\,{\to_E}}{\vdash had : ([\boxplus] \to [\boxminus] \to Z) \to Z}\,{\to_I}}{\vdash \mathbf{H} : \forall Z.([\boxplus] \to [\boxminus] \to Z) \to Z}\,{\forall_I}}{\vdash \mathbf{H}@([\boxplus] + [\boxminus]) : (\forall Z.([\boxplus] \to [\boxminus] \to Z) \to Z)@([\boxplus] + [\boxminus])}\,{@_I}
$$

Also,

$$
\dfrac{\dfrac{\dfrac{\vdash |+\rangle : \boxplus}{\vdash |+\rangle : \frac{1}{\sqrt{2}}.\forall X.\forall Y.X \to Y \to X + \frac{1}{\sqrt{2}}.\forall X.\forall Y.X \to Y \to Y}\,{\preceq}}{\vdash |+\rangle@[\boxplus] : \frac{1}{\sqrt{2}}.\forall Y.[\boxplus] \to Y \to [\boxplus] + \frac{1}{\sqrt{2}}.\forall Y.[\boxplus] \to Y \to Y}\,{\forall_E}}{\vdash (|+\rangle@[\boxplus])@[\boxminus] : \frac{1}{\sqrt{2}}.[\boxplus] \to [\boxminus] \to [\boxplus] + \frac{1}{\sqrt{2}}.[\boxplus] \to [\boxminus] \to [\boxminus]}\,{\forall_E}
$$

Now we prove $\vdash \{(\mathbf{H}@([\boxplus] + [\boxminus]))\ (|+\rangle@[\boxplus])@[\boxminus]\} : \mathbb{T}$.

$$
\dfrac{\dfrac{\dfrac{\vdash \mathbf{H}@([\boxplus] + [\boxminus]) : (\forall Z.([\boxplus] \to [\boxminus] \to Z) \to Z)@([\boxplus] + [\boxminus]) \quad \vdash (|+\rangle@[\boxplus])@[\boxminus] : \frac{1}{\sqrt{2}}.[\boxplus] \to [\boxminus] \to [\boxplus] + \frac{1}{\sqrt{2}}.[\boxplus] \to [\boxminus] \to [\boxminus]}{\vdash (\mathbf{H}@([\boxplus] + [\boxminus]))\ (|+\rangle@[\boxplus])@[\boxminus] : \frac{1}{\sqrt{2}}.[\boxplus] + \frac{1}{\sqrt{2}}.[\boxminus]}\,{\to_E} \qquad \vdash \mathbf{I} : I}{\vdash \{(\mathbf{H}@([\boxplus] + [\boxminus]))\ (|+\rangle@[\boxplus])@[\boxminus]\} : \frac{1}{\sqrt{2}}.\boxplus + \frac{1}{\sqrt{2}}.\boxminus}\,{\to_E}}{\vdash \{(\mathbf{H}@([\boxplus] + [\boxminus]))\ (|+\rangle@[\boxplus])@[\boxminus]\} : \mathbb{T}}\,{\preceq}
$$

7.5 Conclusion

In this chapter, we have defined *Lineal*: an explicitly typed algebraic λ-calculus, based on λ_{lin} and *Vectorial*, with subject reduction and strong normalisation. The language allows making arbitrary linear combination of λ-terms $\alpha.\mathbf{t}+\beta.\mathbf{r}$. The type system is a fine-grained analysis tool describing the "vectorial" properties of the terms: it keeps track of the 'amplitude of a term', *i.e.* if $\mathbf{t}$ and $\mathbf{u}$ both have the same type U, then $\alpha.\mathbf{t} + \beta.\mathbf{u}$ has type $(\alpha + \beta).U$. Also, it keeps track of the 'direction of a term', *i.e.* if $\mathbf{t}$ and $\mathbf{u}$ have types U and V respectively, then $\alpha.\mathbf{t} + \beta.\mathbf{u}$ has type $\alpha.U + \beta.V$. This calculus is able to

encode matrices and vectors just as *Vectorial* does, but unlike *Vectorial*, its type system has the subject reduction property.

The problems depicted in Chapter 5 were solved here by using explicit types in the terms, and a subtyping system.

The resulting system has the property that if a term $\mathbf{t}$ has type $\sum_i \alpha_i.U_i$, then there exists $\mathbf{t}' = \sum_i \alpha_i.\mathbf{b}_i$ such that $\mathbf{t} \to^* \mathbf{t}'$, where each $\mathbf{b}_i$ is a base term of type U_i.

Chapter 8

Conclusions and future work

Résumé du Chapitre

Dans ce chapitre, nous résumons les contributions de cette thèse, et nous proposons quelques pistes pour de futurs travaux. ■

In this thesis we had developed a confluent, strongly normalising, typed algebraic λ-calculus, called *Lineal*. The calculus is an extension of explicitly-typed System F with arbitrary linear combination of terms: if $\mathbf{t}$ and $\mathbf{u}$ are two terms, $\alpha.\mathbf{t} + \beta.\mathbf{u}$ is also a term, where α and β belongs to a commutative ring and "." denotes the scalar multiplication. If we consider the normal form of the terms forming a base of an infinite vectorial space of normalised terms, then linear combinations of these terms in the base, the "base terms", give us the vectors of the space. Since the calculus is strongly normalising (Theorem 7.3.6), any term in the calculus will reduce to a vector in this space. The type system gives us the shape of this vector: a term $\alpha.\mathbf{t} + \beta.\mathbf{r}$, will have type $\alpha.T + \beta.R$ (or something "smaller", according to an ordering which morally just adds zeros), as stated by the following Lemma:

Lemma 7.2.2 (Generation lemmas)**.** Let S be a type, α a scalar, Γ a typing context and $\mathbf{t}$ and $\mathbf{r}$ terms.

5. If $\Gamma \vdash \mathbf{t} + \mathbf{r} : S$, then there exist types T and R such that $\Gamma \vdash \mathbf{t} : T$, $\Gamma \vdash \mathbf{r} : R$ and $T + R \preceq S$.

6. If $\Gamma \vdash \alpha.\mathbf{t} : R$, then there exists a type T such that $\Gamma \vdash \mathbf{t} : T$ and $\alpha.T \preceq R$.

Another feature of *Lineal* is that the types are preserved by reduction:

Theorem 7.2.1 (Subject Reduction)**.** For any terms $\mathbf{t}$, $\mathbf{t}'$, any context Γ and any type T , if $\mathbf{t} \to \mathbf{t}'$ then $\Gamma \vdash \mathbf{t} : T \Rightarrow \Gamma \vdash \mathbf{t}' : T$.

Thus, we can state that:

If a term $\mathbf{t}$ *has normal form* $\sum_{i=1}^{n} \alpha_i.\mathbf{b}_i$*, then its type is* $\sum_{i=1}^{n} \alpha_i.U_i$*, where* U_i *is the type of* $\mathbf{b}_i$*.*

(where the type is padded with some 0s).

Analogously, a trivial induction give us the inverse:

If a term $\mathbf{t}$ *has type* $\sum_{i=1}^{n} \alpha_i.U_i$*, where the* U_i *are not type abstractions or applications, then* $\mathbf{t}$ *reduces to* $\sum_{i=1}^{n} \alpha_i.\mathbf{b}_i$*, where* b_i *has type* U_i*.*

This time, padding the term with some $\mathbf{0}$'s.

8.1 Summary

While seeking the correct definition of *Lineal*, we have developed several type systems, tackling one challenge at a time. Although these type systems were meant as intermediate steps, each is interesting in itself.

The *Scalar* type system. This system may have interesting applications for barycentric or probabilistic computing, *i.e.* when the amplitudes of the normal forms are required to sum to one. Indeed by doing small modifications to the system, obtaining the system $\mathcal{B}$, we were able to prove the following Theorem. **Theorem 3.4.3.** Let $\Gamma \vdash \mathbf{t} : A$ be well-formed, then $\mathbf{t}{\downarrow}$ has the barycentric property.

The *Additive* type system. This system serves one purpose: to show the connection between the additive fragment of the algebraic λ-calculus λ_{lin}, and System F. This connection helped proving strong normalisation for *Additive* itself, and for the λ^{CA} calculus which followed. Since several presentations of calculi associated with the differential lambda calculus [Ehrhard and Regnier, 2003] are carried out with sums but without scalars, this connection becomes an interesting piece of analysis for this strand of algebraic calculi. We have set up a translation where sum types translate into pairs, and the translation shows how the commutative and associative properties of addition can be simulated with pairs, by making them explicit.

The *Vectorial* type system. This system showed the need to move to explicit types. With implicit types, the factorisation reduction rules (*e.g.* $\alpha.\mathbf{t}+\beta.\mathbf{t} \to (\alpha+\beta).\mathbf{t}$) are the cause of a fundamental problem: since this is an extension of System F, there is no principal type, and so $\mathbf{t}$ can have two unrelated types, T and T', which cannot be "unified". Thus, $(\alpha+\beta).\mathbf{t}$ would have to have type $\alpha.T + \beta.T'$, which fails to reveal the structure of the vector. In order to avoid this problem we have set up a type system without subject reduction: $\alpha.\mathbf{t} + \beta.\mathbf{t}$ has type $\alpha.T + \beta.T'$, but $(\alpha+\beta).\mathbf{t}$ has both $(\alpha+\beta).T$ and $(\alpha+\beta).T'$ as types. We thus defined an order relation, which roughly states that if T and T' are types for the same term, then $(\alpha+\beta).T \preceq \alpha.T + \beta.T'$. We then obtained a weak subject reduction where the types are preserved up to this relation. This system is a proof-of-concept, serving to reveal this surprising absence of a one-to-one correspondence between the implicitly-typed and explicitly-typed calculus in this algebraic setting. Another interesting contribution which arose from this setting was a novel proof of strong normalisation: The proof of strong normalisation of *Scalar* is based on the proof of strong normalisation of $\lambda 2^{la}$, which is a straightforward extension of System F. Once this extension was proven strongly normalising, only a translation from typed terms in *Scalar* to typed terms in $\lambda 2^{la}$preserving reducibility, had to be defined. The same is true for *Additive*: the translation to System F was also used to prove that it is strongly normalising. For *Vectorial*, however, there was no easy way to translate the terms to a strongly normalising system, and thus a new proof was developed. This proof is a non trivial extension of the classical proof using reducibility candidates. This have set up the basis for the proofs in the subsequent systems, by showing that they can be translated to *Vectorial* in a reducibility-preserving manner.

The λ^{CA} calculus. The *Lineal* calculus not only gives us a fine-grained type system revealing the structure of terms, but also gives us confluence in a simplified algebraic λ-calculus. The original λ_{lin} had several restrictions to prevent non-terminating terms from causing a loss of confluence in this algebraic

setting, however, if we are interested only in this restriction, the full *Lineal* calculus may seen somewhat an overkill. Still, none of the previous systems achieve this: *Scalar* does not have a vector space of normalised terms, terms being added have to have the same type up to the scalars. *Additive* is for just a fragment of the calculus, and *Vectorial* is again as complex as *Lineal*. Therefore there enters the λ^{CA} calculus, an alternative type system giving us strong normalisation, and hence a simplified set of rules, with a reasonably simple setting. The type system also gives us some information of terms, which, while not being as accurate as *Lineal*, provides lower bounds of their scalars.

8.2 Future directions

The next paragraphs will unravel directions for future work. As such, they contain several unproven intuitions, *i.e.* tentative connections. However we strongly believe that some of them can be formalised, and so listing them is a element of judgement of the potential outcomes of this thesis, as well as an aid for future researchers embracing the topic of vectorial types.

8.2.1 Semantics and Differentiation

In Chapter 2 we have formalised connections between λ_{lin} and λ_{alg}. These connections suggest some work that can be done: λ_{lin} and *Lineal* still lack a full formal denotational semantics. A first step has been made in [Valiron, 2010], however this is preliminary in comparison with the many more results on the denotational semantics of λ_{alg} and related systems, *e.g.* [Ehrhard, 2005, Tasson, 2009]. A study of the invariability of those semantics by the call-by-name/call-by-value simulation proposed in Chapter 2 could therefore lead to a denotational semantics for *Lineal*.

Another possible way to exploit this connection is in order to relate λ_{lin} to the differential λ-calculus [Ehrhard and Regnier, 2003], a calculus that introduces a differential operator which induces a Taylor expansion for terms [Ehrhard, 2010], among other interesting results. It was from this differential λ-calculus that the addition in λ_{alg} came about. Indeed, λ_{alg} can be seen as the differential λ-calculus without a differential operator. Thus the question of whether is possible to program an analogous operator in λ_{lin} arises.

8.2.2 A quantum calculus

The *Lineal* calculus can be seen as a first step towards a quantum calculus, but while in *Lineal* any vector is allowed, in quantum computing we must restrict these vectors to be of norm one: $\alpha.\mathbf{b}_1 + \beta.\mathbf{b}_2$ is a valid vector only if α and β are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$. Notice that this is the case only when $\mathbf{b}_1 \neq \mathbf{b}_2$; indeed if $\mathbf{b}_1 = \mathbf{b}_2$, the restriction would be $|\alpha + \beta|^2 = 1$. Notice that $\mathbf{b}_1$ and $\mathbf{b}_2$ are members of the base of the vectorial space, and two different basis vectors are orthogonal by definition. But in general, for non-basis vectors, the condition that needs to be checked is orthogonality.

Given an orthogonality definition, we would need to change rule $+_I$ in order to account for the restriction to vectors of norm one:

$$\frac{\Gamma \vdash \mathbf{t} : T \quad \Gamma \vdash \mathbf{r} : R \quad |\alpha|^2 + |\beta|^2 = 1 \quad \mathbf{t} \perp \mathbf{u}}{\Gamma \vdash \alpha.\mathbf{t} + \beta.\mathbf{u} : \alpha.T + \beta.R} +_\perp \qquad \frac{\Gamma \vdash \mathbf{t} : T \quad \Gamma \vdash \mathbf{r} : R \quad |\alpha + \beta|^2 = 1 \quad \mathbf{t} \parallel \mathbf{u}}{\Gamma \vdash \alpha.\mathbf{t} + \beta.\mathbf{u} : \alpha.T + \beta.R} +_\parallel$$

A natural way to define orthogonality is to define first an inner product between terms.

Definition 8.2.1 (Inner product between closed terms)**.** *Let* $\mathbf{b}_1$ *and* $\mathbf{b}_2$ *be either closed, reduced base terms, or the term* $\mathbf{0}$*.*

$$\begin{array}{rcl} \langle \mathbf{b}_1 | \mathbf{b}_2 \rangle = \langle \mathbf{b}_2 | \mathbf{b}_1 \rangle & = & \begin{cases} 0 & \text{if } \mathbf{b}_1 \neq \mathbf{b}_2 \\ 1 & \text{if } \mathbf{b}_1 = \mathbf{b}_2 \end{cases} \\ \langle \alpha.\mathbf{t} + \beta.\mathbf{s} | \mathbf{r} \rangle & = & \alpha \times \langle \mathbf{t} | \mathbf{r} \rangle + \beta \times \langle \mathbf{s} | \mathbf{r} \rangle \\ \langle \mathbf{t} | \mathbf{s} \rangle = \overline{\langle \mathbf{s} | \mathbf{t} \rangle} & = & \langle \mathbf{t}{\downarrow} | \mathbf{s}{\downarrow} \rangle \end{array}$$

For example, we have $\langle (\lambda x : U.x)\ \mathbf{v}^U | \mathbf{v}^U \rangle = 1$ and $\langle \mathbf{true} | \mathbf{false} \rangle = 0$. It also works for more elaborated examples:

$$\langle \frac{1}{\sqrt{2}}.\mathbf{true} + \frac{1}{\sqrt{2}}.\mathbf{false} | \mathbf{true} \rangle = \frac{1}{\sqrt{2}} \times \langle \mathbf{true} | \mathbf{true} \rangle + \frac{1}{\sqrt{2}} \times \langle \mathbf{false} | \mathbf{true} \rangle = \frac{1}{\sqrt{2}}$$

Then we can of course define $\mathbf{t} \perp \mathbf{s} \Leftrightarrow \langle \mathbf{t} | \mathbf{s} \rangle = 0$, and some similar definition for parallelism. However, our challenge is a harder one still: we must check this properties using types. Indeed, the definition of inner product needs to reduce the term (remember that the vectorial space is defined only over normal forms terms), and typing is useless if it requires prior reduction. Thus assume we have an analogous definition of orthogonality between types; we want it to have the property: if $\vdash \mathbf{t} : T$ and $\vdash \mathbf{r} : R$, then $\mathbf{t} \perp \mathbf{r} \Leftrightarrow T \perp R$. Alas it cannot be that simple, for example **true** and **false** have both type $X \to X \to X$. However, an encouraging point is that since *Lineal* is explicitly typed, we can expect to have $T \perp R \Rightarrow \mathbf{t} \perp \mathbf{r}$.

Once the norm one terms are properly characterised, one could expect to be able to add a measurement operator *à la* [Díaz-Caro, 2007, Arrighi, Díaz-Caro, Gadella, and Grattage, 2011a].

8.2.3 Logics

The type systems presented in this thesis could also lead to different algebraic logics: the barycentric restriction to *Scalar* could lead to a barycentric or probabilistic logic. *Vectorial* and *Lineal* could induce a vectorial logic, where logic formulae form a vectorial space. The quantum calculus suggested above could also lead to a novel formulation of a quantum logic formally connected with quantum computing. These logics would then need to be compared with linear logic, in particular with respect to the different kinds of linearity involved. Indeed, linear logic introduces a linearity which is based on resources, whereas algebraic calculi uses the classic concept of algebraic linearity. The question is whether these linearities are the same, or if they can be compared in some way.

For the sake of exemplifying, we take *Scalar* (*cf.* Chapter 3) which is one of the simplest systems developed in this thesis. Consider the "logic" defined from *Scalar*: the logical propositions are the types; the sequents are the contexts plus the types; the logical rules are obtained simply by erasing the terms from the typing rules; the proofs are obtained simply by erasing the terms from the type derivation trees – or equivalently by applying the logical rules upon the logical propositions. We call it *Scalar* logic, and denote $\mathcal{SL}$. We now show that proofs in $\mathcal{SL}$ enjoy a no-cloning property.

First we need to define what we mean by proof method. For this, consider R be a $\mathcal{SL}$ rule, where some information is added to make it deterministic: for example, we write $s_I[\alpha]$ for s_I, when the scalar introduced is α, and so on. That is, for any sequents Q_i, Q'_i, with $i = 1, \ldots, n$, such that $\forall i$, $Q_i \equiv Q'_i$, we make sure that

$$\left\{ \frac{Q_1, \ldots, Q_n}{S} R \ \wedge \ \frac{Q'_1, \ldots, Q'_n}{S'} R \right\} \Rightarrow S \equiv S'$$

Hence if Π is a tree with nodes labelled by deterministic names of $\mathcal{SL}$ logical rules, then one may think of Π as a function from sequents to proofs, *i.e.* a proof method:

Definition 8.2.2. *We define recursively the concept of proof method of order n to be the set of functions Π_n which take the following form:*

$$\Pi_0(S) = S$$
$$\Pi_n(S) = \frac{\Pi_{n-1}(S)}{P}\,R \qquad or \qquad \frac{\Pi_k(S) \quad \pi_h}{P}\,R \qquad or \qquad \frac{\pi_k \quad \Pi_h(S)}{P}\,R$$

where S is a sequent, π_n is a constant proof of size n, $\max\{k,h\} = n-1$, R is a logical rule, and P is a sequent such that the resulting proof is well-formed.

We denote by $C(\Pi_n(S))$ the conclusion (root) of the proof $\Pi_n(S)$.

A no-cloning theorem (*cf.* Theorem 1.1.1) can be defined in terms of proof methods, and the way they treat scalars, *i.e.* there is no generic proof method able to take a sequent with a scalar in its type as argument, and return a sequent where such a scalar appears more than once in the type (proof in Appendix G.1).

Theorem 8.2.3 (No-cloning of scalars)**.** *$\nexists\Pi_n$ such that $\forall\alpha, C(\Pi_n(\Gamma \vdash \alpha.U)) = \Delta \vdash (\delta \times \alpha^s + \gamma).V$ with $\delta \neq 0$ and γ constants in $\mathcal{S}$, $s \in \mathbb{N}^{>1}$ and U, V unit* formulas.

Notice that α is a member of a ring and s is a natural number, so α^s is just the multiplication of α by itself s times.

We can reformulate this theorem to look more like a no-cloning theorem in the following way. Let $T \otimes T$ stand for the usual encoding of tuples[1].

Corollary 8.2.4 (No-cloning Theorem)**.** *$\nexists\Pi_n$ such that $\forall T, \Pi_n(\Gamma \vdash T)$ is a witness of $\Gamma \vdash T \Rightarrow \Delta \vdash T \otimes T$.*

Proof. By Lemma 3.2.2, $\exists\alpha, U$ such that $T \equiv \alpha.U$, so $T \otimes T \equiv \alpha.U \otimes \alpha.U \equiv \alpha^2.(U \otimes U) = (1 \times (1 \times \alpha^2 + 0)).(U \otimes U)$. Then by Theorem 8.2.3 the corollary holds. □

Hence our no-cloning allows the existence of a proof method Π such that $\Pi(\Gamma \vdash T)$ has conclusion $\Gamma \vdash T \Rightarrow \Delta \vdash T \otimes T$, but it does not allow the *same* proof method Π to accomplish this for any proposition T. Informally, this property states that $\mathcal{SL}$ has no fixed proof method for duplicating a proposition.

Clearly $\mathcal{SL}$, unlike linear logic ($\mathcal{LL}$) [Girard, 1987], does not refrain us from duplicating resources. Yet we have been able to prove a no-cloning theorem for $\mathcal{SL}$. How can we make sense of this apparent contradiction? Consider the *copying machine* $\vdash \lambda x.x \otimes x : \forall X.X \to X \otimes X$, and let $A \equiv \alpha.U$, then this machine allows:

$$\dfrac{\dfrac{\dfrac{}{\vdash \alpha.\lambda x.x \otimes x : \alpha.\forall X.X \to (X \otimes X)}}{\vdash \alpha.\lambda x.x \otimes x : \alpha.U \to U \otimes U}\,\forall_E \qquad \dfrac{\vdots}{\vdash \mathbf{t} : \alpha.U \equiv A}}{\vdash (\alpha.\lambda x.x \otimes x)\ \mathbf{t} : \alpha^2.(U \otimes U) \equiv A \otimes A}\,\to_E$$

This proof tree yields $A \otimes A$ from a single proof of A, which needs be plugged as the right branch of the tree. However the symbol A appears also in the right branch of the tree; hence the proof method that duplicates A crucially depends on A. It is on this basis that our no-cloning theorem is formulated; our no-cloning allows the existence of a proof method Π such that $\Pi(\Gamma \vdash T)$ has conclusion $\Gamma \vdash T \Rightarrow \Delta \vdash T \otimes T$, but it does not allow the *same* proof method Π to work for any type. This way of phrasing no-cloning must probably hold in $\mathcal{LL}$ as well, but it is not usually contemplated. $\mathcal{SL}$ emphasises this property, which seems more in line with quantum theory than the straightforward non-duplication of resources of $\mathcal{LL}$.

[1] Formally, to allow such an encoding for general types, we need to add the following equivalence $(\alpha.U) \to T \equiv \alpha.(U \to T)$.

Indeed, quantum theory states that it is not possible to have a *universal* cloning machine, but does allow cloning machines of *specific* vectors.

Notice that, because the differentiation between general types and unit types, the correspondence between hypothesis in a derivation tree and direct implication by arrow introduction, is broken. Indeed, take a derivation tree starting with a formula T corresponding to a general type as hypothesis, and concluding with another formula R. In many classical logics one could derive $T \Rightarrow R$, however it is not the case in any of the logics induced by the type systems presented in this thesis. $T \Rightarrow R$ is valid only when T is a unit type. So, we have two kinds of implications, the one coming from arrow types, and the general one obtained from lifting a hypothesis.

In fact, technically speaking what we are saying is that if we assume an unknown formula as hypothesis, we cannot derive an implication from it. Nevertheless, some modifications could be introduced to *Lineal* in order to allow this kind of implications. In particular, we could introduce a new kind of non-base type A^+ and see the consequences. The first point is that, as discussed in Section 3.1, we cannot allow any type at the left hand side of an arrow without breaking the correspondence between scalar in the term and scalar in the type. This is unless the type A^+ is used only once in the function. Arrow introduction could therefore be relaxed to allow this exception. This proposed change seems to relate the A^+ types with the linear resources in linear logic. Formalising this ideas could provide a link between *Lineal* and linear logic, which would connect the concept of resources in $\mathcal{LL}$ to the concept of unknown linear combinations in *Lineal*. This could provide a new algebraic interpretation of linear logic, in an alternative setting to the model approach, *i.e.* within an algebraic logic.

Appendices

Appendix A

Proofs from Chapter 2

A.1 Proof of Lemma 2.2.2

Lemma 2.2.2 (Local confluence)**.** The four languages in Figure 2.1 are locally confluent.

Proof.

- If $\mathbf{t} \to \mathbf{r}$ and $\mathbf{t} \to \mathbf{r}'$, using only algebraic rules, then this has been already proven in Lemma 2.2.1.
- If $\mathbf{t} \to \mathbf{r}$ and $\mathbf{t} \to \mathbf{r}'$, using only beta-reduction, this is a trivial extension of the confluence of lambda calculus.
- If $\mathbf{t} \to \mathbf{r}$ by an algebraic rule and $\mathbf{t} \to \mathbf{r}'$ by beta reduction, then in λ_{lin} a term of the form $(\lambda x.\mathbf{t}')\ \mathbf{b}$ has to be a subterm of $\mathbf{t}$, since $\mathbf{t}$ beta-reduces. Note that $\mathbf{t}'$ cannot reduce since it is under a lambda and $\mathbf{b}$ cannot reduce since it is a base term. Then the beta-reduction and the algebraic-reduction are independent in λ_{lin}, and so this result is trivial. In λ_{alg} a term of the form $(\lambda x.\mathbf{t}')\ \mathbf{r}$ has to be a subterm of $\mathbf{t}$. Note that $\mathbf{t}'$ cannot reduce since it is under a lambda and $\mathbf{r}$ cannot reduce since it is an argument. Then again the beta-reduction and the algebraic-reduction are independent in λ_{alg}, and so this result is trivial.

□

A.2 Proof of Lemma 2.3.5

Lemma 2.3.5. $[\![\mathbf{t}[\mathbf{b}/x]]\!] = [\![\mathbf{t}]\!][\Psi(\mathbf{b})/x]$ with $\mathbf{b}$ a base term.

Proof. Structural induction on $\mathbf{t}$.

- $\mathbf{t} = x$. Cases:
  - $\mathbf{b} = y$. Then $\mathbf{t}[\mathbf{b}/x] = y$, so $[\![\mathbf{t}[\mathbf{b}/x]]\!] = \lambda f.\,(f)\ y = \lambda f.\,(f)\ x[y/x] = [\![\mathbf{t}]\!][\Psi(\mathbf{b})/x]$.
  - $\mathbf{b} = \lambda y.\mathbf{r}$. Then $[\![\mathbf{t}[\mathbf{b}/x]]\!] = \lambda f.\,(f)\ \lambda y.[\![\mathbf{r}]\!] = \lambda f.\,(f)\ x[\lambda y.[\![\mathbf{r}]\!]/x] = [\![\mathbf{t}]\!][\Psi(\mathbf{b})/x]$.
- $\mathbf{t} = y$. Then $[\![\mathbf{t}[\mathbf{b}/x]]\!] = [\![\mathbf{t}]\!][\Psi(\mathbf{b})/x] = [\![\mathbf{t}]\!]$.
- $\mathbf{t} = \mathbf{0}$. Analogous to previous case.

- $\mathbf{t} = \lambda y.\mathbf{r}$. Then

$$\begin{aligned}
[\![(\lambda y.\mathbf{r})[\mathbf{b}/x]]\!] &= [\![\lambda y.(\mathbf{r}[\mathbf{b}/x])]\!] \\
&= \lambda f.\,(f)\ \lambda y.[\![\mathbf{r}[\mathbf{b}/x]]\!] \\
&\quad \text{by the induction hypothesis} \\
&= \lambda f.\,(f)\ \lambda y.[\![\mathbf{r}]\!][\Psi(\mathbf{b})/x] \\
&= (\lambda f.\,(f)\ \lambda y.[\![\mathbf{r}]\!])[\Psi(\mathbf{b})/x] \\
&= [\![\mathbf{t}]\!][\Psi(\mathbf{b})/x]
\end{aligned}$$

- $\mathbf{t} = (\mathbf{r}_1)\ \mathbf{r}_2$. Then

$$\begin{aligned}
[\![\mathbf{t}[\mathbf{b}/x]]\!] &= [\![((\mathbf{r}_1)\ \mathbf{r}_2)[\mathbf{b}/x]]\!] \\
&= [\![(\mathbf{r}_1[\mathbf{b}/x])\ \mathbf{r}_2[\mathbf{b}/x]]\!] \\
&= \lambda f.([\![\mathbf{r}_1[\mathbf{b}/x]]\!])\ \lambda g.([\![\mathbf{r}_2[\mathbf{b}/x]]\!])\ \lambda h.((g)\ h)\ f \\
&\quad \text{by the induction hypothesis} \\
&= \lambda f.([\![\mathbf{r}_1]\!][\Psi(\mathbf{b})/x])\ \lambda g.([\![\mathbf{r}_2]\!][\Psi(\mathbf{b})/x])\ \lambda h.((g)\ h)\ f \\
&= \lambda f.([\![\mathbf{r}_1]\!])\ \lambda g.([\![\mathbf{r}_2]\!])\ \lambda h.((g)\ h)\ f[\Psi(\mathbf{b})/x] \\
&= [\![(\mathbf{r}_1)\ \mathbf{r}_2]\!][\Psi(\mathbf{b})/x] \\
&= [\![\mathbf{t}]\!][\Psi(\mathbf{b})/x]
\end{aligned}$$

- $\mathbf{t} = \alpha.\mathbf{r}$. Then

$$\begin{aligned}
[\![\mathbf{t}[\mathbf{b}/x]]\!] &= [\![(\alpha.\mathbf{r})[\mathbf{b}/x]]\!] \\
&= [\![\alpha.(\mathbf{r}[\mathbf{b}/x])]\!] \\
&= \lambda f.(\alpha.[\![\mathbf{r}[\mathbf{b}/x]]\!]\ f) \\
&\quad \text{by the induction hypothesis} \\
&= \lambda f.(\alpha.[\![\mathbf{r}]\!][\Psi(\mathbf{b})/x]\ f) \\
&= (\lambda f.(\alpha.[\![\mathbf{r}]\!]\ f))[\Psi(\mathbf{b})/x] \\
&= [\![\alpha.\mathbf{r}]\!][\Psi(\mathbf{b})/x] \\
&= [\![\mathbf{t}]\!][\Psi(\mathbf{b})/x]
\end{aligned}$$

- $\mathbf{t} = \mathbf{r}_1 + \mathbf{r}_2$. Then

$$\begin{aligned}
[\![\mathbf{t}[\mathbf{b}/x]]\!] &= [\![(\mathbf{r}_1 + \mathbf{r}_2)[\mathbf{b}/x]]\!] \\
&= [\![\mathbf{r}_1[\mathbf{b}/x] + \mathbf{r}_2[\mathbf{b}/x]]\!] \\
&= \lambda f.(([\![\mathbf{r}_1[\mathbf{b}/x]]\!] + [\![\mathbf{r}_2[\mathbf{b}/x]]\!])\ f) \\
&\quad \text{by the induction hypothesis} \\
&= \lambda f.(([\![\mathbf{r}_1]\!][\Psi(\mathbf{b})/x] + [\![\mathbf{r}_2]\!][\Psi(\mathbf{b})/x])\ f) \\
&= (\lambda f.(([\![\mathbf{r}_1]\!] + [\![\mathbf{r}_2]\!])\ f))[\Psi(\mathbf{b})/x] \\
&= [\![\mathbf{r}_1 + \mathbf{r}_2]\!][\Psi(\mathbf{b})/x] \\
&= [\![\mathbf{t}]\!][\Psi(\mathbf{b})/x]
\end{aligned}$$

□

A.3 Proof of Lemma 2.3.13

Lemma 2.3.13. If $\mathbf{b}$ is a base term, then for any $\mathbf{t}$, $(\llbracket \mathbf{t} \rrbracket)\ \mathbf{b} \to^*_{a+\beta} \mathbf{t} : \mathbf{b}$.

Proof. Structural induction on $\mathbf{t}$.

- $\mathbf{t} = x$. Then $(\llbracket x \rrbracket)\ \mathbf{b} = (\lambda f.\,(f)\ x)\ \mathbf{b} \to_{a+\beta} (\mathbf{b})\ x = x : \mathbf{b}$.
- $\mathbf{t} = \lambda x.\mathbf{r}$. Then $(\llbracket \lambda x.\mathbf{r} \rrbracket)\ \mathbf{b} = (\lambda f.\,(f)\ \lambda x.\llbracket \mathbf{r} \rrbracket)\ \mathbf{b} = (\lambda f.\,(f)\ \Psi(\lambda x.\mathbf{r}))\ \mathbf{b} \to_{a+\beta} (\mathbf{b})\ \Psi(\lambda x.\mathbf{r}) = \lambda x.\mathbf{r} : \mathbf{b}$.
- $\mathbf{t} = \mathbf{0}$. Then $(\llbracket \mathbf{0} \rrbracket)\ \mathbf{b} = (\mathbf{0})\ \mathbf{b} \to_{a+\beta} \mathbf{0} = \mathbf{0} : \mathbf{b}$.
- $\mathbf{t} = \mathbf{t}' + \mathbf{r}$. Then $(\llbracket \mathbf{t}' + \mathbf{r} \rrbracket)\ \mathbf{b} = (\lambda f.(\llbracket \mathbf{t}' \rrbracket + \llbracket \mathbf{r} \rrbracket)\ f)\ \mathbf{b} \to_{a+\beta} (\llbracket \mathbf{t}' \rrbracket + \llbracket \mathbf{r} \rrbracket)\ \mathbf{b} \to_{a+\beta} (\llbracket \mathbf{t}' \rrbracket)\ \mathbf{b} + (\llbracket \mathbf{r} \rrbracket)\ \mathbf{b}$ which $\to_{a+\beta}$-reduces by the induction hypothesis to $\mathbf{t}' : \mathbf{b} + \mathbf{r} : \mathbf{b} = \mathbf{t}' + \mathbf{r} : \mathbf{b}$.
- $\mathbf{t} = \alpha.\mathbf{r}$. Then $(\llbracket \alpha.\mathbf{r} \rrbracket)\ \mathbf{b} = (\lambda f.(\alpha.\llbracket \mathbf{r} \rrbracket)\ f)\ \mathbf{b} \to_{a+\beta} (\alpha.\llbracket \mathbf{r} \rrbracket)\ \mathbf{b} \to_{a+\beta} \alpha.(\llbracket \mathbf{r} \rrbracket)\ \mathbf{b})$ which $\to_{a+\beta}$-reduces by the induction hypothesis to $\alpha.(\mathbf{r} : \mathbf{b}) = \alpha.\mathbf{r} : \mathbf{b}$.
- $\mathbf{t} = (\mathbf{t}')\ \mathbf{r}$. Then $(\llbracket (\mathbf{t}')\ \mathbf{r} \rrbracket)\ \mathbf{b} = (\lambda f.(\llbracket \mathbf{t}' \rrbracket)\ \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ f)\ \mathbf{b} \to_{a+\beta} (\llbracket \mathbf{t}' \rrbracket)\ \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b}$. Note that $\lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b}$ is a value, so by the induction hypothesis the above term reduces to $\mathbf{t}' : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b}$. We do a second induction, over $\mathbf{t}'$, to prove that $\mathbf{t}' : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} \to_{a+\beta} (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}' = (\mathbf{t}_1)\ \mathbf{t}_2$, then $\mathbf{t}' : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} = ((\mathbf{t}_1)\ \mathbf{t}_2)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}'$ is a base term, then $\mathbf{t}' : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} = (\lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b})\ \Psi(\mathbf{t}')$ which $\to_{a+\beta}$-reduces to $(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((\Psi(\mathbf{t}'))\ h)\ \mathbf{b}$ which $\to_{a+\beta}$-reduces by the main induction hypothesis to $\mathbf{r} : \lambda h.((\Psi(\mathbf{t}'))\ h)\ \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}' = \alpha.\mathbf{t}_1$, then $\mathbf{t}' : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} = \alpha.\mathbf{t}_1 : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} = \alpha.(\mathbf{t}_1 : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b})$ which $\to_{a+\beta}$-reduces by the second induction hypothesis to $\alpha.((\mathbf{t}_1)\ \mathbf{r} : \mathbf{b}) = (\alpha.\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}' = \mathbf{t}_1 + \mathbf{t}_2$, then $\mathbf{t}' : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} = \mathbf{t}_1 + \mathbf{t}_2 : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} = \mathbf{t}_1 : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} + \mathbf{t}_2 : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b}$ which $\to_{a+\beta}$-reduces by the second induction hypothesis to $(\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} + (\mathbf{t}_2)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}_1 + \mathbf{t}_2)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}' = \mathbf{0}$ then $\mathbf{t} : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} = \mathbf{0} : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} = \mathbf{0} = (\mathbf{0})\ \mathbf{r} : \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$

□

A.4 Proof of Lemma 2.3.14

Lemma 2.3.14. If $\mathbf{t} \to_\ell \mathbf{r}$ then $\forall \mathbf{b}$ base term, $\mathbf{t} : \mathbf{b} \to^*_a \mathbf{r} : \mathbf{b}$.

Proof. Case by case on the rules $\to_\ell$.

Rules A_r

- $(\mathbf{b}')\ (\mathbf{t} + \mathbf{r}) \to_\ell (\mathbf{b}')\ \mathbf{t} + (\mathbf{b}')\ \mathbf{r}$, with $\mathbf{b}'$ being a base term. Then $(\mathbf{b}')\ (\mathbf{t} + \mathbf{r}) : \mathbf{b} = \mathbf{t} + \mathbf{r} : \lambda f.((\Psi(\mathbf{b}'))\ f)\ \mathbf{b} = \mathbf{t} : \lambda f.((\Psi(\mathbf{b}'))\ f)\ \mathbf{b} + \mathbf{r} : \lambda f.((\Psi(\mathbf{b}'))\ f)\ \mathbf{b} = (\mathbf{b}')\ \mathbf{t} : \mathbf{b} + (\mathbf{b}')\ \mathbf{r} : \mathbf{b} = (\mathbf{b}')\ \mathbf{t} + (\mathbf{b}')\ \mathbf{r} : \mathbf{b}$.

- $(\mathbf{b}')\ \alpha.\mathbf{t}{\to_\ell}\alpha.(\mathbf{b}')\ \mathbf{t}$, with $\mathbf{b}'$ base term. Then $(\mathbf{b}')\ \alpha.\mathbf{t} : \mathbf{b} = \alpha.\mathbf{t} : \lambda f.((\Psi(\mathbf{b}'))\ f)\ \mathbf{b} = \alpha.(\mathbf{t} : \lambda f.((\Psi(\mathbf{b}'))\ f)\ \mathbf{b}) = \alpha.((\mathbf{b}')\ \mathbf{t} : \mathbf{b}) = \alpha.(\mathbf{b}')\ \mathbf{t} : \mathbf{b}$.
- $(\mathbf{b}')\ \mathbf{0}{\to_\ell}\mathbf{0}$, with $\mathbf{b}'$ a base term. Then $(\mathbf{b}')\ \mathbf{0} : \mathbf{b} = \mathbf{0} : \lambda f.((\Psi(\mathbf{b}'))\ f)\ \mathbf{b} = \mathbf{0} = \mathbf{0} : \mathbf{b}$.

Rules A_l

- $(\mathbf{t} + \mathbf{r})\ \mathbf{v}{\to_\ell}(\mathbf{t})\ \mathbf{v} + (\mathbf{r})\ \mathbf{v}$, with $\mathbf{v}$ being a value. Then $(\mathbf{t} + \mathbf{r})\ \mathbf{v} : \mathbf{b} = (\mathbf{t})\ \mathbf{v} + (\mathbf{r})\ \mathbf{v} : \mathbf{b}$.
- $(\alpha.\mathbf{t})\ \mathbf{v}{\to_\ell}\alpha.(\mathbf{t})\ \mathbf{v})$, with $\mathbf{v}$ being a value. Then $(\alpha.\mathbf{t})\ \mathbf{v} : \mathbf{b} = \alpha.(\mathbf{t})\ \mathbf{v} : \mathbf{b}$.
- $(\mathbf{0})\ \mathbf{v}{\to_\ell}\mathbf{0}$, with $\mathbf{v}$ a value. Then $(\mathbf{0})\ \mathbf{v} : \mathbf{b} = \mathbf{0} = \mathbf{0} : \mathbf{b}$.

Rules F **and** S

- $\alpha.(\mathbf{t}+\mathbf{r}){\to_\ell}\alpha.\mathbf{t}+\alpha.\mathbf{r}$. Then $\alpha.(\mathbf{t}+\mathbf{r}) : \mathbf{b} = \alpha.(\mathbf{t} : \mathbf{b}+\mathbf{r} : \mathbf{b}){\to_a}\alpha.(\mathbf{t} : \mathbf{b})+\alpha.(\mathbf{r} : \mathbf{b}) = \alpha.\mathbf{t}+\alpha.\mathbf{r} : \mathbf{b}$.
- $\alpha.\mathbf{t}+\beta.\mathbf{t}{\to_\ell}(\alpha+\beta).\mathbf{t}$. Then $\alpha.\mathbf{t}+\beta.\mathbf{t} : \mathbf{b} = \alpha.(\mathbf{t} : \mathbf{b})+\beta.(\mathbf{t} : \mathbf{b}){\to_a}(\alpha+\beta).(\mathbf{t} : \mathbf{b}) = (\alpha+\beta).\mathbf{t} : \mathbf{b}$.
- $\alpha.\mathbf{t} + \mathbf{t}{\to_\ell}(\alpha + 1).\mathbf{t}$. Then $\alpha.\mathbf{t} + \mathbf{t} : \mathbf{b} = \alpha.\mathbf{t} : \mathbf{b} + \mathbf{t} : \mathbf{b} = \alpha.(\mathbf{t} : \mathbf{b}) + \mathbf{t} : \mathbf{b}{\to_a}(\alpha + 1).(\mathbf{t} : \mathbf{b}) = (\alpha + 1).\mathbf{t} : \mathbf{b}$.
- $\mathbf{t} + \mathbf{t}{\to_\ell}(1 + 1).\mathbf{t}$. Then $\mathbf{t} + \mathbf{t} : \mathbf{b} = \mathbf{t} : \mathbf{b} + \mathbf{t} : \mathbf{b}{\to_a}(1 + 1).(\mathbf{t} : \mathbf{b}) = (1 + 1).\mathbf{t} : \mathbf{b}$.
- $\mathbf{0} + \mathbf{t}{\to_\ell}\mathbf{t}$. Then $\mathbf{0} + \mathbf{t} : \mathbf{b} = (\mathbf{0} : \mathbf{b}) + (\mathbf{t} : \mathbf{b}) = \mathbf{0} + (\mathbf{t} : \mathbf{b}){\to_a}\mathbf{t} : \mathbf{b}$.
- $\alpha.(\beta.\mathbf{t}){\to_\ell}(\alpha \times \beta).\mathbf{t}$. Then $\alpha.(\beta.\mathbf{t}) : \mathbf{b} = \alpha.(\beta.\mathbf{t} : \mathbf{b}) = \alpha.(\beta.(\mathbf{t} : \mathbf{b})){\to_a}(\alpha \times \beta).(\mathbf{t} : \mathbf{b}) = (\alpha \times \beta).\mathbf{t} : \mathbf{b}$.
- $1.\mathbf{t}{\to_\ell}\mathbf{t}$. Then $1.\mathbf{t} : \mathbf{b} = 1.(\mathbf{t} : \mathbf{b}){\to_a}\mathbf{t} : \mathbf{b}$.
- $0.\mathbf{t}{\to_\ell}\mathbf{0}$. Then $0.\mathbf{t} : \mathbf{b} = 0.(\mathbf{t} : \mathbf{b}){\to_a}\mathbf{0} = \mathbf{0} : \mathbf{b}$.
- $\alpha.\mathbf{0}{\to_\ell}\mathbf{0}$. Then $\alpha.\mathbf{0} : \mathbf{b} = \alpha.(\mathbf{0} : \mathbf{b}) = \alpha.\mathbf{0}{\to_a}\mathbf{0} = \mathbf{0} : \mathbf{b}$.

Rules *Asso* **and** *Com*

- $\mathbf{t} + (\mathbf{r} + \mathbf{s}){\to_\ell}(\mathbf{t} + \mathbf{r}) + \mathbf{s}$. Then $\mathbf{t} + (\mathbf{r} + \mathbf{s}) : \mathbf{b} = \mathbf{t} : \mathbf{b} + (\mathbf{r} + \mathbf{s} : \mathbf{b}) = \mathbf{t} : \mathbf{b} + (\mathbf{r} : \mathbf{b} + \mathbf{s} : \mathbf{b}){\to_a}(\mathbf{t} : \mathbf{b} + \mathbf{r} : \mathbf{b}) + \mathbf{s} : \mathbf{b} = \mathbf{t} + \mathbf{r} : \mathbf{b} + \mathbf{s} : \mathbf{b} = (\mathbf{t} + \mathbf{r}) + \mathbf{s} : \mathbf{b}$.
- $\mathbf{t} + \mathbf{r}{\to_\ell}\mathbf{r} + \mathbf{t}$. Then $\mathbf{t} + \mathbf{r} : \mathbf{b} = \mathbf{t} : \mathbf{b} + \mathbf{r} : \mathbf{b}{\to_a}\mathbf{r} : \mathbf{b} + \mathbf{t} : \mathbf{b} = \mathbf{r} + \mathbf{t} : \mathbf{b}$.

Rules ξ **and** $\xi_{\lambda_{lin}}$ Assume $M{\to_\ell}\mathbf{t}'$, and assume that for all $\mathbf{b}$ base term, $\mathbf{t} : \mathbf{b}{\to_a^*}\mathbf{t}' : \mathbf{b}$. We show that the result also holds for each contextual rule.

- $\mathbf{t} + \mathbf{r}{\to_\ell}\mathbf{t}' + \mathbf{r}$. Then $\mathbf{t} + \mathbf{r} : \mathbf{b} = \mathbf{t} : \mathbf{b} + \mathbf{r} : \mathbf{b}{\to_a^*}\mathbf{t}' : \mathbf{b} + \mathbf{r} : \mathbf{b} = \mathbf{t}' + \mathbf{r} : \mathbf{b}$.
- $\mathbf{r} + \mathbf{t}{\to_\ell}\mathbf{r} + \mathbf{t}'$, analogous to previous case.
- $\alpha.\mathbf{t}{\to_\ell}\alpha.\mathbf{t}'$. Then $\alpha.\mathbf{t} : \mathbf{b} = \alpha.(\mathbf{t} : \mathbf{b}){\to_a^*}\alpha.(\mathbf{t}' : \mathbf{b}) = \alpha.\mathbf{t}' : \mathbf{b}$.
- $(\mathbf{v})\ \mathbf{t}{\to_\ell}(\mathbf{v})\ \mathbf{t}'$. Case by case:
  - $\mathbf{v} = \mathbf{b}'$. Then $(\mathbf{b}')\ \mathbf{t} : \mathbf{b} = \mathbf{t} : \lambda f.((\Psi(\mathbf{b}'))\ f)\ \mathbf{b}$ which $\to_a$-reduces by the induction hypothesis to $\mathbf{t}' : \lambda f.((\Psi(\mathbf{b}'))\ f)\ \mathbf{b} = (\mathbf{b}')\ \mathbf{t}' : \mathbf{b}$.
  - $\mathbf{v} = \mathbf{0}$. Then $(\mathbf{0})\ \mathbf{t} : \mathbf{b} = \mathbf{0} = (\mathbf{0})\ \mathbf{t}' : \mathbf{b}$.
  - $\mathbf{v} = \alpha.\mathbf{w}$. Then $(\alpha.\mathbf{w})\ \mathbf{t} : \mathbf{b} = \alpha.(\mathbf{w})\ \mathbf{t} : \mathbf{b} = \alpha.((\mathbf{w})\ \mathbf{t} : \mathbf{b})$ which $\to_a$-reduces by the induction hypothesis to $\alpha.((\mathbf{w})\ \mathbf{t}' : \mathbf{b}) = \alpha.(\mathbf{w})\ \mathbf{t}' : \mathbf{b} = (\alpha.\mathbf{w})\ \mathbf{t}' : \mathbf{b}$.

- $\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_2$. Then $(\mathbf{v}_1 + \mathbf{v}_2)\ \mathbf{t} : \mathbf{b} = (\mathbf{v}_1)\ \mathbf{t} + (\mathbf{v}_2)\ \mathbf{t} : \mathbf{b} = (\mathbf{v}_1)\ \mathbf{t} : \mathbf{b} + (\mathbf{v}_2)\ \mathbf{t} : \mathbf{b}$ which $\to_a$-reduces by the induction hypothesis to $(\mathbf{v}_1)\ \mathbf{t}' : \mathbf{b} + (\mathbf{v}_2)\ \mathbf{t}' : \mathbf{b} = (\mathbf{v}_1)\ \mathbf{t}' + (\mathbf{v}_2)\ \mathbf{t}' : \mathbf{b} = (\mathbf{v}_1 + \mathbf{v}_2)\ \mathbf{t}' : \mathbf{b}$.

• $(\mathbf{t})\ \mathbf{r} \to_\ell (\mathbf{t}')\ \mathbf{r}$ Case by case:

- $\mathbf{t} = \mathbf{b}'$. Absurd since a base term cannot reduce.
- $\mathbf{t} = \alpha.\mathbf{t}_1$. Case by case on the possible $\to_\ell$-reductions of $\mathbf{t}$:
  - $\mathbf{t}' = \alpha.\mathbf{t}_1'$ with $\mathbf{t}_1 \to_\ell \mathbf{t}_1'$. Then $(\alpha.\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} = \alpha.(\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} = \alpha.((\mathbf{t}_1)\ \mathbf{r} : \mathbf{b})$ which by the induction hypothesis $\to_a$-reduces to $\alpha.((\mathbf{t}_1')\ \mathbf{r} : \mathbf{b}) = \alpha.(\mathbf{t}_1')\ \mathbf{r} : \mathbf{b} = (\alpha.\mathbf{t}_1')\ \mathbf{r} : \mathbf{b}$.
  - $\mathbf{t} = \alpha.(\beta.\mathbf{t}_3)$ and $\mathbf{t}' = (\alpha \times \beta).\mathbf{t}_3$. Then $(\alpha.(\beta.\mathbf{t}_3))\ \mathbf{r} : \mathbf{b} = \alpha.(\beta.((\mathbf{t}_3)\ \mathbf{r} : \mathbf{b})) \to_a (\alpha \times \beta).((\mathbf{t}_3)\ \mathbf{r} : \mathbf{b}) = ((\alpha \times \beta).\mathbf{t}_3)\ \mathbf{r} : \mathbf{b}$.
  - $\mathbf{t} = \alpha.(\mathbf{s}_1 + \mathbf{s}_2)$ and $\mathbf{t}' = \alpha.\mathbf{s}_1 + \alpha.\mathbf{s}_2$. Then $(\alpha.(\mathbf{s}_1 + \mathbf{s}_2))\ \mathbf{r} : \mathbf{b} = \alpha.((\mathbf{s}_1)\ \mathbf{r} : \mathbf{b} + (\mathbf{s}_2)\ \mathbf{r} : \mathbf{b}) \to_a \alpha.((\mathbf{s}_1)\ \mathbf{r} : \mathbf{b}) + \alpha.((\mathbf{s}_2)\ \mathbf{r} : \mathbf{b}) = (\alpha.\mathbf{s}_1 + \alpha.\mathbf{s}_2)\ \mathbf{r} : \mathbf{b}$.
  - $\alpha = 1$ and $\mathbf{t}' = \mathbf{t}_1$. Then $(1.\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} = 1.((\mathbf{t}_1)\ \mathbf{r} : \mathbf{b}) \to_a (\mathbf{t}_1)\ \mathbf{r} : \mathbf{b}$.
  - $\alpha = 0$ and $\mathbf{t}' = \mathbf{0}$. Then $(0.\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} = 0.((\mathbf{t}_1)\ \mathbf{r} : \mathbf{b}) \to_a \mathbf{0} = (\mathbf{0})\ \mathbf{r} : \mathbf{b}$.
  - $\mathbf{t}_1 = \mathbf{0}$ and $\mathbf{t}' = \mathbf{0}$. Then $(\alpha.\mathbf{0})\ \mathbf{r} : \mathbf{b} = \alpha.((\mathbf{0})\ \mathbf{r} : \mathbf{b}) = \alpha.\mathbf{0} \to_a \mathbf{0} = (\mathbf{0})\ \mathbf{r} : \mathbf{b}$.
- $\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2$. Case by case on the possible $\to_\ell$-reductions of $\mathbf{t}$:
  - $\mathbf{t}' = \mathbf{t}_1' + \mathbf{t}_2$ with $\mathbf{t}_1 \to_\ell \mathbf{t}_1'$. Then $(\mathbf{t}_1 + \mathbf{t}_2)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} + (\mathbf{t}_2)\ \mathbf{r} : \mathbf{b}$ which by the induction hypothesis $\to_a$-reduces to $(\mathbf{t}_1')\ \mathbf{r} : \mathbf{b} + (\mathbf{t}_2)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}_1' + \mathbf{t}_2)\ \mathbf{r} : \mathbf{b}$.
  - $\mathbf{t}' = \mathbf{t}_1 + \mathbf{t}_2'$ with $\mathbf{t}_2 \to_\ell \mathbf{t}_2'$. Analogous to previous case.
  - $\mathbf{t}_2 = \mathbf{s}_1 + \mathbf{s}_2$ and $\mathbf{t}' = (\mathbf{t}_1 + \mathbf{s}_1) + \mathbf{s}_2$. Then $(\mathbf{t}_1 + (\mathbf{s}_1 + \mathbf{s}_2))\ \mathbf{r} : \mathbf{b} = (\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} + ((\mathbf{s}_1)\ \mathbf{r} : \mathbf{b} + (\mathbf{s}_2)\ \mathbf{r} : \mathbf{b}) \to_a ((\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} + (\mathbf{s}_1)\ \mathbf{r} : \mathbf{b}) + (\mathbf{s}_2)\ \mathbf{r} : \mathbf{b} = ((\mathbf{t}_1 + \mathbf{s}_1) + \mathbf{s}_2)\ \mathbf{r} : \mathbf{b}$.
  - $\mathbf{t}_1 = \mathbf{s}_1 + \mathbf{s}_2$ and $\mathbf{t}' = \mathbf{s}_1 + (\mathbf{s}_2 + \mathbf{t}_2)$. Analogous to previous case.
  - $\mathbf{t}' = \mathbf{t}_2 + \mathbf{t}_1$. Then $(\mathbf{t}_1 + \mathbf{t}_2)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} + (\mathbf{t}_2)\ \mathbf{r} : \mathbf{b} \to_a (\mathbf{t}_2)\ \mathbf{r} : \mathbf{b} + (\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}_2 + \mathbf{t}_1)\ \mathbf{r} : \mathbf{b}$.
  - $\mathbf{t}_1 = \alpha.\mathbf{t}_3$, $\mathbf{t}_2 = \beta.\mathbf{t}_3$ and $\mathbf{t}' = (\alpha + \beta).\mathbf{t}_3$. Then $(\alpha.\mathbf{t}_3 + \beta.\mathbf{t}_3)\ \mathbf{r} : \mathbf{b} = \alpha.((\mathbf{t}_3)\ \mathbf{r} : \mathbf{b}) + \beta.((\mathbf{t}_3)\ \mathbf{r} : \mathbf{b}) \to_a (\alpha + \beta).((\mathbf{t}_3)\ \mathbf{r} : \mathbf{b}) = ((\alpha + \beta).\mathbf{t}_3)\ \mathbf{r} : \mathbf{b}$.
  - $\mathbf{t}_1 = \alpha.\mathbf{t}_3$, $\mathbf{t}_2 = \mathbf{t}_3$ and $\mathbf{t}' = (\alpha + 1).\mathbf{t}_3$. Analogous to previous case.
  - $\mathbf{t}_1 = \mathbf{t}_2$ and $\mathbf{t}' = (1 + 1).\mathbf{t}_1$. Analogous to previous case.
- $\mathbf{t} = \mathbf{0}$. Absurd since $\mathbf{0}$ does not reduce.
- $\mathbf{t} = (\mathbf{t}_1)\ \mathbf{t}_2$. Then $((\mathbf{t}_1)\ \mathbf{t}_2)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}_1)\ \mathbf{t}_2 : \lambda g.([\![\mathbf{r}]\!])\ \lambda h.((g)\ h)\ \mathbf{b}$, which by the induction hypothesis $\to_a$-reduces to $\mathbf{t}' : \lambda g.([\![\mathbf{r}]\!])\ \lambda h.((g)\ h)\ \mathbf{b}$. We do a second induction, over $\mathbf{t}'$, to prove that $\mathbf{t}' : \lambda g.([\![\mathbf{r}]\!])\ \lambda h.((g)\ h)\ \mathbf{b} \to_a (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}' = (\mathbf{t}_1')\ \mathbf{t}_2'$, then $\mathbf{t}' : \lambda g.([\![\mathbf{r}]\!])\ \lambda h.((g)\ h)\ \mathbf{b} = ((\mathbf{t}_1')\ \mathbf{t}_2')\ \mathbf{r} : \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - $\mathbf{t}'$ cannot be a base term since from $(\mathbf{t}_1)\ \mathbf{t}_2$ it is not possible to arrive to a base term using only $\to_\ell$.
  - If $\mathbf{t}' = \alpha.\mathbf{t}_1'$, then $\mathbf{t}' : \lambda g.([\![\mathbf{r}]\!])\ \lambda h.((g)\ h)\ \mathbf{b} = \alpha.\mathbf{t}_1' : \lambda g.([\![\mathbf{r}]\!])\ \lambda h.((g)\ h)\ \mathbf{b} = \alpha.(\mathbf{t}_1' : \lambda g.([\![\mathbf{r}]\!])\ \lambda h.((g)\ h)\ \mathbf{b})$ which $\to_a$-reduces by the induction hypothesis to $\alpha.((\mathbf{t}_1')\ \mathbf{r} : \mathbf{b}) = (\alpha.\mathbf{t}_1')\ \mathbf{r} : \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}' = \mathbf{t}_1' + \mathbf{t}_2'$, then the term $\mathbf{t}' : \lambda g.([\![\mathbf{r}]\!])\ \lambda h.((g)\ h)\ \mathbf{b} = \mathbf{t}_1' + \mathbf{t}_2' : \lambda g.([\![\mathbf{r}]\!])\ \lambda h.((g)\ h)\ \mathbf{b} = \mathbf{t}_1' : \lambda g.([\![\mathbf{r}]\!])\ \lambda h.((g)\ h)\ \mathbf{b} + \mathbf{t}_2' : \lambda g.([\![\mathbf{r}]\!])\ \lambda h.((g)\ h)\ \mathbf{b}$ which $\to_a$-reduces by the induction hypothesis to $(\mathbf{t}_1')\ \mathbf{r} : \mathbf{b} + (\mathbf{t}_2')\ \mathbf{r} : \mathbf{b} = (\mathbf{t}_1' + \mathbf{t}_2')\ \mathbf{r} : \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.

∗ If $\mathbf{t}' = \mathbf{0}$ then $\mathbf{t}' : \lambda g.([\![\mathbf{r}]\!])\ \lambda h.((g)\ h)\ \mathbf{b} = \mathbf{0} : \lambda g.([\![\mathbf{r}]\!])\ \lambda h.((g)\ h)\ \mathbf{b} = \mathbf{0} = (\mathbf{0})\ \mathbf{r} : \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$

□

A.5 Proof of Lemma 2.3.15

Lemma 2.3.15. If $\mathbf{t} \to_{\ell+\beta} \mathbf{r}$ then $\forall \mathbf{b}$ base term, $\mathbf{t} : \mathbf{b} \to^*_{a+\beta} \mathbf{r} : \mathbf{b}$.

Proof. Case by case on the rules of λ_{lin}.

Rule β_b

$$\begin{array}{rcl}
(\lambda x.\mathbf{t})\ \mathbf{b}' : \mathbf{b} & = & \mathbf{b}' : \lambda f.((\Psi(\lambda x.\mathbf{t}))\ f)\ \mathbf{b} \\
& = & (\lambda f.((\Psi(\lambda x.\mathbf{t}))\ f)\ \mathbf{b})\ \Psi(\mathbf{b}') \\
& \to_{\beta_n} & ((\Psi(\lambda x.\mathbf{t}))\,\Psi(\mathbf{b}'))\ \mathbf{b} \\
& = & ((\lambda x.[\![\mathbf{t}]\!])\,\Psi(\mathbf{b}'))\ \mathbf{b} \\
& \to_{\beta_n} & [\![\mathbf{t}]\!][\Psi(\mathbf{b})/x]\ \mathbf{b} \\
\text{(Lemma 2.3.5)} & = & [\![\mathbf{t}[\mathbf{b}/x]]\!]\ \mathbf{b} \\
\text{(Lemma 2.3.13)} & \to^*_{a+\beta} & \mathbf{t}[\mathbf{b}/x] : \mathbf{b}
\end{array}$$

Algebraic rules If $\mathbf{t} \to_\ell \mathbf{r}$, then by Lemma 2.3.14 $\mathbf{t} : \mathbf{b} \to^*_a \mathbf{r} : \mathbf{b}$ which implies that $\mathbf{t} : \mathbf{b} \to^*_{a+\beta} \mathbf{r} : \mathbf{b}$.

Rules ξ **and** $\xi_{\lambda_{lin}}$ If $\mathbf{t} \to_\ell \mathbf{t}'$, then we use Lemma 2.3.14 to close the case. Assume $\mathbf{t} \to_{\beta_b} \mathbf{t}'$, and assume that for all $\mathbf{b}$ base term, $\mathbf{t} : \mathbf{b} \to^*_{a+\beta} \mathbf{t}' : \mathbf{b}$. We show that the result also holds for each contextual rule.

- $\mathbf{t} + \mathbf{r} \to_{\beta_b} \mathbf{t}' + \mathbf{r}$. Then $\mathbf{t} + \mathbf{r} : \mathbf{b} = \mathbf{t} : \mathbf{b} + \mathbf{r} : \mathbf{b} \to^*_{a+\beta} \mathbf{t}' : \mathbf{b} + \mathbf{r} : \mathbf{b} = \mathbf{t}' + \mathbf{r} : \mathbf{b}$.
- $\mathbf{r} + \mathbf{t} \to_{\beta_b} \mathbf{r} + \mathbf{t}'$, analogous to previous case.
- $\alpha.\mathbf{t} \to_{\beta_b} \alpha.\mathbf{t}'$. Then $\alpha.\mathbf{t} : \mathbf{b} = \alpha.(\mathbf{t} : \mathbf{b}) \to^*_{a+\beta} \alpha.(\mathbf{t}' : \mathbf{b}) = \alpha.\mathbf{t}' : \mathbf{b}$.
- $(\mathbf{v})\ \mathbf{t} \to_{\beta_b} (\mathbf{v})\ \mathbf{t}'$. Case by case:
  - $\mathbf{v} = \mathbf{b}'$. Then $(\mathbf{b}')\ \mathbf{t} : \mathbf{b} = \mathbf{t} : \lambda f.((\Psi(\mathbf{b}'))\ f)\ \mathbf{b}$ which $\to_{a+\beta}$-reduces by the induction hypothesis to $\mathbf{t}' : \lambda f.((\Psi(\mathbf{b}'))\ f)\ \mathbf{b} = (\mathbf{b}')\ \mathbf{t}' : \mathbf{b}$.
  - $\mathbf{v} = \mathbf{0}$. Then $(\mathbf{0})\ \mathbf{t} : \mathbf{b} = \mathbf{0} = (\mathbf{0})\ \mathbf{t}' : \mathbf{b}$.
  - $\mathbf{v} = \alpha.\mathbf{w}$. Then $(\alpha.\mathbf{w})\ \mathbf{t} : \mathbf{b} = \alpha.(\mathbf{w})\ \mathbf{t} : \mathbf{b} = \alpha.((\mathbf{w})\ \mathbf{t} : \mathbf{b})$ which $\to_{a+\beta}$-reduces by the induction hypothesis to $\alpha.((\mathbf{w})\ \mathbf{t}' : \mathbf{b}) = \alpha.(\mathbf{w})\ \mathbf{t}' : \mathbf{b} = (\alpha.\mathbf{w})\ \mathbf{t}' : \mathbf{b}$.
  - $\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_2$. Then $(\mathbf{v}_1 + \mathbf{v}_2)\ \mathbf{t} : \mathbf{b} = (\mathbf{v}_1)\ \mathbf{t} + (\mathbf{v}_2)\ \mathbf{t} : \mathbf{b} = (\mathbf{v}_1)\ \mathbf{t} : \mathbf{b} + (\mathbf{v}_2)\ \mathbf{t} : \mathbf{b}$ which $\to_{a+\beta}$-reduces by the induction hypothesis to $(\mathbf{v}_1)\ \mathbf{t}' : \mathbf{b} + (\mathbf{v}_2)\ \mathbf{t}' : \mathbf{b} = (\mathbf{v}_1)\ \mathbf{t}' + (\mathbf{v}_2)\ \mathbf{t}' : \mathbf{b} = (\mathbf{v}_1 + \mathbf{v}_2)\ \mathbf{t}' : \mathbf{b}$.
- $(\mathbf{t})\ \mathbf{r} \to_{\beta_b} (\mathbf{t}')\ \mathbf{r}$ Case by case:
  - $\mathbf{t} = \mathbf{b}'$. Absurd since a base term cannot reduce.
  - $\mathbf{t} = \alpha.\mathbf{t}_1$. The only possible $\to_{\beta_b}$-reduction from $\mathbf{t}$ is $\mathbf{t}' = \alpha.\mathbf{t}'_1$ with $\mathbf{t}_1 \to_{\beta_b} \mathbf{t}'_1$. Then $(\alpha.\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} = \alpha.(\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} = \alpha.((\mathbf{t}_1)\ \mathbf{r} : \mathbf{b})$ which by the induction hypothesis $\to_{a+\beta}$-reduces to $\alpha.((\mathbf{t}'_1)\ \mathbf{r} : \mathbf{b}) = \alpha.(\mathbf{t}'_1)\ \mathbf{r} : \mathbf{b} = (\alpha.\mathbf{t}'_1)\ \mathbf{r} : \mathbf{b}$.

- $\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2$. Case by case on the possible $\rightarrow_{\beta_b}$-reductions of $\mathbf{t}$:
  - $\mathbf{t}' = \mathbf{t}_1' + \mathbf{t}_2$ with $\mathbf{t}_1 \rightarrow_{\beta_b} \mathbf{t}_1'$. Then $(\mathbf{t}_1 + \mathbf{t}_2)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} + (\mathbf{t}_2)\ \mathbf{r} : \mathbf{b}$ which by the induction hypothesis $\rightarrow_{a+\beta}$-reduces to $(\mathbf{t}_1')\ \mathbf{r} : \mathbf{b} + (\mathbf{t}_2)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}_1' + \mathbf{t}_2)\ \mathbf{r} : \mathbf{b}$.
  - $\mathbf{t}' = \mathbf{t}_1 + \mathbf{t}_2'$ with $\mathbf{t}_2 \rightarrow_{\beta_b} \mathbf{t}_2'$. Analogous to previous case.
- $\mathbf{t} = \mathbf{0}$. Absurd since $\mathbf{0}$ does not reduce.
- $\mathbf{t} = (\mathbf{t}_1)\ \mathbf{t}_2$. Then $((\mathbf{t}_1)\ \mathbf{t}_2)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}_1)\ \mathbf{t}_2 : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b}$, which by the induction hypothesis $\rightarrow_{a+\beta}$-reduces to $\mathbf{t}' : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b}$. We do a second induction, over $\mathbf{t}'$, to prove that $\mathbf{t}' : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} \rightarrow_{a+\beta} (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}' = (\mathbf{t}_1')\ \mathbf{t}_2'$, then $\mathbf{t}' : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} = ((\mathbf{t}_1')\ \mathbf{t}_2')\ \mathbf{r} : \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}'$ is a base term, then $\mathbf{t}' : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} = (\lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b})\ \Psi(\mathbf{t}')$ which $\rightarrow_{a+\beta}$-reduces to $(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((\Psi(\mathbf{t}'))\ h)\ \mathbf{b}$ which, by Lemma 2.3.13, $\rightarrow_{a+\beta}$-reduces $\mathbf{r} : \lambda h.((\Psi(\mathbf{t}'))\ h)\ \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}' = \alpha.\mathbf{t}_1'$, then $\mathbf{t}' : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} = \alpha.\mathbf{t}_1' : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} = \alpha.(\mathbf{t}_1' : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b})$ which $\rightarrow_{a+\beta}$-reduces by the induction hypothesis to $\alpha.((\mathbf{t}_1')\ \mathbf{r} : \mathbf{b}) = (\alpha.\mathbf{t}_1')\ \mathbf{r} : \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}' = \mathbf{t}_1' + \mathbf{t}_2'$, then $\mathbf{t}' : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} = \mathbf{t}_1' + \mathbf{t}_2' : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} = \mathbf{t}_1' : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} + \mathbf{t}_2' : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b}$ which $\rightarrow_{a+\beta}$-reduces by the induction hypothesis to $(\mathbf{t}_1')\ \mathbf{r} : \mathbf{b} + (\mathbf{t}_2')\ \mathbf{r} : \mathbf{b} = (\mathbf{t}_1' + \mathbf{t}_2')\ \mathbf{r} : \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}' = \mathbf{0}$ then $\mathbf{t}' : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} = \mathbf{0} : \lambda g.(\llbracket \mathbf{r} \rrbracket)\ \lambda h.((g)\ h)\ \mathbf{b} = \mathbf{0} = (\mathbf{0})\ \mathbf{r} : \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$

□

A.6 Proof of Lemma 2.3.20

Lemma 2.3.20. $\{\!|\mathbf{t}[\mathbf{r}/x]|\!\} = \{\!|\mathbf{t}|\!\}[\{\!|\mathbf{r}|\!\}/x]$.

Proof. Structural induction on $\mathbf{t}$.

- $\mathbf{t} = x$. Then $\{\!|x[\mathbf{r}/x]|\!\} = \{\!|\mathbf{r}|\!\} = x[\{\!|\mathbf{r}|\!\}/x] = \{\!|x|\!\}[\{\!|\mathbf{r}|\!\}/x]$.
- $\mathbf{t} = y$. Then $\{\!|y[\mathbf{r}/x]|\!\} = y = \{\!|y|\!\}[\{\!|\mathbf{r}|\!\}/x]$.
- $\mathbf{t} = \mathbf{0}$. Analogous to previous case.
- $\mathbf{t} = \lambda y.\mathbf{t}'$. Then

$$\begin{aligned}
\{\!|(\lambda y.\mathbf{t}')[\mathbf{r}/x]|\!\} &= \{\!|\lambda y.(\mathbf{t}'[\mathbf{r}/x])|\!\} \\
&= \lambda f.\,(f)\ \lambda y.\{\!|\mathbf{t}'[\mathbf{r}/x]|\!\} \\
&\quad \text{by the induction hypothesis} \\
&= \lambda f.\,(f)\ \lambda y.\{\!|\mathbf{t}'|\!\}[\{\!|\mathbf{r}|\!\}/x] \\
&= (\lambda f.\,(f)\ \lambda y.\{\!|\mathbf{t}'|\!\})[\{\!|\mathbf{r}|\!\}/x] \\
&= \{\!|\mathbf{t}|\!\}[\{\!|\mathbf{r}|\!\}/x]
\end{aligned}$$

- $\mathbf{t} = (\mathbf{r}_1)\ \mathbf{r}_2$. Then

$$\begin{aligned}
\{\!|\mathbf{t}[\mathbf{r}/x]|\!\} &= \{\!|((\mathbf{r}_1)\ \mathbf{r}_2)[\mathbf{r}/x]|\!\} \\
&= \{\!|(\mathbf{r}_1[\mathbf{r}/x])\ \mathbf{r}_2[\mathbf{r}/x]|\!\} \\
&= \lambda f.(\{\!|\mathbf{r}_1[\mathbf{r}/x]|\!\})\ \lambda g.((g)\ \{\!|\mathbf{r}_2[\mathbf{r}/x]|\!\})\ f \\
&\quad \text{by the induction hypothesis} \\
&= \lambda f.(\{\!|\mathbf{r}_1|\!\}[\{\!|\mathbf{r}|\!\}/x])\ \lambda g.((g)\ \{\!|\mathbf{r}_2|\!\}[\{\!|\mathbf{r}|\!\}/x])\ f \\
&= (\lambda f.(\{\!|\mathbf{r}_1|\!\})\ \lambda g.((g)\ \{\!|\mathbf{r}_2|\!\})\ f)[\{\!|\mathbf{r}|\!\}/x] \\
&= \{\!|(\mathbf{r}_1)\ \mathbf{r}_2|\!\}[\{\!|\mathbf{r}|\!\}/x] \\
&= \{\!|\mathbf{t}|\!\}[\{\!|\mathbf{r}|\!\}/x]
\end{aligned}$$

- $\mathbf{t} = \alpha.\mathbf{t}'$. Then

$$\begin{aligned}
\{\!|\mathbf{t}[\mathbf{r}/x]|\!\} &= \{\!|(\alpha.\mathbf{t}')[\mathbf{r}/x]|\!\} \\
&= \{\!|\alpha.(\mathbf{t}'[\mathbf{r}/x])|\!\} \\
&= \lambda f.(\alpha.\{\!|\mathbf{t}'[\mathbf{r}/x]|\!\})\!\}\ f \\
&\quad \text{by the induction hypothesis} \\
&= \lambda f.(\alpha.\{\!|\mathbf{t}'|\!\}[\{\!|\mathbf{r}|\!\}/x])\ f \\
&= \lambda f.(\alpha.\{\!|\mathbf{t}'|\!\})\ f[\{\!|\mathbf{r}|\!\}/x] \\
&= \{\!|\alpha.\mathbf{t}'|\!\}[\{\!|\mathbf{r}|\!\}/x] \\
&= \{\!|\mathbf{t}|\!\}[\{\!|\mathbf{r}|\!\}/x]
\end{aligned}$$

- $\mathbf{t} = \mathbf{r}_1 + \mathbf{r}_2$. Then

$$\begin{aligned}
\{\!|\mathbf{t}[\mathbf{r}/x]|\!\} &= \{\!|(\mathbf{r}_1 + \mathbf{r}_2)[\mathbf{r}/x]|\!\} \\
&= \{\!|\mathbf{r}_1[\mathbf{r}/x] + \mathbf{r}_2[\mathbf{r}/x]|\!\} \\
&= \lambda f.(\{\!|\mathbf{r}_1[\mathbf{r}/x]|\!\} + \{\!|\mathbf{r}_2[\mathbf{r}/x]|\!\})\ f \\
&\quad \text{by the induction hypothesis} \\
&= \lambda f.(\{\!|\mathbf{r}_1|\!\}[\{\!|\mathbf{r}|\!\}/x] + \{\!|\mathbf{r}_2|\!\}[\{\!|\mathbf{r}|\!\}/x])\ f \\
&= \lambda f.((\{\!|\mathbf{r}_1|\!\} + \{\!|\mathbf{r}_2|\!\})\ f[\{\!|\mathbf{r}|\!\}/x] \\
&= \{\!|\mathbf{r}_1 + \mathbf{r}_2|\!\}[\{\!|\mathbf{r}|\!\}/x] \\
&= \{\!|\mathbf{t}|\!\}[\{\!|\mathbf{r}|\!\}/x]
\end{aligned}$$

□

A.7 Proof of Lemma 2.3.26

Lemma 2.3.26. If $\mathbf{b}$ is a base term, then for any closed term $\mathbf{t}$, $(\{\!|\mathbf{t}|\!\})\ \mathbf{b} \to^*_{\ell+\beta} \mathbf{t} : \mathbf{b}$.

Proof. Structural induction on $\mathbf{t}$.

- $\mathbf{t} = \lambda x.\mathbf{r}$. Then $(\{\!|\lambda x.\mathbf{r}|\!\})\ \mathbf{b} = (\lambda f.\,(f)\ \lambda x.\{\!|\mathbf{r}|\!\})\ \mathbf{b} = (\lambda f.\,(f)\ \Phi(\lambda x.\mathbf{r}))\ \mathbf{b} \to_{\ell+\beta} (\mathbf{b})\ \Phi(\lambda x.\mathbf{r}) = \lambda x.\mathbf{r} : \mathbf{b}$.

- $\mathbf{t} = \mathbf{0}$. Then $(\{\!|\mathbf{0}|\!\})\ \mathbf{b} = (\lambda f.(\mathbf{0})\ f)\ \mathbf{b} \to_{\ell+\beta} (\mathbf{0})\ \mathbf{b} \to_{\ell+\beta} \mathbf{0} = \mathbf{0} : \mathbf{b}$.
- $\mathbf{t} = \mathbf{t}' + \mathbf{r}$. Then $(\{\!|\mathbf{t}' + \mathbf{r}|\!\})\ \mathbf{b} = (\lambda f.(\{\!|\mathbf{t}'|\!\} + \{\!|\mathbf{r}|\!\})\ f)\ \mathbf{b} \to_{\ell+\beta} (\{\!|\mathbf{t}'|\!\} + \{\!|\mathbf{r}|\!\})\ \mathbf{b}$ which $\to_{\ell+\beta}$-reduces by the induction hypothesis to $\mathbf{t}' : \mathbf{b} + \mathbf{r} : \mathbf{b} = \mathbf{t}' + \mathbf{r} : \mathbf{b}$.
- $\mathbf{t} = \alpha.\mathbf{r}$. Then $(\{\!|\alpha.\mathbf{r}|\!\})\ \mathbf{b} = (\lambda f.(\alpha.\{\!|\mathbf{r}|\!\})\ f)\ \mathbf{b} \to_{\ell+\beta} (\alpha.\{\!|\mathbf{r}|\!\})\ \mathbf{b} \to_{\ell+\beta} \alpha.(\{\!|\mathbf{r}|\!\})\ \mathbf{b})$ which $\to_{\ell+\beta}$-reduces by the induction hypothesis to $\alpha.(\mathbf{r} : \mathbf{b}) = \alpha.\mathbf{r} : \mathbf{b}$.
- $\mathbf{t} = (\mathbf{t}')\ \mathbf{r}$. Then $(\{\!|(\mathbf{t}')\ \mathbf{r}|\!\})\ \mathbf{b} = (\lambda f.(\{\!|\mathbf{t}'|\!\})\ \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ f)\ \mathbf{b} \to_{\ell+\beta} (\{\!|\mathbf{t}'|\!\})\ \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b}$. Note that $\lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b}$ is a value, so by the induction hypothesis the above term reduces to $\mathbf{t}' : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b}$. We do a second induction, over $\mathbf{t}'$, to prove that $\mathbf{t}' : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} \to^*_{\ell+\beta} (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}' = (\mathbf{t}_1)\ \mathbf{t}_2$, then $\mathbf{t}' : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} = ((\mathbf{t}_1)\ \mathbf{t}_2)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}'$ is a base term, then $\mathbf{t}' : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} = (\lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b})\ \Phi(\mathbf{t}') \to_{\ell+\beta} ((\Phi(\mathbf{t}'))\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}' = \alpha.\mathbf{t}_1$, then $\alpha.\mathbf{t}_1 : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} = \alpha.(\mathbf{t}_1 : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b})$ which $\to^*_{\ell+\beta}$-reduces by the induction hypothesis to $\alpha.((\mathbf{t}_1)\ \mathbf{r} : \mathbf{b}) = (\alpha.\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}' = \mathbf{t}_1 + \mathbf{t}_2$, then $\mathbf{t}' : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} = \mathbf{t}_1 + \mathbf{t}_2 : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} = \mathbf{t}_1 : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} + \mathbf{t}_2 : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b}$ which $\to^*_{\ell+\beta}$-reduces by the induction hypothesis to $(\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} + (\mathbf{t}_2)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}_1 + \mathbf{t}_2)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}' = \mathbf{0}$ then $\mathbf{t}' : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} = \mathbf{0} : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} = \mathbf{0} = (\mathbf{0})\ \mathbf{r} : \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$

□

A.8 Proof of Lemma 2.3.27

Lemma 2.3.27. If $\mathbf{t} \to_{a+\beta} \mathbf{r}$ then $\forall \mathbf{b}$ base term, $\mathbf{t} : \mathbf{b} \to^*_{\ell+\beta} \mathbf{r} : \mathbf{b}$.

Proof. Case by case on the rules of λ_{alg}.

Rule β_b

$$\begin{array}{rcl}
(\lambda x.\mathbf{t})\ \mathbf{r} : \mathbf{b} & = & ((\Phi(\lambda x.\mathbf{t}))\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} \\
& = & ((\lambda x.\{\!|\mathbf{t}|\!\})\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} \\
\text{(since } \{\!|\mathbf{r}|\!\} \text{ is a base term)} & \to_{\ell+\beta} & \{\!|\mathbf{t}|\!\}[\{\!|\mathbf{r}|\!\}/x]\ \mathbf{b} \\
\text{(Lemma 2.3.20)} & = & \{\!|\mathbf{t}[\mathbf{r}/x]|\!\}\ \mathbf{b} \\
\text{(Lemma 2.3.26)} & \to^*_{\ell+\beta} & \mathbf{t}[\mathbf{r}/x] : \mathbf{b}
\end{array}$$

Rules A

- Let $(\mathbf{t} + \mathbf{r})\ \mathbf{s} \to_{a+\beta} (\mathbf{t})\ \mathbf{s} + (\mathbf{r})\ \mathbf{s}$. $(\mathbf{t} + \mathbf{r})\ \mathbf{s} : \mathbf{b} = ((\mathbf{t})\ \mathbf{s} + (\mathbf{r})\ \mathbf{s}) : \mathbf{b}$.
- Let $(\alpha.\mathbf{t})\ \mathbf{r} \to_{a+\beta} \alpha.(\mathbf{t})\ \mathbf{r}$. $(\alpha.\mathbf{t})\ \mathbf{r} : \mathbf{b} = \alpha.(\mathbf{t})\ \mathbf{r} : \mathbf{b}$.
- Let $(\mathbf{0})\ \mathbf{r} \to_{a+\beta} \mathbf{0}$. $(\mathbf{0})\ \mathbf{r} : \mathbf{b} = \mathbf{0} = \mathbf{0} : \mathbf{b}$

Rules F **and** S

- $\alpha.(\mathbf{t} + \mathbf{r}) \to_{a+\beta} \alpha.\mathbf{t} + \alpha.\mathbf{r}$. Then $\alpha.(\mathbf{t} + \mathbf{r}) : \mathbf{b} = \alpha.(\mathbf{t} : \mathbf{b} + \mathbf{r} : \mathbf{b}) \to_{\ell+\beta} \alpha.(\mathbf{t} : \mathbf{b}) + \alpha.(\mathbf{r} : \mathbf{b}) = \alpha.\mathbf{t} + \alpha.\mathbf{r} : \mathbf{b}$.

- $\alpha.\mathbf{t} + \beta.\mathbf{t} \to_{a+\beta} (\alpha+\beta).\mathbf{t}$. Then $\alpha.\mathbf{t} + \beta.\mathbf{t} : \mathbf{b} = \alpha.(\mathbf{t} : \mathbf{b}) + \beta.(\mathbf{t} : \mathbf{b}) \to_{\ell+\beta} (\alpha+\beta).(\mathbf{t} : \mathbf{b}) = (\alpha+\beta).\mathbf{t} : \mathbf{b}$.
- $\alpha.\mathbf{t} + \mathbf{t} \to_{a+\beta} (\alpha+1).\mathbf{t}$. Then $\alpha.\mathbf{t} + \mathbf{t} : \mathbf{b} = \alpha.\mathbf{t} : \mathbf{b} + \mathbf{t} : \mathbf{b} = \alpha.(\mathbf{t} : \mathbf{b}) + \mathbf{t} : \mathbf{b} \to_{\ell+\beta} (\alpha+1).(\mathbf{t} : \mathbf{b}) = (\alpha+1).\mathbf{t} : \mathbf{b}$.
- $\mathbf{t} + \mathbf{t} \to_{a+\beta} (1+1).\mathbf{t}$. Then $\mathbf{t} + \mathbf{t} : \mathbf{b} = \mathbf{t} : \mathbf{b} + \mathbf{t} : \mathbf{b} \to_{\ell+\beta} (1+1).(\mathbf{t} : \mathbf{b}) = (1+1).\mathbf{t} : \mathbf{b}$.
- $\mathbf{0} + \mathbf{t} \to_{a+\beta} \mathbf{t}$. Then $\mathbf{0} + \mathbf{t} : \mathbf{b} = (\mathbf{0} : \mathbf{b}) + (\mathbf{t} : \mathbf{b}) = \mathbf{0} + (\mathbf{t} : \mathbf{b}) \to_{\ell+\beta} \mathbf{t} : \mathbf{b}$.
- $\alpha.(\beta.\mathbf{t}) \to_{a+\beta} (\alpha \times \beta).\mathbf{t}$. Then $\alpha.(\beta.\mathbf{t}) : \mathbf{b} = \alpha.(\beta.\mathbf{t} : \mathbf{b}) = \alpha.(\beta.(\mathbf{t} : \mathbf{b})) \to_{\ell+\beta} (\alpha \times \beta).(\mathbf{t} : \mathbf{b}) = (\alpha \times \beta).\mathbf{t} : \mathbf{b}$.
- $1.\mathbf{t} \to_{a+\beta} M$. Then $1.\mathbf{t} : \mathbf{b} = 1.(\mathbf{t} : \mathbf{b}) \to_{\ell+\beta} \mathbf{t} : \mathbf{b}$.
- $0.\mathbf{t} \to_{a+\beta} \mathbf{0}$. Then $0.\mathbf{t} : \mathbf{b} = 0.(\mathbf{t} : \mathbf{b}) \to_{\ell+\beta} \mathbf{0} = \mathbf{0} : \mathbf{b}$.
- $\alpha.\mathbf{0} \to_{a+\beta} \mathbf{0}$. Then $\alpha.\mathbf{0} : \mathbf{b} = \alpha.(\mathbf{0} : \mathbf{b}) = \alpha.\mathbf{0} \to_{\ell+\beta} \mathbf{0} = \mathbf{0} : \mathbf{b}$.

Rules *Asso* **and** *Com*

- $\mathbf{t} + (\mathbf{r} + \mathbf{s}) \to_{a+\beta} (\mathbf{t} + \mathbf{r}) + \mathbf{s}$. Then $\mathbf{t} + (\mathbf{r} + \mathbf{s}) : \mathbf{b} = \mathbf{t} : \mathbf{b} + (\mathbf{r} + \mathbf{s} : \mathbf{b}) = \mathbf{t} : \mathbf{b} + (\mathbf{r} : \mathbf{b} + \mathbf{s} : \mathbf{b}) \to_{\ell+\beta} (\mathbf{t} : \mathbf{b} + \mathbf{r} : \mathbf{b}) + \mathbf{s} : \mathbf{b} = \mathbf{t} + \mathbf{r} : \mathbf{b} + \mathbf{s} : \mathbf{b} = (\mathbf{t} + \mathbf{r}) + \mathbf{s} : \mathbf{b}$.
- $\mathbf{t} + \mathbf{r} \to_{a+\beta} \mathbf{r} + \mathbf{t}$. Then $\mathbf{t} + \mathbf{r} : \mathbf{b} = \mathbf{t} : \mathbf{b} + \mathbf{r} : \mathbf{b} \to_{\ell+\beta} \mathbf{r} : \mathbf{b} + \mathbf{t} : \mathbf{b} = \mathbf{r} + \mathbf{t} : \mathbf{b}$.

Rules ξ Assume $\mathbf{t} \to_{a+\beta} \mathbf{t}'$, and assume that for all $\mathbf{b}$ base term, $\mathbf{t} : \mathbf{b} \to^*_{\ell+\beta} \mathbf{t}' : \mathbf{b}$. We show that the result also holds for each contextual rule.

- $\mathbf{t} + \mathbf{r} \to_{a+\beta} \mathbf{t}' + \mathbf{r}$. Then $\mathbf{t} + \mathbf{r} : \mathbf{b} = \mathbf{t} : \mathbf{b} + \mathbf{r} : \mathbf{b} \to^*_{\ell+\beta} \mathbf{t}' : \mathbf{b} + \mathbf{r} : \mathbf{b} = \mathbf{t}' + \mathbf{r} : \mathbf{b}$.
- $\mathbf{r} + \mathbf{t} \to_{a+\beta} \mathbf{r} + \mathbf{t}'$, analogous to previous case.
- $\alpha.\mathbf{t} \to_{a+\beta} \alpha.\mathbf{t}'$. Then $\alpha.\mathbf{t} : \mathbf{b} = \alpha.(\mathbf{t} : \mathbf{b}) \to^*_{\ell+\beta} \alpha.(\mathbf{t}' : \mathbf{b}) = \alpha.\mathbf{t}' : \mathbf{b}$.
- $(\mathbf{t})\ \mathbf{r} \to_{a+\beta} (\mathbf{t}')\ \mathbf{r}$ Case by case:
  - $\mathbf{t} = \mathbf{b}'$. Absurd since a base term cannot reduce.
  - $\mathbf{t} = \alpha.\mathbf{t}_1$. Case by case on the possible $\to_{a+\beta}$-reductions of $\mathbf{t}$:
    - $\mathbf{t}' = \alpha.\mathbf{t}'_1$ with $\mathbf{t}_1 \to_{a+\beta} \mathbf{t}'_1$. Then $(\alpha.\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} = \alpha.(\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} = \alpha.((\mathbf{t}_1)\ \mathbf{r} : \mathbf{b})$ which by the induction hypothesis $\to_{\ell+\beta}$-reduces to $\alpha.((\mathbf{t}'_1)\ \mathbf{r} : \mathbf{b}) = \alpha.(\mathbf{t}'_1)\ \mathbf{r} : \mathbf{b} = (\alpha.\mathbf{t}'_1)\ \mathbf{r} : \mathbf{b}$.
    - $\mathbf{t} = \alpha.(\beta.\mathbf{t}_3)$ and $\mathbf{t}' = (\alpha \times \beta).\mathbf{t}_3$. Then $(\alpha.(\beta.\mathbf{t}_3))\ \mathbf{r} : \mathbf{b} = \alpha.(\beta.((\mathbf{t}_3)\ \mathbf{r} : \mathbf{b})) \to_{\ell+\beta} (\alpha \times \beta).((\mathbf{t}_3)\ \mathbf{r} : \mathbf{b}) = ((\alpha \times \beta).\mathbf{t}_3)\ \mathbf{r} : \mathbf{b}$.
    - $\mathbf{t} = \alpha.(\mathbf{s}_1 + \mathbf{s}_2)$ and $\mathbf{t}' = \alpha.\mathbf{s}_1 + \alpha.\mathbf{s}_2$. Then $(\alpha.(\mathbf{s}_1 + \mathbf{s}_2))\ \mathbf{r} : \mathbf{b} = \alpha.((\mathbf{s}_1)\ \mathbf{r} : \mathbf{b} + (\mathbf{s}_2)\ \mathbf{r} : \mathbf{b}) \to_{\ell+\beta} \alpha.((\mathbf{s}_1)\ \mathbf{r} : \mathbf{b}) + \alpha.((\mathbf{s}_2)\ \mathbf{r} : \mathbf{b}) = (\alpha.\mathbf{s}_1 + \alpha.\mathbf{s}_2)\ \mathbf{r} : \mathbf{b}$.
    - $\alpha = 1$ and $\mathbf{t}' = \mathbf{t}_1$. Then $(1.\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} = 1.((\mathbf{t}_1)\ \mathbf{r} : \mathbf{b}) \to_{\ell+\beta} (\mathbf{t}_1)\ \mathbf{r} : \mathbf{b}$.
    - $\alpha = 0$ and $\mathbf{t}' = \mathbf{0}$. Then $(0.\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} = 0.((\mathbf{t}_1)\ \mathbf{r} : \mathbf{b}) \to_{\ell+\beta} \mathbf{0} = (\mathbf{0})\ \mathbf{r} : \mathbf{b}$.
    - $\mathbf{t}_1 = \mathbf{0}$ and $\mathbf{t}' = \mathbf{0}$. Then $(\alpha.\mathbf{0})\ \mathbf{r} : \mathbf{b} = \alpha.((\mathbf{0})\ \mathbf{r} : \mathbf{b}) = \alpha.\mathbf{0} \to_{\ell+\beta} \mathbf{0} = (\mathbf{0})\ \mathbf{r} : \mathbf{b}$.
  - $\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2$. Case by case on the possible $\to_{a+\beta}$-reductions of $\mathbf{t}$:
    - $\mathbf{t}' = \mathbf{t}'_1 + \mathbf{t}_2$ with $\mathbf{t}_1 \to_{a+\beta} \mathbf{t}'_1$. Then $(\mathbf{t}_1 + \mathbf{t}_2)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} + (\mathbf{t}_2)\ \mathbf{r} : \mathbf{b}$ which by the induction hypothesis $\to_{\ell+\beta}$-reduces to $(\mathbf{t}'_1)\ \mathbf{r} : \mathbf{b} + (\mathbf{t}_2)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}'_1 + \mathbf{t}_2)\ \mathbf{r} : \mathbf{b}$.
    - $\mathbf{t}' = \mathbf{t}_1 + \mathbf{t}'_2$ with $\mathbf{t}_2 \to_{a+\beta} \mathbf{t}'_2$. Analogous to previous case.
    - $\mathbf{t}_2 = \mathbf{s}_1 + \mathbf{s}_2$ and $\mathbf{t}' = (\mathbf{t}_1 + \mathbf{s}_1) + \mathbf{s}_2$. Then $(\mathbf{t}_1 + (\mathbf{s}_1 + \mathbf{s}_2))\ \mathbf{r} : \mathbf{b} = (\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} + ((\mathbf{s}_1)\ \mathbf{r} : \mathbf{b} + (\mathbf{s}_2)\ \mathbf{r} : \mathbf{b}) \to_{\ell+\beta} ((\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} + (\mathbf{s}_1)\ \mathbf{r} : \mathbf{b}) + (\mathbf{s}_2)\ \mathbf{r} : \mathbf{b} = ((\mathbf{t}_1 + \mathbf{s}_1) + \mathbf{s}_2)\ \mathbf{r} : \mathbf{b}$.

  - $\mathbf{t}_1 = \mathbf{s}_1 + \mathbf{s}_2$ and $\mathbf{t}' = \mathbf{s}_1 + (\mathbf{s}_2 + \mathbf{t}_2)$. Analogous to previous case.
  - $\mathbf{t}' = \mathbf{t}_2 + \mathbf{t}_1$. Then $(\mathbf{t}_1 + \mathbf{t}_2)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} + (\mathbf{t}_2)\ \mathbf{r} : \mathbf{b} \to_{\ell+\beta} (\mathbf{t}_2)\ \mathbf{r} : \mathbf{b} + (\mathbf{t}_1)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}_2 + \mathbf{t}_1)\ \mathbf{r} : \mathbf{b}$.
  - $\mathbf{t}_1 = \alpha.\mathbf{t}_3$, $\mathbf{t}_2 = \beta.\mathbf{t}_3$ and $\mathbf{t}' = (\alpha + \beta).\mathbf{t}_3$. Then $(\alpha.\mathbf{t}_3 + \beta.\mathbf{t}_3)\ \mathbf{r} : \mathbf{b} = \alpha.((\mathbf{t}_3)\ \mathbf{r} : \mathbf{b}) + \beta.((\mathbf{t}_3)\ \mathbf{r} : \mathbf{b}) \to_{\ell+\beta} (\alpha + \beta).((\mathbf{t}_3)\ \mathbf{r} : \mathbf{b}) = ((\alpha + \beta).\mathbf{t}_3)\ \mathbf{r} : \mathbf{b}$.
  - $\mathbf{t}_1 = \alpha.\mathbf{t}_3$, $\mathbf{t}_2 = \mathbf{t}_3$ and $\mathbf{t}' = (\alpha + 1).\mathbf{t}_3$. Analogous to previous case.
  - $\mathbf{t}_1 = \mathbf{t}_2$ and $\mathbf{t}' = (1 + 1).\mathbf{t}_1$. Analogous to previous case.

- $\mathbf{t} = \mathbf{0}$. Absurd since $\mathbf{0}$ does not reduce.
- $\mathbf{t} = (\mathbf{t}_1)\ \mathbf{t}_2$. Then $((\mathbf{t}_1)\ \mathbf{t}_2)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}_1)\ \mathbf{t}_2 : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b}$, which by the induction hypothesis $\to_{\ell+\beta}$-reduces to $\mathbf{t}' : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b}$. We do a second induction, over $\mathbf{t}'$, to prove that $\mathbf{t}' : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} \to^*_{\ell+\beta} (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}' = (\mathbf{t}'_1)\ \mathbf{t}'_2$, then $\mathbf{t}' : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} = ((\mathbf{t}'_1)\ \mathbf{t}'_2)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}'$ is a base term, then $\mathbf{t}' : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} = (\lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b})\ \Phi(\mathbf{t}')$ which $\to_{\ell+\beta}$-reduces to $((\Phi(\mathbf{t}'))\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}' = \alpha.\mathbf{t}'_1$, then $\alpha.\mathbf{t}'_1 : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} = \alpha.(\mathbf{t}'_1 : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b})$ which $\to^*_{\ell+\beta}$-reduces by the induction hypothesis to $\alpha.((\mathbf{t}'_1)\ \mathbf{r} : \mathbf{b}) = (\alpha.\mathbf{t}'_1)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}' = \mathbf{t}'_1 + \mathbf{t}'_2$, then $\mathbf{t}' : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} = \mathbf{t}'_1 + \mathbf{t}'_2 : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} = \mathbf{t}'_1 : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} + \mathbf{t}'_2 : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b}$ which $\to^*_{\ell+\beta}$-reduces by the induction hypothesis to $(\mathbf{t}'_1)\ \mathbf{r} : \mathbf{b} + (\mathbf{t}'_2)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}'_1 + \mathbf{t}'_2)\ \mathbf{r} : \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$.
  - If $\mathbf{t}' = \mathbf{0}$ then $\mathbf{t}' : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} = \mathbf{0} : \lambda g.((g)\ \{\!|\mathbf{r}|\!\})\ \mathbf{b} = \mathbf{0} = (\mathbf{0})\ \mathbf{r} : \mathbf{b} = (\mathbf{t}')\ \mathbf{r} : \mathbf{b}$

□

A.9 COQ proof of Lemma 2.2.1

The proof of the local confluence of the algebraic fragments of $\lambda^{\to}_{lin}$ and $\lambda^{\to}_{alg}$ are sufficiently monotonous so that one can ask a proof assistant to do them. For this purpose we use the library `LocConf` setting up some convenient tactics. The interested reader can find the whole set of files in [Valiron, 2011a]:

- `RW.v`, `ListTac.v` and `LocConfTac.v` are the files containing the library;
- `Llin.v` and `Lalg.v` respectively contain the proofs for $\lambda^{\to}_{lin}$ and $\lambda^{\to}_{alg}$.

To compile the files, you will need COQ v.8.2pl1. and the Ssreflect extension v.1.2. Proceed with a flavour of:

```
$ coqc RW.v ListTac.v LocConfTac.v
$ coqc Llin.v
$ coqc Lalg.v
```

To check that no particular assumption were made, you can use

```
$ coqchk -o Llin
$ coqchk -o Lalg
```

A.9.1 Summary of the proof.

We summarise the content of `Llin.v` and `Lalg.v`. First we define the set of scalars.

```
Variable scalar : Set.

Variable Sadd : scalar -> scalar -> scalar.
Variable Smul : scalar -> scalar -> scalar.
Variable S0 : scalar.
Variable S1 : scalar.

Notation "A + B" := (Sadd A B) : scalar_scope.
Notation "A * B" := (Smul A B) : scalar_scope.

Open Scope scalar_scope.

Hypothesis S_0_1_dec : ~ S1 = S0.
Hypothesis S_0_lunit : forall a, S0 + a = a.
Hypothesis S_0_lelim : forall a, S0 * a = S0.
Hypothesis S_1_lunit : forall a, S1 * a = a.
Hypothesis S_rdistrib : forall a b c, a*(b+c) = (a*b)+(a*c).
Hypothesis S_ldistrib : forall a b c, (a+b)*c = (a*c)+(b*c).
Hypothesis S_add_assoc : forall a b c, (a+b)+c = a+(b+c).
Hypothesis S_mul_assoc : forall a b c, (a*b)*c = a*(b*c).
Hypothesis S_add_commut : forall a b, a+b = b+a.
Hypothesis S_mul_commut : forall a b, a*b = b*a.

Close Scope scalar_scope.
```

We then define the set of terms. Values and bases are properties on terms defined by induction.

```
Inductive term : Set :=
| T0 : term
| Tadd : term -> term -> term
| Tmul : scalar -> term -> term
| Tvar : nat -> term
| Tlambda : term -> term
| Tapply : term -> term -> term.

Notation "A +s B" := (Sadd A B) (at level 50) : term_scope.
Notation "A *s B" := (Smul A B) (at level 40) : term_scope.

Notation "A + B" := (Tadd A B) : term_scope.
Notation "A '**' B" := (Tmul A B) (at level 35) : term_scope.
Notation "@ A" := (Tvar A) (at level 10) : term_scope.
```

```
Notation "A ; B" := (Tapply A B) (at level 30) : term_scope.
Notation "\ A" := (Tlambda A) (at level 40) : term_scope.

Open Scope term_scope.

Inductive is_value : term -> Prop :=
| valT0 : is_value T0
| valTlambda : forall s, is_value (Tlambda s)
| valTvar : forall n, is_value (Tvar n)
| valTmulbase : forall a s, is_value s -> is_value (a ** s)
| valTadd : forall s t, is_value s -> is_value t -> is_value (s + t).

Inductive is_base : term -> Prop :=
| baseTlambda : forall s, is_base (Tlambda s)
| baseTvar : forall n, is_base (Tvar n).
```

The definition of local confluence is given in the file `RW.v` of the library:

```
Section RW.

(** The relation is on some terms *)
Variable term : Set.

(** It is a binary proposition *)
Variable R : term -> term -> Prop.

(** Transitivity closure of the relation *)
Inductive Rstar : term -> term -> Prop :=
| Rzero : forall r, Rstar r r
| Rcons : forall r t s, (R r s) -> (Rstar s t) -> (Rstar r t).

Definition local_confluent :=
forall r s t, R r s -> R r t -> exists u, Rstar s u /\ Rstar t u.

End RW.
```

A.9.2 $\lambda^{\rightarrow}_{lin}$

We can now set up the rewrite system of $\lambda^{\rightarrow}_{lin}$. We use the notion of base for the right-linearity of the application.

```
Section Term.

Hypothesis R : term -> term -> Prop.
```

```
(** Elementary rules *)
Definition R_T0_runit := forall t, R (t + T0) t.
Definition R_S0_anni := forall t, R (S0 ** t) T0.
Definition R_S1_unit := forall t, R (S1 ** t) t.
Definition R_T0_anni := forall a, R (a ** T0) T0.
Definition R_mul_abs := forall a b t, R (a ** (b ** t)) ((a *s b) ** t).
Definition R_ma_dist := forall a s t, R (a ** (s + t)) (a ** s + a ** t).

(** Factorization *)
Definition R_add_fact := forall a b t, R (a ** t + b ** t) ((a +s b) ** t).
Definition R_add_fact1 := forall a t, R (a ** t + t) ((a +s S1) ** t).
Definition R_add_fact11 := forall t, R (t + t) ((S1 +s S1) ** t).

(** Assoc. and commut. of addition *)
Definition R_add_com := forall s t, R (s + t) (t + s).
Definition R_add_rassoc := forall r s t, R ((r + s) + t) (r + (s + t)).
Definition R_add_lassoc := forall r s t, R (r + (s + t)) ((r + s) + t).

(** Congruence *)
Definition R_cong_mul := forall a s t, R s t -> R (a**s) (a**t).
Definition R_cong_ladd := forall u s t, R s t -> R (s+u) (t+u).
Definition R_cong_radd := forall u s t, R s t -> R (u+ s) (u+t).
Definition R_cong_lapp := forall u s t, R s t -> R (s;u) (t;u).
Definition R_cong_rapp := forall u s t, is_value u -> R s t -> R (u;s) (u;t).

(** Linearity of application *)
Definition R_add_app_ldist := forall r s t, is_value t -> R ((r + s);t) (r;t + s;t).
Definition R_mul_app_ldist := forall a r s, is_value s -> R ((a**r);s) (a**(r;s)).
Definition R_T0_app_ldist := forall s, is_value s -> R (T0;s) T0.
Definition R_add_app_rdist := forall r s t, is_base t -> R (t;(r + s)) (t;r + t;s).
Definition R_mul_app_rdist := forall a r s, is_base s -> R (s;(a**r)) (a**(s;r)).
Definition R_T0_app_rdist := forall s, is_base s -> R (s;T0) T0.

End Term.

Inductive R : term -> term -> Prop :=
| ax1 :R_T0_runit R
| ax2 :R_S0_anni R
| ax3 :R_S1_unit R
| ax4 :R_T0_anni R
| ax5 :R_mul_abs R
| ax6 :R_ma_dist R
```

```
| ax7 :R_add_fact R
| ax8 :R_add_fact1 R
| ax9 :R_add_fact11 R

| ax10 :R_add_com R
| ax11 :R_add_rassoc R
| ax12 :R_add_lassoc R

| ax13 :R_cong_mul R
| ax14 :R_cong_ladd R
| ax15 :R_cong_radd R
| ax16 :R_cong_lapp R
| ax17 :R_cong_rapp R

| ax18 :R_add_app_ldist R
| ax19 :R_mul_app_ldist R
| ax20 :R_T0_app_ldist R
| ax21 :R_add_app_rdist R
| ax22 :R_mul_app_rdist R
| ax23 :R_T0_app_rdist R.
```

The theorem stating the local confluence of `R` reads as follows:

```
Theorem R_local_confluence : forall r s t:term,
(R r s) -> (R r t) -> exists u:term, (Rstar R s u) /\ (Rstar R t u).
```

A.9.3 $\lambda_{alg}^{\rightarrow}$

The rewrite system of $\lambda_{alg}^{\rightarrow}$ is simpler, since it does not consider values.

```
Section Term.

Hypothesis R : term -> term -> Prop.

(** Elementary rules *)
Definition R_T0_runit := forall t, R (t + T0) t.
Definition R_S0_anni := forall t, R (S0 ** t) T0.
Definition R_S1_unit := forall t, R (S1 ** t) t.
Definition R_T0_anni := forall a, R (a ** T0) T0.
Definition R_mul_abs := forall a b t, R (a ** (b ** t)) ((a *s b) ** t).
Definition R_ma_dist := forall a s t, R (a ** (s + t)) (a ** s + a ** t).
(** Factorization *)
Definition R_add_fact := forall a b t, R (a ** t + b ** t) ((a +s b) ** t).
Definition R_add_fact1 := forall a t, R (a ** t + t) ((a +s S1) ** t).
Definition R_add_fact11 := forall t, R (t + t) ((S1 +s S1) ** t).
```

```
(** Assoc. and commut. of addition *)
Definition R_add_com := forall s t, R (s + t) (t + s).
Definition R_add_rassoc := forall r s t, R ((r + s) + t) (r + (s + t)).
Definition R_add_lassoc := forall r s t, R (r + (s + t)) ((r + s) + t).
(** Congruence *)
Definition R_cong_mul := forall a s t, R s t -> R (a**s) (a**t).
Definition R_cong_ladd := forall u s t, R s t -> R (s+u) (t+u).
Definition R_cong_radd := forall u s t, R s t -> R (u+ s) (u+t).
Definition R_cong_lapp := forall u s t, R s t -> R (s;u) (t;u).
(** Linearity of application *)
Definition R_add_app_ldist := forall r s t, R ((r + s);t) (r;t + s;t).
Definition R_mul_app_ldist := forall a r s, R ((a**r);s) (a**(r;s)).
Definition R_T0_app_ldist := forall s, R (T0;s) T0.

End Term.

Inductive R : term -> term -> Prop :=
| ax1 :R_T0_runit R
| ax2 :R_S0_anni R
| ax3 :R_S1_unit R
| ax4 :R_T0_anni R
| ax5 :R_mul_abs R
| ax6 :R_ma_dist R

| ax7 :R_add_fact R
| ax8 :R_add_fact1 R
| ax9 :R_add_fact11 R

| ax10 :R_add_com R
| ax11 :R_add_rassoc R
| ax12 :R_add_lassoc R

| ax13 :R_cong_mul R
| ax14 :R_cong_ladd R
| ax15 :R_cong_radd R
| ax16 :R_cong_lapp R

| ax18 :R_add_app_ldist R
| ax19 :R_mul_app_ldist R
| ax20 :R_T0_app_ldist R.
```

The statement of local confluence for `R` is the same as in the previous section:

```
Theorem R_local_confluence : forall r s t:term,
```

```
(R r s) -> (R r t) -> exists u:term, (Rstar R s u) /\ (Rstar R t u).
```

Appendix B

Proofs from Chapter 3

B.1 Proof of Lemma 3.2.2

Lemma 3.2.2 (α unit)**.** $\forall T \in \mathcal{T},\ \exists U \in \mathcal{U}, \alpha \in \mathcal{S}$ such that $T \equiv \alpha.U$.

Proof. Let $\mu(\cdot) : \mathcal{T} \to \mathbb{N}$ be a map defined inductively by

$$\begin{array}{rclrclrcl} \mu(X) & = & 0 & \mu(\forall X.\mathcal{T}) & = & 1+\mu(\mathcal{T}) & \mu(\overline{0}) & = & 0 \\ \mu(\mathcal{U} \to \mathcal{T}) & = & 0 & \mu(\alpha.\mathcal{T}) & = & 1+\mu(\mathcal{T}) & & & \end{array}$$

Then we proceed by induction over $\mu(T)$.

Basic cases Let $\mu(T) = 0$. Then

1. $T = X$, then $T \in \mathcal{U}$ and $T \equiv 1.T$.
2. $T = \mathcal{U} \to A$, then $T \in \mathcal{U}$ and $T \equiv 1.T$.
3. $T = \overline{0}$, then $\forall U \in \mathcal{U}$, $T \equiv 0.U$.

Inductive cases Let $\mu(T) = n$ and assume the lemma is valid for all A with $\mu(A) < n$. Then, the possible cases are

1. $T = \forall X.A$, then $\mu(A) = n - 1$, and so by the induction hypothesis $\exists U \in \mathcal{U}$ s,t, $A \equiv U$ or $A \equiv \alpha.U$, then $T \equiv \forall X.U \in \mathcal{U}$ or $T \equiv \forall X.\alpha.U \equiv \alpha.\forall X.U$.
2. $T = \alpha.A$, then $\mu(A) = n - 1$, and so by the induction hypothesis $\exists U \in \mathcal{U}$ s,t, $A \equiv U$ or $A \equiv \beta.U$, then $T \equiv \alpha.U$ or $T \equiv \alpha.\beta.U \equiv (\alpha \times \beta).U$.

□

B.2 Proof of Lemma 3.2.6

Lemma 3.2.6 (Arrows comparison)**.** For any types $U, V \in \mathcal{U}$ and $T, R \in \mathcal{T}$, if $V \to R \preceq U \to T$, then $\exists \vec{W}, \vec{X}\ /\ U \to T \equiv (V \to R)[\vec{W}/\vec{X}]$.

Proof. A map $(\cdot)^\circ : \mathcal{T} \to \mathcal{T}$ is defined by

$$X^\circ = X \qquad (U \to T)^\circ = U \to T \qquad (\forall X.T)^\circ = T^\circ \qquad (\alpha.T)^\circ = \alpha.T^\circ \qquad (\overline{0})^\circ = \overline{0}$$

We need two intermediate results.

1. For any types $T \in \mathcal{T}, U \in \mathcal{U}$, exists $V \in \mathcal{U}$ such that $(T[U/X])^\circ \equiv T^\circ[V/X]$.
2. For any types T, R, if $T \preceq R$ then $\exists \vec{U}, \vec{X} \;/\; R^\circ \equiv T^\circ[\vec{U}/\vec{X}]$.

Proofs

1. Structural induction on T

   $T = X$ then $(X[U/X])^\circ = U^\circ = X[U^\circ/X] = X^\circ[U^\circ/X]$.

   $T = Y$ then $(Y[U/X])^\circ = Y = Y^\circ[U/X]$.

   $T = V \to R$ then $((V \to R)[U/X])^\circ = (V[U/X] \to R[U/X])^\circ = V[U/X] \to R[U/X] = (V \to R)[U/X] = (V \to R)^\circ[U/X]$.

   $T = \forall Y.R$ then $((\forall Y.R)[U/X])^\circ = (\forall Y.R[U/X])^\circ = (R[U/X])^\circ$, which, by the induction hypothesis, is equivalent to $R^\circ[V/X] = (\forall Y.R)^\circ[V/X]$.

   $T = \overline{0}$ analogous to $T \equiv Y$.

   $T = \alpha.R$ then $(\alpha.R[U/X])^\circ = \alpha.(R[U/X])^\circ$, which, by the induction hypothesis, is equivalent to $\alpha.(R^\circ)[V/X] = (\alpha.R)^\circ[V/X]$.

2. It suffices to show this for $T \prec R$.

   Case 1 $R \equiv \forall X.T$. Then $R^\circ \equiv T^\circ$.

   Case 2 $T \equiv \forall X.S$ and $R \equiv S[U/X]$ then by the intermediate result 1 one has $R^\circ \equiv S^\circ[U/X] \equiv T^\circ[U/X]$.

Proof of the lemma. $U \to T \equiv (U \to T)^\circ$, by the intermediate result 2, this is equivalent to $(V \to R)^\circ[\vec{U}/\vec{X}] \equiv (V \to R)[\vec{U}/\vec{X}]$. □

B.3 Proof of Lemma 3.2.8

Lemma 3.2.8 (Generation lemma (app))**.** For any terms $\mathbf{t}, \mathbf{r}$, any type T, any scalar γ, any context Γ and any number $n \in \mathbb{N}$, if $S_n = \Gamma \vdash (\mathbf{t})\ \mathbf{r} : \gamma.T$, then $\exists \alpha, \beta \in \mathcal{S}$, $r, s \in \mathbb{N}$ with $\max(r, s) < n$, $U \in \mathcal{U}$ and $R \preceq T$ such that $S_r = \Gamma \vdash \mathbf{r} : \alpha.U$ and $S_s = \Gamma \vdash \mathbf{t} : \beta.U \to R$ with $\alpha \times \beta = \gamma$.

Proof. Induction on n.

1. $\dfrac{\Gamma \vdash \mathbf{t} : \beta.(U \to T) \qquad \Gamma \vdash \mathbf{r} : \alpha.U}{\Gamma \vdash (\mathbf{t})\ \mathbf{r} : (\alpha \times \beta).T} \to_E$ This is the trivial case.

2. $\dfrac{\Gamma \vdash (\mathbf{t})\ \mathbf{r} : \forall X.\gamma.T}{\Gamma \vdash (\mathbf{t})\ \mathbf{r} : \gamma.T[U/X]} \forall_E$ Since $\forall X.\gamma.T \equiv \gamma.\forall X.T$, by the induction hypothesis $\exists \alpha, \beta, r, s, U$ and $T' \geq \forall X.T$ such that $S_r = \Gamma \vdash \mathbf{t} : \beta.U \to T'$, $S_s = \Gamma \vdash \mathbf{r} : \alpha.U$, $\alpha \times \beta = \gamma$ and $\max(r, s) < n - 1$. Since $T' \preceq \forall X.T \prec T[U/X]$ then by transitivity $T' \preceq T[U/X]$.

3. $\dfrac{\dfrac{\Gamma \vdash (\mathbf{t})\ \mathbf{r} : \gamma.T}{\Gamma \vdash (\mathbf{t})\ \mathbf{r} : \forall X.\gamma.T} \forall_I}{\Gamma \vdash (\mathbf{t})\ \mathbf{r} : \gamma.\forall X.T} \equiv$ By the induction hypothesis $\exists \alpha, \beta, r, s, U$ and $T' \geq T$ such that $S_r = \Gamma \vdash \mathbf{t} : \beta.U \to T'$, $S_s = \Gamma \vdash \mathbf{r} : \alpha.U$, $\alpha \times \beta = \gamma$ and $\max(r, s) < n - 1$. By definition $T \prec \forall X.T$, so by transitivity $T' \preceq \forall X.T$.

□

B.4 Proof of Lemma 3.2.9

Lemma 3.2.9 (Generation lemma (abs))**.** For any term $\mathbf{t}$, any type T, any context Γ and any number $n \in \mathbb{N}$, if $S_n = \Gamma \vdash \lambda x.\mathbf{t} : T$ then $\exists U \in \mathcal{U}$, $R \in \mathcal{T}$ and $m < n$ such that $S_m = \Gamma, x : U \vdash \mathbf{t} : R$ and $U \to R \preceq T$.

Proof. Induction on n.

1. $\dfrac{\Gamma, x : U \vdash \mathbf{t} : T}{\Gamma \vdash \lambda x.\mathbf{t} : U \to T} \to_I$ This is the trivial case.
2. $\dfrac{\Gamma \vdash \lambda x.\mathbf{t} : \forall X.T}{\Gamma \vdash \lambda x.\mathbf{t} : T[V/X]} \forall_E$ By the induction hypothesis $\exists U, R$ such that $\Gamma, x : U \vdash \mathbf{t} : R$ and $U \to R \preceq \forall X.T \prec T[V/X]$.
3. $\dfrac{\Gamma \vdash \lambda x.\mathbf{t} : T}{\Gamma \vdash \lambda x.\mathbf{t} : \forall X.T} \forall_I$ By the induction hypothesis $\exists U, R$ such that $\Gamma, x : U \vdash \mathbf{t} : R$ and $U \to R \preceq T \prec \forall X.T$.

□

B.5 Proof of Lemma 3.2.10

Lemma 3.2.10 (Generation lemma (sc))**.** For any scalar $\alpha \neq 0$, any context Γ, any term $\mathbf{t}$, any type T and any number $n \in \mathbb{N}$, if $S_n = \Gamma \vdash \alpha.\mathbf{t} : \alpha.T$, then $\exists m < n$ such that $S_m = \Gamma \vdash \mathbf{t} : T$.

Proof. Induction on n.

1. $\dfrac{\Gamma \vdash \alpha.\mathbf{t} : \forall X.\alpha.T}{\Gamma \vdash \alpha.\mathbf{t} : \alpha.T[U/X]} \forall_E$ Since $\forall X.\alpha.T \equiv \alpha.\forall X.T$, by the induction hypothesis $\exists m < n-1$ such that $S_m = \Gamma \vdash \mathbf{t} : \forall X.T$, then by using $\forall_E$ rule, $\Gamma \vdash \mathbf{t} : T[U/X]$ and notice that $m < n-1 \Rightarrow m+1 < n$.
2. $\dfrac{\dfrac{\Gamma \vdash \alpha.\mathbf{t} : \alpha.T}{\Gamma \vdash \alpha.\mathbf{t} : \forall X.\alpha.T} \forall_I}{\Gamma \vdash \alpha.\mathbf{t} : \alpha.\forall X.T} \equiv$ By the induction hypothesis $\exists m < n-1$ such that $S_m = \Gamma \vdash \mathbf{t} : T$, then by using $\forall_I$ rule, $\Gamma \vdash \mathbf{t} : \forall X.T$ and notice that $m < n-1 \Rightarrow m+1 < n$.
3. $\dfrac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \alpha.T} s_I$ This is the trivial case.

□

B.6 Proof of Lemma 3.2.11

Lemma 3.2.11 (Generation lemma (sc-0))**.** For any context Γ, any term $\mathbf{t}$, any type T and any number $n \in \mathbb{N}$, if $S_n = \Gamma \vdash 0.\mathbf{t} : T$, then $\exists R$ and $m < n$ such that $S_m = \Gamma \vdash \mathbf{t} : R$.

Proof. Induction on n.

1. $\dfrac{\Gamma \vdash 0.\mathbf{t} : \forall X.0.T}{\Gamma \vdash 0.\mathbf{t} : 0.T[U/X]} \forall_E$ Since $\forall X.0.T \equiv 0.\forall X.T$, by the induction hypothesis $\exists R$ and $m < n-1$ such that $S_m = \Gamma \vdash \mathbf{t} : R$.
2. $\dfrac{\dfrac{\Gamma \vdash 0.\mathbf{t} : \alpha.T}{\Gamma \vdash 0.\mathbf{t} : \forall X.0.T} \forall_I}{\Gamma \vdash 0.\mathbf{t} : 0.\forall X.T} \equiv$ By the induction hypothesis $\exists R$ and $m < n-1$ such that $S_m = \Gamma \vdash \mathbf{t} : R$.
3. $\dfrac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash 0.\mathbf{t} : 0.T} s_I$ This is the trivial case.

□

B.7 Proof of Lemma 3.2.12

Lemma 3.2.12 (Generation lemma (sum))**.** For any terms $\mathbf{t}$, $\mathbf{r}$, any scalar α, any type $U \in \mathcal{U}$, any context T and any number $n \in \mathbb{N}$, if $S_n = \Gamma \vdash \mathbf{t} + \mathbf{r} : \alpha.U$, then $\exists \delta, \gamma \in \mathcal{S}$ and $r, s \in \mathbb{N}$ with $\max(r,s) < n$ such that $S_r = \Gamma \vdash \mathbf{t} : \delta.U$ and $S_s = \Gamma \vdash \mathbf{r} : \gamma.U$ with $\delta + \gamma = \alpha$.

Proof. Induction on n.

1. $\dfrac{\Gamma \vdash \mathbf{t} : \delta.T \qquad \Gamma \vdash \mathbf{r} : (\alpha - \delta).T}{\Gamma \vdash \mathbf{t} + \mathbf{r} : \alpha.T} +_I$ Then take $\gamma = \alpha - \delta$, $S_r = \Gamma \vdash \mathbf{t} : \delta.T$, $S_s = \Gamma \vdash \mathbf{r} : (\alpha - \delta).T$ and notice that $\max(r,s) < n$ and $\delta + \alpha - \delta = \alpha$.

2. $\dfrac{\Gamma \vdash \mathbf{t} + \mathbf{r} : \forall X.\alpha.T}{\Gamma \vdash \mathbf{t} + \mathbf{r} : \alpha.T[U/X]} \forall_E$ Since $\forall X.\alpha.T \equiv \alpha.\forall X.T$, by the induction hypothesis $\exists \delta, \gamma, r$ and s such that $S_s = \Gamma \vdash \mathbf{t} : \delta.\forall X.T$, $S_r = \Gamma \vdash \mathbf{r} : \gamma.\forall X.T$, $\delta + \gamma = \alpha$ and $\max(r,s) < n - 1$. Then by using $\forall_E$ rule, $\Gamma \vdash \mathbf{t} : \delta.T[U/X]$ and $\Gamma \vdash \mathbf{r} : \gamma.T[U/X]$. So, $S_{r+1} = \Gamma \vdash \mathbf{t} : \delta.T[U/X]$, $S_{s+1} = \Gamma \vdash \mathbf{r} : \gamma.T[U/X]$ and $\max(r+1, s+1) = \max(r,s) + 1 < n$.

3. $\dfrac{\dfrac{\Gamma \vdash \mathbf{t} + \mathbf{r} : \alpha.T}{\Gamma \vdash \mathbf{t} + \mathbf{r} : \forall X.\alpha.T} \forall_I}{\Gamma \vdash \mathbf{t} + \mathbf{r} : \alpha.\forall X.T} \equiv$ By the induction hypothesis $\exists \delta, \gamma, r$ and s such that $S_r = \Gamma \vdash \mathbf{t} : \delta.T$, $S_s = \Gamma \vdash \mathbf{r} : \gamma.T$, $\delta + \gamma = \alpha$ and $\max(r,s) < n - 1$. Then, by using $\forall_I$ rule, $\Gamma \vdash \mathbf{t} : \forall X.\delta.T \equiv \delta.\forall X.T$ and $\Gamma \vdash \mathbf{r} : \forall X.\gamma.T \equiv \gamma.\forall X.T$. So, $S_{r+1} = \Gamma \vdash \mathbf{t} : \delta.\forall X.T$, $S_{s+1} = \Gamma \vdash \mathbf{r} : \gamma.\forall X.T$ and $\max(r+1, s+1) = \max(r,s) + 1 < n$. □

B.8 Proof of Lemma 3.2.13

Lemma 3.2.13 (Substitution)**.** For any term $\mathbf{t}$, any base terms $\mathbf{b}$, any types $T \in \mathcal{T}$, $\vec{U} \in \mathcal{U}^n$ and any context Γ,

1. $\Gamma \vdash \mathbf{t} : T \Rightarrow \Gamma[\vec{U}/\vec{X}] \vdash \mathbf{t} : T[\vec{U}/\vec{X}]$.
2. $\{\ \Gamma, x : U \vdash \mathbf{t} : T \text{ and } \Gamma \vdash \mathbf{b} : U\ \} \Rightarrow \Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T$.

Proof of item 1. Let $S_n = \Gamma \vdash \mathbf{t} : T$. Induction on n.

1. $\dfrac{}{\Gamma, x : V \vdash x : V}\, ax$ Notice that $(\Gamma, x : V)[\vec{U}/\vec{X}] = \Gamma[\vec{U}/\vec{X}], x : V[\vec{U}/\vec{X}]$, then by ax rule, $(\Gamma, x : V)[\vec{U}/\vec{X}] \vdash x : V[\vec{U}/\vec{X}]$.

2. $\dfrac{}{\Gamma \vdash \mathbf{0} : \overline{0}}\, ax_{\overline{0}}$ Notice that $\overline{0} = \overline{0}[\vec{U}/\vec{X}]$, then by $ax_{\overline{0}}$, $\Gamma[\vec{U}/\vec{X}] \vdash \mathbf{0} : \overline{0}[\vec{U}/\vec{X}]$.

3. $\dfrac{\Gamma \vdash \mathbf{t} : \alpha.(V \to T) \qquad \Gamma \vdash \mathbf{r} : \beta.V}{\Gamma \vdash (\mathbf{t})\ \mathbf{r} : (\alpha \times \beta).T} \to_E$ By the induction hypothesis $\Gamma[\vec{U}/\vec{X}] \vdash \mathbf{t} : (\alpha.(V \to T))[\vec{U}/\vec{X}] = \alpha.V[\vec{U}/\vec{X}] \to T[\vec{U}/\vec{X}]$ also by the induction hypothesis, $\Gamma[\vec{U}/\vec{X}] \vdash \mathbf{r} : (\beta.V)[\vec{U}/\vec{X}] = \beta.V[\vec{U}/\vec{X}]$. Then by rule $\to_E$, $\Gamma[U/X] \vdash (\mathbf{t})\ \mathbf{r} : (\alpha \times \beta).T[\vec{U}/\vec{X}]$ and this type is equal to $((\alpha \times \beta).T)[\vec{U}/\vec{X}]$.

4. $\dfrac{\Gamma, x : V \vdash \mathbf{t} : T}{\Gamma \vdash \lambda x.\mathbf{t} : V \to T} \to_I$ By the induction hypothesis $(\Gamma, x : V)[\vec{U}/\vec{X}] \vdash \mathbf{t} : T[\vec{U}/\vec{X}]$. Notice that $(\Gamma, x : V)[\vec{U}/\vec{X}] = \Gamma[\vec{U}/\vec{X}], x : V[\vec{U}/\vec{X}]$, then using rule $\to_I$, one has $\Gamma[\vec{U}/\vec{X}] \vdash \lambda x.\mathbf{t} : V[\vec{U}/\vec{X}] \to T[\vec{U}/\vec{X}] = (V \to T)[\vec{U}/\vec{X}]$.

5. $\dfrac{\Gamma \vdash \mathbf{t} : \forall Y.T}{\Gamma \vdash \mathbf{t} : T[V/Y]} \forall_E$ By the induction hypothesis $\Gamma[\vec{U}/\vec{X}] \vdash \mathbf{t} : (\forall Y.T)[\vec{U}/\vec{X}]$ where $Y \notin \vec{X}$ and $Y \notin FV(\vec{U})$. Then $(\forall Y.T)[\vec{U}/\vec{X}] = \forall Y.T[\vec{U}/\vec{X}]$, and so by using $\forall_E$ rule, $\Gamma[\vec{U}/\vec{X}] \vdash \mathbf{t} : (T[\vec{U}/\vec{X}])[W/Y]$. As $Y \notin FV(\vec{U})$, then $(T[\vec{U}/\vec{X}])[W/Y] = T[\vec{U}/\vec{X}, W/Y]$. Take $W = V[\vec{U}/\vec{X}]$, then $T[\vec{U}/\vec{X}, W/Y] = (T[V/Y])[\vec{U}/\vec{X}]$.

6. $\dfrac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \mathbf{t} : \forall Y.T}\,\forall_I$ Let Z be a fresh variable. By the induction hypothesis $\Gamma[Z/Y] \vdash \mathbf{t} : T[Z/Y]$, but since $Y \notin FV(\Gamma)$, we can just write $\Gamma \vdash \mathbf{t} : T[Z/Y]$. By the induction hypothesis again $\Gamma[\vec{U}/\vec{X}] \vdash \mathbf{t} : (T[Z/Y])[\vec{U}/\vec{X}]$. Since Z is fresh, it does not appears in $\Gamma[\vec{U}/\vec{X}]$. Then by using $\forall_I$ rule, $\Gamma[\vec{U}/\vec{X}] \vdash \mathbf{t} : \forall Z.((T[Z/Y])[\vec{U}/\vec{X}])$. Notice that $\forall Z.((T[Z/Y])[\vec{U}/\vec{X}]) = (\forall Z.T[Z/Y])[\vec{U}/\vec{X}] = (\forall Y.T)[\vec{U}/\vec{X}]$.

7. $\dfrac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \alpha.T}\,s_I$ By the induction hypothesis $\Gamma[\vec{U}/\vec{X}] : \mathbf{t} : T[\vec{U}/\vec{X}]$, then by rule s_I, one has $\Gamma[\vec{U}/\vec{X}] : \alpha.\mathbf{t} : \alpha.T[\vec{U}/\vec{X}] = (\alpha.T)[\vec{U}/\vec{X}]$.

8. $\dfrac{\Gamma \vdash \mathbf{t} : \alpha.T \qquad \Gamma \vdash \mathbf{r} : \beta.T}{\Gamma \vdash \mathbf{t} + \mathbf{r} : (\alpha + \beta).T}\,+_I$ By the induction hypothesis $\Gamma[\vec{U}/\vec{X}] \vdash \mathbf{t} : (\alpha.T)[\vec{U}/\vec{X}] = \alpha.T[\vec{U}/\vec{X}]$ and also $\Gamma[\vec{U}/\vec{X}] \vdash \mathbf{r} : (\beta.T)[\vec{U}/\vec{X}] = \beta.T[\vec{U}/\vec{X}]$. So, by rule $+_I$, $\Gamma[\vec{U}/\vec{X}] \vdash \mathbf{t} + \mathbf{r} : (\alpha+\beta).T[\vec{U}/\vec{X}]$ and notice that $(\alpha+\beta).T[\vec{U}/\vec{X}] = ((\alpha + \beta).T)[\vec{U}/\vec{X}]$.

□

Proof of item 2. Let $S_n = \Gamma, x : U \vdash \mathbf{t} : T$. Induction on n.

1. $\dfrac{}{\Gamma, x : U \vdash x : U}\,ax$ Notice that $x[\mathbf{b}/x] = \mathbf{b}$, so $\Gamma \vdash \mathbf{b} : U$.

2. $\dfrac{}{\Gamma, y : V, x : U \vdash y : V}\,ax$ Notice that $y[\mathbf{b}/x] = y$, so $\Gamma, y : V \vdash y[\mathbf{b}/x] : V$ by rule ax.

3. $\dfrac{}{\Gamma, x : U \vdash \mathbf{0} : \overline{0}}\,ax_{\overline{0}}$ Notice that $\mathbf{0}[\mathbf{b}/x] = \mathbf{0}$, so $\Gamma \vdash \mathbf{0}[\mathbf{b}/x] : \overline{0}$ by rule $ax_{\overline{0}}$.

4. $\dfrac{\Gamma, x : U \vdash \mathbf{t} : \alpha.V \to T \quad \Gamma, x : U \vdash \mathbf{r} : \beta.V}{\Gamma, x : U \vdash (\mathbf{t})\ \mathbf{r} : (\alpha \times \beta).T}\,\to_E$ By the induction hypothesis $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : \alpha.V \to T$ and also $\Gamma \vdash \mathbf{r}[\mathbf{b}/x] : \beta.V$, so by rule $\to_E$, $\Gamma \vdash (\mathbf{t}[\mathbf{b}/x])\ \mathbf{r}[\mathbf{b}/x] : (\alpha \times \beta).T$. Notice that $(\mathbf{t}[\mathbf{b}/x])\ \mathbf{r}[\mathbf{b}/x]$ is equal to $((\mathbf{t})\ \mathbf{r})[\mathbf{b}/x]$.

5. $\dfrac{\Gamma, y : V, x : U \vdash \mathbf{t} : T}{\Gamma, x : U \vdash \lambda y.\mathbf{t} : V \to T}\,\to_I$ By the induction hypothesis $\Gamma, y : V \vdash \mathbf{t}[\mathbf{b}/x] : T$, then by rule $\to_I$, $\Gamma \vdash \lambda y.(\mathbf{t}[\mathbf{b}/x]) : V \to T$. Notice that $\lambda y.(\mathbf{t}[\mathbf{b}/x]) = (\lambda y.\mathbf{t})[\mathbf{b}/x]$.

6. $\dfrac{\Gamma, x : U \vdash \mathbf{t} : \forall X.T}{\Gamma, x : U \vdash \mathbf{t} : T[V/X]}\,\forall_E$ By the induction hypothesis one has $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : \forall X.T$, then by rule $\forall_E$, $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T[V/X]$.

7. $\dfrac{\Gamma, x : U \vdash \mathbf{t} : T}{\Gamma, x : U \vdash \mathbf{t} : \forall X.T}\,\forall_I$ By the induction hypothesis one has $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T$, then by rule $\forall_I$, $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : \forall X.T$.

8. $\dfrac{\Gamma, x : U \vdash \mathbf{t} : T}{\Gamma, x : U \vdash \alpha.\mathbf{t} : \alpha.T}\,s_I$ By the induction hypothesis $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T$, then by rule s_I, $\Gamma \vdash \alpha.(\mathbf{t}[\mathbf{b}/x]) : \alpha.T$. Notice that $\alpha.(\mathbf{t}[\mathbf{b}/x]) = (\alpha.\mathbf{t})[\mathbf{b}/x]$.

9. $\dfrac{\Gamma, x : U \vdash \mathbf{t} : \alpha.T \quad \Gamma, x : U \vdash \mathbf{r} : \beta.T}{\Gamma, x : U \vdash \mathbf{t} + \mathbf{r} : (\alpha + \beta).T}\,+_I$ By the induction hypothesis one has $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : \alpha.T$ and $\Gamma \vdash \mathbf{r}[\mathbf{b}/x] : \beta.T$, so by rule $+_I$, it can be deduced $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] + \mathbf{r}[\mathbf{b}/x] : (\alpha + \beta).T$. Notice that $\mathbf{t}[\mathbf{b}/x] + \mathbf{r}[\mathbf{b}/x] = (\mathbf{t} + \mathbf{r})[\mathbf{b}/x]$.

□

B.9 Proof of Lemma 3.2.15

Lemma 3.2.15 (Scaling unit)**.** For any term $\mathbf{t}$, scalar α, type T and context Γ, if $\Gamma \vdash \alpha.\mathbf{t} : T$ then there exists $U \in \mathcal{U}$ and $\gamma \in \mathcal{S}$ such that $T \equiv \alpha.\gamma.U$.

Proof. Let $S_n = \Gamma \vdash \alpha.\mathbf{t} : T$. Induction on n.

1. $\dfrac{\Gamma \vdash \mathbf{t}:T}{\Gamma \vdash \alpha.\mathbf{t}:\alpha.T}\, s_I$ By Lemma 3.2.2, $\exists U \in \mathcal{U}, \gamma \in \mathcal{S}$ such that $T \equiv \gamma.U$. Then $\alpha.T \equiv \alpha.\gamma.U$.

2. $\dfrac{\Gamma \vdash \alpha.\mathbf{t}:\forall X.T}{\Gamma \vdash \alpha.\mathbf{t}:T[V/X]}\, \forall_E$ By Lemma 3.2.2, $\exists U \in \mathcal{U}, \delta \in \mathcal{S}$ such that $T \equiv \delta.U$, then $T[V/X] \equiv \delta.U[V/X]$ and also $\forall X.T \equiv \forall X.\delta.U \equiv \delta.\forall X.U$. In addition, by the induction hypothesis $\exists U' \in \mathcal{U}, \gamma \in \mathcal{S}$ such that $\forall X.T \equiv \alpha.\gamma.U'$. Summarising: $\alpha.\gamma.U' \equiv \delta.\forall X.U$. Then, by Lemma 3.2.3, $\delta = \alpha \times \gamma$, so $T[V/X] \equiv \alpha.\gamma.U[V/X]$.

3. $\dfrac{\Gamma \vdash \alpha.\mathbf{t}:T}{\Gamma \vdash \alpha.\mathbf{t}:\forall X.T}\, \forall_I$ By the induction hypothesis $\exists U \in \mathcal{U}, \gamma \in \mathcal{S}$ such that $T \equiv \alpha.\gamma.U$, then $\forall X.T \equiv \forall X.\alpha.\gamma.U \equiv \alpha.\gamma.\forall X.U$.

□

B.10 Proof of Lemma 3.2.16

Lemma 3.2.16 (Base terms in unit)**.** For any base term $\mathbf{b}$, context Γ and type T, $\Gamma \vdash \mathbf{b}:T \Rightarrow \exists U \in \mathcal{U}$ such that $T \equiv U$.

Proof. Let $S_n = \Gamma \vdash \mathbf{b}:T$. Induction on n

1. $\dfrac{}{\Gamma, x:U \vdash x:U}\, ax$ Trivial case.

2. $\dfrac{\Gamma, x:U \vdash \mathbf{t}:T}{\Gamma \vdash \lambda x.\mathbf{t}:U \to T}\, \to_I$ Trivial case.

3. $\dfrac{\Gamma \vdash \mathbf{b}:\forall X.T}{\Gamma \vdash \mathbf{b}:T[U/X]}\, \forall_E$ By the induction hypothesis $\forall X.T \in \mathcal{U}$, so $T \in \mathcal{U}$ and then $T[U/X] \in \mathcal{U}$.

4. $\dfrac{\Gamma \vdash \mathbf{t}:T}{\Gamma \vdash \mathbf{t}:\forall X.T}\, \forall_I$ By the induction hypothesis $T \in \mathcal{U}$, so $\forall X.T \in \mathcal{U}$.

□

B.11 Proof of Lemma 3.2.17

Lemma 3.2.17 (Type for $\mathbf{0}$)**.** For any context Γ and type T, $\Gamma \vdash \mathbf{0}:T \Rightarrow T \equiv \overline{0}$.

Proof. Let $S_n = \Gamma \vdash \mathbf{0}:T$. Induction on n.

1. $\dfrac{}{\Gamma \vdash \mathbf{0}:\overline{0}}\, ax_{\overline{0}}$ Trivial case

2. $\dfrac{\Gamma \vdash \mathbf{0}:\forall X.T}{\Gamma \vdash \mathbf{0}:T[U/X]}\, \forall_E$ By the induction hypothesis $\forall X.T \equiv \overline{0}$, so $T \equiv \overline{0}$ and also $T[U/X] \equiv \overline{0}$.

3. $\dfrac{\Gamma \vdash \mathbf{0}:T}{\Gamma \vdash \mathbf{0}:\forall X.T}\, \forall_I$ By the induction hypothesis $T \equiv \overline{0}$, so $\forall X.T \equiv \overline{0}$.

□

B.12 Proof of Theorem 3.2.1

Theorem 3.2.1 (Subject Reduction)**.** For any terms $\mathbf{t}$, $\mathbf{t}'$, any context Γ and any type T, if $\mathbf{t} \to \mathbf{t}'$, then $\Gamma \vdash \mathbf{t} : T \Rightarrow \Gamma \vdash \mathbf{t}' : T$.

Proof. We proceed by checking that every reduction rule preserves the type. Let $\mathbf{t} \to \mathbf{r}$ and $\Gamma \vdash \mathbf{t} : T$. To show that $\Gamma \vdash \mathbf{r} : T$, we proceed by induction on the derivation of the typing judgement.

Elementary rules

rule $\mathbf{t} + \mathbf{0} \to \mathbf{t}$ Let $\Gamma \vdash \mathbf{t} + \mathbf{0} : T$. By Lemma 3.2.12,$\exists \alpha, \beta \in \mathcal{S}$ such that$\Gamma \vdash \mathbf{t} : \alpha.T$ and $\Gamma \vdash \mathbf{0} : \beta.T$ with $\alpha + \beta = 1$. Then, by Lemma 3.2.17, $\beta.T \equiv \overline{0}$, so either $\beta = 0$, and then $\alpha = 1$ or $T \equiv \overline{0}$, and then $\alpha.T \equiv \overline{0} \equiv T$.

rule $0.\mathbf{t} \to \mathbf{0}$ Let $\Gamma \vdash 0.\mathbf{t} : T$. By Lemma 3.2.15, $\exists R$ such that $T \equiv 0.R \equiv \overline{0}$, and by rule $ax_{\overline{0}}$, $\Gamma \vdash \mathbf{0} : \overline{0}$.

rule $1.\mathbf{t} \to \mathbf{t}$ Let $\Gamma \vdash 1.\mathbf{t} : T \equiv 1.T$. By Lemma 3.2.10, $\Gamma \vdash \mathbf{t} : T$.

rule $\alpha.\mathbf{0} \to \mathbf{0}$ Let $\Gamma \vdash \alpha.\mathbf{0} : T$. By Lemma 3.2.15, $\exists R$ such that $T \equiv \alpha.R$. Cases

$\alpha \neq 0$ By Lemma 3.2.10, $\Gamma \vdash \mathbf{0} : R$. Thus, by Lemma 3.2.17, $R \equiv \overline{0}$ and so $T \equiv \alpha.R \equiv \overline{0}$.

$\alpha = 0$ By Lemma 3.2.11, $\exists S$ such that $\Gamma \vdash \mathbf{0} : S$ and by Lemma 3.2.17, $S \equiv \overline{0}$.

rule $\alpha.(\beta.\mathbf{t}) \to (\alpha \times \beta).\mathbf{t}$ True by Lemma 3.2.21.

rule $\alpha.(\mathbf{t} + \mathbf{r}) \to \alpha.\mathbf{t} + \alpha.\mathbf{r}$ True by Lemma 3.2.22.

Factorisation rules

rule $\alpha.\mathbf{t} + \beta.\mathbf{t} \to (\alpha + \beta).\mathbf{t}$ True by Lemma 3.2.23.

rule $\alpha.\mathbf{t} + \mathbf{t} \to (\alpha + 1).\mathbf{t}$ Let $\Gamma \vdash \alpha.\mathbf{t} + \mathbf{t} : T$. Using rule s_I one can derive $\Gamma \vdash 1.(\alpha.\mathbf{t} + \mathbf{t}) : 1.T$. Then by Lemma 3.2.22, $\Gamma \vdash 1.\alpha.\mathbf{t} + 1.\mathbf{t} : 1.T$. Moreover, by Lemma 3.2.12, $\Gamma \vdash 1.\alpha.\mathbf{t} : \gamma.T$ and $\Gamma \vdash 1.\mathbf{t} : \delta.T$ with $\gamma + \delta = 1$. So, by Lemma 3.2.21, $\Gamma \vdash \alpha.\mathbf{t} : \gamma.T$. Then using rule $+_I$ one can derive $\Gamma \vdash \alpha.\mathbf{t} + 1.\mathbf{t} : 1.T$. We conclude, by Lemma 3.2.23, with $\Gamma \vdash (\alpha + 1).\mathbf{t} : 1.T \equiv T$.

rule $\mathbf{t} + \mathbf{t} \to (1 + 1).\mathbf{t}$ Let $\Gamma \vdash \mathbf{t} + \mathbf{t} : T$. Then by rule s_I, $\Gamma \vdash 1.(\mathbf{t} + \mathbf{t}) : 1.T$. By Lemma 3.2.22, $\Gamma \vdash 1.\mathbf{t} + 1.\mathbf{t} : 1.T$ and by Lemma 3.2.23, $\Gamma \vdash (1 + 1).\mathbf{t} : 1.T \equiv T$.

Application rules

rule $(\mathbf{t} + \mathbf{r})\ \mathbf{u} \to (\mathbf{t})\ \mathbf{u} + (\mathbf{r})\ \mathbf{u}$ Let $\Gamma \vdash (\mathbf{t} + \mathbf{r})\ \mathbf{u} : T \equiv 1.T$. Then, by Lemma 3.2.8, $\exists \alpha, \beta, U$ and $T' \preceq T$ such that $\Gamma \vdash \mathbf{u} : \alpha.U$ and $\Gamma \vdash \mathbf{t} + \mathbf{r} : \beta.U \to T' \equiv 1.\beta.U \to T'$ with $\alpha \times \beta = 1$. Then by Lemma 3.2.12, $\exists \delta$ and γ such that $\Gamma \vdash \mathbf{t} : \delta.\beta.U \to T' \equiv (\delta \times \beta).U \to T'$ and $\Gamma \vdash \mathbf{r} : \gamma.\beta.U \to T' \equiv (\gamma \times \beta).U \to T'$ with $\delta + \gamma = 1$. Then by rule $\to_E$, $\Gamma \vdash (\mathbf{t})\ \mathbf{u} : (\delta \times \beta \times \alpha).T'$ and $\Gamma \vdash (\mathbf{r})\ \mathbf{u} : (\gamma \times \beta \times \alpha).T'$. Notice that $(\delta \times \beta \times \alpha).T' = (\delta \times 1).T' = \delta.T'$ and $(\gamma \times \beta \times \alpha).T' = (\gamma \times 1).T' = \gamma.T'$. Then by Lemmas 3.2.5 and 3.2.7 $\Gamma \vdash (\mathbf{t})\ \mathbf{u} : \delta.T$ and $\Gamma \vdash (\mathbf{r})\ \mathbf{u} : \gamma.T$, from which, using rule $+_I$, one can derive $\Gamma \vdash (\mathbf{t})\ \mathbf{u} + (\mathbf{r})\ \mathbf{u} : (\delta + \gamma).T \equiv T$.

rule $(\mathbf{u})\ (\mathbf{t} + \mathbf{r}) \to (\mathbf{u})\ \mathbf{t} + (\mathbf{u})\ \mathbf{r}$ Analogous to the previous case.

rule $(\alpha.\mathbf{t})\ \mathbf{r} \to \alpha.(\mathbf{t})\ \mathbf{r}$ Let $\Gamma \vdash (\alpha.\mathbf{t})\ \mathbf{r} : T \equiv 1.T$. Then by Lemma 3.2.8, $\exists \gamma, \beta, U$ and $T' \preceq T$ such that $\Gamma \vdash \mathbf{r} : \gamma.U$ and $\Gamma \vdash \alpha.\mathbf{t} : \beta.U \to T'$ with $\gamma \times \beta = 1$. Moreover, by Lemma 3.2.15,

$\beta.U \to T' \equiv \alpha.\delta.U'$ then by Lemma 3.2.3, $U \to T' \equiv U'$ and $\beta = \alpha \times \delta$ (notice that $\beta \neq 0$ because $\gamma \times \beta = 1$). So by Lemma 3.2.10, $\Gamma \vdash \mathbf{t} : \delta.(U \to T')$. By using rule $\to_E$ one can derive $\Gamma \vdash (\mathbf{t})\ \mathbf{r} : \delta.\gamma.T'$ from which, using rule s_I one can deduce $\Gamma \vdash \alpha.(\mathbf{t})\ \mathbf{r} : \alpha.\delta.\gamma.T'$. Notice that $\alpha.\delta.\gamma.T' \equiv (\alpha \times \delta \times \gamma).T' = (\beta \times \gamma).T' = 1.T \equiv T'$, so by Lemma 3.2.7, $\Gamma \vdash \alpha.(\mathbf{t})\ \mathbf{r} : 1.T$.

rule $(\mathbf{r})\ (\alpha.\mathbf{t}) \to \alpha.(\mathbf{r})\ \mathbf{t}$ Analogous to the previous case.

rule $(\mathbf{0})\ \mathbf{t} \to \mathbf{0}$ True by Lemma 3.2.20, and rule $ax_{\overline{0}}$.

rule $(\mathbf{t})\ \mathbf{0} \to \mathbf{0}$ True by Lemma 3.2.20, and rule $ax_{\overline{0}}$.

Beta reduction

rule $(\lambda x.\mathbf{t})\ \mathbf{b} \to \mathbf{t}[\mathbf{b}/x]$ Let $\Gamma \vdash (\lambda x.\mathbf{t})\ \mathbf{b} : T$. By rule $\equiv$, $\Gamma \vdash (\lambda x.\mathbf{t})\ \mathbf{b} : 1.T$, so by Lemma 3.2.8, $\exists \alpha, \beta, U, T' \preceq T$ such that $\Gamma \vdash \lambda x.\mathbf{t} : \beta.U \to T'$ and $\Gamma \vdash \mathbf{b} : \alpha.U$ with $\alpha \times \beta = 1$. Since $\mathbf{b}$ is a base term, by Lemma 3.2.16, $\alpha = 1$ and so $\beta = 1$. Then by Corollary 3.2.14, $\Gamma, x : U \vdash \mathbf{t} : T'$. Thus, by Lemma 3.2.13, $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T'$, from which, by Lemma 3.2.7, one obtain $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T$.

AC equivalences

Commutativity Let $\Gamma \vdash \mathbf{t} + \mathbf{r} : T$. Then, by Lemma 3.2.12, $\exists \delta$ and γ such that $\Gamma \vdash \mathbf{t} : \delta.T$ and $\Gamma \vdash \mathbf{r} : \gamma.T$ with $\delta + \gamma = 1$. Then using rules $+_I$ and $\equiv$, one can derive $\Gamma \vdash \mathbf{r} + \mathbf{t} : T$.

Associativity Let $\Gamma \vdash (\mathbf{t} + \mathbf{r}) + \mathbf{u} : T$. Then, by Lemma 3.2.12, $\exists \delta$ and γ such that $\Gamma \vdash \mathbf{t} + \mathbf{r} : \delta.T$ and $\Gamma \vdash \mathbf{u} : \gamma.T$ with $\delta + \gamma = 1$. Then, by Lemma 3.2.12 again, $\exists \delta'$ and γ' such that $\Gamma \vdash \mathbf{t} : \delta'.T$ and $\Gamma \vdash \mathbf{r} : \gamma'.T$ with $\delta' + \gamma' = \delta$. Then with rule $+_I$ one can deduce $\Gamma \vdash \mathbf{r} + \mathbf{u} : (\gamma' + \gamma).T$ and with the same rule, $\Gamma \vdash \mathbf{t} + (\mathbf{r} + \mathbf{u}) : (\delta' + \gamma' + \gamma).T \equiv T$. The inverse is analogous: if $\Gamma \vdash \mathbf{t} + (\mathbf{r} + \mathbf{u}) : T$ then $\Gamma \vdash (\mathbf{t} + \mathbf{r}) + \mathbf{u} : T$.

Contextual rules Let $\mathbf{t} \to \mathbf{r}$ and assume as induction hypothesis that for any context Γ and type T, if $\Gamma \vdash \mathbf{t} : T$ then $\Gamma \vdash \mathbf{r} : T$.

$(\mathbf{t})\ \mathbf{u} \to (\mathbf{r})\ \mathbf{u}$ Let $\Gamma \vdash (\mathbf{t})\ \mathbf{u} : T$. By Lemma 3.2.7, $\Gamma \vdash (\mathbf{t})\ \mathbf{u} : 1.T$. Then by Lemma 3.2.8, $\Gamma \vdash \mathbf{t} : \alpha.U \to R$ and $\Gamma \vdash \mathbf{u} : \beta.U$ with $R \preceq T$ and $\alpha \times \beta = 1$. By the induction hypothesis $\Gamma \vdash \mathbf{r} : \alpha.U \to R$, from which, using rule $\to_E$, one can deduce $\Gamma \vdash (\mathbf{r})\ \mathbf{u} : \alpha \times \beta.R$. Notice that $\alpha \times \beta.R = 1.R \equiv R \preceq T$, so by Lemma 3.2.7, $\Gamma \vdash (\mathbf{r})\ \mathbf{u} : T$.

$(\mathbf{u})\ \mathbf{t} \to (\mathbf{u})\ \mathbf{r}$ Analogous to previous case.

$\mathbf{t} + \mathbf{u} \to \mathbf{r} + \mathbf{u}$ Let $\Gamma \vdash \mathbf{t} + \mathbf{u} : T$. By Lemma 3.2.2, $T \equiv \alpha.U$, so by rule $\equiv$, $\Gamma \vdash \mathbf{t} + \mathbf{u} : \alpha.U$. Then by Lemma 3.2.12, $\Gamma \vdash \mathbf{t} : \delta.U$ and $\Gamma \vdash \mathbf{u} : \gamma.U$ with $\delta + \gamma = \alpha$. By the induction hypothesis $\Gamma \vdash \mathbf{r} : \delta.U$, so using rule $+_I$ one can deduce $\Gamma \vdash \mathbf{r} + \mathbf{u} : (\delta + \gamma).U$. Notice that $(\delta + \gamma).U = \alpha.U \equiv T$.

$\mathbf{u} + \mathbf{t} \to \mathbf{u} + \mathbf{r}$ Analogous to previous case.

$\alpha.\mathbf{t} \to \alpha.\mathbf{r}$ Let $\Gamma \vdash \alpha.\mathbf{t} : T$. By Lemma 3.2.15, $\exists \gamma$ and U such that $T \equiv \alpha.\gamma.U$. Cases

- $\alpha \neq 0$ By Lemma 3.2.10, $\Gamma \vdash \mathbf{t} : \gamma.U$. Moreover, by the induction hypothesis $\Gamma \vdash \mathbf{r} : \gamma.U$, and using the rule s_I one can derive $\Gamma \vdash \alpha.\mathbf{r} : \alpha.\gamma.U \equiv T$.
- $\alpha = 0$ By Lemma 3.2.11, $\exists R$ such that $\Gamma \vdash \mathbf{t} : R$. Moreover, by the induction hypothesis $\Gamma \vdash \mathbf{r} : R$, and using the rule s_I one can derive $\Gamma \vdash \alpha.\mathbf{r} : \alpha.R \equiv \overline{0} \equiv T$.

$\lambda x.\mathbf{t} \to \lambda x.\mathbf{r}$ Let $\Gamma \vdash \lambda x.\mathbf{t} : T$. By Lemma 3.2.9, $\exists U$ and R such that $\Gamma, x : U \vdash \mathbf{t} : R$ with $U \to R \preceq T$. Then by the induction hypothesis $\Gamma, x : U \vdash \mathbf{r} : R$, and using the rule $\to_I$ one can derive $\Gamma \vdash \lambda x.\mathbf{r} : U \to R$. By Lemma 3.2.7, $\Gamma \vdash \lambda x.\mathbf{r} : T$.

□

B.13 Proof of Lemma 3.3.5

Lemma 3.3.5.

1. $SN \in SAT$,
2. $A, B \in SAT \Rightarrow A \to B \in SAT$,
3. Let $\{A_i\}_{i\in I}$ be a collection of members of SAT, $\bigcap_{i\in I} A_i \in SAT$,
4. Given a valuation ξ in SAT and a A in $\mathbb{T}(\lambda 2^{la})$, then $[\![A]\!]_\xi \in SAT$.

Proof.

1. Obviously $SN \subseteq SN$. We need to prove it satisfies each point of the definition of saturation.
   (a) $\mathbf{0} \in SN$.
   (b) $\forall x, \vec{\mathbf{t}}, (x)\ \vec{\mathbf{t}} \in SN$.
   (c) Assume $(\mathbf{t}[\mathbf{b}/x])\ \vec{\mathbf{r}} \in SN$, then the term
   $$((\lambda x.\mathbf{t})\ \mathbf{b})\ \vec{\mathbf{r}} \tag{B.1}$$
   must terminate because $\mathbf{t}, \mathbf{b}$ and $\vec{\mathbf{r}}$ terminate since they are SN by assumption ($\mathbf{t}[\mathbf{b}/x]$ is a sub-term of a term in SN, hence itself is SN; but then $\mathbf{t}$ is also SN), After finitely many steps reducing terms in B.1 we obtain $((\lambda x.\mathbf{t}')\ \mathbf{b}')\ \vec{\mathbf{r}'}$ with $\mathbf{t} \to^* \mathbf{t}'$, $\mathbf{b} \to \mathbf{b}'$ and $\forall i, \mathbf{r}_i \to \mathbf{r}'_i$. Then the contraction of $((\lambda x.\mathbf{t}')\ \mathbf{b}')\ \vec{\mathbf{r}'}$ gives
   $$(\mathbf{t}'[\mathbf{b}'/x])\ \vec{\mathbf{r}'} \tag{B.2}$$
   This is a reduct of $(\mathbf{t}[\mathbf{b}/x])\ \vec{\mathbf{r}}$ and since this is SN, also B.2 and $((\lambda x.\mathbf{t})\ \mathbf{b})\ \vec{\mathbf{r}}$ are SN.
   (d) First notice that if $\mathbf{t}, \mathbf{u} \in SN$, then $\mathbf{t} + \mathbf{u} \in SN$. Now, assume $\forall i \in I$, $(\mathbf{t}_i)\ \vec{\mathbf{r}} \in SN$, which implies that $\mathbf{t}_i$ and $\vec{\mathbf{r}}$ are SN. Also, notice that $(\sum_{i\in I} \mathbf{t}_i)\ \vec{\mathbf{r}} \to^* \sum_{i\in I}(\mathbf{t}_i)\ \vec{\mathbf{r}}$ which is the sum of SN terms, so is SN. We need to prove that any other reduction starting from $(\sum_{i\in I} \mathbf{t}_i)\ \vec{\mathbf{r}}$ is also SN. We proceed by induction on I. To simplify the notation, we take $I = \{1, \ldots, n\}$ with $n \geq 1$.
   - If $I = \{1\}$, then we are done, since $\forall i \in I$, $(\mathbf{t}_i)\ \vec{\mathbf{r}} \in SN$.
   - Assume it is true for $I = \{1, \ldots, n\}$, that is $(\sum_{i=1}^{n} \mathbf{t}_i)\ \vec{\mathbf{r}} \in SN$.
   - Let $I = \{1, \ldots, n+1\}$, then we must prove that $(\sum_{i=1}^{n} \mathbf{t}_i + \mathbf{t}_{n+1})\ \vec{\mathbf{r}} \in SN$. Case by case on its possible reductions. Notice that any reduction in $\vec{\mathbf{t}}$ or $\vec{\mathbf{r}}$ is finite since these terms are in SN, so the amount of addends is the same.

     Elementary rules

- One $\mathbf{t}_k = \mathbf{0}$ and rule $\mathbf{t} + \mathbf{0} \to \mathbf{t}$ applies. Then the induction hypothesis closes the case.
- One $\mathbf{t}_k = \alpha.(\mathbf{t}_{k_1} + \mathbf{t}_{k_2})$ and reduces to $\alpha.\mathbf{t}_{k_1} + \alpha.\mathbf{t}_{k_2}$, it still can be considered as one addend since $\mathbf{t}_k$ is in SN.
- In other case, it is just a reduction in one $\mathbf{t}_i$ or one of $\vec{\mathbf{r}}$.

Factorisation rules Induction hypothesis.

Application rules

- Again, reductions on any $\mathbf{t}_k$ are no considered since these cases are trivial by the strong normalisation of $\mathbf{t}_k$.
- $(\sum_{i=1}^{n} \mathbf{t}_i + \mathbf{t}_{n+1})\ \vec{\mathbf{r}} \to (\sum_{i=1}^{n} \mathbf{t}_i)\ \vec{\mathbf{r}} + (\mathbf{t}_{n+1})\ \vec{\mathbf{r}}$, then since by induction hypothesis $(\sum_{i=1}^{n} \mathbf{t}_i)\ \vec{\mathbf{r}} \in SN$ and by hypothesis also $(\mathbf{t}_{n+1})\ \vec{\mathbf{r}} \in SN$, its sum is in SN.
- $(\sum_{i=1}^{n} \mathbf{t}_i + \mathbf{t}_{n+1})\ \vec{\mathbf{r}} \to (\sum_{i=1}^{k} \mathbf{t}_i)\ \vec{\mathbf{r}} + (\sum_{i=k+1}^{n+1} \mathbf{t}_i)\ \vec{\mathbf{r}}$ the induction hypothesis applies to both addends.
- Any other case does not involves a sum, so they are either included in the case of reduction of one $\mathbf{t}_k$ or one from $\vec{\mathbf{r}}$, or in the basic case of the induction.

Beta reduction This is either the basic case or a reduction in one $\mathbf{t}_k$ or one of $\vec{\mathbf{r}}$.

(e) $\forall i \in I$, $((\mathbf{u})\ \mathbf{t}_i)\ \vec{\mathbf{r}} \in SN$. Notice that $((\mathbf{u})\ \sum_{i\in I} \mathbf{t}_i)\ \vec{\mathbf{r}} \to^* \sum_{i\in I}((\mathbf{u})\ \mathbf{t}_i)\ \vec{\mathbf{r}}$ which is the sum of SN terms. This case is analogous to 1d.

(f) $\mathbf{t} \in SN$, then $\forall \alpha \in \mathcal{S}, \alpha.\mathbf{t} \in SN$ and *vice versa*.

(g) $\alpha.(((\mathbf{t}_1)\ \mathbf{t}_2)\dots)\ \mathbf{t}_n) \in SN$ then $\forall k, ((((\mathbf{t}_1)\dots)\ \alpha.\mathbf{t}_k)\dots)\ \mathbf{t}_n$ must terminate because $\vec{\mathbf{t}}$ terminate since these terms are SN by assumption, so after infinitely many reduction steps reducing $((((\mathbf{t}_1)\dots)\ \alpha.\mathbf{t}_k)\dots)\ \mathbf{t}_n$ we obtain $\alpha.\mathbf{u}$, with $((\mathbf{t}_1)\dots)\ \mathbf{t}_n \to^* \mathbf{u}$. So $\alpha.\mathbf{u}$ is a reduct of $\alpha.(((\mathbf{t}_1)\dots)\ \mathbf{t}_n)$ and since this term is SN, $\forall k, ((((\mathbf{t}_1)\dots)\ \alpha.\mathbf{t}_k)\dots)\ \mathbf{t}_n$ are SN.

(h) $(\mathbf{0})\ \vec{\mathbf{t}} \to^* (\mathbf{0})\ \vec{\mathbf{t}'}$ and since $\vec{\mathbf{t}}$ is SN, assume $\vec{\mathbf{t}'}$ is in normal form, so $(\mathbf{0})\ \vec{\mathbf{t}'}$ can only reduce to $\mathbf{0}$, then $(\mathbf{0})\ \vec{\mathbf{t}} \in SN$.

(i) $((\mathbf{t})\ \mathbf{0})\ \vec{\mathbf{u}} \to^* ((\mathbf{t}')\ \mathbf{0})\ \vec{\mathbf{u}'}$, since $\mathbf{t}$ and $\vec{\mathbf{u}}$ are in SN, assume $\mathbf{t}', \vec{\mathbf{u}'}$ are in normal form, then $((\mathbf{t}')\ \mathbf{0})\ \vec{\mathbf{u}'}$ can only reduce to $(\mathbf{0})\ \vec{\mathbf{u}'}$ which can only reduce to $\mathbf{0}$, so it is in SN.

2. Let $A, B \in SAT$, then $x \in A$ by definition of saturated sets. $\forall \mathbf{t} \in A \to B$, $(\mathbf{t})\ x \in B$. Since $B \in SAT$, then $B \subseteq SN$, so $(\mathbf{t})\ x \in SN$ and so $\mathbf{t}$ is strongly normalising. Therefore $A \to B \subseteq SN$. Now we need to show $A \to B$ is saturated by showing each point at the definition of saturated sets.

(a) By saturation of B, $\forall \mathbf{u} \in A$, $(\mathbf{0})\ \mathbf{u} \in B$, then $\mathbf{0} \in A \to B$.

(b) Let $\vec{\mathbf{t}} \in SN$, we need to show that $(x)\ \vec{\mathbf{t}} \in A \to B$, *i.e.* $\forall \mathbf{u} \in A$, $((x)\ \vec{\mathbf{t}})\ \mathbf{u} \in B$, which is true since $A \in SAT$ implies that $\mathbf{u} \in SN$, so $B \in SAT$ implies that $((x)\ \vec{\mathbf{t}})\ \mathbf{u} \in B$.

(c) Let $(\mathbf{t}[\mathbf{b}/x])\ \vec{\mathbf{r}} \in A \to B$, then $\forall \mathbf{u} \in A$, $((\mathbf{t}[\mathbf{b}/x])\ \vec{\mathbf{r}})\ \mathbf{u} \in B$ and since B is saturated, $(((\lambda x.\mathbf{t})\ \mathbf{b})\ \vec{\mathbf{r}})\ \mathbf{u} \in B$, so $((\lambda x.\mathbf{t})\ \mathbf{b})\ \vec{\mathbf{r}} \in A \to B$.

(d) Let $\forall i \in I$, $(\mathbf{t}_i)\ \vec{\mathbf{r}} \in A \to B$, then $\forall \mathbf{u} \in A$ and $\forall i \in I$, $((\mathbf{t}_i)\ \vec{\mathbf{r}})\ \mathbf{u} \in B$, then by the saturation of B, $((\sum_{i\in I} \mathbf{t}_i)\ \vec{\mathbf{r}})\ \mathbf{u} \in B$, so $(\sum_{i\in I} \mathbf{t}_i)\ \vec{\mathbf{r}} \in A \to B$.

(e) Let $\forall i \in I$, $((\mathbf{u})\ \mathbf{t}_i)\ \vec{\mathbf{r}} \in A \to B$, then $\forall \mathbf{t}' \in A$, $(((\mathbf{u})\ \mathbf{t}_i)\ \vec{\mathbf{r}})\ \mathbf{t}' \in B$, then by saturation of B, $(((\mathbf{u})\ \sum_{i\in I} \mathbf{t}_i)\ \vec{\mathbf{r}})\ \mathbf{t}' \in B$, so $((\mathbf{u})\ \sum_{i\in I} \mathbf{t}_i)\ \vec{\mathbf{r}} \in A \to B$.

(f) Let $\mathbf{t} \in A \to B$ then $\forall \mathbf{u} \in A$, $(\mathbf{t})\ \mathbf{u} \in B$, then by the saturation of B, $\forall \alpha \in \mathcal{S}$, $\alpha.(\mathbf{t})\ \mathbf{u} \in B$, then also by the saturation of B, $(\alpha.\mathbf{t})\ \mathbf{u} \in B$, so $\alpha.\mathbf{t} \in A \to B$.

Let $\alpha.\mathbf{t} \in A \to B$, then $\forall \mathbf{u} \in A$, $(\alpha.\mathbf{t})\ \mathbf{u} \in B$, so by the saturation of B, $\alpha.(\mathbf{t})\ \mathbf{u} \in B$, and again, by the saturation of B, $(\mathbf{t})\ \mathbf{u} \in B$, so $\mathbf{t} \in A \to B$.

(g) Let $\alpha.(((\mathbf{t}_1)\dots)\ \mathbf{t}_n) \in A \to B$, then $\forall \mathbf{u} \in A$, $(\alpha.(((\mathbf{t}_1)\dots)\ \mathbf{t}_n))\ \mathbf{u} \in B$, then by the saturation of B, $\alpha.((((\mathbf{t}_1)\dots)\ \mathbf{t}_n)\ \mathbf{u}) \in B$, and so, again by the saturation of B, one also has $\forall k, (((((\mathbf{t}_1)\dots)\ \alpha.\mathbf{t}_k)\dots)\ \mathbf{t}_n)\ \mathbf{u} \in B$, then $\forall k, ((((\mathbf{t}_1)\dots)\ \alpha.\mathbf{t}_k)\dots)\ \mathbf{t}_n \in A \to B$.

The inverse follows analogously: let $((((\mathbf{t}_1)\dots)\ \alpha.\mathbf{t}_k)\ \dots)\ \mathbf{t}_n \in A \to B$, then $\forall \mathbf{u} \in A$, $(((((\mathbf{t}_1)\dots)\ \alpha.\mathbf{t}_k)\ \dots)\ \mathbf{t}_n)\ \mathbf{u} \in B$, so by the saturation of B, $\alpha.((((\mathbf{t}_1)\dots)\ \mathbf{t}_n)\ \mathbf{u}) \in B$ and then, also by the saturation of B, one has $(\alpha.((\mathbf{t}_1)\dots)\ \mathbf{t}_n)\ \mathbf{u} \in B$, then $\alpha.((\mathbf{t}_1)\dots)\ \mathbf{t}_n \in A \to B$.

(h) $\forall \mathbf{u} \in A$, $\mathbf{u} \in SN$ and then, by the saturation of B, $\forall \vec{\mathbf{t}} \in SN$, $((\mathbf{0})\ \vec{\mathbf{t}})\ \mathbf{u} \in B$. Then $(\mathbf{0})\ \vec{\mathbf{t}} \in A \to B$.

(i) $\forall \mathbf{r} \in A$, $\mathbf{r} \in SN$ and then, by the saturation of B, $\forall \mathbf{t}, \vec{\mathbf{u}} \in SN$, $(((\mathbf{t})\ \mathbf{0})\ \vec{\mathbf{u}})\ \mathbf{r} \in B$. Then $((\mathbf{t})\ \mathbf{0})\ \vec{\mathbf{u}} \in A \to B$.

3. Let $\{A_i\}_{i\in I}$ be a collection of members of SAT, then $\forall i \in I$, $A_i \subseteq SN$, so $\bigcap_{i\in I} A_i \subseteq SN$. We have to show that $\bigcap_{i\in I} A_i$ is saturated.

   - Conditions (a), (b), (h) and (i) follows trivially: all these conditions have the form "$\mathbf{t} \in X$". Since by the saturation of A_i, $\forall i \in I, \mathbf{t} \in A_i$, then $\mathbf{t} \in \bigcap_{i\in I} A_i$.
   - Conditions (c), (d), (e), (f) and (g) are also straightforward: all these conditions have the form "If $\mathbf{t}$ in X, then $\mathbf{r}$ in X". Let $\mathbf{t} \in \bigcap_{i\in I} A_i$, then $\forall i \in I, \mathbf{t} \in A_i$ and so, by the saturation of A_i, $\mathbf{r} \in A_i$, from where one can deduce $\mathbf{r} \in \bigcap_{i\in I} A_i$.

4. By structural induction on A.

   $A := X$**:** Then $[\![A]\!]_\xi = \xi(X) \in SAT$.

   $A := B \to C$**:** Then $[\![A]\!]_\xi = [\![B]\!]_\xi \to [\![C]\!]_\xi$. by the induction hypothesis $[\![B]\!]_\xi$ and $[\![C]\!]_\xi \in SAT$, then by Lemma 3.3.5(2), $[\![B]\!]_\xi \to [\![C]\!]_\xi \in SAT$.

   $A := \forall X.A'$**:** Then $[\![A]\!]_\xi = \bigcap_{Y\in SAT} [\![A']\!]_{\xi(X:=Y)}$. By the induction hypothesis one has that $\forall Y$ in SAT, $[\![A']\!]_{\xi(X:=Y)}$ is also in SAT, then by Lemma 3.3.5(3), $\bigcap_{Y\in SAT} [\![A']\!]_{\xi(X:=Y)} \in SAT$.

□

B.14 Proof of Theorem 3.3.7

Theorem 3.3.7 (Soundness)**.** $\Gamma \Vdash \mathbf{t} : A \Rightarrow \Gamma \vDash \mathbf{t} : A$.

Proof. We proceed by induction on the derivation of $\Gamma \Vdash \mathbf{t} : T$.

1. $\dfrac{}{\Gamma, x : A \Vdash x : A}\ ax^{\triangleleft}$ Notice that if $\rho, \xi \vDash \Gamma, x : A$, then by definition $\rho, \xi \vDash x : A$.

2. $\dfrac{}{\Gamma \Vdash \mathbf{0} : A}\ ax_0^{\triangleleft}$ Then $\forall \xi, \rho$, by the saturation of $[\![A]\!]_\xi$, $\mathbf{0} \in [\![A]\!]_\xi$. Since $[\![\mathbf{0}]\!]_\rho = \mathbf{0}$, then $\rho, \xi \vDash \mathbf{0} : A$, and so $\forall \Gamma$, $\Gamma \vDash \mathbf{0} : A$.

3. $\dfrac{\Gamma \Vdash \mathbf{t}: A \to B \qquad \Gamma \Vdash \mathbf{r}: A}{\Gamma \Vdash (\mathbf{t})\ \mathbf{r}: B} \to_E^{\triangleleft}$

By the induction hypothesis, $\Gamma \vDash \mathbf{t}: A \to B$ and $\Gamma \vDash \mathbf{r}: A$. Assume $\rho, \xi \vDash \Gamma$ in order to show $\rho, \xi \vDash (\mathbf{t})\ \mathbf{r}: B$. Then $\rho, \xi \vDash \mathbf{t}: A \to B$, *i.e.* $[\![\mathbf{t}]\!]_\rho \in [\![A \to B]\!]_\xi = [\![A]\!]_\xi \to [\![B]\!]_\xi$ and $[\![\mathbf{r}]\!]_\rho \in [\![A]\!]_\xi$. Then $[\![(\mathbf{t})\ \mathbf{r}]\!]_\rho = [\![\mathbf{t}]\!]_\rho\ [\![\mathbf{r}]\!]_\rho \in [\![B]\!]_\xi$, so $\rho, \xi \vDash (\mathbf{t})\ \mathbf{r}: B$.

4. $\dfrac{\Gamma, x: A \Vdash \mathbf{t}: B}{\Gamma \Vdash \lambda x.\mathbf{t}: A \to B} \to_I^{\triangleleft}$

Assume $\rho, \xi \vDash \Gamma$ in order to show $\rho, \xi \vDash \lambda x.\mathbf{t}: A \to B$. That is, we must show $([\![\lambda x.\mathbf{t}]\!]_\rho)\ \mathbf{u} \in [\![B]\!]_\xi$ for all $\mathbf{u} \in [\![A]\!]_\xi$. Assume $\mathbf{u} \in [\![A]\!]_\xi$, then $\rho(x := \mathbf{u}) \vDash \Gamma, x: A$ and hence by the induction hypothesis $[\![\mathbf{t}]\!]_{\rho(x:=\mathbf{u})} \in B$. Since $([\![\lambda x.\mathbf{t}]\!]_\rho)\ \mathbf{u} = ((\lambda x.\mathbf{t})[\vec{y} := \rho(\vec{y})])\ \mathbf{u} \to_\beta \mathbf{t}[\vec{y} := \rho(\vec{y}), x := \mathbf{u}] = [\![\mathbf{t}]\!]_{\rho(x:=\mathbf{u})}$, it follows from the saturation of $[\![B]\!]_\xi$ that $([\![\lambda x.\mathbf{t}]\!]_\rho)\ \mathbf{u} \in [\![B]\!]_\xi$.

5. $\dfrac{\Gamma \Vdash \mathbf{t}: \forall X.A}{\Gamma \Vdash \mathbf{t}: A[B/X]} \forall_E^{\triangleleft}$

Assume $\rho, \xi \vDash \Gamma$ in order to show $\rho, \xi \vDash \mathbf{t}: A[B/X]$. By the induction hypothesis $[\![\mathbf{t}]\!]_\rho \in [\![\forall X.A]\!]_\xi$ and this set is equal to $\bigcap_{Y \in SAT} [\![A]\!]_{\xi(X:=Y)}$, hence $[\![\mathbf{t}]\!]_\rho \in [\![A]\!]_{\xi(X:=[\![B]\!]_\xi)} = [\![A[B/X]]\!]_\xi$.

6. $\dfrac{\Gamma \Vdash \mathbf{t}: A \qquad X \notin FV(\Gamma)}{\Gamma \Vdash \mathbf{t}: \forall X.A} \forall_I^{\triangleleft}$

Assume $\rho, \xi \vDash \Gamma$ in order to show $\rho, \xi \vDash \mathbf{t}: \forall X.A$. Since $X \notin FV(\Gamma)$, one also has $\forall Y \in SAT$ that $\rho, \xi(X := Y) \vDash \Gamma$, therefore $\forall Y \in SAT$, $[\![\mathbf{t}]\!]_\rho \in [\![A]\!]_{\xi(X:=Y)}$, then by the induction hypothesis $[\![\mathbf{t}]\!]_\rho \in [\![\forall X.A]\!]_\xi$, *i.e.* $\rho, \xi \vDash \mathbf{t}: \forall X.A$.

7. $\dfrac{\Gamma \Vdash \mathbf{t}: A \qquad \Gamma \Vdash \mathbf{r}: A}{\Gamma \Vdash \mathbf{t} + \mathbf{r}: A} +_I^{\triangleleft}$

Assume $\rho, \xi \vDash \Gamma$ in order to show $\rho, \xi \vDash \mathbf{t} + \mathbf{r}: A$. By the induction hypothesis one has $\Gamma \vDash \mathbf{t}: A$ and $\Gamma \vDash \mathbf{r}: A$, so $[\![\mathbf{t}]\!]_\rho \in [\![A]\!]_\xi$ and $[\![\mathbf{r}]\!]_\rho \in [\![A]\!]_\xi$. Since $[\![\mathbf{t} + \mathbf{r}]\!]_\rho = (\mathbf{t} + \mathbf{r})[\vec{x} := \rho(\vec{x})] = \mathbf{t}[\vec{x} := \rho(\vec{x})] + \mathbf{r}[\vec{x} := \rho(\vec{x})] = [\![\mathbf{t}]\!]_\rho + [\![\mathbf{r}]\!]_\rho$, it follows from the saturation of $[\![A]\!]_\xi$ that $[\![\mathbf{t} + \mathbf{r}]\!]_\rho \in [\![A]\!]_\xi$.

8. $\dfrac{\Gamma \Vdash \mathbf{t}: A}{\Gamma \Vdash \alpha.\mathbf{t}: A} s_I^{\triangleleft}$

Suppose $\rho, \xi \vDash \Gamma$ in order to show $\rho, \xi \vDash \alpha.\mathbf{t}: A$. By the induction hypothesis $\Gamma \vDash \mathbf{t}: A$, then $[\![\mathbf{t}]\!]_\rho \in [\![A]\!]_\xi$. Since $[\![\alpha.\mathbf{t}]\!]_\rho = (\alpha.\mathbf{t})[\vec{x} := \rho(\vec{x})] = \alpha.(\mathbf{t}[\vec{x} := \rho(\vec{x})]) = \alpha.[\![\mathbf{t}]\!]_\rho$, it follows from the saturation of $[\![A]\!]_\xi$ that $[\![\alpha.\mathbf{t}]\!]_\rho \in [\![A]\!]_\xi$.

□

B.15 Proof of Lemma 3.3.10

Lemma 3.3.10 (Correspondence with $\lambda 2^{la}$)**.** $\Gamma \vdash \mathbf{t}: T \Rightarrow \Gamma^\natural \Vdash \mathbf{t}: T^\natural$.

Proof. We proceed by induction on the derivation of $\Gamma \vdash \mathbf{t}: T$.

1. $\dfrac{}{\Gamma, x: U \vdash x: U}\, ax$ $\quad (\Gamma, x: U)^\natural = \Gamma^\natural, x: U^\natural$, so by $ax^{\triangleleft}$, $(\Gamma, x: U)^\natural \Vdash x: U^\natural$.

2. $\dfrac{}{\Gamma \vdash \mathbf{0}: \overline{0}}\, ax_{\overline{0}}$ $\quad$ By $ax_0{}^{\triangleleft}$, $\Gamma^\natural \Vdash \mathbf{0}: A$ for any $A \in \mathbb{T}(\lambda 2^{la})$, so take $A = \overline{0}^\natural$.

3. $\dfrac{\Gamma \vdash \mathbf{t}: \alpha.U \to T \qquad \Gamma \vdash \mathbf{r}: \beta.U}{\Gamma \vdash (\mathbf{t})\ \mathbf{r}: (\alpha \times \beta).T} \to_E$ $\quad$ By the induction hypothesis $\Gamma^\natural \Vdash \mathbf{t}: U^\natural \to T^\natural$ and $\Gamma^\natural \Vdash \mathbf{r}: U^\natural$, so by rule $\to_E^{\triangleleft}$, $\Gamma^\natural \Vdash (\mathbf{t})\ \mathbf{r}: T^\natural = ((\alpha \times \beta).T)^\natural$.

4. $\dfrac{\Gamma, x: U \vdash \mathbf{t}: T}{\Gamma \vdash \lambda x.\mathbf{t}: U \to T} \to_I$ $\quad$ By the induction hypothesis one has $\Gamma^\natural, x: U^\natural \Vdash \mathbf{t}: T^\natural$, so by rule $\to_I^{\triangleleft}$, $\Gamma^\natural \Vdash \lambda x.\mathbf{t}: U^\natural \to T^\natural = (U \to T)^\natural$.

5. $\dfrac{\Gamma \vdash \mathbf{t}: \forall X.T}{\Gamma \vdash \mathbf{t}: T[U/X]} \forall_E$ $\quad$ By the induction hypothesis one has $\Gamma^\natural \Vdash \mathbf{t}: (\forall X.T)^\natural = \forall X.T^\natural$, so by rule $\forall_E^{\triangleleft}$, $\Gamma^\natural \Vdash \mathbf{t}: T^\natural[U^\natural/X]$.

6. $\dfrac{\Gamma \vdash \mathbf{t}:T}{\Gamma \vdash \mathbf{t}:\forall X.T}\,\forall_I$ By the induction hypothesis $\Gamma^\natural \Vdash \mathbf{t}:T^\natural$, so by rule $\forall_I^\triangleleft$, $\Gamma^\natural \Vdash \mathbf{t}:\forall X.T^\natural = (\forall X.T)^\natural$.

7. $\dfrac{\Gamma \vdash \mathbf{t}:\alpha.T \qquad \Gamma \vdash \mathbf{r}:\beta.T}{\Gamma \vdash \mathbf{t}+\mathbf{r}:(\alpha+\beta).T}\,+_I$ By the induction hypothesis $\Gamma^\natural \Vdash \mathbf{t}:T^\natural$ and $\Gamma^\natural \Vdash \mathbf{r}:T^\natural$, so by rule $+_I^\triangleleft$, $\Gamma^\natural \Vdash \mathbf{t}+\mathbf{r}:T^\natural = ((\alpha+\beta).T)^\natural$.

8. $\dfrac{\Gamma \vdash \mathbf{t}:T}{\Gamma \vdash \alpha.\mathbf{t}:\alpha.T}\,s_I$ By the induction hypothesis $\Gamma^\natural \Vdash \mathbf{t}:T^\natural$, so by rule $s_I^\triangleleft$, $\Gamma^\natural \Vdash \alpha.\mathbf{t}:T^\natural = (\alpha.T)^\natural$.

□

B.16 Proof of Corollary 3.3.13(3)

Corollary 3.3.13 (Item 3 in the proof: Commutation)**.** Algebraic rules and β-reduction commutes, *i.e.* if $\mathbf{t} \to_a \mathbf{u}$ and $\mathbf{t} \to_\beta \mathbf{r}$, then there exists a term $\mathbf{t}'$ such that $\mathbf{u} \to^* \mathbf{t}'$ and $\mathbf{r} \to^* \mathbf{t}'$, where $\to^*$ is a reduction sequence of zero or more steps involving any rules.

Proof. Let $\mathbf{t} \to_a \mathbf{u}$ and $\mathbf{t} \to_\beta \mathbf{r}$. We proceed by structural induction on $\mathbf{t}$.

1. $\mathbf{t}$ cannot be a variable or $\mathbf{0}$, since it reduces.

2. $\mathbf{t} = \lambda x.\mathbf{t}_1$. Then $\mathbf{u} = \lambda x.\mathbf{u}_1$ and $\mathbf{r} = \lambda x.\mathbf{r}_1$ with $\mathbf{t}_1 \to_a \mathbf{u}_1$ and $\mathbf{t}_1 \to_\beta \mathbf{r}_1$. Then by the induction hypothesis $\exists \mathbf{t}_1'$ such that $\mathbf{u}_1 \to_\beta \mathbf{t}_1'$ and $\mathbf{r}_1 \to_a \mathbf{t}_1'$. Take $\mathbf{t}' = \lambda x.\mathbf{t}'$.

3. $\mathbf{t} = (\mathbf{t}_1)\ \mathbf{t}_2$. Cases

   - $\mathbf{r} = (\mathbf{r}_1)\ \mathbf{t}_2$ where $\mathbf{t}_1 \to_\beta \mathbf{r}_1$. Cases
     - $\mathbf{u} = (\mathbf{u}_1)\ \mathbf{t}_2$ where $\mathbf{t}_1 \to_a \mathbf{u}_1$. Then by the induction hypothesis, $\exists \mathbf{t}_1'$ such that $\mathbf{r}_1 \to_a \mathbf{t}_1'$ and $\mathbf{u}_1 \to_\beta \mathbf{t}_1'$. Take $\mathbf{t}' = (\mathbf{t}_1')\ \mathbf{t}_2$.
     - $\mathbf{u} = (\mathbf{t}_1)\ \mathbf{u}_2$ where $\mathbf{t}_2 \to_a \mathbf{u}_2$. Take $\mathbf{t}' = (\mathbf{r}_1)\ \mathbf{u}_2$.
   - $\mathbf{r} = (\mathbf{t}_1)\ \mathbf{r}_2$. Analogous to previous case.
   - $\mathbf{r} = \mathbf{r}_1[\mathbf{t}_2/x]$ where $\mathbf{t}_1 = \lambda x.\mathbf{r}_1$ and $\mathbf{t}_2$ is a base term. Cases
     - $\mathbf{u} = (\lambda x.\mathbf{u}_1)\ \mathbf{t}_2$ where $\mathbf{r}_1 \to_a \mathbf{u}_1$. Then notice that $\mathbf{u} \to_\beta \mathbf{u}_1[\mathbf{t}_2/x]$. We proceed by structural induction over $\mathbf{r}_1$ to prove that $\mathbf{r}_1 \to_a \mathbf{u}_1$ implies $\mathbf{r}_1[\mathbf{t}_2/x] \to_a \mathbf{u}_1[\mathbf{t}_2/x]$ for any $\mathbf{t}_2$ base term.
       - $\mathbf{r}_1$ cannot be $\mathbf{0}$ or a variable, since it reduces.
       - $\mathbf{r}_1 = \lambda y.\mathbf{r}_1'$. Then $\mathbf{u}_1 = \lambda y.\mathbf{u}_1'$ where $\mathbf{r}_1' \to_a \mathbf{u}_1'$. Then by the induction hypothesis $\mathbf{r}_1'[\mathbf{t}_2/x] \to_a \mathbf{u}_1'[\mathbf{t}_2/x]$ which implies that $\lambda y.\mathbf{r}_1'[\mathbf{t}_2/x] \to_a$-reduces to $(\lambda y.\mathbf{u}_1')[\mathbf{t}_2/x]$.
       - $\mathbf{r}_1 = (\mathbf{r}_1')\ \mathbf{r}_1''$. Cases
         - $\mathbf{u}_1 = (\mathbf{u}_1')\ \mathbf{r}_1''$ with $\mathbf{r}_1' \to_a \mathbf{u}_1'$, then by the induction hypothesis $\mathbf{r}_1'[\mathbf{t}_2/x] \to_a \mathbf{u}_1'[\mathbf{t}_2/x]$ which implies that $((\mathbf{r}_1')\ \mathbf{r}_1'')[\mathbf{t}_2/x] \to_a ((\mathbf{u}_1')\ \mathbf{r}_1'')[\mathbf{t}_2/x]$.
         - $\mathbf{u}_1 = (\mathbf{r}_1')\ \mathbf{u}_1''$ with $\mathbf{r}_1'' \to_a \mathbf{u}_1''$. Analogous to previous case.
       - $\mathbf{r}_1 = \mathbf{r}_1' + \mathbf{r}_1''$. Cases
         - $\mathbf{u}_1 = \mathbf{u}_1' + \mathbf{r}_1''$ with $\mathbf{r}_1' \to_a \mathbf{u}_1'$, then by the induction hypothesis $\mathbf{r}_1'[\mathbf{t}_2/x] \to_a \mathbf{u}_1'[\mathbf{t}_2/x]$ which implies that $(\mathbf{r}_1' + \mathbf{r}_1'')[\mathbf{t}_2/x] \to_a (\mathbf{u}_1' + \mathbf{r}_1'')[\mathbf{t}_2/x]$.
         - $\mathbf{u}_1 = \mathbf{r}_1' + \mathbf{u}_1''$ with $\mathbf{r}_1'' \to_a \mathbf{u}_1''$. Analogous to previous case.

· $\mathbf{u}_1 = (\alpha + \beta).\mathbf{r}_1'''$ where $\mathbf{r}_1' = \alpha.\mathbf{r}_1'''$ and $\mathbf{r}_1'' = \beta.\mathbf{r}_1'''$. Then notice that $(\alpha.\mathbf{r}_1''' + \beta.\mathbf{r}_1''')[\mathbf{t}_2/x] = \alpha.\mathbf{r}_1'''[\mathbf{t}_2/x] + \beta.\mathbf{r}_1'''[\mathbf{t}_2/x] \to_a (\alpha + \beta).\mathbf{r}_1'''[\mathbf{t}_2/x] = ((\alpha + \beta).\mathbf{r}_1''')[\mathbf{t}_2/x]$.

· $\mathbf{u}_1 = (\alpha + 1).\mathbf{r}_1'''$ where $\mathbf{r}_1' = \alpha.\mathbf{r}_1'''$ and $\mathbf{r}_1'' = \mathbf{r}_1'''$. Analogous to previous case.

· $\mathbf{u}_1 = (1 + 1).\mathbf{r}_1'$ where $\mathbf{r}_1' = \mathbf{r}_1''$. Analogous to previous case.

· $\mathbf{u}_1 = \mathbf{r}_1'$ and $\mathbf{r}_1'' = \mathbf{0}$. Then $(\mathbf{r}_1' + \mathbf{0})[\mathbf{t}_2/x] = \mathbf{r}_1'[\mathbf{t}_2/x] + \mathbf{0} \to_a \mathbf{r}_1'[\mathbf{t}_2/x]$.

∗ $\mathbf{r}_1 = \alpha.\mathbf{r}_1'$. Cases

· $\mathbf{u}_1 = \mathbf{u}_1'$ with $\mathbf{r}_1' \to_a \mathbf{u}_1'$. Then by the induction hypothesis one has $\mathbf{r}_1'[\mathbf{t}_2/x] \to_a \mathbf{u}_1'[\mathbf{t}_2/x]$, so $(\alpha.\mathbf{r}_1')[\mathbf{t}_2/x] \to_a (\alpha.\mathbf{u}_1')[\mathbf{t}_2/x]$.

· $\mathbf{u}_1 = (\alpha \times \beta).\mathbf{r}_1''$ with $\mathbf{r}_1' = \beta.\mathbf{r}_1''$. Then $(\alpha.(\beta.\mathbf{r}_1''))[\mathbf{t}_2/x]$ is equal to $\alpha.(\beta.\mathbf{r}_1''[\mathbf{t}_2/x]) \to_a (\alpha \times \beta).\mathbf{r}_1''[\mathbf{t}_2/x] = ((\alpha \times \beta).\mathbf{r}_1'')[\mathbf{t}_2/x]$.

· $\mathbf{u} = \alpha.\mathbf{r}_1'' + \alpha.\mathbf{r}_1'''$ and $\mathbf{r}_1' = \mathbf{r}_1'' + \mathbf{r}_1'''$. Then $(\alpha.(\mathbf{r}_1'' + \mathbf{r}_1'''))[\mathbf{t}_2/x] = \alpha.(\mathbf{r}_1''[\mathbf{t}_2/x] + \mathbf{r}_1'''[\mathbf{t}_2/x]) \to_a \alpha.\mathbf{r}_1''[\mathbf{t}_2/x] + \alpha.\mathbf{r}_1'''[\mathbf{t}_2/x] = (\alpha.\mathbf{r}_1'' + \alpha.\mathbf{r}_1''')[\mathbf{t}_2/x]$.

· $\mathbf{u} = \mathbf{r}_1'$ and $\alpha = 1$. Then $(1.\mathbf{r}_1')[\mathbf{t}_2/x] = 1.\mathbf{r}_1'[\mathbf{t}_2/x] \to_a \mathbf{r}_1'[\mathbf{t}_2/x]$.

· $\mathbf{u} = \mathbf{0}$ and $\alpha = 0$. Then $(0.\mathbf{r}_1')[\mathbf{t}_2/x] = 0.\mathbf{r}_1'[\mathbf{t}_2/x] \to_a \mathbf{0} = \mathbf{0}[\mathbf{t}_2/x]$.

· $\mathbf{u} = \mathbf{0}$ and $\mathbf{r}_1' = \mathbf{0}$. Then $(\alpha.\mathbf{0})[\mathbf{t}_2/x] = \alpha.\mathbf{0}[\mathbf{t}_2/x] = \alpha.\mathbf{0} \to_a \mathbf{0} = \mathbf{0}[\mathbf{t}_2/x]$.

– $\mathbf{u} = (\lambda x.\mathbf{r}_1)\ \mathbf{u}_2$ where $\mathbf{t}_2 \to_a \mathbf{u}_2$. Then $\mathbf{u} \to_\beta \mathbf{r}_1[\mathbf{u}_2/x]$. We proceed by structural induction over $\mathbf{r}_1$ to show that if $\mathbf{t}_2 \to_a \mathbf{u}_2$, then $\mathbf{r}_1[\mathbf{t}_2/x] \to_a^* \mathbf{r}_1[\mathbf{u}_2/x]$.

∗ $\mathbf{r}_1 = x$, then $x[\mathbf{t}_2/x] = \mathbf{t}_2 \to_a \mathbf{u}_2 = x[\mathbf{u}_2/x]$.

∗ $\mathbf{r}_1 = y$, then $y[\mathbf{t}_2/x] = y = y[\mathbf{u}_2/x]$.

∗ $\mathbf{r}_1 = \mathbf{0}$, analogous to previous case.

∗ $\mathbf{r}_1 = \lambda y.\mathbf{r}_1'$, then $(\lambda y.\mathbf{r}_1')[\mathbf{t}_2/x] = \lambda y.\mathbf{r}_1'[\mathbf{t}_2/x]$. By the induction hypothesis $\mathbf{r}_1'[\mathbf{t}_2/x] \to_a^* \mathbf{r}_1'[\mathbf{u}_2/x]$, so $\lambda y.\mathbf{r}_1'[\mathbf{t}_2/x] \to_a^* \lambda y.\mathbf{r}_1'[\mathbf{u}_2/x]$.

∗ $\mathbf{r}_1 = (\mathbf{r}_1')\ \mathbf{r}_1''$. Then $((\mathbf{r}_1')\ \mathbf{r}_1'')[\mathbf{t}_2/x] = (\mathbf{r}_1'[\mathbf{t}_2/x])\ \mathbf{r}_1''[\mathbf{t}_2/x]$. By induction hypothesis $\mathbf{r}_1'[\mathbf{t}_2/x] \to_a^* \mathbf{r}_1'[\mathbf{u}_2/x]$ and $\mathbf{r}_1''[\mathbf{u}_2/x]$, so $(\mathbf{r}_1'[\mathbf{t}_2/x])\ \mathbf{r}_1''[\mathbf{t}_2/x] \to_a^* (\mathbf{r}_1'[\mathbf{u}_2/x])\ \mathbf{r}_1''[\mathbf{u}_2/x] = ((\mathbf{r}_1')\ \mathbf{r}_1'')[\mathbf{u}_2/x]$.

∗ $\mathbf{r}_1 = \mathbf{r}_1' + \mathbf{r}_1''$. Then $(\mathbf{r}_1' + \mathbf{r}_1'')[\mathbf{t}_2/x] = \mathbf{r}_1'[\mathbf{t}_2/x] + \mathbf{r}_1''[\mathbf{t}_2/x]$. By induction hypothesis $\mathbf{r}_1'[\mathbf{t}_2/x] \to_a^* \mathbf{r}_1'[\mathbf{u}_2/x]$ and $\mathbf{r}_1''[\mathbf{u}_2/x]$, so one has $\mathbf{r}_1'[\mathbf{t}_2/x] + \mathbf{r}_1''[\mathbf{t}_2/x] \to_a^* \mathbf{r}_1'[\mathbf{u}_2/x] + \mathbf{r}_1''[\mathbf{u}_2/x] = (\mathbf{r}_1' + \mathbf{r}_1'')[\mathbf{u}_2/x]$.

∗ $\mathbf{r}_1 = \alpha.\mathbf{r}_1'$. Then $(\alpha.\mathbf{r}_1')[\mathbf{t}_2/x] = \alpha.\mathbf{r}_1'[\mathbf{t}_2/x]$. By the induction hypothesis $\mathbf{r}_1'[\mathbf{t}_2/x] \to_a^* \mathbf{r}_1'[\mathbf{u}_2/x]$, so $\alpha.\mathbf{r}_1'[\mathbf{t}_2/x] \to_a^* \alpha.\mathbf{r}_1'[\mathbf{u}_2/x] = (\alpha.\mathbf{r}_1')[\mathbf{u}_2/x]$.

4. $\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2$. Cases

- $\mathbf{u} = (\alpha + \beta).\mathbf{u}'$ with $\mathbf{t}_1 = \alpha.\mathbf{u}'$ and $\mathbf{t}_2 = \beta.\mathbf{u}'$. Cases
  – $\mathbf{r} = \alpha.\mathbf{r}' + \beta.\mathbf{u}'$ with $\mathbf{u}' \to_\beta \mathbf{r}'$. Then notice that $\mathbf{u} = (\alpha + \beta).\mathbf{u}' \to_\beta (\alpha + \beta).\mathbf{r}'$ and $\alpha.\mathbf{r}' + \beta.\mathbf{u}' \to_\beta \alpha.\mathbf{r}' + \beta.\mathbf{r}' \to_a (\alpha + \beta).\mathbf{r}'$.
  – $\mathbf{r} = \alpha.\mathbf{u}' + \beta.\mathbf{r}'$ with $\mathbf{u}' \to_\beta \mathbf{r}'$. Analogous to previous case.
- $\mathbf{u} = (\alpha + 1).\mathbf{u}'$ with $\mathbf{t}_1 = \alpha.\mathbf{u}'$ and $\mathbf{t}_2 = \mathbf{u}'$. Analogous to previous case.
- $\mathbf{u} = (1 + 1).\mathbf{u}'$ with $\mathbf{t}_1 = \mathbf{t}_2$. Analogous to previous case.
- $\mathbf{u} = \mathbf{t}_1$ with $\mathbf{t}_2 = \mathbf{0}$. Then the only possibility for $\mathbf{r}$ is to be $\mathbf{r}' + \mathbf{0}$ where $\mathbf{t}_1 \to_\beta \mathbf{r}'$. Then $\mathbf{r}' + \mathbf{0} \to_a \mathbf{r}'$, which closes the case.

5. $\mathbf{t} = \alpha.\mathbf{t}_1$. The only possibility for $\mathbf{r}$ is to be $\alpha.\mathbf{r}'$ where $\mathbf{t}_1 \to_\beta \mathbf{r}'$. Cases

   - $\mathbf{u} = (\alpha \times \beta).\mathbf{u}_1$ with $\mathbf{t}_1 = \beta.\mathbf{u}_1$, then $\mathbf{r}' = \beta.\mathbf{r}''$ with $\mathbf{u}_1 \to_\beta \mathbf{r}''$. Then $(\alpha \times \beta).\mathbf{u}_1 \to (\alpha \times \beta).\mathbf{r}''$ and $\alpha.(\beta.\mathbf{r}'') \to_a (\alpha \times \beta).\mathbf{r}''$.
   - $\mathbf{u} = \alpha.\mathbf{u}_1 + \alpha.\mathbf{u}_2$ with $\mathbf{t}_1 = \alpha.(\mathbf{u}_1 + \mathbf{u}_2)$. Cases
     - $\mathbf{r}' = \mathbf{r}_1 + \mathbf{u}_2$ with $\mathbf{u}_1 \to_\beta \mathbf{r}_1$, then $\alpha.(\mathbf{r}_1 + \mathbf{u}_2) \to_a \alpha.\mathbf{r}_1 + \alpha.\mathbf{u}_2$ and $\alpha.\mathbf{u}_1 + \alpha.\mathbf{u}_2 \to_\beta \alpha.\mathbf{r}_1 + \alpha.\mathbf{u}_2$.
     - $\mathbf{r}' = \mathbf{u}_1 + \mathbf{r}_2$ with $\mathbf{u}_2 \to_\beta \mathbf{r}_2$, analogous to previous case.
   - $\mathbf{u} = \mathbf{t}_1$ with $\alpha = 1$. Notice that $1.\mathbf{r}' \to_a \mathbf{r}'$.
   - $\mathbf{u} = \mathbf{0}$ with $\alpha = 0$. Notice that $\alpha.\mathbf{r}' \to_a \mathbf{0}$.
   - $\mathbf{u} = \mathbf{0}$ with $\mathbf{t}_1 = \mathbf{0}$. Absurd since $\mathbf{t}_1 \to_\beta \mathbf{r}'$.

□

B.17 Proof of Theorem 3.4.3

Theorem 3.4.3 (Normal-form of terms in $\mathcal{B}$ have weight 1)**.** Let $\Gamma \vdash \mathbf{t} : A$ be well-formed, then $\omega(\mathbf{t}\downarrow) = 1$.

We need two preliminary lemmas:

Lemma B.17.1. *If* $(\mathbf{t}_1)\ \mathbf{t}_2$ *is in normal form, then* $\mathbf{t}_1 = (x)\ \vec{\mathbf{r}}$.

Proof. Structural induction on $\mathbf{t}_1$.

1. $\mathbf{t}_1 = x$. Done.
2. $\mathbf{t}_1 = \lambda x.\mathbf{r}$, then $(\mathbf{t}_1)\ \mathbf{t}_2 \to \mathbf{r}[\mathbf{t}_2/x]$, which is a contradiction.
3. $\mathbf{t}_1 = \mathbf{0}$, then $(\mathbf{t}_1)\ \mathbf{t}_2 \to \mathbf{0}$, which is a contradiction.
4. $\mathbf{t}_1 = \alpha.\mathbf{r}$, then $(\mathbf{t}_1)\ \mathbf{t}_2 \to \alpha.(\mathbf{r})\ \mathbf{t}_2$, which is a contradiction.
5. $\mathbf{t}_1 = \mathbf{r} + \mathbf{u}$, then $\mathbf{t}_1\ \mathbf{t}_2 \to (\mathbf{r})\ \mathbf{t}_2 + (\mathbf{u})\ \mathbf{t}_2$, which is a contradiction.
6. $\mathbf{t}_1 = (\mathbf{u}_1)\ \mathbf{u}_2$, then by the induction hypothesis $\mathbf{u}_1 = (x)\ \vec{\mathbf{r}}$, so $(\mathbf{u}_1)\ \mathbf{u}_2 = (x)\ \vec{\mathbf{r}}'$, where $\vec{\mathbf{r}}' = \vec{\mathbf{r}}, \mathbf{u}_2$.

□

Lemma B.17.2. $\Gamma \vdash (x)\ \vec{\mathbf{r}} : T$ *and* $(x)\ \vec{\mathbf{r}}$ *is in normal form, then* $\exists A \in \mathbb{T}(\lambda 2^{la}), \alpha \in \mathcal{S}$ *such that* $T \equiv \alpha.A$.

Proof. Induction on the derivation of $\Gamma \vdash (x)\ \vec{\mathbf{r}} : T$.

1. $\dfrac{}{\Gamma, x : T \vdash x : T}\ ax$ — Then $T \in \mathbb{T}(\lambda 2^{la})$, because in $\mathcal{B}$ contexts have only classic types.
2. $\dfrac{\Gamma \vdash (x)\ \vec{\mathbf{r}} : \alpha.(U \to T) \qquad \Gamma \vdash \mathbf{t} : \beta.U}{\Gamma \vdash ((x)\ \vec{\mathbf{r}})\ \mathbf{t} : (\alpha \times \beta).T}\ \to_E$ — Then by the induction hypothesis, $U \to T \in \mathbb{T}(\lambda 2^{la})$, so $T \in \mathbb{T}(\lambda 2^{la})$.
3. $\dfrac{\Gamma \vdash (x)\ \vec{\mathbf{r}} : \forall X.T}{\Gamma \vdash (x)\ \vec{\mathbf{r}} : T[A/X]}\ \forall_E$ — Then by the induction hypothesis $\exists B \in \mathbb{T}(\lambda 2^{la}), \alpha \in \mathcal{S}$ such that $\forall X.T \equiv \alpha.B$, so $\exists C \in \mathbb{T}(\lambda 2^{la})$ such that $T \equiv \alpha.C$, then $T[A/X] \equiv (\alpha.C)[A/X] \equiv \alpha.C[A/X]$. Notice that $C[A/X] \in \mathbb{T}(\lambda 2^{la})$.

4. $\dfrac{\Gamma \vdash (x)\ \vec{\mathbf{r}} : T}{\Gamma \vdash (x)\ \vec{\mathbf{r}} : \forall X.T}\ \forall_I$ Then by the induction hypothesis $\exists A \in \mathbb{T}(\lambda 2^{la}), \alpha \in \mathcal{S}$ such that $T \equiv \alpha.A$, so $\forall X.T \equiv \forall X.\alpha.A \equiv \alpha.\forall X.A$.

□

Now we can prove the theorem. However, instead, we prove the most general case: $\Gamma \vdash \mathbf{t} : \alpha.A \Rightarrow \omega(\mathbf{t}{\downarrow}) = \alpha$

Proof. Structural induction on $\mathbf{t}{\downarrow}$. We take $\Gamma \vdash \mathbf{t}{\downarrow} : \alpha.A$, which is true by Theorem 3.2.1.

1. $\mathbf{t}{\downarrow} = \mathbf{0}$. Then $\omega(\mathbf{t}{\downarrow}) = 0$. In addition, by Lemma 3.2.17, $\alpha = 0$.

2. $\mathbf{t}{\downarrow} = x$ or $\mathbf{t}{\downarrow} = \lambda x.\mathbf{t}'$. Then $\omega(\mathbf{t}{\downarrow}) = 1$. In addition, by Lemma 3.2.16, $\alpha = 1$.

3. $\mathbf{t}{\downarrow} = \gamma.\mathbf{t}'$. Then $\omega(\mathbf{t}{\downarrow}) = \gamma.\omega(\mathbf{t}')$. By Lemma 3.2.15, $\exists U \in \mathcal{U}, \delta \in \mathcal{S}$ such that $\alpha.A \equiv \gamma.\delta.U$, and by Lemma 3.2.3, $\alpha = \gamma \times \delta$. Cases:

   $\alpha = 0$ Cases:

   $\gamma = 0$ Then $\omega(\gamma.\mathbf{t}') = 0 \times \omega(\mathbf{t}') = 0$, or

   $\gamma \neq 0, \delta = 0$ Then by Lemma 3.2.10, $\Gamma \vdash \mathbf{t}' : 0.U \equiv 0.A$, so by the induction hypothesis $\omega(\mathbf{t}') = 0$, and then $\omega(\gamma.\mathbf{t}') = \gamma \times 0 = 0$.

   $\alpha \neq 0$ Then $A \equiv U$, so by Lemma 3.2.10, $\Gamma \vdash \mathbf{t}' : \delta.A$. Then by the induction hypothesis $\omega(\mathbf{t}') = \delta$. Notice that $\omega(\mathbf{t}{\downarrow}) = \gamma \times \omega(\mathbf{t}') = \gamma \times \delta = \alpha$.

4. $\mathbf{t}{\downarrow} = \mathbf{t}_1 + \mathbf{t}_2$. Then $\omega(\mathbf{t}{\downarrow}) = \omega(\mathbf{t}_1) + \omega(\mathbf{t}_2)$. By Lemma 3.2.12, $\exists \sigma, \phi \in \mathcal{S}$ such that $\Gamma \vdash \mathbf{t}_1 : \sigma.A$ and $\Gamma \vdash \mathbf{t}_2 : \phi.A$ with $\sigma + \phi = \alpha$. Then by the induction hypothesis $\omega(\mathbf{t}_1) = \sigma$ and $\omega(\mathbf{t}_2) = \phi$, so $\omega(\mathbf{t}_1) + \omega(\mathbf{t}_2) = \sigma + \phi = \alpha$.

5. $\mathbf{t}{\downarrow} = (\mathbf{t}_1)\ \mathbf{t}_2$. Then $\omega(\mathbf{t}{\downarrow}) = \omega(\mathbf{t}_1) \times \omega(\mathbf{t}_2)$. By Lemma 3.2.8, $\exists U \in \mathcal{U}, T \in \mathcal{T}, \beta, \delta \in \mathcal{S}$ such that $\Gamma \vdash \mathbf{t}_1 : \beta.U \to T$ and $\Gamma \vdash \mathbf{t}_2 : \delta.U$ with $T \preceq A$ and $\beta \times \delta = \alpha$. Since $A \in \mathbb{T}(\lambda 2^{la})$, $T \in \mathbb{T}(\lambda 2^{la})$ and since $(\mathbf{t}_1)\ \mathbf{t}_2$ is in normal form, by Lemma B.17.1, $\mathbf{t}_1$ is a variable applied to something else, so by Lemma B.17.2, $U \to T \in \mathbb{T}(\lambda 2^{la})$, which implies that $U \in \mathbb{T}(\lambda 2^{la})$. Then by the induction hypothesis, $\omega(\mathbf{t}_1) = \beta$ and $\omega(\mathbf{t}_2) = \delta$, so $\omega(\mathbf{t}{\downarrow}) = \omega(\mathbf{t}_1) \times \omega(\mathbf{t}_2) = \beta \times \delta = \alpha$.

□

Appendix C

Proofs from Chapter 4

C.1 Proof of Lemma 4.2.3

Lemma 4.2.3 (Arrows comparison)**.** For any types T, R and any unit types U, V, if $V \to R \preceq U \to T$, then $\exists \vec{W}, \vec{X}$ such that $U \to T \equiv (V \to R)[\vec{W}/\vec{X}]$.

Proof. The proof is analogous to the proof of Lemma 3.2.6 (*cf.* Appendix B.2), however simpler, since the order in *Additive* is given only between unit types.

A map $(\cdot)^\circ : \Lambda_{\mathcal{U}} \mapsto \Lambda_{\mathcal{U}}$ is defined by

$$X^\circ = X \qquad (U \to T)^\circ = U \to T \qquad (\forall X.U)^\circ = U^\circ$$

We need two intermediate results.

1. For any types U, V, $\exists W$ such that $(U[V/X])^\circ \equiv U^\circ[W/X]$.
2. For any types U, V, if $U \preceq V$ then $\exists \vec{W}, \vec{X} \ / \ V^\circ \equiv U^\circ[\vec{W}/\vec{X}]$.

Proofs

1. Structural induction on U

   $U = X$ then $(X[V/X])^\circ = V^\circ = X[V^\circ/X] = X^\circ[V^\circ/X]$.

   $U = Y$ then $(Y[V/X])^\circ = Y = Y^\circ[V/X]$.

   $U = W \to T$ then $((W \to T)[V/X])^\circ = (W[V/X] \to T[V/X])^\circ = W[V/X] \to T[V/X] = (W \to T)[V/X] = (W \to T)^\circ[V/X]$.

   $U = \forall Y.U'$ then $((\forall Y.U')[V/X])^\circ = (\forall Y.U'[V/X])^\circ = (U'[V/X])^\circ$, which is, by the induction hypothesis, equivalent to $U'^\circ[W/X] = (\forall Y.U')^\circ[W/X]$.

2. It suffices to show this for $U \prec V$.

   Case 1 $V \equiv \forall X.U$. Then $V^\circ \equiv U^\circ$.

   Case 2 $U \equiv \forall X.U'$ and $V \equiv U'[W/X]$ then by the intermediate result 1 one has $V^\circ \equiv U'^\circ[W/X] \equiv U^\circ[W/X]$.

Proof of the lemma. $U \to T \equiv (U \to T)^\circ$, by the intermediate result 2, this is equivalent to $(V \to R)^\circ[\vec{W}/\vec{X}] \equiv (V \to R)[\vec{W}/\vec{X}]$. □

C.2 Proof of Lemma 4.2.4

Lemma 4.2.4 ($\preccurlyeq$-subsumption)**.** For any context Γ, any term $\mathbf{t}$ and any unit types U, V such that $U \preccurlyeq V$ and no free type variable in U occurs in Γ, if $\Gamma \vdash \mathbf{t} : U$ then $\Gamma \vdash \mathbf{t} : V$.

Proof. Analogous to proof of Lemma 3.2.13, by replacing all the occurrences of general types for unit types. □

C.3 Proof of Lemma 4.2.5

Lemma 4.2.5 (Generation lemma (app))**.** For any $\Gamma, T, \mathbf{t}$ and $\mathbf{u}$, if $\Gamma \vdash (\mathbf{t})\ \mathbf{u} : T$ is derivable, then there are $\alpha, \beta \in \mathbb{N}_0$, and some types U (in $\mathcal{U}$), $T_1, \ldots, T_\alpha$ such that $\Gamma \vdash \mathbf{t} : \sum_{i=1}^{\alpha}(U \to T_i)$ and $\Gamma \vdash \mathbf{u} : \sum_{j=1}^{\beta} U$ with $\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i \preccurlyeq T$ and each derivation of typing judgement for $\mathbf{t}$ and $\mathbf{u}$ smaller than the one for $(\mathbf{t})\ \mathbf{u}$.

Proof. Let $S_n = \Gamma \vdash (\mathbf{t})\ \mathbf{u} : T$. Induction on n.

1. $\dfrac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{\alpha} U \to T_i \qquad \Gamma \vdash \mathbf{u} : \sum_{j=1}^{\beta} U}{\Gamma \vdash (\mathbf{t})\ \mathbf{u} : \sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i}\ {\to_E}$ Trivial case.

2. $\dfrac{\Gamma \vdash (\mathbf{t})\ \mathbf{u} : \forall X.U}{\Gamma \vdash (\mathbf{t})\ \mathbf{u} : U[V/X]}\ \forall_E$ By the induction hypothesis $\exists r, s, \alpha, \beta \in \mathbb{N}, W \in \mathcal{U}, T_1, \ldots, T_n \in \mathcal{T}$ such that $S_r = \Gamma \vdash \mathbf{t} : \sum_{i=1}^{\alpha}(W \to T_i)$ and $S_s = \Gamma \vdash \mathbf{u} : \sum_{j=1}^{\beta} W$ with $\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i \preceq \forall X.U$ and where $\max(r, s) < n - 1$. Since $\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i \preceq \forall X.U \prec U[V/X]$ then by transitivity $\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i \preceq U[V/X]$. Remark that there is no conflict with the order defined only for unit types, it just means that $\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i$ is equivalent to a unit type.

3. $\dfrac{\Gamma \vdash (\mathbf{t})\ \mathbf{u} : U}{\Gamma \vdash (\mathbf{t})\ \mathbf{u} : \forall X.U}\ \forall_I$ By the induction hypothesis $\exists r, s, \alpha, \beta \in \mathbb{N}, V \in \mathcal{U}, T_1, \ldots, T_n \in \mathcal{T}$ such that $S_r = \Gamma \vdash \mathbf{t} : \sum_{i=1}^{\alpha}(V \to T_i)$ and $S_s = \Gamma \vdash \mathbf{u} : \sum_{j=1}^{\beta} V$ with $\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i \preceq U$ and where $\max(r, s) < n - 1$. Since $\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i \preceq U \prec \forall X.U$ then by transitivity $\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i \preceq \forall X.U$.

□

C.4 Proof of Lemma 4.2.6

Lemma 4.2.6 (Generation lemma (abs))**.** For any Γ, $\mathbf{t}$ and T, if $\Gamma \vdash \lambda x.\mathbf{t} : T$ is derivable, then there exist a unit type U and a type R such that $\Gamma, x : U \vdash \mathbf{t} : R$ and $U \to R \preccurlyeq T$.

Proof. Let $S_n = \Gamma \vdash \lambda x.\mathbf{t} : T$. Induction on n.

1. $\dfrac{\Gamma, x : U \vdash \mathbf{t} : T}{\Gamma \vdash \lambda x.\mathbf{t} : U \to T}\ {\to_I}$ This is the trivial case.

2. $\dfrac{\Gamma \vdash \lambda x.\mathbf{t} : \forall X.U}{\Gamma \vdash \lambda x.\mathbf{t} : U[V/X]}\ \forall_E$ By the induction hypothesis $\exists W \in \mathcal{U}, R \in \mathcal{T}$ such that $\Gamma, x : V \vdash \mathbf{t} : R$ and $V \to R \preceq \forall X.U \prec U[V/X]$.

3. $\dfrac{\Gamma \vdash \lambda x.\mathbf{t} : U}{\Gamma \vdash \lambda x.\mathbf{t} : \forall X.U}\,\forall_I$ By the induction hypothesis $\exists V \in \mathcal{U}, R \in \mathcal{T}$ such that $\Gamma, x : V \vdash \mathbf{t} : R$ and $V \to R \preceq U \prec \forall X.U$.

□

C.5 Proof of Lemma 4.2.7

Lemma 4.2.7 (Generation lemma (sum))**.** For any Γ, T, $\mathbf{r}$ and $\mathbf{u}$, if $\Gamma \vdash \mathbf{r} + \mathbf{u} : T$, then there are some types R, S such that $\Gamma \vdash \mathbf{r} : R$ and $\Gamma \vdash \mathbf{u} : S$ with $R + S \equiv T$ and typing derivations for $\mathbf{r}$ and $\mathbf{u}$ smaller than the one for $\mathbf{r} + \mathbf{u}$.

Proof. Let $\Gamma \vdash \mathbf{r} + \mathbf{u} : T$. Induction on n.

1. $\dfrac{\Gamma \vdash \mathbf{r} + \mathbf{u} : \forall X.U}{\Gamma \vdash \mathbf{r} + \mathbf{u} : U[V/X]}\,\forall_E$ Then by the induction hypothesis $\exists r, s \in \mathbb{N}, R, S \in \mathcal{T}$ such that $S_r = \Gamma \vdash \mathbf{r} : R$ and $S_s = \Gamma \vdash \mathbf{u} : S$ with $R + S \equiv \forall X.U \prec U[V/X]$ and where $\max(r, s) < n - 1$.
2. $\dfrac{\Gamma \vdash \mathbf{r} + \mathbf{u} : U}{\Gamma \vdash \mathbf{r} + \mathbf{u} : \forall X.U}\,\forall_I$ Analogous to previous case.
3. $\dfrac{\Gamma \vdash \mathbf{r} : R \qquad \Gamma \vdash \mathbf{u} : S}{\Gamma \vdash \mathbf{r} + \mathbf{u} : R + S}\,+_I$ Then take $S_r = \Gamma \vdash \mathbf{r} : R$ and $S_s = \Gamma \vdash \mathbf{u} : S$ and notice that $\max(r, s) = n - 1 < n$.

□

C.6 Proof of Lemma 4.2.8

Lemma 4.2.8 (Substitution)**.** For any Γ, T, U, $\mathbf{b}$ and $\mathbf{t}$,

1. $\Gamma \vdash \mathbf{t} : T \Rightarrow \Gamma[\vec{U}/\vec{X}] \vdash \mathbf{t} : T[\vec{U}/\vec{X}]$.
2. $\{\ \Gamma, x : U \vdash \mathbf{t} : T$ and $\Gamma \vdash \mathbf{b} : U\ \} \Rightarrow \Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T$.

Proof of item 1. We re-use the proof of Lemma 3.2.13 (*cf.* Appendix B.8)

- Cases 1, 2 and 4 remain the same.
- Case 3: replace all the occurrences of α by $\sum_{i=1}^{\alpha}$ and β by $\sum_{i=1}^{\beta}$.
- Cases 5 and 6: take T as unit type.
- Remove cases 7 and 8
- Add the following case:

  $\dfrac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{r} : R}{\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R}\,+_I$ By the induction hypothesis $\Gamma[\vec{U}/\vec{X}] \vdash \mathbf{t} : T[\vec{U}/\vec{X}]$ and also $\Gamma[\vec{U}/\vec{X}] \vdash \mathbf{r} : R[\vec{U}/\vec{X}]$, so by rule $+_I$, $\Gamma[\vec{U}/\vec{X}] \vdash \mathbf{t} + \mathbf{r} : T[\vec{U}/\vec{X}] + R[\vec{U}/\vec{X}]$. Notice that $T[\vec{U}/\vec{X}] + R[\vec{U}/\vec{X}] = (T + R)[\vec{U}/\vec{X}]$.

□

Proof of item 2. We re-use the proof of Lemma 3.2.13 (*cf.* Appendix B.8)

- Cases 1, 2, 3 and 5 remain the same.
- Case 4: replace all the occurrences of α by $\sum_{i=1}^{\alpha}$ and β by $\sum_{i=1}^{\beta}$.

- Cases 6 and 7: take T as unit type.
- Remove cases 8 and 9
- Add the following case:

$$\frac{\Gamma, x : U \vdash \mathbf{t} : T \qquad \Gamma \vdash, x : U \mathbf{r} : R}{\Gamma, x : U \vdash \mathbf{t} + \mathbf{r} : T + R} +_I$$

By the induction hypothesis $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T$ and also $\Gamma \vdash \mathbf{r}[\mathbf{b}/x] : R$, so by rule $+_I$, $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] + \mathbf{r}[\mathbf{b}/x] : T + R$. Notice that $\mathbf{t}[\mathbf{b}/x] + \mathbf{r}[\mathbf{b}/x] = (\mathbf{t} + \mathbf{r})[\mathbf{b}/x]$.

□

C.7 Proof of Theorem 4.2.1

Theorem 4.2.1 (Subject reduction)**.** For any terms $\mathbf{t}, \mathbf{t}'$, any context Γ and any type T, if $\mathbf{t} \to \mathbf{t}'$ then $\Gamma \vdash \mathbf{t} : T \Rightarrow \Gamma \vdash \mathbf{t}' : T$.

Proof. We treat separately every rule which can be applied in $\mathbf{t} \to \mathbf{t}'$.

Rule $(\mathbf{u} + \mathbf{t})\ \mathbf{r} \to (\mathbf{u})\ \mathbf{r} + (\mathbf{t})\ \mathbf{r}$**:** Let $\Gamma \vdash (\mathbf{u}+\mathbf{t})\ \mathbf{r} : T$. Then by Lemma 4.2.5, there exist $\alpha, \beta \in \mathbb{N}$, $U \in \mathcal{U}$ and $T_1, \ldots, T_n$ such that $\Gamma \vdash \mathbf{u} + \mathbf{t} : \sum_{i=1}^{\alpha}(U \to T_i)$, $\Gamma \vdash \mathbf{r} : \sum_{j=1}^{\beta} U$ and $\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i \preccurlyeq T$. By Lemma 4.2.7, there are some types R, S such that $\Gamma \vdash \mathbf{u} : R$, $\Gamma \vdash \mathbf{t} : S$ and $R + S \equiv \sum_{i=1}^{\alpha}(U \to T_i)$. In addition, by remark 4.1.2, $R \equiv \sum_{j=1}^{\delta} V_j$ and $S \equiv \sum_{j=\delta+1}^{\gamma} V_j$ for some $V_1, \ldots, V_\delta, \ldots, V_\gamma$ in $\mathcal{U}$ or equal to $\overline{0}$ (with $0 \leq \delta \leq \gamma$). So $\sum_{j=1}^{\delta} V_j + \sum_{j=\delta+1}^{\gamma} V_j \equiv \sum_{i=1}^{\alpha} U \to T_i$. Thus $\gamma = \alpha$ and (possibly after re-labelling the T_j's), $\sum_{j=1}^{\delta} V_j \equiv \sum_{j=1}^{\delta} U \to T_j$ and $\sum_{j=\delta+1}^{\alpha} V_j \equiv \sum_{j=\delta+1}^{\alpha} U \to T_j$ Then, by rule $\equiv$, $\Gamma \vdash \mathbf{u} : \sum_{j=1}^{\delta} U \to T_i$ and $\Gamma \vdash \mathbf{t} : \sum_{j=\delta+1}^{\alpha} U \to T_j$. So using $\to_E$ we derive $\Gamma \vdash (\mathbf{u})\ \mathbf{r} : \sum_{i=1}^{\delta}\sum_{j=1}^{\beta} T_i$ and $\Gamma \vdash (\mathbf{t})\ \mathbf{r} : \sum_{k=\delta+1}^{\alpha}\sum_{j=1}^{\beta} T_k$ and then, using $+_I$, one gets $\Gamma \vdash (\mathbf{u})\ \mathbf{r} + (\mathbf{t})\ \mathbf{r} : \sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i$. Since $\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i \preccurlyeq T$, we can conclude with Lemma 4.2.4 that $\Gamma \vdash (\mathbf{u})\ \mathbf{r} + (\mathbf{t})\ \mathbf{r} : T$.

Rule $(\mathbf{r})\ (\mathbf{u} + \mathbf{t}) \to (\mathbf{r})\ \mathbf{u} + (\mathbf{r})\ \mathbf{t}$**.** This is analogous to the previous case.

Rule $(\mathbf{0})\ \mathbf{t} \to \mathbf{0}$**.** Let $\Gamma \vdash (\mathbf{0})\ \mathbf{t} : T$. Then by Lemma 4.2.5 $\exists \alpha, \beta, U, T_1, \ldots, T_\alpha$ such that $\Gamma \vdash \mathbf{0} : \sum_{i=1}^{\alpha}(U \to T_i)$ and $\Gamma \vdash \mathbf{t} : \sum_{j=1}^{\beta} U$ with $\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i \preccurlyeq T$. Then by Lemma 4.2.10, $\alpha = 0$, and so Then $\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i \equiv \overline{0}$. Thus $0 \equiv \sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i \preccurlyeq T$. Notice that by $ax_{\overline{0}}$, $\Gamma \vdash \mathbf{0} : \overline{0}$. Then by Lemma 4.2.4, $\Gamma \vdash \mathbf{0} : T$.

Rules $(\mathbf{t})\ \mathbf{0} \to \mathbf{0}$**.** This is analogous to the previous case.

Rules $\mathbf{t} + \mathbf{0} \to \mathbf{t}$**.** Let $\Gamma \vdash \mathbf{t} + \mathbf{0} : T$. By Lemma 4.2.7, there are some types R and S such that $\Gamma \vdash \mathbf{t} : R$ and $\Gamma \vdash \mathbf{0} : S$, with $R + S \equiv T$. Then by Lemma 4.2.10, $S \equiv \overline{0}$ and so $T \equiv R + S \equiv R$. By rule $\equiv$, $\Gamma \vdash \mathbf{t} : T$.

Rule $(\lambda x.\mathbf{t})\ \mathbf{b} \to \mathbf{t}[\mathbf{b}/x]$**.** Let $\Gamma \vdash (\lambda x.\mathbf{t})\ \mathbf{b} : T$. Then by Lemma 4.2.5, $\exists \alpha$, β, U, $T_1, \ldots, T_n$ such that $\Gamma \vdash \lambda x.\mathbf{t} : \sum_{i=1}^{\alpha}(U \to T_i)$ and $\Gamma \vdash \mathbf{b} : \sum_{j=1}^{\beta} U$ with $\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} T_i \preccurlyeq T$. By Lemma 4.2.11, $\alpha = \beta = 1$, so $\Gamma \vdash \lambda x.\mathbf{t} : U \to T_1$, and $\Gamma \vdash \mathbf{b} : U$. Then by Corollary 4.2.9, $\Gamma, x : U \vdash \mathbf{t} : T_1$ and hence by Lemma 4.2.8, $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T_1$. In addition, since $T_1 \preccurlyeq T$, by Lemma 4.2.4, $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T$.

Rule $\mathbf{t} + \mathbf{r} \to \mathbf{r} + \mathbf{t}$**.** Let $\Gamma \vdash \mathbf{t} + \mathbf{r} : T$. Then by Lemma 4.2.7, $\exists R, S$ such that $\Gamma \vdash \mathbf{t} : R$ and $\Gamma \vdash \mathbf{r} : S$ with $R + S \equiv T$. Then by rule $+_I$, $\Gamma \vdash \mathbf{r} + \mathbf{t} : S + R$ and since $S + R \equiv R + S \equiv T$, by rule $\equiv$, $\Gamma \vdash \mathbf{r} + \mathbf{t} : T$.

Rule $(\mathbf{t}+\mathbf{r})+\mathbf{u}\to\mathbf{t}+(\mathbf{r}+\mathbf{u})$**.** Let $\Gamma\vdash(\mathbf{t}+\mathbf{r})+\mathbf{u}:T$. Then by Lemma 4.2.7, $\exists R_1,R_2$ such that $\Gamma\vdash\mathbf{t}+\mathbf{r}:R_1$ and $\Gamma\vdash\mathbf{u}:R_2$ with $R_1+R_2\equiv T$. Then by Lemma 4.2.7 again, $\exists S_1,S_2$ such that $\Gamma\vdash\mathbf{t}:S_1$ and $\Gamma\vdash\mathbf{r}:S_2$ with $S_1+S_2\equiv R_1$. So using rule $+_I$ twice in the correct order, we can derive $\Gamma\vdash\mathbf{t}+(\mathbf{r}+\mathbf{u}):S_1+(S_2+R_2)$. Since $S_1+(S_2+R_2)\equiv(S_1+S_2)+R_2\equiv R_1+R_2\equiv T$, then using rule $\equiv$, $\Gamma\vdash\mathbf{t}+(\mathbf{r}+\mathbf{u}):T$.

Rule $\mathbf{t}+\mathbf{r}\to\mathbf{t}'+\mathbf{r}$ **as a consequence of** $\mathbf{t}\to\mathbf{t}'$**.** Let $\Gamma\vdash\mathbf{t}+\mathbf{r}:T$, then by Lemma 4.2.7, $\exists S,R\in\mathcal{T}$ such that $\Gamma\vdash\mathbf{t}:S$ and $\Gamma\vdash\mathbf{r}:R$ with $R+S\preccurlyeq T$. Then by the induction hypothesis, $\Gamma\vdash\mathbf{t}':S$, so using rule $+_I$ one can derive $\Gamma\vdash\mathbf{t}'+\mathbf{r}:R+S$. Notice that by Lemma 4.2.4,$\Gamma\vdash\mathbf{t}'+\mathbf{r}:T$.

Rule $\mathbf{r}+\mathbf{t}\to\mathbf{r}+\mathbf{t}'$ **as a consequence of** $\mathbf{t}\to\mathbf{t}'$**.** Analogous to previous case.

Rule $(\mathbf{r})\ \mathbf{t}\to(\mathbf{r})\ \mathbf{t}'$ **as a consequence of** $\mathbf{t}\to\mathbf{t}'$**.** Let $\Gamma\vdash(\mathbf{r})\ \mathbf{t}:T$, then by Lemma 4.2.5, $\exists\alpha,\beta\in\mathbb{N},U\in\mathcal{U},T_1,\ldots,T_\alpha\in\mathcal{T}\ /\ \Gamma\vdash\mathbf{r}:\sum_{i=1}^{\alpha}(U\to T_i)$ and $\Gamma\vdash\mathbf{t}:\sum_{j=1}^{\beta}U$ with $\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta}T_i\preccurlyeq T$. By the induction hypothesis, $\Gamma\vdash\mathbf{t}':\sum_{j=1}^{\beta}U$, so using rule $\to_E$, one has $\Gamma\vdash(\mathbf{r})\ \mathbf{t}':\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta}T_i$. Notice that by Lemma 4.2.4, $\Gamma\vdash(\mathbf{r})\ \mathbf{t}':T$.

Rule $(\mathbf{t})\ \mathbf{r}\to(\mathbf{t}')\ \mathbf{r}$ **as a consequence of** $\mathbf{t}\to\mathbf{t}'$**.** Analogous to previous case.

Rule $\lambda x.\mathbf{t}\to\lambda x.\mathbf{t}'$ **as a consequence of** $\mathbf{t}\to\mathbf{t}'$**.** Let $\Gamma\vdash\lambda x.\mathbf{t}:T$, then by Lemma 4.2.6, $\exists U\in\mathcal{U},R\in\mathcal{T}$ such that $U\to R\preccurlyeq T$ and $\Gamma,x:U\vdash\mathbf{t}:R$. Then by the induction hypothesis, $\Gamma,x:U\vdash\mathbf{t}':R$, so using rule $\to_I$, one obtain $\Gamma\vdash\lambda x.\mathbf{t}':U\to R$. Notice that by Lemma 4.2.4, $\Gamma\vdash\lambda x.\mathbf{t}':T$.

□

C.8 Proof of Lemma 4.3.4

Lemma 4.3.4. $\mathsf{T}[w\mapsto\mathsf{T}'[s]]\equiv\mathsf{T}\circ\mathsf{T}'[wv\mapsto s(v)]$ where w denotes a ℓ-leaf of T, and v a ℓ-leaf of T'.

Proof. Induction on T.

$\mathsf{T}=\mathtt{Z}$**.** Then $\mathtt{Z}[w\mapsto\mathsf{T}'[s]]=\overline{0}$ and $\mathtt{Z}\circ\mathsf{T}'[wv\mapsto s(v)]=\mathtt{Z}[wv\mapsto s(v)]=\overline{0}$.

$\mathsf{T}=\ell$**.** Then $\ell[w\mapsto\mathsf{T}'[s]]=\mathsf{T}'[s]$ and $\ell\circ\mathsf{T}'[wv\mapsto s(v)]=\mathsf{T}'[wv\mapsto s(v)]=\mathsf{T}'[s]$.

$\mathsf{T}=\mathtt{S}(\mathsf{T}_1,\mathsf{T}_2)$**.** Then $\mathtt{S}(\mathsf{T}_1,\mathsf{T}_2)[w\mapsto\mathsf{T}'[s]]=\mathsf{T}_1[\mathtt{l}w\mapsto\mathsf{T}'[s]]+\mathsf{T}_2[\mathtt{r}w\mapsto\mathsf{T}'[s]]$ which by the induction hypothesis is equal to $\mathsf{T}_1\circ\mathsf{T}'[\mathtt{l}wv\mapsto s(v)]+\mathsf{T}_2\circ\mathsf{T}'[\mathtt{r}wv\mapsto s(v)]=\mathtt{S}(\mathsf{T}_1\circ\mathsf{T}',\mathsf{T}_2\circ\mathsf{T}')[wv\mapsto s(v)]=\mathtt{S}(\mathsf{T}_1,\mathsf{T}_2)\circ\mathsf{T}'[wv\mapsto s(v)]$.

□

C.9 Proof of Lemma 4.3.12

Lemma 4.3.12. If $T=\mathsf{T}[w\mapsto U_w]$ is a type of Add_{struct}, then $|T|=\mathsf{T}[w\mapsto|U_w|]$.

Proof. Induction on T.

1. $T=U$. Then $\mathsf{T}=\ell$ and so $\mathsf{T}[w\mapsto U_w]$ is just U_ε. Since $U=\ell[\varepsilon\mapsto U_\varepsilon]$, one has $|U|=|\ell[\varepsilon\mapsto U_\varepsilon]|=|U_\varepsilon|=\ell[\varepsilon\mapsto|U_\varepsilon|]$.
2. $T=\overline{0}$. Then $\mathsf{T}=\mathtt{Z}$ and so $\mathtt{Z}[w\mapsto U_w]=\overline{0}$ and $|T|=|\mathtt{Z}[w\mapsto U_w]|=|\overline{0}|=\star=\mathtt{Z}[w\mapsto|U_w|]$.

3. $T = R + S$. Then $\mathsf{T} = \mathtt{S}(\mathsf{T}_1, \mathsf{T}_2)$ where $R = \mathsf{T}_1[w \mapsto U_{\mathtt{1}w}]$ and $S = \mathsf{T}_2[w \mapsto U_{\mathtt{r}w}]$. Then by the induction hypothesis $|R| = \mathsf{T}_1[w \mapsto |U_{\mathtt{1}w}|]$ and $|S| = \mathsf{T}_2[w \mapsto |U_{\mathtt{r}w}|]$. So, $|T| = |R+S| = |R| \times |S| = \mathsf{T}_1[w \mapsto |U_{\mathtt{1}w}|] \times \mathsf{T}_2[w \mapsto |U_{\mathtt{r}w}|] = \mathtt{S}(\mathsf{T}_1, \mathsf{T}_2)[w \mapsto |U_w|] = \mathsf{T}[w \mapsto |U_w|]$.

□

C.10 Proof of Theorem 4.3.21

First we need the following intermediate result.

Lemma C.10.1. *Let* $\mathcal{D}_1 = \Gamma, x\!:\!U \vdash \mathbf{t}\!:\!T$ *and* $\mathcal{D}_2 = \Gamma \vdash \mathbf{b}\!:\!U$*, then* $\exists \mathcal{D}_3$ *such that* $[\mathbf{t}]_{\mathcal{D}_1}[[\mathbf{b}]_{\mathcal{D}_2}/x] = [\mathbf{t}[\mathbf{b}/x]]_{\mathcal{D}_3}$.

Proof. We call $\mathcal{D}_1'$ to the last derivation from where to obtain $\mathcal{D}_1$, using rule R. If there are two sequents in the last step, we call them $\mathcal{D}_1'$ and $\mathcal{D}_1''$. If $R = \forall_I$, then by the induction hypothesis, $\exists \mathcal{D}_3$ such that $[\mathbf{t}]_{\mathcal{D}_1'}[[\mathbf{b}]_{\mathcal{D}_2}/x] = [\mathbf{t}[\mathbf{b}/x]]_{\mathcal{D}_3}$. Notice that $[\mathbf{t}]_{\mathcal{D}_1}[[\mathbf{b}]_{\mathcal{D}_2}/x] = [\mathbf{t}]_{\mathcal{D}_1'}[[\mathbf{b}]_{\mathcal{D}_2}/x]$. The case $R = \forall_E$ is similar.

For any other case, we proceed by structural induction on $\mathbf{t}$.

1. $\mathbf{t} = x$. Then $[x]_{\mathcal{D}_1} = x$ and so $[x]_{\mathcal{D}_1}[[\mathbf{b}]_{\mathcal{D}_2}/x] = [\mathbf{b}]_{\mathcal{D}_2} = [x[\mathbf{b}/x]]_{\mathcal{D}_2}$. Take $\mathcal{D}_3 = \mathcal{D}_2$.

2. $\mathbf{t} = y$. Then $[y]_{\mathcal{D}_1} = y$ and so $[y]_{\mathcal{D}_1}[[\mathbf{b}]_{\mathcal{D}_2}/x] = y = [y[\mathbf{b}/x]]_{\mathcal{D}_1}$. Take $\mathcal{D}_3 = \mathcal{D}_1$.

3. $\mathbf{t} = \mathbf{0}$. Then $[\mathbf{0}]_{\mathcal{D}_1} = \star$ and so $[\mathbf{0}]_{\mathcal{D}_1}[[\mathbf{b}]_{\mathcal{D}_2}/x] = \star = [\mathbf{0}[\mathbf{b}/x]]_{\mathcal{D}_1}$. Take $\mathcal{D}_3 = \mathcal{D}_1$.

4. $\mathbf{t} = \lambda y.\mathbf{r}$. Then $R = \!\rightarrow_I$. By the induction hypothesis, $\exists \mathcal{D}_3$ such that $[\mathbf{r}]_{\mathcal{D}_1'}[[\mathbf{b}]_{\mathcal{D}_2}/x] = [\mathbf{r}[\mathbf{b}/x]]_{\mathcal{D}_3}$. Then $[\mathbf{t}]_{\mathcal{D}_1}[[\mathbf{b}]_{\mathcal{D}_2}/x] = (\lambda y.[\mathbf{r}]_{\mathcal{D}_1'})[[\mathbf{b}]_{\mathcal{D}_2}/x] = \lambda y.([\mathbf{r}]_{\mathcal{D}_1'}[[\mathbf{b}]_{\mathcal{D}_2}/x]) = \lambda y.[\mathbf{r}[\mathbf{b}/x]]_{\mathcal{D}_3} = [\mathbf{t}[\mathbf{b}/x]]_{\mathcal{D}_3'}$, for some $\mathcal{D}_3'$ which is either obtained by applying $\rightarrow_I$ from $\mathcal{D}_3'$, if y is in the context of the last sequent of $\mathcal{D}_3'$, or by adding it to the context of each sequent in $\mathcal{D}_3'$.

5. $\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2$. By the induction hypothesis $\exists \mathcal{D}_3$ and $\mathcal{D}_3'$ such that $[\mathbf{t}_1]_{\mathcal{D}_1'}[[\mathbf{b}]_{\mathcal{D}_2}/x] = [\mathbf{t}_1[\mathbf{b}/x]]_{\mathcal{D}_3}$ and $[\mathbf{t}_2]_{\mathcal{D}_1''}[[\mathbf{b}]_{\mathcal{D}_2}/x] = [\mathbf{t}_2[\mathbf{b}/x]]_{\mathcal{D}_3'}$. Then

$$\begin{aligned}
[\mathbf{t}_1 + \mathbf{t}_2]_{\mathcal{D}_1}[[\mathbf{b}]_{\mathcal{D}_2}/x] &= \langle [\mathbf{t}_1]_{\mathcal{D}_1'}, [\mathbf{t}_1]_{\mathcal{D}_1'}\rangle[[\mathbf{b}]_{\mathcal{D}_2}/x] \\
&= \langle [\mathbf{t}_1]_{\mathcal{D}_1'}[[\mathbf{b}]_{\mathcal{D}_2}/x], [\mathbf{t}_1]_{\mathcal{D}_1'}[[\mathbf{b}]_{\mathcal{D}_2}/x]\rangle \\
&= \langle [\mathbf{t}_1[\mathbf{b}/x]]_{\mathcal{D}_3}, [\mathbf{t}_2[\mathbf{b}/x]]_{\mathcal{D}_3'}\rangle \\
&= [\mathbf{t}_1[\mathbf{b}/x] + \mathbf{t}_2[\mathbf{b}/x]]_{\mathcal{D}_3''}
\end{aligned}$$

where $\mathcal{D}_3''$ is obtained from $\mathcal{D}_3$ and $\mathcal{D}_3'$ by rule $+_I$. Notice that $\mathbf{t}_1[\mathbf{b}/x] + \mathbf{t}_2[\mathbf{b}/x] = (\mathbf{t}_1 + \mathbf{t}_2)[\mathbf{b}/x]$.

6. $\mathbf{t} = (\mathbf{t}_1)\ \mathbf{t}_2$. By the induction hypothesis $\exists \mathcal{D}_3$ and $\mathcal{D}_3'$ such that $[\mathbf{t}_1]_{\mathcal{D}_1'}[[\mathbf{b}]_{\mathcal{D}_2}/x] = [\mathbf{t}_1[\mathbf{b}/x]]_{\mathcal{D}_3}$ and $[\mathbf{t}_2]_{\mathcal{D}_1''}[[\mathbf{b}]_{\mathcal{D}_2}/x] = [\mathbf{t}_2[\mathbf{b}/x]]_{\mathcal{D}_3'}$. Then

$$\begin{aligned}
[(\mathbf{t}_1)\ \mathbf{t}_2]_{\mathcal{D}_1}[[\mathbf{b}]_{\mathcal{D}_2}/x] &= \mathsf{T} \circ \mathsf{T}'[wv \mapsto \pi_{\overline{w}}([\mathbf{t}_1]_{\mathcal{D}_1'})\pi_{\overline{v}}([\mathbf{t}_2]_{\mathcal{D}_1''})][[\mathbf{b}]_{\mathcal{D}_2}/x] \\
&= \mathsf{T} \circ \mathsf{T}'[wv \mapsto \pi_{\overline{w}}([\mathbf{t}_1]_{\mathcal{D}_1'}[[\mathbf{b}]_{\mathcal{D}_2}/x])\pi_{\overline{v}}([\mathbf{t}_2]_{\mathcal{D}_1''}[[\mathbf{b}]_{\mathcal{D}_2}/x])] \\
&= \mathsf{T} \circ \mathsf{T}'[wv \mapsto \pi_{\overline{w}}([\mathbf{t}_1[\mathbf{b}/x]]_{\mathcal{D}_3})\pi_{\overline{v}}([\mathbf{t}_2[\mathbf{b}/x]]_{\mathcal{D}_3'})] \\
&= [(\mathbf{t}_1[\mathbf{b}/x])\ \mathbf{t}_2[\mathbf{b}/x]]_{\mathcal{D}_3''}
\end{aligned}$$

where $\mathcal{D}_3''$ is obtained from $\mathcal{D}_3$ and $\mathcal{D}_3'$ by rule $\rightarrow_{E'}$. Notice that $(\mathbf{t}_1[\mathbf{b}/x])\ \mathbf{t}_2[\mathbf{b}/x] = ((\mathbf{t}_1)\ \mathbf{t}_2)[\mathbf{b}/x]$.

□

Now we can prove the theorem.

Theorem 4.3.21. Let $\Gamma \vdash \mathbf{t} : T$ be derivable (by $\mathcal{D}$) in Add_{struct}, and $\mathbf{t} \to \mathbf{r}$. If the reduction is not due to rule $\mathbf{t} + \mathbf{0} \to \mathbf{t}$, then there is $\mathcal{D}$' deriving $\Gamma \vdash \mathbf{r} : T$, and $[\mathbf{t}]_{\mathcal{D}} \to^* [\mathbf{r}]_{\mathcal{D}'}$.

Proof. Induction over $\mathcal{D}$.

1. $\mathcal{D} = ax$ or $ax_{\overline{0}}$. Impossible since nor x nor $\mathbf{0}$ can reduce.
2. $\mathcal{D} = \dfrac{\Gamma \vdash \mathbf{t} : \mathsf{T}[w \mapsto (U \to T_w)] \quad \Gamma \vdash \mathbf{u} : \mathsf{T}'[v \mapsto U]}{\Gamma \vdash (\mathbf{t})\ \mathbf{u} : \mathsf{T} \circ \mathsf{T}'[wv \mapsto T_w]} \to_{E'}$

   Then $[(\mathbf{t})\ \mathbf{u}]_{\mathcal{D}} = \mathsf{T} \circ \mathsf{T}'[wv \mapsto \pi_{\bar{w}}([\mathbf{t}]_{\mathcal{D}_1})\pi_{\bar{v}}([\mathbf{u}]_{\mathcal{D}_2})]$.

   Consider $(\mathbf{t})\ \mathbf{u} \to \mathbf{r}$.

   - If $\mathbf{r} = (\mathbf{t}')\ \mathbf{u}$ with $\mathbf{t} \to \mathbf{t}'$, then by the induction hypothesis there is $\mathcal{D}_3 = \Gamma \vdash \mathbf{t}' : \mathsf{T}[w \mapsto (U \to T_w)]$ such that $[\mathbf{t}]_{\mathcal{D}_1} \to^* [\mathbf{t}']_{\mathcal{D}_3}$. Then take

     $$\mathcal{D}_4 = \frac{\Gamma \vdash \mathbf{t}' : \mathsf{T}[w \mapsto (U \to T_w)] \qquad \Gamma \vdash \mathbf{u} : \mathsf{T}'[v \mapsto U]}{\Gamma \vdash (\mathbf{t}')\ \mathbf{u} : \mathsf{T} \circ \mathsf{T}'[wv \mapsto T_w]} \to_{E'}$$

     Notice that $[(\mathbf{t})\ \mathbf{u}]_{\mathcal{D}} = \mathsf{T} \circ \mathsf{T}'[wv \mapsto \pi_{\bar{w}}([\mathbf{t}]_{\mathcal{D}_1})\pi_{\bar{v}}([\mathbf{u}]_{\mathcal{D}_2})] \to^* \mathsf{T} \circ \mathsf{T}'[wv \mapsto \pi_{\bar{w}}([\mathbf{t}']_{\mathcal{D}_3})\pi_{\bar{v}}([\mathbf{u}]_{\mathcal{D}_2})] = [(\mathbf{t}')\ \mathbf{u}]_{\mathcal{D}_4}$.
   - If $\mathbf{r} = (\mathbf{t})\ \mathbf{u}'$ with $\mathbf{u} \to \mathbf{u}'$, this is analogous to the previous case.
   - If $\mathbf{r} = (\mathbf{t}_1)\ \mathbf{u} + (\mathbf{t}_2)\ \mathbf{u}$ with $\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2$, then by Lemma 4.2.7, $\exists T_1, T_2$ such that $\Gamma \vdash \mathbf{t}_1 : T_1$ and $\Gamma \vdash \mathbf{t}_2 : T_2$ with $T_1 + T_2 = \mathsf{T}[w \mapsto (U \to T_w)]$. Notice that this lemma has been proved for *Additive*, not for Add_{struct}, however to prove the equivalent result is trivial and we use it without distinction. Let $T_1 = \mathsf{T}_1[u_1]$ and $T_2 = \mathsf{T}_2[u_2]$. Then $T_1 + T_2 = \mathsf{S}(\mathsf{T}_1, \mathsf{T}_2)[\mathtt{l}w \mapsto u_1(w), \mathtt{r}w \mapsto u_2(w)] = \mathsf{T}[w \mapsto (U \to T_w)]$. Also notice that

     $$\mathsf{T} \circ \mathsf{T}'[u] = \mathsf{S}(\mathsf{T}_1 \circ \mathsf{T}', \mathsf{T}_2 \circ \mathsf{T}')[u] = \mathsf{T}_1 \circ \mathsf{T}'[u] + \mathsf{T}_2 \circ \mathsf{T}'[u] \tag{C.1}$$

     Take $u = wv \mapsto \pi_{\bar{w}}([\mathbf{t}_1 + \mathbf{t}_2]_{\mathcal{D}_1})\pi_{\bar{v}}([\mathbf{u}]_{\mathcal{D}_2})$.

     On the other hand, for any $\mathcal{D}'$, $[(\mathbf{t}_1)\ \mathbf{u} + (\mathbf{t}_2)\ \mathbf{u}]_{\mathcal{D}'} = [(\mathbf{t}_1)\ \mathbf{u}]_{\mathcal{D}'_1} + [(\mathbf{t}_2)\ \mathbf{u}]_{\mathcal{D}'_2}$. Since we know the type for $\mathbf{t}_1$ and the type for $\mathbf{t}_2$, we can derive

     $$\frac{\Gamma \vdash \mathbf{t}_1 : \mathsf{T}_1[w_1 \mapsto (U \to T_{w_1})] \quad \Gamma \vdash \mathbf{u} : \mathsf{T}'[v \mapsto U]}{\Gamma \vdash (\mathbf{t}_1)\ \mathbf{u} : \mathsf{T}_1 \circ \mathsf{T}'[w_1 v \mapsto T_{w_1}]} \to_{E'}$$

     and

     $$\frac{\Gamma \vdash \mathbf{t}_2 : \mathsf{T}_2[w_2 \mapsto (U \to T_{w_2})] \quad \Gamma \vdash \mathbf{u} : \mathsf{T}'[v \mapsto U]}{\Gamma \vdash (\mathbf{t}_2)\ \mathbf{u} : \mathsf{T}_2 \circ \mathsf{T}'[w_2 v \mapsto T_{w_2}]} \to_{E'}$$

     So, using those derivation trees as $\mathcal{D}'_1$ and $\mathcal{D}'_2$ the following translation follows:

     $[(\mathbf{t}_1)\ \mathbf{u}]_{\mathcal{D}'_1} = \mathsf{T}_1 \circ \mathsf{T}'[w_1 v \mapsto \pi_{\bar{w}}([\mathbf{t}_1]_{\mathcal{D}'_{11}})\pi_{\bar{v}}([\mathbf{u}]_{\mathcal{D}_2})]$ and

     $[(\mathbf{t}_1)\ \mathbf{u}]_{\mathcal{D}'_2} = \mathsf{T}_1 \circ \mathsf{T}'[w_2 v \mapsto \pi_{\bar{w}}([\mathbf{t}_2]_{\mathcal{D}'_{21}})\pi_{\bar{v}}([\mathbf{u}]_{\mathcal{D}_2})]$.

     So,

     $$[(\mathbf{t}_1)\ \mathbf{u} + (\mathbf{t}_2)\ \mathbf{u}]_{\mathcal{D}'} = \mathsf{T}_1 \circ \mathsf{T}'[w_1 v \mapsto \pi_{\bar{w}}([\mathbf{t}_1]_{\mathcal{D}'_{11}})\pi_{\bar{v}}([\mathbf{u}]_{\mathcal{D}_2})] + \mathsf{T}_1 \circ \mathsf{T}'[w_2 v \mapsto \pi_{\bar{w}}([\mathbf{t}_2]_{\mathcal{D}'_{21}})\pi_{\bar{v}}([\mathbf{u}]_{\mathcal{D}_2})] \tag{C.2}$$

     Notice that $wv \mapsto \pi_{\bar{w}}([\mathbf{t}_1 + \mathbf{t}_2]_{\mathcal{D}_1})\pi_{\bar{v}}([\mathbf{u}]_{\mathcal{D}_2})$ is equal to $\mathtt{l}w_1 v \mapsto \pi_{\bar{w}}([\mathbf{t}_1]_{\mathcal{D}'_{11}})\pi_{\bar{v}}([\mathbf{u}]_{\mathcal{D}_2}), \mathtt{r}w_2 v \mapsto \pi_{\bar{w}}([\mathbf{t}_2]_{\mathcal{D}'_{21}})\pi_{\bar{v}}([\mathbf{u}]_{\mathcal{D}_2})$ making (C.1)=(C.2).

- If $\mathbf{r} = (\mathbf{t})\ \mathbf{u}_1 + (\mathbf{t})\ \mathbf{u}_2$ with $\mathbf{u} = \mathbf{u}_1 + \mathbf{u}_2$, this is analogous to the previous case.
- If $\mathbf{r} = \mathbf{0}$ with $\mathbf{t} = \mathbf{0}$, then by Lemma 4.2.10, $\mathsf{T}[w \mapsto (U \to T_w)] = \overline{0} = \mathsf{Z}[w \mapsto (U \to T_w)]$. Then $[(\mathbf{0})\ \mathbf{u}]_{\mathcal{D}} = \mathsf{Z}\circ\mathsf{T}'[wv \mapsto \pi_{\bar{w}}([\mathbf{0}]_{\mathcal{D}_1})\pi_{\bar{v}}([\mathbf{u}]_{\mathcal{D}_2})] = \mathsf{Z}[wv \mapsto \pi_{\bar{w}}([\mathbf{0}]_{\mathcal{D}_1})\pi_{\bar{v}}([\mathbf{u}]_{\mathcal{D}_2})] = \star$. Take $\mathcal{D}' = \dfrac{}{\Gamma \vdash \mathbf{0} : \overline{0}}\, ax_{\overline{0}}$, so $[\mathbf{0}]_{\mathcal{D}'} = \star$.
- If $\mathbf{r} = \mathbf{0}$ with $\mathbf{u} = \mathbf{0}$, this is analogous to the previous case.
- If $\mathbf{r} = \mathbf{t}'[\mathbf{b}/x]$ with $\mathbf{t} = \lambda x.\mathbf{t}'$ and $\mathbf{u} = \mathbf{b}$, then by Lemma 4.2.11 $\mathsf{T} = \mathsf{T}' = \ell$, so

$$\ell\circ\ell[wv \mapsto \pi_{\bar{w}}([\lambda x.\mathbf{t}']_{\mathcal{D}_1})\pi_{\bar{b}}([\mathbf{b}]_{\mathcal{D}_2})] = \ell[wv \mapsto \pi_{\bar{w}}([\lambda x.\mathbf{t}']_{\mathcal{D}_1})\pi_{\bar{b}}([\mathbf{b}]_{\mathcal{D}_2})] = [\lambda x.\mathbf{t}']_{\mathcal{D}_1}[\mathbf{b}]_{\mathcal{D}_2}$$

  In the case where the last rule in $\mathcal{D}_1$ is $\forall_I$ or $\forall_E$, take $\mathcal{D}'$ as the same derivation until the previous sequent (before the use of this last rule). Repeat the process until you obtain a $\mathcal{D}'$ such that the last derivation is done by rule $\to_I$. Then $[\lambda x.\mathbf{t}']_{\mathcal{D}_1}[\mathbf{b}]_{\mathcal{D}_2} = [\lambda x.\mathbf{t}']_{\mathcal{D}'}[\mathbf{b}]_{\mathcal{D}_2} = \lambda x.[\mathbf{t}']_{\mathcal{D}''}[\mathbf{b}]_{\mathcal{D}_2} \to [\mathbf{t}']_{\mathcal{D}''}[[\mathbf{b}]_{\mathcal{D}_2}/x]$. By Lemma C.10.1, $\exists\mathcal{D}_3$ such that this last expression is equal to $[\mathbf{t}[\mathbf{b}/x]]_{\mathcal{D}_3}$.

3. $\mathcal{D} = \dfrac{\Gamma, x : U \vdash \mathbf{t} : T}{\Gamma \vdash \lambda x.\mathbf{t} : T}\to_I$

   Then $[\lambda x.\mathbf{t}]_{\mathcal{D}} = \lambda x.[\mathbf{t}]_{\mathcal{D}'}$. Notice that the only possible reduction for $\lambda x.\mathbf{t}$ is when $\mathbf{t} \to \mathbf{t}'$, then by the induction hypothesis $\exists D_1 = \Gamma, x : U \vdash \mathbf{t}' : T$ such that $[\mathbf{t}]_{\mathcal{D}'} \to^* [\mathbf{t}']_{\mathcal{D}_1}$. Let $\mathcal{D}_2$ be obtained from $\mathcal{D}_1$ by rule $\to_I$, then $[\lambda x.\mathbf{t}]_{\mathcal{D}} = \lambda x.[\mathbf{t}]_{\mathcal{D}'} \to^* \lambda x.[\mathbf{t}']_{\mathcal{D}_1} = [\lambda x.\mathbf{t}']_{\mathcal{D}_2}$.

4. $\mathcal{D} = \dfrac{\Gamma \vdash \mathbf{t} : \forall X.U}{\Gamma \vdash \mathbf{t} : U[V/X]}\forall_E$

   Then $[\mathbf{t}]_{\mathcal{D}} = [\mathbf{t}]_{\mathcal{D}'}$. By the induction hypothesis, $\exists\mathcal{D}_1 = \Gamma \vdash \mathbf{t}' : \forall X.U$ such that $[\mathbf{t}]_{\mathcal{D}'} \to^* [\mathbf{t}']_{\mathcal{D}_1}$.

5. $\mathcal{D} = \forall_I$. This is analogous to the previous case.

6. $\mathcal{D} = \dfrac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{u} : R}{\Gamma \vdash \mathbf{t} + \mathbf{u} : T + R} +_I$

   Then $[\mathbf{t} + \mathbf{u}]_{\mathcal{D}} = \langle [\mathbf{t}]_{\mathcal{D}_1}, [\mathbf{u}]_{\mathcal{D}_2} \rangle$. Cases

   - $\mathbf{t}' = \mathbf{r} + \mathbf{u}$ with $\mathbf{t} \to \mathbf{r}$. Then by the induction hypothesis, $\exists D' = \Gamma \vdash \mathbf{r} : T$ such that $[\mathbf{t}]_{\mathcal{D}_1} \to^* [\mathbf{r}]_{\mathcal{D}'}$. Let $\mathcal{D}_3$ be obtained from $\mathcal{D}'$ and $\mathcal{D}_2$ by rule $+_I$. So $[\mathbf{t}+\mathbf{u}]_{\mathcal{D}} = \langle [\mathbf{t}]_{\mathcal{D}_1}, [\mathbf{u}]_{\mathcal{D}_2} \rangle \to^* \langle [\mathbf{r}]_{\mathcal{D}'}, [\mathbf{u}]_{\mathcal{D}_2} \rangle = [\mathbf{r} + \mathbf{u}]_{\mathcal{D}_3}$.
   - $\mathbf{t}' = \mathbf{t} + \mathbf{r}$ with $\mathbf{u} \to \mathbf{r}$. This is analogous to the previous case.

□

C.11 Proof of Corollary 4.3.22

Corollary 4.3.22 (Strong normalisation)**.** If $\Gamma \vdash \mathbf{t} : T$ is derivable in *Additive*, then $\mathbf{t}$ is strongly normalising.

Proof. First notice that if $\Gamma \vdash \mathbf{t} : T$ is derivable in *Additive*, then by Proposition 4.3.7, $\exists T' \equiv T$ such that $\Gamma \vdash \mathbf{t} : T'$ is derivable in Add_{struct}. Let us call this derivation $\mathcal{D}$. Then by Theorem 4.3.14, $|\Gamma| \vdash_F [\mathbf{t}]_{\mathcal{D}} : |T'|$ is derivable in System F_P.

Assume $\mathbf{t}$ is not strongly normalising, say $\mathbf{t} \to \mathbf{t}_1 \to \mathbf{t}_2 \to \cdots$. For a first approximation, consider that none of these reductions happens by rule $\mathbf{t}+\mathbf{0} \to \mathbf{t}$. Then by Theorem 4.3.21 there exists derivations $\mathcal{D}_1, \mathcal{D}_1, \ldots$, such that $[\mathbf{t}]_{\mathcal{D}} \to^* [\mathbf{t}_1]_{\mathcal{D}_1} \to^* [\mathbf{t}_2]_{\mathcal{D}_1} \to^* \cdots$. However, due to the strong normalisation of F_p, there exists a natural number n such that, $\forall i \geq n$, $[\mathbf{t}_i]_{\mathcal{D}_i} = [\mathbf{t}_{i+1}]_{\mathcal{D}_{i+1}}$.

We proceed in two steps:

1. We prove that, if $\mathbf{t}_i \to \mathbf{t}_{i+1}$ and $[\mathbf{t}_i]_{\mathcal{D}_i} = [\mathbf{t}_{i+1}]_{\mathcal{D}_{i+1}}$, then the reduction is an algebraic rule, (*i.e.* not a beta-reduction).

2. Then we show that algebraic rules are strictly decreasing with respect to the following measure, which is always positive: $|\mathbf{0}| = 0$, $|x| = 1$, $|\lambda x.\mathbf{t}| = |\mathbf{t}|$, $|\mathbf{t}+\mathbf{r}| = 2+|\mathbf{t}|+|\mathbf{r}|$, $|(\mathbf{t})\ \mathbf{r}| = (3|\mathbf{t}|+2)(3|\mathbf{r}|+2)$.

By item 1 only algebraic rules happen, which are strictly decreasing in the positive measure of item 2. Since $\mathbf{t} + \mathbf{0} \to \mathbf{t}$ is also strictly decreasing, then $\mathbf{t}$ has to be strongly normalising.

We proceed with the proofs:

1. We show that if $\mathbf{t}_i \to_\beta \mathbf{t}_{i+1}$, then $[\mathbf{t}_i]_{\mathcal{D}_i} \neq [\mathbf{t}_{i+1}]_{\mathcal{D}_{i+1}}$. Induction on the structure of $\mathbf{t}_i$. Notice that we do not consider the cases of $\mathcal{D}_i$ ending with $\forall_I$ or $\forall_E$, since these rules do not change the translations.

   $(\lambda x.\mathbf{r})\ \mathbf{b} \to_\beta \mathbf{r}[\mathbf{b}/x]$**:** We can assume that $\mathcal{D}_i$ ends with $\to_E$, so $[(\lambda x.\mathbf{r})\ \mathbf{b}]_{\mathcal{D}_i} = \mathsf{T} \circ \mathsf{T}'[wv \mapsto \pi_{\overline{w}}([\lambda x.\mathbf{r}]_{\mathcal{D}_{i1}})\pi_{\overline{v}}([\mathbf{b}]_{\mathcal{D}_{i2}})]$, where $\mathcal{D}_{i1}$ ends with the sequent assigning a type with the structure of T to the term $\lambda x.\mathbf{r}$ and $\mathcal{D}_{i2}$ ends with the sequent assigning a type with the structure of T' to $\mathbf{b}$. Notice that by Lemma 4.2.11, $\mathsf{T} = \mathsf{T}' = \ell$, so $[(\lambda x.\mathbf{r})\ \mathbf{b}]_{\mathcal{D}_i} = \mathsf{T} \circ \mathsf{T}'[wv \mapsto \pi_{\overline{w}}([\lambda x.\mathbf{r}]_{\mathcal{D}_{i1}})\pi_{\overline{v}}([\mathbf{b}]_{\mathcal{D}_{i2}})] = ([\lambda x.\mathbf{r}]_{\mathcal{D}_{i1}})\ [\mathbf{b}]_{\mathcal{D}_{i2}} = (\lambda x.[\mathbf{r}]_{\mathcal{D}'_{i1}})\ [\mathbf{b}]_{\mathcal{D}_{i2}}$, where $\mathcal{D}'_{i1}$ is the derivation from where $\mathcal{D}_{i1}$ is obtained with $\to_I$. We can trivially ensure that this term cannot be equal to $[\mathbf{r}[\mathbf{b}/x]]_{\mathcal{D}_{i+1}}$, for any $\mathcal{D}_{i+1}$.

   $\mathbf{r}_1 + \mathbf{r}_2 \to \mathbf{r}'_1 + \mathbf{r}_2$ **with** $\mathbf{r}_1 \to_\beta \mathbf{r}'_1$**:** We can assume that $\mathcal{D}_i$ ends with $+_I$. Then $[\mathbf{r}_1 + \mathbf{r}_2]_{\mathcal{D}_i} = \langle [\mathbf{r}_1]_{\mathcal{D}_{i1}}, [\mathbf{r}_2]_{\mathcal{D}_{i2}} \rangle$, where $[\mathbf{r}_j]_{\mathcal{D}_{ij}}$ is the derivation of the type for $\mathbf{r}_j$. Analogously, $[\mathbf{r}'_1 + \mathbf{r}_2]_{\mathcal{D}_{i+1}} = \langle [\mathbf{r}'_1]_{\mathcal{D}_{i+1,1}}, [\mathbf{r}_2]_{\mathcal{D}_{i+1,2}} \rangle$. By the induction hypothesis, $[\mathbf{r}_1]_{\mathcal{D}_{i1}} \neq [\mathbf{r}'_1]_{\mathcal{D}_{i+1,1}}$, so $\langle [\mathbf{r}_1]_{\mathcal{D}_{i1}}, [\mathbf{r}_2]_{\mathcal{D}_{i2}} \rangle \neq \langle [\mathbf{r}'_1]_{\mathcal{D}_{i+1,1}}, [\mathbf{r}_2]_{\mathcal{D}_{i+1,2}} \rangle$.

   $\mathbf{r}_1 + \mathbf{r}_2 \to \mathbf{r}_1 + \mathbf{r}'_2$ **with** $\mathbf{r}_2 \to_\beta \mathbf{r}'_2$**:** Analogous to previous case.

   $\lambda x.\mathbf{r} \to \lambda x.\mathbf{r}'$ **with** $\mathbf{r} \to_\beta \mathbf{r}'$**:** We can assume that $\mathcal{D}_i$ ends with $\to_I$, so there is a derivation $\mathcal{D}'_i$ for $\mathbf{r}$, where x is in the context and $[\lambda x.\mathbf{r}]_{\mathcal{D}_i} = \lambda x.[\mathbf{r}]_{\mathcal{D}'_i}$, which by induction hypothesis is not equal to $\lambda x.[\mathbf{r}']_{\mathcal{D}'_{i+1}} = [\lambda x.\mathbf{r}']_{\mathcal{D}_{i+1}}$.

   $(\mathbf{r}_1)\ \mathbf{r}_2 \to (\mathbf{r}'_1)\ \mathbf{r}_2$ **with** $\mathbf{r}_1 \to_\beta \mathbf{r}'_1$**:** We can assume that $\mathcal{D}_i$ ends with $\to_E$. Then $[(\mathbf{r}_1)\ \mathbf{r}_2]_{\mathcal{D}_i} = \mathsf{T} \circ \mathsf{T}'[wv \mapsto \pi_{\overline{w}}([\mathbf{r}_1]_{\mathcal{D}_{i1}})\pi_{\overline{v}}([\mathbf{r}_2]_{\mathcal{D}_{i2}})]$, where $\mathcal{D}_{i1}$ ends with the sequent assigning a type with the structure of T to the term $\mathbf{r}_1$ and $\mathcal{D}_{i2}$ ends with the sequent assigning a type with the structure of T' to $\mathbf{r}_2$. By the induction hypothesis, there is no $\mathcal{D}_{i+1,1}$ such that $[\mathbf{r}_1]_{\mathcal{D}_{i1}} = [\mathbf{r}'_1]_{\mathcal{D}_{i+1,1}}$, so there is no $\mathcal{D}_{i+1}$ such that $[(\mathbf{r}_1)\ \mathbf{r}_2]_{\mathcal{D}_i} = [(\mathbf{r}'_1)\ \mathbf{r}_2]_{\mathcal{D}_{i+1}}$.

   $(\mathbf{r}_1)\ \mathbf{r}_2 \to (\mathbf{r}_1)\ \mathbf{r}'_2$ **with** $\mathbf{r}_2 \to_\beta \mathbf{r}'_2$**:** Analogous to previous case.

2. Rule by rule analysis.

   Rule $(\mathbf{u}+\mathbf{t})\ \mathbf{r} \to (\mathbf{u})\ \mathbf{r} + (\mathbf{t})\ \mathbf{r}$**:** $|(\mathbf{u}+\mathbf{t})\ \mathbf{r}| = (4+3|\mathbf{u}|+3|\mathbf{t}|)(3|\mathbf{r}|+2)+2+6(2|\mathbf{r}|+1) > (4+3|\mathbf{u}|+3|\mathbf{t}|)(3|\mathbf{r}|+2)+2 = |(\mathbf{u})\ \mathbf{r} + (\mathbf{t})\ \mathbf{r}|$.

Rule $(\mathbf{r})\ (\mathbf{u}+\mathbf{t}) \to (\mathbf{r})\ \mathbf{u} + (\mathbf{r})\ \mathbf{t}$**:** $|(\mathbf{r})\ (\mathbf{u}+\mathbf{t})| = (3|\mathbf{r}|+2)(4+3|\mathbf{u}|+3|\mathbf{t}|)+2+6(2|\mathbf{r}|+1) > (3|\mathbf{r}|+2)(4+3|\mathbf{u}|+3|\mathbf{t}|)+2 = |(\mathbf{r})\ \mathbf{u} + (\mathbf{r})\ \mathbf{t}|$.

Rule $(\mathbf{0})\ \mathbf{t} \to \mathbf{0}$**:** $|(\mathbf{0})\ \mathbf{t}| = 6|\mathbf{t}|+4 > 0 = |\mathbf{0}|$.

Rule $(\mathbf{t})\ \mathbf{0} \to \mathbf{0}$**:** $|(\mathbf{0})\ \mathbf{t}| = 6|\mathbf{t}|+4 > 0 = |\mathbf{0}|$.

Rule $\mathbf{t}+\mathbf{0} \to \mathbf{t}$**:** $|\mathbf{t}+\mathbf{0}| = 2+|\mathbf{t}| > |\mathbf{t}|$.

□

Appendix D

Proofs from Chapter 5

D.1 Proof of Lemma 5.3.2

Lemma 5.3.2 ($\preceq$-subsumption)**.** For any context Γ, any term $\mathbf{t}$ and any types T, R such that $T \preceq R$ and no free type variable in T occurs in Γ. Then $\Gamma \vdash \mathbf{t} : T$ implies $\Gamma \vdash \mathbf{t} : R$.

Proof. One can assume $\exists S_1, \ldots, S_n \,/\, T \equiv S_1 \prec S_2 \prec \cdots \prec S_n \equiv R$ (if not, there must be an equivalence instead, so the lemma would hold due to the $\equiv$-rule). So $\forall i$ one has $S_i \equiv \sum_{j=1}^n \alpha_j.U_j^i$, then $\Gamma \vdash \mathbf{t} : \sum_{j=1}^n \alpha_j.U_j^i$ and using $\forall_E$ or $\forall_I$, we get $\Gamma \vdash \mathbf{t} : : \sum_{j=1}^n \alpha_j.U_j^{i+1}$. Since $\sum_{j=1}^n \alpha_j.U_j^{i+1} \equiv S_{i+1}$ we finally get $\Gamma \vdash \mathbf{t} : S_{i+1}$, and so

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash \mathbf{t} : T \quad T \equiv S_1}{\Gamma \vdash \mathbf{t} : S_1}\equiv}{\Gamma \vdash \mathbf{t} : S_n} \qquad S_n \equiv R}{\Gamma \vdash \mathbf{t} : R}\equiv$$

□

D.2 Proof of Lemma 5.3.5

Lemma 5.3.5 (Order characterisation)**.** For any type R, unit types $V_1, \ldots, V_m$ and scalars $\beta_1, \ldots, \beta_m$, if $R \sqsubseteq \sum_{j=1}^m \beta_j.V_j$, then there exist a scalar δ, a natural number k, a set $N \subseteq \{1, \ldots, m\}$ and a unit type $W \preceq V_k$ such that $R \equiv \delta.W + \sum_{j \in N} \beta_j.V_j$ and $\sum_{j=1}^m \beta_j = \delta + \sum_{j \in N} \beta_j$.

Proof. Structural induction on R.

- $R \equiv U$. Then by definition of $\sqsubseteq$, $\exists k \,/\, U \preceq V_k$ and $\sum_{j=1}^m \beta_j = 1$.
- $R \equiv \alpha.T$. Then $\sum_{j=1}^m \beta_j.V_j \equiv \alpha.\sum_{j=1}^m \gamma_j.V_j$, so by the induction hypothesis $T \equiv \delta.W + \sum_{j \in N} \gamma_j.V_j$ with $N \subseteq \{1, \ldots, m\}$ and $W \preceq V_k$ for some k. So $R \equiv \alpha.T \equiv \alpha.\delta.W + \alpha.\sum_{j \in N} \gamma_j.V_j \equiv (\alpha \times \delta).W + \sum_{j \in N} \beta_j.V_j$. Notice that $(\alpha \times \delta) + \sum_{j \in N} \beta_j \equiv \alpha.(\delta + \sum_{j \in N} \gamma_j) = \alpha.\sum_{j=1}^m \gamma_j \equiv \sum_{j=1}^m \beta_j$.
- $R \equiv T + S$. Then $\exists m' \leq m$ such that $\sum_{j=1}^m \beta_j.V_j \equiv \sum_{j=1}^{m'} \beta_j.V_j + S$ with $T \sqsubseteq \sum_{j=1}^{m'} \beta_j.V_j$, so by the induction hypothesis $T \equiv \delta.W + \sum_{j \in N} \beta_j.V_j$ with $N \subseteq \{1, \ldots, m'\}$ and $W \preceq V_k$ for some k. So $R \equiv T + S \equiv \delta.W + \sum_{j \in N} \beta_j.V_j + S \equiv \delta.W + \sum_{j \in N'} \beta_j.V_j$ with $N' = N \cup \{m' + 1, \ldots, m\}$. Notice that $\sum_{j=1}^m \beta_j = \sum_{j=1}^{m'} \beta_j + \sum_{j=m'+1}^m \beta_j = \delta + \sum_{j \in N} \beta_j + \sum_{j=m'+1}^m \beta_j = \delta + \sum_{j \in N'} \beta_j$.

□

D.3 Proof of Lemma 5.3.6

Lemma 5.3.6 (Scalars)**.** For any context Γ, term $\mathbf{t}$, type T and scalar α, if $\Gamma \vdash \alpha.\mathbf{t} : T$, then there exists a type R such that $T \equiv \alpha.R$ and if $\alpha \neq 0$, $\Gamma \vdash \mathbf{t} : R$. Moreover, if $\Gamma \vdash \alpha.\mathbf{t} : \alpha.T$, then $\Gamma \vdash \mathbf{t} : T$.

Proof. Induction on the typing derivation to prove the first part of the Lemma.

- Let $\Gamma \vdash \alpha.\mathbf{t} : \alpha.T$ as a consequence of $\Gamma \vdash \mathbf{t} : T$ and rule s_I. Trivial.
- Let $\Gamma \vdash \alpha.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i$ with $X \notin FV(\Gamma)$ as a consequence of $\Gamma \vdash \alpha.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.U_i$ and rule $\forall_I$. By the induction hypothesis, $\exists R$ such that $\sum_{i=1}^{n} \alpha_i.U_i \equiv \alpha.R$ and if $\alpha \neq 0$, $\Gamma \vdash \mathbf{t} : R$. Then by Lemma 5.2.2, $R \equiv \sum_{j=1}^{m} \beta_j.V_j$. So, $\sum_{i=1}^{n} \alpha_i.U_i \equiv \sum_{j=1}^{m} \alpha \times \beta_j.V_j$, then by Lemma 5.2.3[1], we have $\sum_{i=1}^{n} \alpha_i.\forall X.U_i \equiv \sum_{j=1}^{m} \alpha \times \beta_j.\forall X.V_j \equiv \alpha.\sum_{j=1}^{m} \beta_j.\forall X.V_j$. Also, if $\alpha \neq 0$, since $R \equiv \sum_{j=1}^{m} \beta_j.V_j$, by rule $\equiv$, we get $\Gamma \vdash \mathbf{t} : \sum_{j=1}^{m} \beta_j.V_j$, so using rule $\forall_I$, we conclude with $\Gamma \vdash \mathbf{t} : \sum_{j=1}^{m} \beta_j.\forall X.V_j$.
- Let $\Gamma \vdash \alpha.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.U_i[V/X]$ as a consequence of $\Gamma \vdash \alpha.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i$ and rule $\forall_E$. Then by the induction hypothesis, $\exists R$ such that $\sum_{i=1}^{n} \alpha_i.\forall X.U_i \equiv \alpha.R$ and if $\alpha \neq 0$, $\Gamma \vdash \mathbf{t} : R$. Then by Lemma 5.2.2, $R \equiv \sum_{j=1}^{m} \beta_j.V_j$. So, $\sum_{i=1}^{n} \alpha_i.\forall X.U_i \equiv \sum_{j=1}^{m} \alpha \times \beta_j.V_j$, then by Lemma 5.2.3, $\forall j, V_j \equiv \forall X.W_j$ and by Lemma 5.2.3[1], $\sum_{i=1}^{n} \alpha_i.U_i \equiv \sum_{j=1}^{m} \alpha \times \beta_j.W_j$. So by Lemma 5.2.3, $\sum_{i=1}^{n} \alpha_i.U_i[V/X] \equiv (\sum_{i=1}^{n} \alpha_i.U_i)[V/X] \equiv \left(\sum_{j=1}^{m} \alpha \times \beta_j.W_j\right)[V/X] \equiv \alpha.\sum_{j=1}^{m} \beta_j.W_j[V/X]$. Also, if $\alpha \neq 0$, and since $R \equiv \sum_{j=1}^{m} \beta_j.\forall X.W_j$, by rule $\equiv$, we get $\Gamma \vdash \mathbf{t} : \sum_{j=1}^{m} \beta_j.\forall X.W_j$, so using rule $\forall_E$, we conclude with $\Gamma \vdash \mathbf{t} : \sum_{j=1}^{m} \beta_j.W_j[V/X]$.
- Let $\Gamma \vdash \alpha.\mathbf{t} : S$ as a consequence of $\Gamma \vdash \alpha.\mathbf{t} : T$ with $T \equiv S$ and rule $\equiv$. Then by the induction hypothesis, $T \equiv \alpha.R$ and if $\alpha \neq 0$, $\Gamma \vdash \mathbf{t} : R$. By transitivity of the equivalence, $S \equiv \alpha.R$.

The second part of the Lemma, $\Gamma \vdash \alpha.\mathbf{t} : \alpha.T \Rightarrow \Gamma \vdash \mathbf{t} : T$ follows as corollary. If $\Gamma \vdash \alpha.\mathbf{t} : \alpha.T$, we have just proved that there exists R such that $\alpha.T \equiv \alpha.R$ and $\Gamma \vdash \mathbf{t} : R$. It is easy to check that $\alpha.T \equiv \alpha.R \Rightarrow T \equiv R$, so using rule $\equiv$, $\Gamma \vdash \mathbf{t} : T$. □

D.4 Proof of Lemma 5.3.7

Lemma 5.3.7 (Zeros)**.** For any context Γ, term $\mathbf{t}$, unit types $U_1, \ldots, U_n$ and scalars $\alpha_1, \ldots \alpha_n$, if $\Gamma \vdash 0.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.U_i$, then $\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \delta_i.U_i$ and $\forall i$, $\alpha_i = 0$.

Proof. Induction on the typing derivation.

- Let $\Gamma \vdash 0.\mathbf{u} : 0.T$ as a consequence of $\Gamma \vdash \mathbf{u} : T$ and rule s_I. Notice that $T \equiv \sum_{i=1}^{n} \delta_i.U_i$, so $0.T \equiv \sum_{i=1}^{n} 0.U_i$ which proves the case.
- Let $\Gamma \vdash 0.\mathbf{u} : \sum_{i=1}^{n} \alpha_i.U_i[V/X]$ as a consequence of $\Gamma \vdash 0.\mathbf{u} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i$ and rule $\forall_E$. Then by the induction hypothesis $\Gamma \vdash \mathbf{u} : \sum_{i=1}^{n} \delta_i.\forall X.U_i$ and $\forall i, \alpha_i = 0$, so by rule $\forall_E$ we conclude $\Gamma \vdash 0.\mathbf{u} : \sum_{i=1}^{n} \delta_i.U_i[V/X]$.
- Let $\Gamma \vdash 0.\mathbf{u} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i$ as a consequence of $\Gamma \vdash 0.\mathbf{u} : \sum_{i=1}^{n} \alpha_i.U_i$ and rule $\forall_I$. Then by the induction hypothesis $\Gamma \vdash \mathbf{u} : \sum_{i=1}^{n} \delta_i.U_i$ and $\forall i, \alpha_i = 0$, so by rule $\forall_I$ we conclude $\Gamma \vdash 0.\mathbf{u} : \sum_{i=1}^{n} \delta_i.\forall X.U_i$.

□

D.5 Proof of Lemma 5.3.8

Lemma 5.3.8 (Basis terms)**.** For any context Γ, type T and basis term $\mathbf{b}$, if $\Gamma \vdash \mathbf{b} : T$ then there exists a unit type U such that $T \equiv U$.

Proof. Induction on the typing derivation.

- Let $\Gamma, x : U \vdash x : U$ as a consequence of rule ax. Trivial.
- Let $\Gamma \vdash \mathbf{b} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i$ as a consequence of $\Gamma \vdash \mathbf{b} : \sum_{i=1}^{n} \alpha_i.U_i$ and rule $\forall_I$. Then by the induction hypothesis $\exists V$ such that $\sum_{i=1}^{n} \alpha_i.U_i \equiv V \equiv \sum_{j=1}^{1} 1.V$, and then by Lemma 5.2.3[1], $\sum_{i=1}^{n} \alpha_i.\forall X.U_i \equiv \sum_{j=1}^{1} 1.\forall X.V \equiv \forall X.V$.
- Let $\Gamma \vdash \mathbf{b} : \sum_{i=1}^{n} \alpha_i.U_i[V/X]$ as a consequence of $\Gamma \vdash \mathbf{b} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i$ and rule $\forall_I$. Then by the induction hypothesis $\exists W$ such that $\sum_{i=1}^{n} \alpha_i.\forall X.U_i \equiv W \equiv \sum_{j=1}^{1} 1.W$, then by Lemma 5.2.3, $\sum_{j=1}^{1} 1.W \equiv \sum_{j=1}^{1} 1.\forall X.W'$. Then by Lemma 5.2.3[1], $\sum_{i=1}^{n} \alpha_i.U_i \equiv \sum_{j=1}^{1} 1.W'$. Thus by Lemma 5.2.3, $\sum_{i=1}^{n} \alpha_i.U_i[V/X]$ is equivalent to $\sum_{j=1}^{1} 1.W'[V/X] \equiv W'[V/X]$.
- Let $\Gamma \vdash \mathbf{b} : R$ as a consequence of $\Gamma \vdash \mathbf{b} : T$ with $T \equiv R$ and rule $\equiv$. Then by the induction hypothesis $\exists U$ such that $T \equiv U$. So $R \equiv T \equiv U$.

□

D.6 Proof of Lemma 5.3.10

Lemma 5.3.10 (Substitution lemma)**.** For any term $\mathbf{t}$, basis term $\mathbf{b}$, term variable x, context Γ, types T, U, $\vec{W}$ and type variables $\vec{X}$,

1. if $\Gamma \vdash \mathbf{t} : T$, then $\Gamma[U/X] \vdash \mathbf{t} : T[U/X]$;
2. if $\Gamma, x : U \vdash \mathbf{t} : T$, $\Gamma \vdash \mathbf{b} : U[\vec{W}/\vec{X}]$ and $\vec{X} \notin FV(\Gamma)$, then $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T[\vec{W}/\vec{X}]$.

Proof.

1. Induction on the typing derivation.
   - Let $\Gamma, x : V \vdash x : V$ as a consequence of rule ax. By rule ax, one has $\Gamma[U/X], x : V[U/X] \vdash x : V[U/X]$.
   - Let $\Gamma \vdash \mathbf{0} : 0.T$ as a consequence of $\Gamma \vdash \mathbf{t} : T$ and rule 0_I. Then by the induction hypothesis $\Gamma[U/X] \vdash \mathbf{t} : T[U/X]$, so by rule 0_I, $\Gamma[U/X] \vdash \mathbf{0} : 0.T[U/X]$. Notice that $0.T[U/X] = (0.T)[U/X]$.
   - Let $\Gamma \vdash (\mathbf{t})\ \mathbf{r} : \sum_{i=1}^{n'} \sum_{j=1}^{m} \alpha_i \times \beta_j.T_i[\vec{W}_j/\vec{X}]$ as a consequence of $\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n'} \alpha_i.\forall \vec{Y}.(V \to T_i)$, $\Gamma \vdash \mathbf{r} : \sum_{j=1}^{m} \beta_j.W_j$ where $\forall W_j, \exists \vec{U'_j}$ such that $V[\vec{U'}_j/\vec{Y}] = W_j$ and rule $\to_E$. Then by the induction hypothesis $\Gamma[U/X] \vdash \mathbf{t} : (\sum_{i=1}^{n'} \alpha_i.\forall \vec{Y}.(V \to T_i))[U/X] = \sum_{i=1}^{n'} \alpha_i.\forall \vec{Y}.(V[U/X] \to T_i[U/X])$ and $\Gamma[U/X] \vdash \mathbf{r} : (\sum_{j=1}^{m} \beta_j.W_j)[U/X] = \sum_{j=1}^{m} \beta_j.W_j[U/X]$ Notice that, up to variable renaming, $W_j[U/X] = (V[\vec{U'}_j/\vec{Y}])[U/X] = (V[U/X])[\vec{U'}_j/\vec{Y}]$ Then using rule $\to_E$, $\Gamma[U/X] \vdash (\mathbf{t})\ \mathbf{r} : \sum_{i=1}^{n'} \sum_{j=1}^{m} \alpha_i \times \beta_j.(T_i[U/X])[\vec{U'}_j/\vec{Y}]$. Notice that, up to variable renaming, we can conclude the case by realising that $\sum_{i=1}^{n'} \sum_{j=1}^{m} \alpha_i \times \beta_j.(T_i[U/X])[\vec{U'}_j/\vec{Y}]$ is equal to $(\sum_{i=1}^{n'} \sum_{j=1}^{m} \alpha_i \times \beta_j.T_i[\vec{U'}_j/\vec{Y}])[U/X]$.

- Let $\Gamma \vdash \lambda x.\mathbf{t} : V \to T$ as a consequence of $\Gamma, x:V \vdash \mathbf{t}:T$ and rule $\to_I$. Then by the induction hypothesis $\Gamma[U/X], x:V[U/X] \vdash \mathbf{t}:T[U/X]$. So using rule $\to_I$, $\Gamma[U/X] \vdash \lambda x.\mathbf{t}: V[U/X] \to T[U/X]$. Notice that $V[U/X] \to T[U/X] = (V \to T)[U/X]$.
- Let $\Gamma \vdash \mathbf{t}: \sum_{i=1}^{n} \alpha_i.V_i[W/Y]$ as a consequence of $\Gamma \vdash \mathbf{t}: \sum_{i=1}^{n} \alpha_i.\forall Y.V_i$ and rule $\forall_E$. Then by the induction hypothesis one has that $\Gamma[U/X] \vdash \mathbf{t}: (\sum_{i=1}^{n} \alpha_i.\forall Y.V_i)[U/X]$. Notice that $(\sum_{i=1}^{n} \alpha_i.\forall Y.V_i)[U/X]$ is equal to $\sum_{i=1}^{n} \alpha_i.\forall Y.V_i[U/X]$. So by rule $\forall_E$, we can derive the following sequent $\Gamma[U/X] \vdash \mathbf{t}: (\sum_{i=1}^{n} \alpha_i.V_i[U/X])[W/Y]$. Notice that, up to variable renaming $(\sum_{i=1}^{n} \alpha_i.V_i[U/X])[W/Y] = (\sum_{i=1}^{n} \alpha_i.V_i[W/Y])[U/X]$.
- Let $\Gamma \vdash \mathbf{t}: \sum_{i=1}^{n} \alpha_i.\forall Y.V_i$ as a consequence of $\Gamma \vdash \mathbf{t}: \sum_{i=1}^{n} \alpha_i.V_i$ and rule $\forall_I$. Then by the induction hypothesis $\Gamma[U/X] \vdash \mathbf{t}: \sum_{i=1}^{n} \alpha_i.V_i[U/X]$. So by rule $\forall_I$, we can derive the sequent $\Gamma[U/X] \vdash \mathbf{t}: \sum_{i=1}^{n} \alpha_i.\forall Y.V_i[U/X]$ Notice that, $\sum_{i=1}^{n} \alpha_i.\forall Y.V_i[U/X] = (\sum_{i=1}^{n} \alpha_i.\forall Y.V_i)[U/X]$.
- Let $\Gamma \vdash \mathbf{t} + \mathbf{r}: T + R$ as a consequence of $\Gamma \vdash \mathbf{t}:T$ and $\Gamma \vdash \mathbf{r}:R$ with rule $+_I$. Then by the induction hypothesis $\Gamma[U/X] \vdash \mathbf{t}:T[U/X]$ and $\Gamma[U/X] \vdash \mathbf{r}:R[U/X]$. So by rule $+_I$, $\Gamma[U/X] \vdash \mathbf{t}+\mathbf{r}: T[U/X] + R[U/X]$. Notice that $T[U/X] + R[U/X] = (T+R)[U/X]$.
- Let $\Gamma \vdash \alpha.\mathbf{t}: \alpha.T$ as a consequence of $\Gamma \vdash \mathbf{t}:T$ and rule s_I. The by the induction hypothesis $\Gamma[U/X] \vdash \mathbf{t}:T[U/X]$. So by rule s_I, $\Gamma[U/X] \vdash \alpha.\mathbf{t}: \alpha.T[U/X]$. Notice that $\alpha.T[U/X] = (\alpha.T)[U/X]$.
- Let $\Gamma \vdash \mathbf{t}:R$ as a consequence of $\Gamma \vdash \mathbf{t}:T$ with $T \equiv R$, and rule $\equiv$. Then by the induction hypothesis $\Gamma[U/X] \vdash \mathbf{t}:T[U/X]$. By Lemma 5.2.3 $T \equiv R \Rightarrow T[U/X] \equiv R[U/X]$. So by rule $\equiv$, $\Gamma[U/X] \vdash \mathbf{t}:R[U/X]$

2. Induction on the typing derivation of $\Gamma, x:U \vdash \mathbf{t}:T$.

- Let $\Gamma, x:U \vdash x:U$ as a consequence of rule ax. Trivial since $x[\mathbf{b}/x] = \mathbf{b}$.
- Let $\Gamma, x:U \vdash \mathbf{0}:0.T$ as a consequence of $\Gamma, x:U \vdash \mathbf{t}:T$ and rule 0_I. Then by the induction hypothesis $\Gamma \vdash \mathbf{t}[\mathbf{b}/x]: T[\vec{W}/\vec{X}]$. Then by rule 0_I, $\Gamma \vdash \mathbf{0}: 0.T[\vec{W}/\vec{X}]$. Notice that $\mathbf{0} = \mathbf{0}[\mathbf{b}/x]$ and $0.T[\vec{W}/\vec{X}] = (0.T)[\vec{W}/\vec{X}]$.
- Let $\Gamma, x:U \vdash (\mathbf{t})\ \mathbf{r}: \sum_{i=1}^{n'} \sum_{j=1}^{m} \alpha_i \times \beta_j.T_i[\vec{W}_j/\vec{Y}]$ as a consequence of $\Gamma, x:U \vdash \mathbf{r}: \sum_{j=1}^{m} \beta_j.V'_j$ and $\Gamma, x:U \vdash \mathbf{t}: \sum_{i=1}^{n'} \alpha_i.\forall \vec{Y}.(V \to T_i)$ where $\forall V'_j,\ \exists \vec{W}_j \ /\ V[\vec{W}_j/\vec{Y}] = V'_j$ by rule $\to_E$. Then by the induction hypothesis $\Gamma \vdash \mathbf{t}[\mathbf{b}/x]: (\sum_{i=1}^{n'} \alpha_i.\forall \vec{Y}.(V \to T_i))[\vec{W}/\vec{X}]$ which is equal to $\sum_{i=1}^{n'} \alpha_i.\forall \vec{Y}.(V[\vec{W}/\vec{X}] \to T_i[\vec{W}/\vec{X}])$. Also we can derive $\Gamma \vdash \mathbf{r}[\mathbf{b}/x]: (\sum_{j=1}^{m} \beta_j.V'_j)[\vec{W}/\vec{X}] = \sum_{j=1}^{m} \beta_j.V'_j[\vec{W}/\vec{X}]$. Notice also that, up to variable renaming, $(V[\vec{W}/\vec{X}])[\vec{W}'_j/\vec{Y}]$ is equal to $(V[\vec{W}'_j/\vec{Y}])[\vec{W}/\vec{X}] = V'_j[\vec{W}/\vec{X}]$. Then by rule $\to_E$, $\Gamma \vdash (\mathbf{t}[\mathbf{b}/x])\ \mathbf{r}[\mathbf{b}/x]: \sum_{i=1}^{n'} \sum_{j=1}^{m} \alpha_i \times \beta_j.(T_i[\vec{W}/\vec{X}])[\vec{W}'_j/\vec{Y}]$. Notice that $(\mathbf{t}[\mathbf{b}/x])\ \mathbf{r}[\mathbf{b}/x]$ is equal to $((\mathbf{t})\ \mathbf{r})[\mathbf{b}/x]$ and, up to variable renaming, $\sum_{i=1}^{n'} \sum_{j=1}^{m} \alpha_i \times \beta_j.(T_i[\vec{W}/\vec{X}])[\vec{W}'_j/\vec{Y}] = (\sum_{i=1}^{n'} \sum_{j=1}^{m} \alpha_i \times \beta_j.T_i[\vec{W}'_j/\vec{Y}])[\vec{W}/\vec{X}]$.
- Let $\Gamma, x:U \vdash \lambda y.\mathbf{t}: V \to T$ as a consequence of $\Gamma, x:U, y:V \vdash \mathbf{t}:T$ and rule $\to_I$. Then by item 1, $\Gamma[\vec{W}/\vec{X}], x:U[\vec{W}/\vec{X}], y:V[\vec{W}/\vec{X}] \vdash \mathbf{t}:T[\vec{W}/\vec{X}]$. Since $\vec{X} \notin FV(\Gamma)$, we do not need to replace anything on Γ and then this sequent is the same to $\Gamma, x:U[\vec{W}/\vec{X}], y:V[\vec{W}/\vec{X}] \vdash \mathbf{t}:T[\vec{W}/\vec{X}]$. Notice that $y \notin \Gamma$, so $\Gamma, y:V[\vec{W}/\vec{X}] \vdash \mathbf{b}:U[\vec{W}/\vec{X}]$. Let $\vec{Z}$ be set of fresh variables. Then $U[\vec{W}/\vec{X}] = (U[\vec{W}/\vec{X}])[\vec{W}/\vec{Z}]$. So, by the induction hypothesis $\Gamma, y:V[\vec{W}/\vec{X}] \vdash \mathbf{t}[\mathbf{b}/x]: (T[\vec{W}/\vec{X}])[\vec{W}/\vec{Z}] = T[\vec{W}/\vec{X}]$. So by rule $\to_I$, $\Gamma \vdash \lambda y.\mathbf{t}[\mathbf{b}/x]: V[\vec{W}/\vec{X}] \to T[\vec{W}/\vec{X}]$. Notice that $\lambda y.\mathbf{t}[\mathbf{b}/x]$ is equal to $(\lambda y.\mathbf{t})[\mathbf{b}/x]$ and $V[\vec{W}/\vec{X}] \to T[\vec{W}/\vec{X}]$ is just $(V \to T)[\vec{W}/\vec{X}]$.

- Let $\Gamma, x:U \vdash \mathbf{t}: \sum_{i=1}^n \alpha_i.V_i[W'/Y]$ as a consequence of $\Gamma, x:U \vdash \mathbf{t}: \sum_{i=1}^n \alpha_i.\forall Y.V_i$ and rule $\forall_E$. Then by the induction hypothesis $\Gamma \vdash \mathbf{t}[\mathbf{b}/x]: (\sum_{i=1}^n \alpha_i.\forall Y.V_i)[\vec{W}/\vec{X}] = \sum_{i=1}^n \alpha_i.\forall Y.V_i[\vec{W}/\vec{X}]$. So by rule $\forall_E$, $\Gamma \vdash \mathbf{t}[\mathbf{b}/x]: (\sum_{i=1}^n \alpha_i.V_i[\vec{W}/\vec{X}])[W'/Y]$. Notice that, up to renaming of variables, the type $(\sum_{i=1}^n \alpha_i.V_i[\vec{W}/\vec{X}])[W'/Y]$ is equal to $(\sum_{i=1}^n \alpha_i.V_i[W'/Y])[\vec{W}/\vec{X}]$.
- Let $\Gamma, x:U \vdash \mathbf{t}: \sum_{i=1}^n \alpha_i.\forall Y.V_i$ as a consequence of $\Gamma, x:U \vdash \mathbf{t}: \sum_{i=1}^n \alpha_i.V_i$ and rule $\forall_I$. Then by the induction hypothesis we have $\Gamma \vdash \mathbf{t}[\mathbf{b}/x]: \sum_{i=1}^n \alpha_i.V_i[\vec{W}/\vec{X}]$. So by rule $\forall_I$, we can conclude $\Gamma \vdash \mathbf{t}[\mathbf{b}/x]: \sum_{i=1}^n \alpha_i.\forall Y.V_i[\vec{W}/\vec{X}]$. Notice that $\sum_{i=1}^n \alpha_i.\forall Y.V_i[\vec{W}/\vec{X}] = (\sum_{i=1}^n \alpha_i.\forall Y.V_i)[\vec{W}/\vec{X}]$.
- Let $\Gamma, x:U \vdash \mathbf{t} + \mathbf{r}: T + R$ as a consequence of $\Gamma, x:U \vdash \mathbf{t}:T$ and $\Gamma, x:U \vdash \mathbf{r}:R$ by rule $+_I$. Then by the induction hypothesis $\Gamma \vdash \mathbf{t}[\mathbf{b}/x]:T[\vec{W}/\vec{X}]$ and $\Gamma \vdash \mathbf{r}[\mathbf{b}/x]:R[\vec{W}/\vec{X}]$. So by rule $+_I$, $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] + \mathbf{r}[\mathbf{b}/x]:T[\vec{W}/\vec{X}] + R[\vec{W}/\vec{X}]$. Notice that $\mathbf{t}[\mathbf{b}/x] + \mathbf{r}[\mathbf{b}/x] = (\mathbf{t} + \mathbf{r})[\mathbf{b}/x]$ and $T[\vec{W}/\vec{X}] + R[\vec{W}/\vec{X}] = (T + R)[\vec{W}/\vec{X}]$.
- Let $\Gamma, x:U \vdash \alpha.\mathbf{t}:\alpha.T$ as a consequence of $\Gamma, x:U \vdash \mathbf{t}:T$ and rule s_I. Then by the induction hypothesis $\Gamma \vdash \mathbf{t}[\mathbf{b}/x]:T[\vec{W}/\vec{X}]$. So by rule s_I, $\Gamma \vdash \alpha.\mathbf{t}[\mathbf{b}/x]:\alpha.T[\vec{W}/\vec{X}]$. Notice that $\alpha.\mathbf{t}[\mathbf{b}/x] = (\alpha.\mathbf{t})[\mathbf{b}/x]$ and $\alpha.T[\vec{W}/\vec{X}] = (\alpha.T)[\vec{W}/\vec{X}]$.
- Let $\Gamma, x:U \vdash \mathbf{t}:R$ as a consequence of $\Gamma, x:U \vdash \mathbf{t}:T$ where $T \equiv R$, and rule $\equiv$. Then by the induction hypothesis $\Gamma \vdash \mathbf{t}[\mathbf{b}/x]:T[\vec{W}/\vec{X}]$. By 5.2.3, $T[\vec{W}/\vec{X}] \equiv R[\vec{W}/\vec{X}]$. So by rule $\equiv$, $\Gamma \vdash \mathbf{t}[\mathbf{b}/x]:R[\vec{W}/\vec{X}]$.

□

D.7 Proof of Lemma 5.3.11

Lemma 5.3.11 (Generation lemma (app))**.** For any terms $\mathbf{t}$, $\mathbf{r}$, any context Γ and any type T, if $\Gamma \vdash (\mathbf{t})\ \mathbf{r}:T$, then there exist natural numbers n, m, unit types $U, V_1, \ldots, V_m$, types $T_1, \ldots, T_n$ and scalars $\alpha_1, \ldots, \alpha_n$ and $\beta_1, \ldots, \beta_m$, such that $\Gamma \vdash \mathbf{t}: \sum_{i=1}^n \alpha_i.\forall \vec{X}.(U \to T_i)$, $\Gamma \vdash \mathbf{r}: \sum_{j=1}^m \beta_j.V_j$, where for all V_j, there exists $\vec{W}_j$ such that $U[\vec{W}_j/\vec{X}] = V_j$ and $\sum_{i=1}^n \sum_{j=1}^m \alpha_i \times \beta_j.T_i[\vec{W}_j/\vec{X}] \preceq T$.

Proof. Induction on the typing derivation.

- Let $\Gamma \vdash (\mathbf{t})\ \mathbf{r}: \sum_{i=1}^n \sum_{j=1}^m \alpha_i \times \beta_j.T_i[\vec{W}_j/\vec{X}]$ as a consequence of $\Gamma \vdash \mathbf{t}: \sum_{i=1}^n \alpha_i.\forall \vec{X}.(U \to T_i)$ and $\Gamma \vdash \mathbf{r}: \sum_{j=1}^m \beta_j.V_j$, where $\forall V_j,\ \exists \vec{W}_j$ such that $U[\vec{W}_j/\vec{X}] = V_j$ by rule $\to_E$. Trivial.
- Let $\Gamma \vdash (\mathbf{t})\ \mathbf{r}:S$ as a consequence of $\Gamma \vdash (\mathbf{t})\ \mathbf{r}:R$ where $R \equiv S$ and rule $\equiv$. The by the induction hypothesis $\Gamma \vdash \mathbf{t}: \sum_{i=1}^n \alpha_i.\forall \vec{X}.(U \to T_i)$, $\Gamma \vdash \mathbf{r}: \sum_{j=1}^m \beta_j.V_j$, $\forall V_j,\ \exists \vec{W}_j$ such that $U[\vec{W}_j/\vec{X}] = V_j$ and $\sum_{i=1}^n \sum_{j=1}^m \alpha_i \times \beta_j.T_i[\vec{W}_j/\vec{X}] \preceq R \equiv S$.
- Let $\Gamma \vdash (\mathbf{t})\ \mathbf{r}: \sum_{i=1}^n \alpha_i.U_i[V/X]$ as a consequence of $\Gamma \vdash (\mathbf{t})\ \mathbf{r}: \sum_{i=1}^n \alpha_i.\forall X.U_i$ and rule $\forall_E$. Then by the induction hypothesis $\Gamma \vdash \mathbf{t}: \sum_{i=1}^n \alpha_i.\forall \vec{X}.(U \to T_i)$, $\Gamma \vdash \mathbf{r}: \sum_{j=1}^m \beta_j.V_j$, $\forall V_j,\ \exists \vec{W}_j\ /\ U[\vec{W}_j/\vec{X}] = V_j$ and $\sum_{i=1}^n \sum_{j=1}^m \alpha_i \times \beta_j.T_i[\vec{W}_j/\vec{X}] \preceq \sum_{i=1}^n \alpha_i.\forall X.U_i \prec \sum_{i=1}^n \alpha_i.U_i[V/X]$.
- Let $\Gamma \vdash (\mathbf{t})\ \mathbf{r}: \sum_{i=1}^n \alpha_i.\forall X.U_i$ as a consequence of $\Gamma \vdash (\mathbf{t})\ \mathbf{r}: \sum_{i=1}^n \alpha_i.U_i$ and rule $\forall_I$. Then by the induction hypothesis $\Gamma \vdash \mathbf{t}: \sum_{i=1}^n \alpha_i.\forall \vec{X}.(U \to T_i)$, $\Gamma \vdash \mathbf{r}: \sum_{j=1}^m \beta_j.V_j$, $\forall V_j,\ \exists \vec{W}_j\ /\ U[\vec{W}_j/\vec{X}] = V_j$ and $\sum_{i=1}^n \sum_{j=1}^m \alpha_i \times \beta_j.T_i[\vec{W}_j/\vec{X}] \preceq \sum_{i=1}^n \alpha_i.U_i \prec \sum_{i=1}^n \alpha_i.\forall X.U_i$.

□

D.8 Proof of Lemma 5.3.12

Lemma 5.3.12 (Generation lemma (abs))**.** For any term variable x, term $\mathbf{t}$, context Γ and type T, if $\Gamma \vdash \lambda x.\mathbf{t} : R$, there exist types U and T such that $U \to T \preceq R$ and $\Gamma, x:U \vdash \mathbf{t} : T$.

Proof. Induction on the typing derivation.

- Let $\Gamma \vdash \lambda x.\mathbf{t} : U \to T$ as a consequence of $\Gamma, x:U \vdash \mathbf{t} : T$ and rule $\to_I$. This is the trivial case.
- Let $\Gamma \vdash \lambda x.\mathbf{t} : R$ as a consequence of $\Gamma \vdash \lambda x.\mathbf{t} : S \qquad S \equiv R$ and rule $\equiv$. Then by the induction hypothesis $U \to T \preceq S \equiv R$ and $\Gamma, x:U \vdash \mathbf{t} : T$.
- Let $\Gamma \vdash \lambda x.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.U_i[V/X]$ as a consequence of $\Gamma \vdash \lambda x.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i$ and rule $\forall_E$. Then by the induction hypothesis $W \to T \preceq \sum_{i=1}^{n} \alpha_i.\forall X.U_i \prec \sum_{i=1}^{n} \alpha_i.U_i[V/X]$ and $\Gamma, x:W \vdash \mathbf{t} : T$.
- Let $\Gamma \vdash \lambda x.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i$ as a consequence of $\Gamma \vdash \lambda x.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.U_i$ and rule $\forall_I$. Then by the induction hypothesis $V \to T \preceq \sum_{i=1}^{n} \alpha_i.U_i \prec \sum_{i=1}^{n} \alpha_i.\forall X.U_i$ and $\Gamma, x:V \vdash \mathbf{t} : T$.

□

D.9 Proof of Lemma 5.3.13

Lemma 5.3.13 (Arrows comparison)**.** For any types T, R and any unit types U, V, if $V \to R \preceq U \to T$, then there exist $\vec{W}, \vec{X}$ such that $U \to T \equiv (V \to R)[\vec{W}/\vec{X}]$.

Proof. The proof is analogous to the proof of Lemma 3.2.6 (*cf.* Appendix B.2) and 4.2.3 (*cf.* Appendix C.1).

A map $(\cdot)^\circ$ from types to types is defined by

$$X^\circ = X \qquad (\alpha.T)^\circ = \alpha.T^\circ \qquad (U \to T)^\circ = U \to T \qquad (T + R)^\circ = T^\circ + R^\circ \qquad (\forall X.U)^\circ = U^\circ$$

We need two intermediate results.

1. For any type T and unit type U, there exists a unit type V such that $(T[U/X])^\circ \equiv T^\circ[V/X]$
2. For any types T, R, if $T \preceq R$ then $\exists \vec{U}, \vec{X} \;/\; R^\circ \equiv T^\circ[\vec{U}/\vec{X}]$

Proofs

1. Structural induction on T.
   - If $T = X$, then $(X[U/X])^\circ = U^\circ = X[U^\circ/X] = X^\circ[U^\circ/X]$.
   - If $T = Y$, then $(Y[U/X])^\circ = Y = Y^\circ[U/X]$.
   - If $T = V \to R$, then $((V \to R)[U/X])^\circ = (V[U/X] \to R[U/X])^\circ = V[U/X] \to R[U/X] = (V \to R)[U/X] = (V \to R)^\circ[U/X]$.
   - If $T = \forall Y.R$, then $((\forall Y.R)[U/X])^\circ = (\forall Y.R[U/X])^\circ = (R[U/X])^\circ$, and by the induction hypothesis $(R[U/X])^\circ \equiv R^\circ[V/X] = (\forall Y.R)^\circ[V/X]$.
   - If $T = \alpha.R$, then $(\alpha.R[U/X])^\circ = \alpha.(R[U/X])^\circ$, which, by the induction hypothesis, is equivalent to $\alpha.(R^\circ)[V/X] = (\alpha.R)^\circ[V/X]$.

- If $T = R + S$, then $((R + S)[U/X])^\circ = (R[U/X] + S[U/X])^\circ$ which is equal to $(R[U/X])^\circ + (S[U/X])^\circ$, which, by the induction hypothesis, is equivalent to $R^\circ[U/X] + S^\circ[U/X] = (R^\circ + S^\circ)[U/X] = (R + S)^\circ[U/X]$.

2. It suffices to show this just for $T \prec R$. Cases:

   - Let $T = \sum_{i=1}^n \alpha_i.U_i$ and $R = \sum_{i=1}^n \alpha_i.\forall X.U_i$. Then $T^\circ = (\sum_{i=1}^n \alpha_i.U_i)^\circ$ which is equal to $\sum_{i=1}^n \alpha_i.U_i^\circ = \sum_{i=1}^n \alpha_i.(\forall X.U_i)^\circ = (\sum_{i=1}^n \alpha_i.\forall X.U_i)^\circ$ which is just R°.
   - The other possible case is $T = \sum_{i=1}^n \alpha_i.\forall X.U_i$ and $R = \sum_{i=1}^i \alpha_i.U_i[V/X]$, in such case $R^\circ = (\sum_{i=1}^n \alpha_i.U_i[V/X])^\circ = \sum_{i=1}^n \alpha_i.(U_i[V/X])^\circ$. This latter type, by item 2, is equivalent to $\sum_{i=1}^n \alpha_i.U_i^\circ[W/X] = \sum_{i=1}^n \alpha_i.(\forall X.U_i)^\circ[W/X] = (\sum_{i=1}^n \alpha_i.\forall X.U_i)^\circ[W/X] = T^\circ[W/X]$.

Proof of the lemma: $U \to T = (U \to T)^\circ$ which by 2 is equivalent to $(V \to R)^\circ[\vec{W}/\vec{X}] = (V \to R)[\vec{W}/\vec{X}]$. □

D.10 Proof of Corollary 5.3.14

Corollary 5.3.14 (of Lemma 5.3.12)**.** For any context Γ, term variable x, term $\mathbf{t}$, type variables $\vec{X}$ and types U and T, if $\Gamma \vdash \lambda x.\mathbf{t} : \forall \vec{X}.(U \to T)$ then the typing judgement $\Gamma, x : U \vdash \mathbf{t} : T$ is valid.

Proof. This result is proved in a completely analogous way to Corollaries 3.2.14 and 4.2.9. By Lemma 5.3.12, $\exists V, R, V \to R \preceq \forall \vec{X}.(U \to T)$ and $\Gamma, x : V \vdash \mathbf{t} : R$. Note that $V \to R \preceq \forall \vec{X}.(U \to T) \preceq U \to T$, so by Lemma 5.3.13, $\exists \vec{W}, \vec{Y}$ such that $U \to T \equiv (V \to R)[\vec{W}/\vec{Y}] \equiv V[\vec{W}/\vec{Y}] \to R[\vec{W}/\vec{Y}]$ so $U \equiv V[\vec{W}/\vec{Y}]$ and $T \equiv R[\vec{W}/\vec{Y}]$. Also by Lemma 5.3.10, $\Gamma[\vec{W}/\vec{Y}], x : V[\vec{W}/\vec{Y}] \vdash \mathbf{t} : R[\vec{W}/\vec{Y}]$. By Lemma 5.3.9 and rule $\equiv$, $\Gamma[\vec{W}/\vec{Y}], x : U \vdash \mathbf{t} : T$. If $\Gamma[\vec{W}/\vec{Y}] \equiv \Gamma$, then we are finished. In the other case, $\vec{Y}$ appears free in Γ. Since $V \to R \preceq U \to T$ and $\Gamma \vdash \lambda x.\mathbf{t} : V \to R$, according to Lemma 5.3.2, $U \to T$ can be obtained from $V \to R$ as a type for $\lambda x.\mathbf{t}$; then we would need to use the rule $\forall_I$; thus $\vec{Y}$ cannot appear free in Γ, which constitutes a contradiction. So, $\Gamma, x : U \vdash \mathbf{t} : T$. □

D.11 Proof of Lemma 5.3.15

Lemma 5.3.15 (Generation lemma (linear combinations))**.** For any context Γ, scalar α, terms $\mathbf{t}$ and $\mathbf{r}$ and types S and T:

1. if $\Gamma \vdash \mathbf{t} + \mathbf{r} : S$ then there exist types R and R' such that $\Gamma \vdash \mathbf{t} : R$, $\Gamma \vdash \mathbf{r} : R'$ and $R + R' \preceq S$;
2. if $\Gamma \vdash \alpha.\mathbf{t} : T$, then there exists a type R such that $\alpha.R \preceq T$ and $\Gamma \vdash \alpha.\mathbf{t} : \alpha.R$;
3. if $\Gamma \vdash \mathbf{0} : T$, then there exists a type R such that $T \equiv 0.R$.

Proof.

1. Induction on the typing derivation.

   - Let $\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R$ as a consequence of $\Gamma \vdash \mathbf{t} : T$, $\Gamma \vdash \mathbf{r} : R$ and rule $+_I$. Trivial case.
   - Let $\Gamma \vdash \mathbf{t} + \mathbf{r} : S$ as a consequence of $\Gamma \vdash \mathbf{t} + \mathbf{r} : S'$ where $S' \equiv S$, and rule $\equiv$. Then by the induction hypothesis $\Gamma \vdash \mathbf{t} : T$, $\Gamma \vdash \mathbf{r} : R$ and $T + R \preceq S' \equiv S$.

- Let $\Gamma \vdash \mathbf{t} + \mathbf{r} \colon \sum_{i=1}^{n} \alpha_i.U_i[V/X]$ as a consequence of $\Gamma \vdash \mathbf{t} + \mathbf{r} \colon \sum_{i=1}^{n} \alpha_i.\forall X.U_i$ and rule $\forall_E$. Then by the induction hypothesis $\Gamma \vdash \mathbf{t} \colon T$, $\Gamma \vdash \mathbf{r} \colon R$ and $T + R \preceq \sum_{i=1}^{n} \alpha_i.\forall X.U_i \prec \sum_{i=1}^{n} \alpha_i.U_i[V/X]$.
- Let $\Gamma \vdash \mathbf{t} + \mathbf{r} \colon \sum_{i=1}^{n} \alpha_i.\forall X.U_i$ as a consequence of $\Gamma \vdash \mathbf{t} + \mathbf{r} \colon \sum_{i=1}^{n} \alpha_i.U_i$ and rule $\forall_I$. Then by the induction hypothesis $\Gamma \vdash \mathbf{t} \colon T$, $\Gamma \vdash \mathbf{r} \colon R$ and $T + R \preceq \sum_{i=1}^{n} \alpha_i.U_i \prec \sum_{i=1}^{n} \alpha_i.\forall X.U_i$.

2. Induction on the typing derivation.

- Let $\Gamma \vdash \alpha.\mathbf{t} \colon \alpha.T$ as a consequence of $\Gamma \vdash \mathbf{t} \colon T$ and rule s_I. Trivial case
- Let $\Gamma \vdash \alpha.\mathbf{t} \colon T$ as a consequence of $\Gamma \vdash \alpha.\mathbf{t} \colon S$ where $S \equiv T$, and rule $\equiv$. Then by the induction hypothesis $\Gamma \vdash \alpha.\mathbf{t} \colon \alpha.R$ with $\alpha.R \preceq S \equiv T$.
- Let $\Gamma \vdash \alpha.\mathbf{t} \colon \sum_{i=1}^{n} \alpha_i.U_i[V/X]$ as a consequence of $\Gamma \vdash \alpha.\mathbf{t} \colon \sum_{i=1}^{n} \alpha_i.\forall X.U_i$ and rule $\forall_E$. Then by the induction hypothesis $\Gamma \vdash \alpha.\mathbf{t} \colon \alpha.R$ with $\alpha.R \preceq \sum_{i=1}^{n} \alpha_i.\forall X.U_i \prec \sum_{i=1}^{n} \alpha_i.U_i[V/X]$.
- Let $\Gamma \vdash \alpha.\mathbf{t} \colon \sum_{i=1}^{n} \alpha_i.\forall X.U_i$ as a consequence of $\Gamma \vdash \alpha.\mathbf{t} \colon \sum_{i=1}^{n} \alpha_i.U_i$ and rule $\forall_I$. The by the induction hypothesis $\Gamma \vdash \alpha.\mathbf{t} \colon \alpha.R$ with $\alpha.R \preceq \sum_{i=1}^{n} \alpha_i.U_i \prec \sum_{i=1}^{n} \alpha_i.\forall X.U_i$.

3. Induction on the typing derivation.

- Let $\Gamma \vdash \mathbf{0} \colon 0.T$ as a consequence of $\Gamma \vdash \mathbf{t} \colon T$ and rule 0_I. Trivial case.
- Let $\Gamma \vdash \mathbf{0} \colon T$ as a consequence of $\Gamma \vdash \mathbf{0} \colon S$ where $S \equiv T$, and rule $\equiv$. Then by the induction hypothesis $0.R \equiv S \equiv T$.
- Let $\Gamma \vdash \mathbf{0} \colon \sum_{i=1}^{n} \alpha_i.U_i[V/X]$ as a consequence of $\Gamma \vdash \mathbf{0} \colon \sum_{i=1}^{n} \alpha_i.\forall X.U_i$ and rule $\forall_E$. Then by the induction hypothesis $\sum_{i=1}^{n} \alpha_i.\forall X.U_i \equiv 0.R$. By Lemma 5.2.2, $R \equiv \sum_{j=1}^{m} \beta_j.V_j$ and so $0.R \equiv \sum_{j=1}^{m} 0.V_j$. Thus by Lemma 5.2.3, $\forall V_j, \exists W_j \;/\; V_j \equiv \forall X.W_j$. Then by Lemma 5.2.3, $\sum_{i=1}^{n} \alpha_i.U_i \equiv \sum_{j=1}^{m} 0.W_j$ and by Lemma 5.2.3, $\sum_{i=1}^{n} \alpha_i.U_i[V/X] = (\sum_{i=1}^{n} \alpha_i.U_i)[V/X]$ which is equivalent to $(\sum_{j=1}^{m} 0.W_j)[V/X] = 0.\sum_{j=1}^{m} W_j[V/X]$.
- Let $\Gamma \vdash \mathbf{0} \colon \sum_{i=1}^{n} \alpha_i.\forall X.U_i$ as a consequence of $\Gamma \vdash \mathbf{0} \colon \sum_{i=1}^{n} \alpha_i.U_i$ and rule $\forall_I$. Then by the induction hypothesis $\sum_{i=1}^{n} \alpha_i.U_i \equiv 0.R$. By Lemma 5.2.2 $R \equiv \sum_{j=1}^{m} \beta_j.V_j$ so $0.R \equiv \sum_{j=1}^{m} 0.V_j$. Then by Lemma 5.2.3, $\sum_{i=1}^{n} \alpha_i.\forall X.U_i \equiv 0.\sum_{j=1}^{m} \forall X.W_j$.

□

D.12 Proof of Theorem 5.3.4

Theorem 5.3.4 (Weak subject reduction)**.** For any terms $\mathbf{t}$, $\mathbf{t}'$, any context Γ and any type T, if $\mathbf{t} \to_R \mathbf{t}'$ and $\Gamma \vdash \mathbf{t} \colon T$, then

- If $R \notin$ Factorisation rules, then $\Gamma \vdash \mathbf{t}' \colon T$.
- If $R \in$ Factorisation rules, then $\exists S \sqsubseteq T$ such that $\Gamma \vdash \mathbf{t}' \colon S$ and $\Gamma \vdash \mathbf{t} \colon S$.

Proof. Let $\mathbf{t} \to_R \mathbf{t}'$ and $\Gamma \vdash \mathbf{t} \colon T$. We proceed by induction. We treat separately every rule R.

Elementary rules

rule $0.\mathbf{t} \to \mathbf{0}$**.** Let $\Gamma \vdash 0.\mathbf{t} \colon T$. Then by Lemma 5.3.15, $\exists R \;/\; 0.R \preceq T$ and $\Gamma \vdash 0.\mathbf{t} \colon 0.R$, then by rule 0_I, $\Gamma \vdash \mathbf{0} \colon 0.(0.R)$. Since $0.(0.R) \equiv 0.R \preceq T$, by Lemma 5.3.2, $\Gamma \vdash \mathbf{0} \colon T$.

rule $1.\mathbf{t} \to \mathbf{t}$**.** Let $\Gamma \vdash 1.\mathbf{t} : T$, then by Lemma 5.3.15, $\exists R \;/\; 1.R \preceq T$ and $\Gamma \vdash 1.\mathbf{t} : 1.R$. Then by Lemma 5.3.6, $\Gamma \vdash \mathbf{t} : R$ and by $\equiv$-rule, $\Gamma \vdash \mathbf{t} : 1.R$, so by Lemma 5.3.2 $\Gamma \vdash \mathbf{t} : T$.

rule $\alpha.\mathbf{0} \to \mathbf{0}$**.** Let $\Gamma \vdash \alpha.\mathbf{0} : T$, then by Lemma 5.3.15, $\exists R \;/\; \alpha.R \preceq T$ and $\Gamma \vdash \alpha.\mathbf{0} : \alpha.R$. Cases:

- If $\alpha \neq 0$, then by Lemma 5.3.6, $\Gamma \vdash \mathbf{0} : R$, and, by Lemma 5.3.15, $\exists S \;/\; R \equiv 0.S$. Notice that $0.S = \alpha \times 0.S \equiv \alpha.(0.S) \equiv \alpha.R \preceq T$, so by Lemma 5.3.2, $\Gamma \vdash \mathbf{0} : T$.
- If $\alpha = 0$, then by rule 0_I, $\Gamma \vdash \mathbf{0} : 0.(0.R)$, and notice that $0.(0.R) \equiv 0.R$. Then by Lemma 5.3.2, $\Gamma \vdash \mathbf{0} : T$.

rule $\alpha.(\beta.\mathbf{t}) \to (\alpha \times \beta).\mathbf{t}$**.** Let $\Gamma \vdash \alpha.(\beta.\mathbf{t}) : T$. Then by Lemma 5.3.15, $\exists R$ such that $\alpha.R \preceq T$ and $\Gamma \vdash \alpha.(\beta.\mathbf{t}) : \alpha.R$ Cases:

- If $\alpha \neq 0$, then by Lemma 5.3.6, $\Gamma \vdash \beta.\mathbf{t} : R$. Then by Lemma 5.3.15 again, $\exists S \;/\; \beta.S \preceq R$ and $\Gamma \vdash \beta.\mathbf{t} : \beta.S$. If $\beta = 0$, then $(\alpha \times \beta).\mathbf{t} = \beta.\mathbf{t}$ and $0.S \equiv (\alpha \times 0).S \equiv \alpha.(0.S) \preceq \alpha.R \preceq T$ so by Lemma 5.3.2, $\Gamma \vdash \beta.\mathbf{t} : T$. If $\beta \neq 0$, then by Lemma 5.3.6, $\Gamma \vdash \mathbf{t} : S$, then by rule s_I, $\Gamma \vdash (\alpha \times \beta).\mathbf{t} : (\alpha \times \beta).S$.
  Note that $(\alpha \times \beta).S \equiv \alpha.(\beta.S) \preceq \alpha.R \preceq T$, so by Lemma 5.3.2, $\Gamma \vdash (\alpha \times \beta).\mathbf{t} : T$.
- If $\alpha = 0$, first we prove that $\Gamma \vdash 0.\beta.\mathbf{t} : T \Rightarrow \Gamma \vdash 0.\mathbf{t} : T$. We proceed by induction on the typing derivation.
  - Let $\Gamma \vdash 0.\beta.\mathbf{t} : 0.T$ as a consequence of $\Gamma \vdash \beta.\mathbf{t} : T$ and rule s_I. Then by Lemma 5.3.15, there exists a type R such that $\beta.R \preceq T$ and $\Gamma \vdash \beta.\mathbf{t} : \beta.R$. Cases:
    - If $\beta \neq 0$, then by Lemma 5.3.6 $\Gamma \vdash \mathbf{t} : R$ so by rule s_I we get $\Gamma \vdash 0.\mathbf{t} : 0.R$. Notice that $0.R = (0 \times \beta).R \equiv 0.\beta.R \preceq 0.T$, so by Lemma 5.3.2, $\Gamma \vdash 0.\mathbf{t} : 0.T$.
    - If $\beta = 0$, then $\Gamma \vdash 0.\mathbf{t} : 0.R$. Notice that $0.R = (0 \times 0).R \equiv 0.(0.R) \preceq 0.T$, so by Lemma 5.3.2, $\Gamma \vdash 0.\mathbf{t} : 0.T$.
  - Let $\Gamma \vdash 0.\beta.\mathbf{t} : T$ as a consequence of $\Gamma \vdash 0.\beta.\mathbf{t} : R$ where $R \equiv T$ and rule $\equiv$. Then by the induction hypothesis $\Gamma \vdash 0.\mathbf{t} : R$, so by rule $\equiv$, $\Gamma \vdash 0.\mathbf{t} : T$.
  - Let $\Gamma \vdash 0.\beta.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.U_i[V/X]$ as a consequence of $\Gamma \vdash 0.\beta.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i$ and rule $\forall_E$. Then by the induction hypothesis $\Gamma \vdash 0.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i$. So by rule $\forall_E$, $\Gamma \vdash 0.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.U_i[V/X]$.
  - Let $\Gamma \vdash 0.\beta.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i$ as a consequence of $\Gamma \vdash 0.\beta.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.U_i$ and rule $\forall_I$. Then by the induction hypothesis $\Gamma \vdash 0.\mathbf{t} : U$. So by rule $\forall_I$, $\Gamma \vdash 0.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i$

  With this result we can deduce that $\Gamma \vdash 0.\mathbf{t} : 0.R$. Notice that $0.\mathbf{t} = (0 \times \beta).\mathbf{t}$, so by Lemma 5.3.2, $\Gamma \vdash (0 \times \beta).\mathbf{t} : T$.

rule $\alpha.(\mathbf{t} + \mathbf{r}) \to \alpha.\mathbf{t} + \alpha.\mathbf{r}$**.** Let $\Gamma \vdash \alpha.(\mathbf{t} + \mathbf{r}) : T$. Then by Lemma 5.3.15, $\exists R$ such that $\alpha.R \preceq T$ and $\Gamma \vdash \alpha.(\mathbf{t} + \mathbf{r}) : \alpha.R$. Cases:

- If $\alpha \neq 0$, then by Lemma 5.3.6, $\Gamma \vdash \mathbf{t} + \mathbf{r} : R$. So by Lemma 5.3.15, $\exists S_1, S_2$ such that $\Gamma \vdash \mathbf{t} : S_1$, $\Gamma \vdash \mathbf{r} : S_2$ and $S_1 + S_2 \preceq R$. Then by rule s_I, $\Gamma \vdash \alpha.\mathbf{t} : \alpha.S_1$ and $\Gamma \vdash \alpha.\mathbf{r} : \alpha.S_2$, and so by rule $+_I$, $\Gamma \vdash \alpha.\mathbf{t} + \alpha.\mathbf{r} : \alpha.S_1 + \alpha.S_2$. Notice that $\alpha.S_1 + \alpha.S_2 \equiv \alpha.(S_1 + S_2) \preceq \alpha.R \preceq T$, so by Lemma 5.3.2, $\Gamma \vdash \alpha.\mathbf{t} + \alpha.\mathbf{r} : T$.
- If $\alpha = 0$, then $\Gamma \vdash 0.(\mathbf{t} + \mathbf{r}) : 0.R$. We show by induction the most general case: $\Gamma \vdash 0.(\mathbf{t} + \mathbf{r})$ implies $S \Rightarrow \Gamma \vdash 0.\mathbf{t} + 0.\mathbf{r} : S$. Then since $0.R \preceq T$, by Lemma 5.3.2, $\Gamma \vdash 0.\mathbf{t} + 0.\mathbf{r} : T$.
  - Let $\Gamma \vdash 0.(\mathbf{t} + \mathbf{r}) : 0.R$ as a consequence of $\Gamma \vdash \mathbf{t} + \mathbf{r} : R$ and rule s_I. Then by Lemma 5.3.15, $\exists T, S$ such that $\Gamma \vdash \mathbf{t} : T$, $\Gamma \vdash \mathbf{r} : S$ and $T + S \preceq R$. So by rule

s_I, $\Gamma \vdash 0.\mathbf{t} : 0.T$ and $\Gamma \vdash 0.\mathbf{r} : 0.S$ and so by rule $+_I$, $\Gamma \vdash 0.\mathbf{t} + 0.\mathbf{r} : 0.T + 0.S$. Notice that $0.T + 0.S \equiv 0.(T + S) \preceq 0.R$, so by Lemma 5.3.2, $\Gamma \vdash 0.\mathbf{t} + 0.\mathbf{r} : 0.R$.

- Let $\Gamma \vdash 0.(\mathbf{t} + \mathbf{r}) : S$ as a consequence of $\Gamma \vdash 0.(\mathbf{t} + \mathbf{r}) : R \qquad R \equiv S$ and rule $\equiv$. Then by the induction hypothesis $\Gamma \vdash 0.\mathbf{t} + 0.\mathbf{r} : R$. So by rule $\equiv$, $\Gamma \vdash 0.\mathbf{t} + 0.\mathbf{r} : S$.
- Let $\Gamma \vdash 0.(\mathbf{t} + \mathbf{r}) : \sum_{i=1}^{n} \alpha_i.U_i[V/X]$ as a consequence of rule $\forall_E$ and the sequent $\Gamma \vdash 0.(\mathbf{t} + \mathbf{r}) : \sum_{i=1}^{n} \alpha_i.\forall X.U_i$. By the induction hypothesis $\Gamma \vdash 0.\mathbf{t} + 0.\mathbf{r} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i$. So by rule $\forall_E$, $\Gamma \vdash 0.\mathbf{t} + 0.\mathbf{r} : \sum_{i=1}^{n} \alpha_i.U_i[V/X]$.
- Let $\Gamma \vdash 0.(\mathbf{t} + \mathbf{r}) : \sum_{i=1}^{n} \alpha_i.\forall X.U_i$ as a consequence of $\Gamma \vdash 0.(\mathbf{t} + \mathbf{r}) : \sum_{i=1}^{n} \alpha_i.U_i$ and rule $\forall_I$. Then by the induction hypothesis $\Gamma \vdash 0.\mathbf{t} + 0.\mathbf{r} : \sum_{i=1}^{n} \alpha_i.U_i$. So by rule $\forall_I$, $\Gamma \vdash 0.\mathbf{t} + 0.\mathbf{r} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i$.

Factorisation rules

rule $\alpha.\mathbf{t} + \beta.\mathbf{t} \to (\alpha + \beta).\mathbf{t}$**.** Let $\Gamma \vdash \alpha.\mathbf{t} + \beta.\mathbf{t} : T$. Then by Lemma 5.3.15, $\exists R, S$ such that $\Gamma \vdash \alpha.\mathbf{t} : R$, $\Gamma \vdash \beta.\mathbf{t} : S$ and $R + S \preceq T$. Then by Lemma 5.3.15, $\exists R' \,/\, \alpha.R' \preceq R$ and $\Gamma \vdash \alpha.\mathbf{t} : \alpha.R'$, also $\exists S' \,/\, \beta.S' \preceq S$ and $\Gamma \vdash \beta.\mathbf{t} : \beta.S'$. Cases:

- If $\alpha \neq 0$ (or analogously $\beta \neq 0$), then by Lemma 5.3.6, $\Gamma \vdash \mathbf{t} : R'$ and so by s_I we conclude $\Gamma \vdash (\alpha + \beta).\mathbf{t} : (\alpha + \beta).R'$. Notice that $(\alpha + \beta).R' \sqsubseteq \alpha.R' + \beta.S' \sqsubseteq R + S \sqsubseteq T$. Also using rules $+_I$ and $\equiv$ we conclude $\Gamma \vdash \alpha.\mathbf{t} + \beta.\mathbf{t} : (\alpha + \beta).R'$.
- If $\alpha = \beta = 0$, then notice that $\Gamma \vdash 0.\mathbf{t} : 0.R'$ and $0.R' \sqsubseteq 0.R' + 0.S' \sqsubseteq R + S \sqsubseteq T$. Also, using rules $+_I$ and $\equiv$, we conclude $\Gamma \vdash 0.\mathbf{t} + 0.\mathbf{t} : 0.R'$.

rules $\alpha.\mathbf{t} + \mathbf{t} \to (\alpha + 1).\mathbf{t}$ **and** $\mathbf{t} + \mathbf{t} \to (1 + 1).\mathbf{t}$**.** This cases are analogous to the previous case.

rule $\mathbf{t} + \mathbf{0} \to \mathbf{t}$**.** Let $\Gamma \vdash \mathbf{t} + \mathbf{0} : T$. Then by Lemma 5.3.15, $\exists R, S$ such that $\Gamma \vdash \mathbf{t} : R$, $\Gamma \vdash \mathbf{0} : S$ and $R + S \preceq T$. So, by Lemma 5.3.15, $\exists S' \,/\, S \equiv 0.S'$. Notice that $R \sqsubseteq R + 0.S' \equiv R + S \sqsubseteq T$.

Application rules

rule $(\mathbf{t} + \mathbf{r})\ \mathbf{u} \to (\mathbf{t})\ \mathbf{u} + (\mathbf{r})\ \mathbf{u}$**.** Let $\Gamma \vdash (\mathbf{t} + \mathbf{r})\ \mathbf{u} : T$. Then by Lemma 5.3.11, $\exists n$, m, U, $T_1, \ldots, T_n$, $\alpha_1, \ldots, \alpha_n$, $\beta_1, \ldots, \beta_m$, $V_1, \ldots, V_m$ such that $\Gamma \vdash \mathbf{t} + \mathbf{r} : \sum_{i=1}^{n} \alpha_i.\forall \vec{X}.(U \to T_i)$, $\Gamma \vdash \mathbf{u} : \sum_{j=1}^{m} \beta_j.V_j$, $\forall V_j,\ \exists \vec{W_j} \,/\, U[\vec{W_j}/\vec{X}] = V_j$ and $\sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j.T_i[\vec{W_j}/\vec{X}] \preceq T$. Then by Lemma 5.3.15, $\exists R, S$ such that $\Gamma \vdash \mathbf{t} : R$, $\Gamma \vdash \mathbf{r} : S$ and $R + S$ which is $\preceq$ than $\sum_{i=1}^{n} \alpha_i.\forall \vec{X}.(U \to T_i)$. Then $\exists N_1, N_2 \subseteq \{1, \ldots, n\}$ such that $R \preceq \sum_{i \in N_1 \setminus N_2} \alpha_i.\forall \vec{X}.(U \to T_i) + \sum_{i \in N_1 \cap N_2} \delta_i.\forall \vec{X}.(U \to T_i)$, $S \preceq \sum_{i \in N_2 \setminus N_1} \alpha_i.\forall \vec{X}.(U \to T_i) + \sum_{i \in N_1 \cap N_2} \gamma_i.\forall \vec{X}.(U \to T_i)$ and $\forall i \in N_1 \cap N_2,\ \delta_i + \gamma_i = \alpha_i$. Then by Lemma 5.3.2,$\Gamma \vdash \mathbf{t} : \sum_{i \in N_1 \setminus N_2} \alpha_i.\forall \vec{X}.(U \to T_i) + \sum_{i \in N_1 \cap N_2} \delta_i.\forall \vec{X}.(U \to T_i)$ and $\Gamma \vdash \mathbf{r} : \sum_{i \in N_2 \setminus N_1} \alpha_i.\forall \vec{X}.(U \to T_i) + \sum_{i \in N_1 \cap N_2} \gamma_i.\forall \vec{X}.(U \to T_i)$. Then by rule $\to_E$, $\Gamma \vdash (\mathbf{t})\ \mathbf{u} : \sum_{i \in N_1 \setminus N_2} \sum_{j=1}^{m} \alpha_i \times \beta_j.T_i[\vec{W_j}/\vec{X}] + \sum_{i \in N_1 \cap N_2} \sum_{j=1}^{m} \delta_i \times \beta_j.T_i[\vec{W_j}/\vec{X}]$, and analogously, $\Gamma \vdash (\mathbf{r})\ \mathbf{u} : \sum_{i \in N_2 \setminus N_1} \sum_{j=1}^{m} \alpha_i \times \beta_j.T_i[\vec{W_j}/\vec{X}] + \sum_{i \in N_1 \cap N_2} \sum_{j=1}^{m} \gamma_i \times \beta_j.T_i[\vec{W_j}/\vec{X}]$. So using $+_I$, $\Gamma \vdash (\mathbf{t})\ \mathbf{u} + (\mathbf{r})\ \mathbf{u} : \sum_{i \in N_1 \cup N_2 \setminus N_1 \cap N_2} \sum_{j=1}^{m} \alpha_i \times \beta_j.T_i[\vec{W_j}/\vec{X}] + \sum_{i \in N_1 \cap N_2} \sum_{j=1}^{m} (\delta_i + \gamma_i) \times \beta_j.T_i[\vec{W_j}/\vec{X}] \equiv \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j.T_i[\vec{W_j}/\vec{X}] \preceq T$. Then by Lemma 5.3.2, $\Gamma \vdash (\mathbf{t})\ \mathbf{u} + (\mathbf{r})\ \mathbf{u} : T$.

rule $(\mathbf{u})\ (\mathbf{t} + \mathbf{r}) \to (\mathbf{u})\ \mathbf{t} + (\mathbf{u})\ \mathbf{r}$**.** Let $\Gamma \vdash (\mathbf{u})\ (\mathbf{t} + \mathbf{r}) : T$. By Lemma 5.3.11, $\exists n$, m, U, T_i, α_i, β_j and V_j, with $i = 1, \ldots, n$ and $j = 1, \ldots, n$, such that $\Gamma \vdash \mathbf{u} : \sum_{i=1}^{n} \alpha_i.\forall \vec{X}.(U \to T_i)$ is valid and also $\Gamma \vdash \mathbf{t} + \mathbf{r} : \sum_{j=1}^{m} \beta_j.V_j$ is valid, and $\forall V_j,\ \exists \vec{W_j}$ such that $U[\vec{W_j}/\vec{X}] \equiv V_j$ and $\sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j.T_i[\vec{W_j}/\vec{X}] \preceq T$. Then by Lemma 5.3.15, $\exists R, S$ such that $\Gamma \vdash \mathbf{t} : R$, $\Gamma \vdash \mathbf{r} : S$ and $R + S \preceq \sum_{j=1}^{m} \beta_j.V_j$. Then, $\exists N_1, N_2 \subseteq \{1, \ldots, n\} \,/\, R \preceq \sum_{j \in N_1 \setminus N_2} \beta_j.V_j + \sum_{j \in N_1 \cap N_2} \delta_j.V_j,\ S \preceq$

$\sum_{j\in N_2\setminus N_1}\beta_j.V_j + \sum_{j\in N_1\cap N_2}\gamma_j.V_j$ and $\forall j \in N_1\cap N_2,\ \delta_j+\gamma_j=\beta_j$. So by Lemma 5.3.2, $\Gamma\vdash \mathbf{t}\colon \sum_{j\in N_1\setminus N_2}\beta_j.V_j + \sum_{j\in N_1\cap N_2}\delta_j.V_j$ and $\Gamma\vdash \mathbf{r}\colon \sum_{j\in N_2\setminus N_1}\beta_j.V_j+\sum_{j\in N_1\cap N_2}\gamma_j.V_j$. Then by rule $\to_E$, we have $\Gamma\vdash (\mathbf{u})\,\mathbf{t}\colon \sum_{i=1}^n\sum_{j\in N_1\setminus N_2}\alpha_i\times\beta_j.T_i[\vec{W_j}/\vec{X}]+\sum_{i=1}^n\sum_{\substack{j\in\\ N_1\cap N_2}}\alpha_i\times\delta_j.T_i[\vec{W_j}/\vec{X}]$ and $\Gamma\vdash (\mathbf{u})\,\mathbf{r}\colon \sum_{i=1}^n\sum_{j\in N_2\setminus N_1}\alpha_i\times\beta_j.T_i[\vec{W_j}/\vec{X}]+\sum_{i=1}^n\sum_{\substack{j\in\\ N_1\cap N_2}}\alpha_i\times\gamma_j.T_i[\vec{W_j}/\vec{X}]$. So with $+_I$, $\Gamma\vdash (\mathbf{u})\ \mathbf{t}+(\mathbf{u})\ \mathbf{r}\colon \sum_{i=1}^n\sum_{j\in N_1\cup N_2\setminus N_1\cap N_2}\alpha_i\times\beta_j.T_i[\vec{W_j}/\vec{X}]+\sum_{i=1}^n\sum_{j\in N_1\cap N_2}\alpha_i\times(\delta_j+\gamma_j).T_i[\vec{W_j}/\vec{X}]\equiv\sum_{i=1}^n\sum_{j=1}^m\alpha_i\times\beta_j.T_i[\vec{W_j}/\vec{X}]\preceq T$. Then by Lemma 5.3.2, $\Gamma\vdash(\mathbf{u})\,\mathbf{t}+(\mathbf{u})\,\mathbf{r}\colon T$.

rule $(\alpha.\mathbf{t})\ \mathbf{r}\to\alpha.(\mathbf{t})\ \mathbf{r}$. Let $\Gamma\vdash(\alpha.\mathbf{t})\ \mathbf{r}\colon T$. Then by Lemma 5.3.11, $\exists n, m, U, T_i,\alpha_i,\beta_j$ and V_j with $i=1\dots n$ and $j=1\dots m$ such that $\Gamma\vdash\alpha.\mathbf{t}\colon\sum_{i=1}^n\alpha_i.\forall\vec{X}.(U\to T_i)$ and $\Gamma\vdash\mathbf{r}\colon\sum_{j=1}^m\beta_j.V_j$ with $\sum_{i=1}^n\sum_{j=1}^m\alpha_i\times\beta_j.T_i[\vec{W_j}/\vec{X}]\preceq T$ where $\forall V_j$, $\vec{W_j}$ is such that $U[\vec{W_j}/\vec{X}]\equiv V_j$. Cases:

- If $\alpha\neq 0$, by Lemma 5.3.15, $\exists R$ such that $\alpha.R\preceq\sum_{i=1}^n\alpha_i.\forall\vec{X}.(U\to T_i)$ and $\Gamma\vdash\alpha.\mathbf{t}\colon\alpha.R$, then by Lemma 5.3.6, $\Gamma\vdash\mathbf{t}\colon R$. Notice also that $R\preceq\sum_{i=1}^n\delta_i.\forall\vec{X}.(U\to T_i)$ where $\forall i,\ \delta_i\times\alpha=\alpha_i$. Then by Lemma 5.3.2, $\Gamma\vdash\mathbf{t}\colon\sum_{i=1}^n\delta_i.\forall\vec{X}.(U\to T_i)$, so using $\to_E$, we get $\Gamma\vdash(\mathbf{t})\ \mathbf{r}\colon\sum_{i=1}^n\sum_{j=1}^m\delta_i\times\beta_j.T_i[\vec{W_j}/\vec{X}]$ and so with s_I we conclude $\Gamma\vdash\alpha.(\mathbf{t})\ \mathbf{r}\colon\alpha.\sum_{i=1}^n\sum_{j=1}^m\delta_i\times\beta_j.T_i[\vec{W_j}/\vec{X}]$. Notice that $\alpha.\sum_{i=1}^n\sum_{j=1}^m\delta_i\times\beta_j.T_i[\vec{W_j}/\vec{X}]$ is equivalent to $\sum_{i=1}^n\sum_{j=1}^m\alpha\times\delta_i\times\beta_j.T_i[W_j/X]\equiv\sum_{i=1}^n\sum_{j=1}^m\alpha_i\times\beta_j.T_i[W_j/X]\preceq T$. Then by Lemma 5.3.2, $\Gamma\vdash\alpha.(\mathbf{t})\ \mathbf{r}\colon T$.
- If $\alpha=0$, then by Lemma 5.3.7 we have $\Gamma\vdash\mathbf{t}\colon\sum_{i=1}^n\delta_i.\forall\vec{X}.(U\to T_i)$ and $\forall i,\ \alpha_i=0$, so with $\to_E$ followed by s_I we conclude $\Gamma\vdash 0.(\mathbf{t})\ \mathbf{r}\colon 0.\sum_{i=1}^n\sum_{j=1}^m\delta_i\times\beta_j.T_i[\vec{W_j}/\vec{X}]$ Notice that $0.\sum_{i=1}^n\sum_{j=1}^m\delta_i\times\beta_j.T_i[\vec{W_j}/\vec{X}]\equiv\sum_{i=1}^n\sum_{j=1}^m\alpha_i\times\beta_j.T_i[W_j/X]\preceq T$, so by Lemma 5.3.2, $\Gamma\vdash 0.(\mathbf{t})\ \mathbf{r}\colon T$.

rule $(\mathbf{t})\ \alpha.\mathbf{r}\to\alpha.(\mathbf{t})\ \mathbf{r}$. Let $\Gamma\vdash(\mathbf{t})\ \alpha.\mathbf{r}\colon T$. Then by Lemma 5.3.11, $\exists n, m, U, T_i,\alpha_i,\beta_j$ and V_j with $i=1\dots n$ and $j=1\dots m$ such that $\Gamma\vdash\mathbf{t}\colon\sum_{i=1}^n\alpha_i.\forall\vec{X}.(U\to T_i)$ and $\Gamma\vdash\alpha.\mathbf{r}\colon\sum_{j=1}^m\beta_j.V_j$ with $\sum_{i=1}^n\sum_{j=1}^m\alpha_i\times\beta_j.T_i[\vec{W_j}/\vec{X}]\preceq T$ where $\forall V_j$, $\vec{W_j}$ is such that $U[\vec{W_j}/\vec{X}]\equiv V_j$. Cases:

- If $\alpha\neq 0$, then by Lemma 5.3.15, $\exists R\ /\ \alpha.R\preceq\sum_{j=1}^m\beta_j.V_j$ and $\Gamma\vdash\alpha.\mathbf{r}\colon\alpha.R$, then by Lemma 5.3.6, $\Gamma\vdash\mathbf{r}\colon R$. Notice also that $R\preceq\sum_{j=1}^m\delta_j.V_j$ where $\forall j,\ \delta_j\times\alpha=\beta_j$. Then by Lemma 5.3.2, $\Gamma\vdash\mathbf{r}\colon\sum_{j=1}^m\delta_j.V_j$, so using rule $\to_E$, followed by s_I we get $\Gamma\vdash\alpha.(\mathbf{t})\ \mathbf{r}\colon\alpha.\sum_{i=1}^n\sum_{j=1}^m\alpha_i\times\delta_j.T_i[\vec{W_j}/\vec{X}_1$. Notice that $\alpha.\sum_{i=1}^n\sum_{j=1}^m\alpha_i\times\delta_j.T_i[\vec{W_j}/\vec{X}]\equiv\sum_{i=1}^n\sum_{j=1}^m\alpha_i\times\alpha\times\delta_j.T_i[\vec{W_j}/\vec{X}]\equiv\sum_{i=1}^n\sum_{j=1}^m\alpha_i\times\beta_j.T_i[\vec{W_j}/\vec{X}]\preceq T$. By Lemma 5.3.2, $\Gamma\vdash\alpha.(\mathbf{t})\ \mathbf{r}\colon T$.
- If $\alpha=0$, then by Lemma 5.3.7 we have $\Gamma\vdash\mathbf{r}\colon\sum_{j=1}^m\delta_j.V_j$ and $\forall j,\ \beta_j=0$, so using rule $\to_E$ followed by rule s_I, we conclude $\Gamma\vdash 0.(\mathbf{t})\ \mathbf{r}\colon 0.\sum_{i=1}^n\sum_{j=1}^m\alpha_i\times\delta_j.T_i[\vec{W_j}/\vec{X}]$. Notice that $0.\sum_{i=1}^n\sum_{j=1}^m\alpha_i\times\delta_j.T_i[\vec{W_j}/\vec{X}]\equiv\sum_{i=1}^n\sum_{j=1}^m\alpha_i\times\beta_j.T_i[\vec{W_j}/\vec{X}]\preceq T$. Then by Lemma 5.3.2, $\Gamma\vdash\alpha.(\mathbf{t})\ \mathbf{r}\colon T$.

rule $(\mathbf{0})\ \mathbf{t}\to\mathbf{0}$. Let $\Gamma\vdash(\mathbf{0})\ \mathbf{t}\colon T$. Then by Lemma 5.3.11, $\exists n, m, U, T_i,\alpha_i$ and β_j with $i=1\dots n$ and $j=1\dots m$ such that $\Gamma\vdash\mathbf{0}\colon\sum_{i=1}^n\alpha_i.\forall\vec{X}.(U\to T_i)$ and $\Gamma\vdash\mathbf{t}\colon\sum_{j=1}^m\beta_j.V_j$, where $\forall V_j,\ \exists\vec{W_j}\ /\ U[\vec{W_j}/\vec{X}]\equiv V_j$ and $\sum_{i=1}^n\sum_{j=1}^m\alpha_i\times\beta_j.T_i[\vec{W_j}/\vec{X}]\preceq T$. By Lemma 5.3.15, $\exists R\ /\ \sum_{i=1}^n\alpha_i.\forall\vec{X}.(U\to T_i)\equiv 0.R$, then it is equivalent to $\sum_{i=1}^{n'}0.\forall\vec{X}.(U\to T_i)$ with $n'\leq n$, that is $\exists N_1,\dots,N_{n'}\subseteq\{1,\dots,n\}$ disjoint sets, such that $\forall k,\ \forall i,j\in N_k,\ T_i=T_j$ and $\sum_{i\in N_k}\alpha_i=0$. Then by rule $\to_E$, $\Gamma\vdash(\mathbf{0})\ \mathbf{t}\colon\sum_{i=1}^{n'}\sum_{j=1}^m 0\times\beta_j.T_i[\vec{W_j}/\vec{X}]$ and so by 0_I, $\Gamma\vdash\mathbf{0}\colon 0.\sum_{i=1}^{n'}\sum_{j=1}^m 0\times\beta_j.T_i[\vec{W_j}/\vec{X}]$. Notice that $0.\sum_{i=1}^{n'}\sum_{j=1}^m 0\times\beta_j.T_i[\vec{W_j}/\vec{X}]\equiv\sum_{i=1}^{n'}\sum_{j=1}^m\alpha_i\times\beta_j.T_i[\vec{W_j}/\vec{X}]\preceq T$, then by Lemma 5.3.2, $\Gamma\vdash\mathbf{0}\colon T$.

rule $(\mathbf{t})\ \mathbf{0} \to \mathbf{0}$**.** Let $\Gamma \vdash (\mathbf{t})\ \mathbf{0} : T$. Then by Lemma 5.3.11, $\exists n, m,\ U,\ T_i$,α_i and β_j with $i = 1 \dots n$ and $j = 1 \dots m$ such that $\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall \vec{X}.(U \to T_i)$ and $\Gamma \vdash \mathbf{0} : \sum_{j=1}^{m} \beta_j.V_j$ where $\forall V_j,\ \exists \vec{W_j}\ /\ U[\vec{W_j}/\vec{X}] \equiv V_j$ and $\sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \times \beta_j.T_i[\vec{W_j}/\vec{X}] \preceq T$. By Lemma 5.3.15, $\exists R\ /\ \sum_{j=1}^{m} \beta_j.V_j \equiv 0.R$, so it is equivalent to $\sum_{j=1}^{m'} 0.V_j$ with $m' \leq n$, *i.e.* $\exists M_1, \dots, M_{m'} \subseteq \{1, \dots, m\}$ disjoint sets, such that $\forall k,\ \forall i, j \in M_k,\ V_i = V_j$ and $\sum_{j \in M_k} \beta_j = 0$. Then using rule $\to_E$ we get $\Gamma \vdash (\mathbf{t})\ \mathbf{0} : \sum_{i=1}^{n}\sum_{j=1}^{m'} \alpha_i \times 0.T_i[\vec{W_j}/\vec{X}]$ and by rule 0_I, $\Gamma \vdash \mathbf{0} : 0.\sum_{i=1}^{n}\sum_{j=1}^{m'} \alpha_i \times 0.T_i[\vec{W_j}/\vec{X}]$. Notice that $0.\sum_{i=1}^{n}\sum_{j=1}^{m'} \alpha_i \times 0.T_i[\vec{W_j}/\vec{X}] \equiv \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \times \beta_j.T_i[\vec{W_j}/\vec{X}] \preceq T$, then by Lemma 5.3.2, $\Gamma \vdash \mathbf{0} : T$.

Beta reduction

rule $(\lambda x.\mathbf{t})\ \mathbf{b} \to \mathbf{t}[\mathbf{b}/x]$**.** Let $\Gamma \vdash (\lambda x.\mathbf{t})\ \mathbf{b} : T$. Then by Lemma 5.3.11, there exist numbers n, m, scalars $\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_m$, a unit type U, and general types $T_1, \dots, T_n$ such that the following sequents can be derived: $\Gamma \vdash \lambda x.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall \vec{X}.(U \to T_i)$ and $\Gamma \vdash \mathbf{b} : \sum_{j=1}^{m} \beta_j.V_j$ with $\sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \times \beta_j.T_i[\vec{W_j}/\vec{X}] \preceq T$, where $\forall V_j$, $\vec{W_j}$ is such that $U[\vec{W_j}/\vec{X}] \equiv V_j$. By Lemma 5.3.8, $\sum_{i=1}^{n} \alpha_i.\forall \vec{X}.(U \to T_i) \equiv \forall \vec{X}.(U \to T_i)$ and $\forall i, k,\ T_i = T_k$, analogously $\sum_{j=1}^{m} \beta_j.V_j \equiv V_j$ where $\forall j, h,\ V_j = V_h$. So $\sum_{i=1}^{n} \alpha_i = 1$ and $\sum_{j=1}^{m} \beta_i = 1$. Then by rule $\equiv$, $\Gamma \vdash \lambda x.\mathbf{t} : \forall \vec{X}.(U \to T_i)$, and $\Gamma \vdash \mathbf{b} : V_i$. Thus, by Corollary 5.3.14, $\Gamma, x : U \vdash \mathbf{t} : T_i$. Notice that $V_i \equiv U[\vec{W_i}/\vec{X}]$, then, by Lemma 5.3.10[2], we have $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T_i[\vec{W_i}/\vec{X}]$. Since $T_i[\vec{W_j}/\vec{X}] \equiv (1 \times 1).T_i[\vec{W_j}/\vec{X}] = (\sum_{i=1}^{n} \alpha_i) \times (\sum_{j=1}^{m} \beta_j).T_i[\vec{W_j}/\vec{X}] = (\sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \times \beta_j).T_i[\vec{W_j}/\vec{X}]$, and as all the T_i are equivalents between them, this type is equivalent to $\sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \times \beta_j.T_i[\vec{W_j}/\vec{X}] \preceq T$. By Lemma 5.3.2, $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T$.

AC equivalences

rule $\mathbf{t} + \mathbf{r} = \mathbf{r} + \mathbf{t}$**.** Let $\Gamma \vdash \mathbf{t} + \mathbf{r} : T$. Then by Lemma 5.3.15 $\exists R, S\ /\ R + S \preceq T$, $\Gamma \vdash \mathbf{t} : R$ and $\Gamma \vdash \mathbf{r} : S$. So using $+_I$, $\Gamma \vdash \mathbf{r} + \mathbf{t} : S + R$. Note that $S + R \equiv R + S \preceq T$, then by Lemma 5.3.2 $\Gamma \vdash \mathbf{r} + \mathbf{t} : T$.

rule $(\mathbf{t} + \mathbf{r}) + \mathbf{u} = \mathbf{t} + (\mathbf{r} + \mathbf{u})$**.** Let $\Gamma \vdash (\mathbf{t} + \mathbf{r}) + \mathbf{u} : T$. Then by Lemma 5.3.15 $\exists R, S$ such that $\Gamma \vdash \mathbf{t}+\mathbf{r} : R$, $\Gamma \vdash \mathbf{u} : S$ and $R+S \preceq T$. Then, by Lemma 5.3.15 again, $\exists R', S'$ such that $\Gamma \vdash \mathbf{t} : R'$, $\Gamma \vdash \mathbf{r} : S'$ and $R' + S' \preceq R$. So using $+_I$ in the correct order, we get $\Gamma \vdash \mathbf{t}+(\mathbf{r}+\mathbf{u}) : R'+(S'+S)$. Note that $R' + (S' + S) \equiv (R' + S') + S \preceq R + S \preceq T$, then by Lemma 5.3.2 $\Gamma \vdash \mathbf{t} + (\mathbf{r} + \mathbf{u}) : T$.

Contextual rules

- Let $\alpha.\mathbf{s} \to \alpha.\mathbf{t}$ as a consequence of $\mathbf{s} \to \mathbf{t}$. Let $\Gamma \vdash \alpha.\mathbf{s} : T$, then by Lemma 5.3.15 $\exists R\ /\ \alpha.R \preceq T$ and $\Gamma \vdash \alpha.\mathbf{s} : \alpha.R$. Cases: If $\alpha \neq 0$, then by Lemma 5.3.6, $\Gamma \vdash \mathbf{s} : R$, so by the induction hypothesis $\Gamma \vdash \mathbf{t} : R'$ with $R' \sqsubseteq R$, then using s_I, $\Gamma \vdash \alpha.\mathbf{t} : \alpha.R'$. Notice that $\alpha.R' \sqsubseteq \alpha.R \sqsubseteq T$. If $\alpha = 0$, then notice that $T \equiv \sum_{i=1}^{n} \beta_i.U_i$, so by Lemma 5.3.7, $\Gamma \vdash \mathbf{s} : \sum_{i=1}^{n} \delta_i.U_i$ and $\forall i,\ \beta_i = 0$. Then by the induction hypothesis $\Gamma \vdash \mathbf{t} : R$ with $R \sqsubseteq \sum_{i=1}^{n} \delta_i.U_i$. So using s_I, $\Gamma \vdash 0.\mathbf{t} : 0.R$. Notice that $0.R \sqsubseteq 0.\sum_{i=1}^{n} \delta_i.U_i \equiv \sum_{i=1}^{n} 0.U_i \equiv T$.
- Let $\mathbf{r} + \mathbf{s} \to \mathbf{r} + \mathbf{t}$ as a consequence of $\mathbf{s} \to \mathbf{t}$. Let $\Gamma \vdash \mathbf{r} + \mathbf{s} : T$, then by Lemma 5.3.15, $\exists R, S$ such that $\Gamma \vdash \mathbf{r} : R$, $\Gamma \vdash \mathbf{s} : S$ and $R + S \preceq T$. Then by the induction hypothesis $\Gamma \vdash \mathbf{t} : S' \sqsubseteq S$, so using $+_I$ we can conclude $\Gamma \vdash \mathbf{r} + \mathbf{t} : R + S'$. Notice that $R + S' \sqsubseteq R + S \preceq T$.

- Let $(\mathbf{r})\ \mathbf{s} \to (\mathbf{r})\ \mathbf{t}$ as a consequence of $\mathbf{s} \to \mathbf{t}$. Let $\Gamma \vdash (\mathbf{r})\ \mathbf{s} : T$, Then by Lemma 5.3.11, $\exists n, n, o \geq 1, U, T_1, \ldots, T_n, \alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_m, V_1, \ldots, V_m$ such that $\Gamma \vdash \mathbf{r} : \sum_{i=1}^{n} \alpha_i . \forall \vec{X} . (U \to T_i)$ and $\Gamma \vdash \mathbf{s} : \sum_{j=1}^{m} \beta_j . V_j$ where for all V_j, there exists $\vec{W}_j$ such that $U[\vec{W}_j/\vec{X}] \equiv V_j$ and $\sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j . T_i[\vec{W}_j/\vec{X}] \preceq T$. Cases:
  - If $\mathbf{s} \to_R \mathbf{t}$ with $R \notin$ factorisation rules, then by the induction hypothesis $\Gamma \vdash \mathbf{t} : \sum_{j=1}^{m} \beta_j . V_j$, so using rule $\to_E$ we obtain $\Gamma \vdash (\mathbf{r})\ \mathbf{t} : \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j . T_i[\vec{W}_j/\vec{X}]$. By Lemma 5.3.2, $\Gamma \vdash (\mathbf{r})\ \mathbf{t} : T$.
  - If $\mathbf{s} \to_R \mathbf{t}$ with $R \in$ factorisation rules, then by the induction hypothesis $\Gamma \vdash \mathbf{t} : R$ with $R \sqsubseteq \sum_{j=1}^{m} \beta_j . V_j$. By Lemma 5.3.5, $\exists \delta, k, N \subseteq \{1 \ldots, m\}, W \preceq V_k$ such that $R \equiv \delta . W + \sum_{j \in N} \beta_j . V_j$. Notice that since we have obtained $\mathbf{t}$ from $\mathbf{s}$ by applying one of the factorisation rules, we can safely take $W \equiv V_k$. Then notice that $\sum_{i=1}^{n} \alpha_i \times \delta . T_i[\vec{W}_j/\vec{X}] + \sum_{i=1}^{n} \sum_{j \in N} \alpha_i \times \beta_j . T_i[\vec{W}_j/\vec{X}] \sqsubseteq \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j . T_i[\vec{W}_j/\vec{X}] \preceq T$.
- Let $(\mathbf{s})\ \mathbf{r} \to (\mathbf{t})\ \mathbf{r}$ as a consequence of $\mathbf{s} \to \mathbf{t}$. Let $\Gamma \vdash (\mathbf{s})\ \mathbf{r} : T$, Then by Lemma 5.3.11, $\exists n, n, o \geq 1, U, T_1, \ldots, T_n, \alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_m, V_1, \ldots, V_m$ such that $\Gamma \vdash \mathbf{s} : \sum_{i=1}^{n} \alpha_i . \forall \vec{X} . (U \to T_i)$ and $\Gamma \vdash \mathbf{r} : \sum_{j=1}^{m} \beta_j . V_j$ where for all V_j, there exists $\vec{W}_j$ such that $U[\vec{W}_j/\vec{X}] \equiv V_j$ and $\sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j . T_i[\vec{W}_j/\vec{X}] \preceq T$. Cases:
  - If $\mathbf{s} \to_R \mathbf{t}$ with R not a factorisation rule, then by the induction hypothesis one has $\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i . \forall \vec{X} . (U \to T_i)$, so using rule $\to_E$, $\Gamma \vdash (\mathbf{t})\ \mathbf{r} : \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j . T_i[\vec{W}_j/\vec{X}]$. Then by Lemma 5.3.2, $\Gamma \vdash (\mathbf{t})\ \mathbf{r} : T$.
  - If $\mathbf{s} \to_R \mathbf{t}$ with $R \in$ factorisation rules, then by the induction hypothesis $\Gamma \vdash \mathbf{t} : R$ where $R \sqsubseteq \sum_{i=1}^{n} \alpha_i . \forall \vec{X} . (U \to T_i)$. By Lemma 5.3.5, $\exists \delta, k, N \subseteq \{1, \ldots, n\}, W \preceq \forall \vec{X} . (U \to T_k)$ such that $R \equiv \delta . W + \sum_{i \in N} \alpha_i . \forall \vec{X} . (U \to T_i)$. Notice that since we have obtained $\mathbf{t}$ from $\mathbf{s}$ by applying one of the factorisation rules, we can safely take $W \equiv \forall \vec{X} . (U \to T_k)$. Then using rule $\to_E$, $\Gamma \vdash (\mathbf{t})\ \mathbf{r} : \sum_{j=1}^{m} \delta \times \beta_j . T_k[\vec{W}_j/\vec{X}]$. Notice that $\sum_{j=1}^{m} \delta \times \beta_j . T_k[\vec{W}_j/\vec{X}] + \sum_{i \in N} \sum_{j=1}^{m} \alpha_i \times \beta_j . T_i[\vec{W}_j/\vec{X}] \sqsubseteq \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j . T_i[\vec{W}_j/\vec{X}] \preceq T$.
- Let $\lambda x . \mathbf{s} \to \lambda x . \mathbf{t}$ as a consequence of $\mathbf{s} \to \mathbf{t}$. Let $\Gamma \vdash \lambda x . \mathbf{s} : T$. By Lemma 5.3.12, $\exists U, R$ such that $U \to R \preceq T$ and $\Gamma, x : U \vdash \mathbf{s} : R$. Then by the induction hypothesis $\Gamma, x : U \vdash \mathbf{t} : S$, with $S \sqsubseteq R$. So using rule $\to_I$, $\Gamma \vdash \lambda x . \mathbf{t} : U \to S$. Notice that since $S \sqsubseteq R$, then $U \to S \sqsubseteq U \to R \sqsubseteq T$.

□

D.13 Proof of Lemma 5.4.1

We need an intermediate result first, showing that the linear combination of strongly normalising terms, is strong normalising.

Lemma D.13.1. *If $\{\mathbf{t}_i\}_i$ are strongly normalising, then so is any linear combination of term made of the $\mathbf{t}_i$*

Proof. Let $\vec{\mathbf{t}} = \mathbf{t}_1, \ldots, \mathbf{t}_n$. We define the *algebraic context* $F(\cdot)$ by the following grammar:

$$F(\vec{\mathbf{t}}) \quad ::= \quad \mathbf{t}_i \mid F(\vec{\mathbf{t}}) + F(\vec{\mathbf{t}}) \mid \alpha \cdot F(\vec{\mathbf{t}}) \mid \mathbf{0}.$$

We claim that for all algebraic contexts $F(\cdot)$ and all strongly normalising terms $\mathbf{t}_i$ that are not linear combinations (that is, of the form x, $\lambda x . \mathbf{r}$ or $(\mathbf{s})\ \mathbf{r}$), the term $F(\vec{\mathbf{t}})$ is also strongly normalising.

The claim is proven by induction on $s(\vec{\mathbf{t}})$, the sum over i of the sum of the lengths of all the possible rewrite sequences starting with $\mathbf{t}_i$.

- If $s(\vec{\mathbf{t}}) = 0$. Then none of the $\mathbf{t}_i$ reduces. We define an algebraic measure $a(\cdot)$ inductively as follows: $a(\mathbf{t}_i) = 1$, $a(F(\vec{\mathbf{t}}) + F(\vec{\mathbf{t}'})) = 2 + a(F(\vec{\mathbf{t}})) + a(F(\vec{\mathbf{t}'}))$, $a(\alpha \cdot F(\vec{\mathbf{t}})) = 1 + 2.a(F(\vec{\mathbf{t}}))$, $a(\mathbf{0}) = 0$. It is now possible to show by induction on $a(F(\vec{\mathbf{t}}))$ that $F(\vec{\mathbf{t}})$ is SN.
  - If $a(F(\vec{\mathbf{t}})) = 0$, then $F(\vec{\mathbf{t}}) = \mathbf{0}$ which is SN.
  - Suppose it is true for all $F(\vec{\mathbf{t}})$ of algebraic measure less or equal to m, and consider $F(\vec{\mathbf{t}})$ such that $a(F(\vec{\mathbf{t}})) = m + 1$. Since the $\mathbf{t}_i$ are not linear combinations and they are in normal form, $F(\vec{\mathbf{t}})$ can only reduce with a rule from Group E or a rule from group F. We show that those reductions are strictly decreasing on the algebraic measure, by a rule by rule analysis, and so, we can conclude by induction hypothesis.
    - $0.F(\vec{\mathbf{t}}) \to \mathbf{0}$. Note that $a(0.F(\vec{\mathbf{t}})) = 1 \geq 0 = a(\mathbf{0})$.
    - $1.F(\vec{\mathbf{t}}) \to F(\vec{\mathbf{t}})$. Note that $a(1.F(\vec{\mathbf{t}})) = 1 + 2.a(F(\vec{\mathbf{t}})) \geq a(F(\vec{\mathbf{t}}))$.
    - $\alpha.\mathbf{0} \to \mathbf{0}$. Note that $a(\alpha.\mathbf{0}) = 1 \geq 0 = a(\mathbf{0})$.
    - $\alpha.(\beta.F(\vec{\mathbf{t}})) \to (\alpha \times \beta).F(\vec{\mathbf{t}})$. Note that $a(\alpha.(\beta.F(\vec{\mathbf{t}}))) = 1 + 2.(1 + 2.a(F(\vec{\mathbf{t}}))) \geq 1 + 2.a(F(\vec{\mathbf{t}})) = a((\alpha \times \beta).F(\vec{\mathbf{t}}))$.
    - $\alpha.(F(\vec{\mathbf{t}}) + F(\vec{\mathbf{t}'})) \to \alpha.F(\vec{\mathbf{t}}) + \alpha.F(\vec{\mathbf{t}'})$. Note that $a(\alpha.(F(\vec{\mathbf{t}}) + F(\vec{\mathbf{t}'}))) = 5 + 2.a(F(\vec{\mathbf{t}})) + 2.a(F(\vec{\mathbf{t}'})) > 4 + 2.a(F(\vec{\mathbf{t}})) + 2.a(F(\vec{\mathbf{t}'})) = a(\alpha.F(\vec{\mathbf{t}}) + \alpha.F(\vec{\mathbf{t}'}))$.
    - $\alpha.F(\vec{\mathbf{t}}) + \beta.F(\vec{\mathbf{t}}) \to (\alpha + \beta).F(\vec{\mathbf{t}})$. Note that $a(\alpha.F(\vec{\mathbf{t}}) + \beta.F(\vec{\mathbf{t}})) = 4 + 4.a(F(\vec{\mathbf{t}})) \geq 1 + 2.a(F(\vec{\mathbf{t}})) = a((\alpha + \beta).F(\vec{\mathbf{t}}))$.
    - $\alpha.F(\vec{\mathbf{t}}) + F(\vec{\mathbf{t}}) \to (\alpha+1).F(\vec{\mathbf{t}})$. Note that $a(\alpha.F(\vec{\mathbf{t}}) + F(\vec{\mathbf{t}})) = 3 + 3.a(F(\vec{\mathbf{t}})) \geq 1 + 2.a(F(\vec{\mathbf{t}})) = a.((\alpha + 1).F(\vec{\mathbf{t}}))$.
    - $F(\vec{\mathbf{t}}) + F(\vec{\mathbf{t}}) \to (1+1).F(\vec{\mathbf{t}})$. Note that $a.(F(\vec{\mathbf{t}}) + F(\vec{\mathbf{t}})) = 2 + 2.a(F(\vec{\mathbf{t}})) \geq 1 + 2.a(F(\vec{\mathbf{t}})) = a.((1 + 1).F(\vec{\mathbf{t}}))$.
    - $F(\vec{\mathbf{t}}) + \mathbf{0} \to F(\vec{\mathbf{t}})$. Note that $a.(F(\vec{\mathbf{t}}) + \mathbf{0}) = 2 + a(F(\vec{\mathbf{t}})) \geq a(F(\vec{\mathbf{t}}))$.
- Suppose it is true for n, then consider $\vec{\mathbf{t}}$ such that $s(\vec{\mathbf{t}}) = n + 1$. Again, we show that $F(\vec{\mathbf{t}})$ is SN by induction on $a(F(\vec{\mathbf{t}}))$.
  - If $a(F(\vec{\mathbf{t}})) = 0$, then $F(\vec{\mathbf{t}}) = \mathbf{0}$ which is SN.
  - Suppose it is true for all $F(\vec{\mathbf{t}})$ of algebraic measure less or equal to m, and consider $F(\vec{\mathbf{t}})$ such that $a(F(\vec{\mathbf{t}})) = m + 1$. Since the $\mathbf{t}_i$ are not linear combinations, $F(\vec{\mathbf{t}})$ can reduce in two ways:
    - $F(\mathbf{t}_1, \dots \mathbf{t}_i, \dots \mathbf{t}_k) \to F(\mathbf{t}_1, \dots \mathbf{t}'_i, \dots \mathbf{t}_k)$ with $\mathbf{t}_i \to \mathbf{t}'_i$. Then write $\mathbf{t}'_i$ as $H(\mathbf{r}_1, \dots \mathbf{r}_l)$ for some algebraic context H, where the r_j's are not linear combinations. Note that

$$\sum_{j=1}^{l} s(\mathbf{r}_j) \leq s(\mathbf{t}'_i) < s(\mathbf{t}_i).$$

      Define the context

$$G(\mathbf{t}_1, \dots, \mathbf{t}_{i-1}, \mathbf{u}_1, \dots \mathbf{u}_l, \mathbf{t}_{i+1}, \dots \mathbf{t}_k) = F(\mathbf{t}_1, \dots, \mathbf{t}_{i-1}, H(\mathbf{u}_1, \dots \mathbf{u}_l), \mathbf{t}_{i+1}, \dots \mathbf{t}_k).$$

      The term $F(\vec{\mathbf{t}})$ then reduces to the term

$$G(\mathbf{t}_1, \dots, \mathbf{t}_{i-1}, \mathbf{r}_1, \dots \mathbf{r}_l, \mathbf{t}_{i+1} \dots \mathbf{t}_k),$$

where

$$s(\mathbf{t}_1, \ldots, \mathbf{t}_{i-1}, \mathbf{r}_1, \ldots \mathbf{r}_l, \mathbf{t}_{i+1} \ldots \mathbf{t}_k) < s(\vec{\mathbf{t}}).$$

Using the top induction hypothesis, we can conclude that $F(\mathbf{t}_1, \ldots \mathbf{t}'_i, \ldots \mathbf{t}_k)$ is SN.

- $F(\vec{\mathbf{t}}) \to G(\vec{\mathbf{t}})$, with $a(G(\vec{\mathbf{t}})) < a(F(\vec{\mathbf{t}}))$. Using the second induction hypothesis, we conclude that $G(\vec{\mathbf{t}})$ is SN

All the possible reducts of $F(\vec{\mathbf{t}})$ are SN: so is $F(\vec{\mathbf{t}})$.

This closes the proof of the claim. Now, consider any SN terms $\{\mathbf{t}_i\}_i$ and any linear combination made of them. It can be written as $F(\vec{\mathbf{t}})$. The hypotheses of the claim are satisfied: $F(\vec{\mathbf{t}})$ is SN. □

Now we can prove the Lemma.

Lemma 5.4.1. If $\mathcal{A}$, $\mathcal{B}$ and all the $\mathcal{A}_i$'s are in $\mathcal{RC}$, then so are $\mathcal{A} \to \mathcal{B}$, $\sum_i \mathcal{A}_i$ and $\cap_i \mathcal{A}_i$.

Proof.

$\mathcal{A} \to \mathcal{B}$

$\mathbf{RC}_1$: Assume that $\mathbf{t} \in \mathcal{A} \to \mathcal{B}$ is not in SN. Then there is an infinite sequence of reduction $(\mathbf{t}_n)_n$ with $\mathbf{t}_0 = \mathbf{t}$. So there is an infinite sequence of reduction $((\mathbf{t}_n)\ \mathbf{b})_n$ starting with $(\mathbf{t})\mathbf{b}$, for all base terms $\mathbf{b}$. This contradicts the definition of $\mathcal{A} \to \mathcal{B}$.

$\mathbf{RC}_2$: We must show that if $\mathbf{t} \to \mathbf{t}'$ and $\mathbf{t} \in \mathcal{A} \to \mathcal{B}$, then $\mathbf{t}' \in \mathcal{A} \to \mathcal{B}$. Let $\mathbf{t}$ such that $\forall \mathbf{b} \in \mathcal{A}$, $(\mathbf{t})\ \mathbf{b} \in \mathcal{B}$. Then by $\mathbf{RC}_2$ in $\mathcal{B}$, $(\mathbf{t}')\ \mathbf{b} \in \mathcal{B}$, and so $\mathbf{t}' \in \mathcal{A} \to \mathcal{B}$. If $\mathbf{t}$ is neutral and $\mathrm{Red}(\mathbf{t}) \subseteq \mathcal{A} \to \mathcal{B}$, then $\mathbf{t}' \in \mathcal{A} \to \mathcal{B}$ since $\mathbf{t}' \in \mathrm{Red}(\mathbf{s})$. If $\mathbf{t} = \mathbf{0}$, it does not reduce.

$\mathbf{RC}_3$ and $\mathbf{RC}_4$: Trivially true by definition.

$\sum_i \mathcal{A}_i$

$\mathbf{RC}_1$: If $\mathbf{t} \in \{\sum_i \alpha_i.\mathbf{t}_i \mid \mathbf{t}_i \in \mathcal{A}_i\}$, the result is trivial by condition $\mathbf{RC}_1$ on the $\mathcal{A}_i$ and Lemma D.13.1. If $\mathbf{t}$ is neutral and $\mathrm{Red}(\mathbf{t}) \subseteq \mathcal{A} + \mathcal{B}$, then $\mathbf{t}$ is strongly normalising since all elements of $\mathrm{Red}(\mathbf{t})$ are strongly normalising.

$\mathbf{RC}_2$ and $\mathbf{RC}_3$: Trivially true by definition.

$\mathbf{RC}_4$: Since $\sum_i 0 \cdot \mathbf{t}_i \in \sum_i \mathcal{A}_i$, by $\mathbf{RC}_2$, $\mathbf{0}$ is also in the set.

$\cap_i \mathcal{A}_i$

$\mathbf{RC}_1$: Trivial since $\forall i$, $\mathcal{A}_i \subseteq SN$.

$\mathbf{RC}_2$: Let $\mathbf{t} \in \cap_i \mathcal{A}_i$, then $\forall i$, $\mathbf{t} \in \mathcal{A})_i$ and so by $\mathbf{RC}_2$ in $\mathcal{A}_i$, $\mathrm{Red}(\mathbf{t}) \subseteq \mathcal{A}_i$. Thus $\mathrm{Red}(\mathbf{t}) \subseteq \cap_i \mathcal{A}_i$.

$\mathbf{RC}_3$: Let $\mathbf{t} \in \mathcal{N}$ and $\mathrm{Red}(\mathbf{t}) \subseteq \cap_i \mathcal{A}$. Then $\forall_i$, $\mathrm{Red}(\mathbf{t}) \subseteq A_i$, and thus, by $\mathbf{RC}_3$ in $\mathcal{A}_i$, $\mathbf{t} \in \mathcal{A}_i$, which implies $\mathbf{t} \in \cap_i \mathcal{A}_i$.

$\mathbf{RC}_4$: By $\mathbf{RC}_4$, $\forall i$, $\mathbf{0} \in \mathcal{A}_i$, then $\mathbf{0} \in \cap_i \mathcal{A}_i$.

□

D.14 Proof of Corollary 5.4.2

Corollary 5.4.2 (of Lemma 5.2.2)**.** Any type T has a decomposition $T \equiv \sum_{i=1}^{n} \alpha_i.U_i$ such that $\forall j,k,\, U_j \not\equiv U_k$.

Proof. By Lemma 5.2.2, $T \equiv \sum_{i=1}^{n} \alpha_i.U_i$. Assume $\exists j,k$ such that $U_j \equiv U_k$, then notice that $T \equiv (\alpha_j + \alpha_k).U_j + \sum_{\substack{i=1 \\ i\neq j \vee k}}^{n} \alpha_i.U_i.$. Repeat the process until there is no j,k such that $U_j \not\equiv U_k$. □

D.15 Proof of Lemma 5.4.3

First we need a few auxiliary lemmas:

Lemma D.15.1. *Given a (valid) valuation $\rho = (\rho_+, \rho_-)$, for all types T we have $\llbracket T \rrbracket_{\bar{\rho}} \subseteq \llbracket T \rrbracket_{\rho}$.*

Proof. Structural induction on T.

- $T = X$. Then $\llbracket T \rrbracket_{\bar{\rho}} = \rho_-(X) \subseteq \rho_+(X) = \llbracket T \rrbracket_{\rho}$.
- $T = U \to R$. Then $\llbracket U \to R \rrbracket_{\bar{\rho}} = \llbracket U \rrbracket_{\rho} \to \llbracket R \rrbracket_{\bar{\rho}}$. By the induction hypothesis $\llbracket U \rrbracket_{\bar{\rho}} \subseteq \llbracket U \rrbracket_{\rho}$ and $\llbracket R \rrbracket_{\bar{\rho}} \subseteq \llbracket R \rrbracket_{\rho}$. We must show that $\forall \mathbf{t} \in \llbracket U \to R \rrbracket_{\bar{\rho}}$, $\mathbf{t} \in \llbracket U \to R \rrbracket_{\rho}$. Let $\mathbf{t} \in \llbracket U \to R \rrbracket_{\bar{\rho}} = \llbracket U \rrbracket_{\rho} \to \llbracket R \rrbracket_{\bar{\rho}}$. We proceed by induction on the definition of $\to$.
  - Let $\mathbf{t} \in \{\mathbf{r} \,|\, \forall \mathbf{b} \in \llbracket U \rrbracket_{\rho}, (\mathbf{r})\ \mathbf{b} \in \llbracket R \rrbracket_{\bar{\rho}}\}$. Notice that forall $\mathbf{b} \in \llbracket U \rrbracket_{\bar{\rho}}$, $\mathbf{b} \in \llbracket U \rrbracket_{\rho}$, and so $(\mathbf{t})\ \mathbf{b} \in \llbracket R \rrbracket_{\bar{\rho}} \subseteq \llbracket R \rrbracket_{\rho}$. Thus $\mathbf{t} \in \llbracket U \rrbracket_{\bar{\rho}} \to \llbracket R \rrbracket_{\rho} = \llbracket U \to R \rrbracket_{\rho}$.
  - Let $\mathrm{Red}(\mathbf{t}) \in \llbracket U \to R \rrbracket_{\bar{\rho}}$ and $\mathbf{t} \in \mathcal{N}$. By the induction hypothesis $\mathrm{Red}(\mathbf{t}) \in \llbracket U \to R \rrbracket_{\rho}$ and so, by $\mathbf{RC}_3$, $\mathbf{t} \in \llbracket U \to R \rrbracket_{\rho}$.
  - Let $\mathbf{t} = \mathbf{0}$. By $\mathbf{RC}_4$, $\mathbf{0}$ is in any reducibility candidate, in particular it is in $\llbracket U \to R \rrbracket_{\rho}$.
- $T = \forall X.S$, where S is a unit type. Then $\llbracket \forall X.S \rrbracket_{\bar{\rho}} = \cap_{\mathcal{A}\subseteq\mathcal{B}\in\mathcal{RC}} \llbracket S \rrbracket_{\bar{\rho},(X_+,X_-)\mapsto(\mathcal{A},\mathcal{B})}$. By the induction hypothesis $\llbracket S \rrbracket_{\bar{\rho}} \subseteq \llbracket S \rrbracket_{\rho}$, then $\forall \mathcal{A},\mathcal{B}$, $\llbracket S \rrbracket_{\bar{\rho},(X_+,X_-)\mapsto(\mathcal{A},\mathcal{B})} \subseteq \llbracket S \rrbracket_{\rho,(X_+,X_-)\mapsto(\mathcal{A},\mathcal{B})}$. Thus we have $\cap_{\mathcal{A}\subseteq\mathcal{B}\in\mathcal{RC}} \llbracket S \rrbracket_{\bar{\rho},(X_+,X_-)\mapsto(\mathcal{A},\mathcal{B})} \subseteq \cap_{\mathcal{A}\subseteq\mathcal{B}\in\mathcal{RC}} \llbracket S \rrbracket_{\rho,(X_+,X_-)\mapsto(\mathcal{A},\mathcal{B})} = \llbracket \forall X.S \rrbracket_{\rho}$.
- T is not a unit type and $T \equiv \sum_i \alpha_i \cdot U_i$. Then $\llbracket T \rrbracket_{\bar{\rho}} = \sum_i \llbracket U_i \rrbracket_{\bar{\rho}}$. By the induction hypothesis $\llbracket U_i \rrbracket_{\bar{\rho}} \subseteq \llbracket U_i \rrbracket_{\rho}$. We proceed by induction on the definition of $\sum_i \llbracket U_i \rrbracket_{\bar{\rho}}$.
  - Let $\mathbf{t} \in \{\sum_i \alpha_i \cdot \mathbf{r}_i \,|\, \mathbf{r}_i \in \llbracket U_i \rrbracket_{\bar{\rho}}\}$. Note that $\forall \mathbf{r} \in \llbracket U_i \rrbracket_{\bar{\rho}}$, $\mathbf{r} \in \llbracket U_i \rrbracket_{\rho}$ and so the result holds.
  - Let $\mathrm{Red}(\mathbf{t}) \in \sum_i \llbracket U_i \rrbracket_{\bar{\rho}}$ and $\mathbf{t} \in \mathcal{N}$. By the induction hypothesis $\mathrm{Red}(\mathbf{t}) \in \sum_i \llbracket U_i \rrbracket_{\rho}$ and so, by $\mathbf{RC}_3$, $\mathbf{t} \in \sum_i \llbracket U_i \rrbracket_{\rho}$.
  - Let $\mathbf{t} = \mathbf{0}$. By $\mathbf{RC}_4$, $\mathbf{0}$ is in any reducibility candidate.

□

Lemma D.15.2. *Let $\rho = (\rho_+, \rho_-)$ and $\rho' = (\rho'_+, \rho'_-)$ be two valuations such that $\forall X$, $\rho'_-(X) \subseteq \rho_-(X)$ and $\rho_+(X) \subseteq \rho'_+(X)$. Then for any type T we have $\llbracket T \rrbracket_{\rho} \subseteq \llbracket T \rrbracket_{\rho'}$ and $\llbracket T \rrbracket_{\bar{\rho}'} \subseteq \llbracket T \rrbracket_{\bar{\rho}}$.*

Proof. Structural induction on T.

- $T = X$. Then $\llbracket X \rrbracket_{\rho} = \rho_+(X) \subseteq \rho'_+(X) = \llbracket X \rrbracket_{\rho'}$ and $\llbracket X \rrbracket_{\bar{\rho}'} = \rho'_-(X) \subseteq \rho_-(X) = \llbracket X \rrbracket_{\bar{\rho}}$.

- $T = U \to R$. Then $\llbracket U \to R \rrbracket_\rho = \llbracket U \rrbracket_{\bar{\rho}} \to \llbracket R \rrbracket_\rho$ and $\llbracket U \to R \rrbracket_{\bar{\rho}'} = \llbracket U \rrbracket_{\rho'} \to \llbracket R \rrbracket_{\bar{\rho}'}$. By the induction hypothesis $\llbracket U \rrbracket_{\bar{\rho}'} \subseteq \llbracket U \rrbracket_{\bar{\rho}}$, $\llbracket U \rrbracket_\rho \subseteq \llbracket U \rrbracket_{\rho'}$, $\llbracket R \rrbracket_\rho \subseteq \llbracket R \rrbracket_{\rho'}$ and $\llbracket R \rrbracket_{\bar{\rho}'} \subseteq \llbracket R \rrbracket_{\bar{\rho}}$. We proceed by induction on the definition of $\to$ to show that $\forall \mathbf{t} \in \llbracket U \rrbracket_{\bar{\rho}} \to \llbracket R \rrbracket_\rho$, then $\mathbf{t} \in \llbracket U \rrbracket_{\bar{\rho}'} \to \llbracket R \rrbracket_{\rho'} = \llbracket U \to R \rrbracket_{\rho'}$
  - Let $\mathbf{t} \in \{\mathbf{r} \,|\, \forall \mathbf{b} \in \llbracket U \rrbracket_{\bar{\rho}}, (\mathbf{r})\ \mathbf{b} \in \llbracket R \rrbracket_\rho\}$. $\forall \mathbf{b} \in \llbracket U \rrbracket_{\bar{\rho}'}$, $\mathbf{b} \in \llbracket U \rrbracket_{\bar{\rho}}$ and then $(\mathbf{t})\ \mathbf{b} \in \llbracket R \rrbracket_\rho \subseteq \llbracket R \rrbracket_{\rho'}$.
  - Let $\mathrm{Red}(\mathbf{t}) \in \llbracket U \to R \rrbracket_\rho$ and $\mathbf{t} \in \mathcal{N}$. By the induction hypothesis $\mathrm{Red}(\mathbf{t}) \in \llbracket U \to R \rrbracket_{\rho'}$ and so, by $\mathbf{RC}_3$, $\mathbf{t} \in \llbracket U \to R \rrbracket_{\rho'}$.
  - Let $\mathbf{t} = \mathbf{0}$. By $\mathbf{RC}_4$, $\mathbf{0}$ is in any reducibility candidate, in particular it is in $\llbracket U \to R \rrbracket_{\rho'}$.

  Analogously, $\forall \mathbf{t} \in \llbracket U \rrbracket_{\rho'} \to \llbracket R \rrbracket_{\bar{\rho}'}$, $\mathbf{t} \in \llbracket U \rrbracket_\rho \to \llbracket R \rrbracket_{\bar{\rho}} = \llbracket U \to R \rrbracket_\rho$.

- $T = \forall X.S$. Then $\llbracket \forall X.S \rrbracket_\rho = \cap_{\mathcal{A} \subseteq \mathcal{B} \in \mathcal{RC}} \llbracket S \rrbracket_{\rho,(X_+,X_-)\mapsto(\mathcal{A},\mathcal{B})}$. By the induction hypothesis $\llbracket S \rrbracket_\rho \subseteq \llbracket S \rrbracket_{\rho'}$, then $\forall \mathcal{A}, \mathcal{B}$, $\llbracket S \rrbracket_{\rho,(X_+,X_-)\mapsto(\mathcal{A},\mathcal{B})} \subseteq \llbracket S \rrbracket_{\rho',(X_+,X_-)\mapsto(\mathcal{A},\mathcal{B})}$. Thus $\cap_{\mathcal{A} \subseteq \mathcal{B} \in \mathcal{RC}} \llbracket S \rrbracket_{\rho,(X_+,X_-)\mapsto(\mathcal{A},\mathcal{B})} \subseteq \cap_{\mathcal{A} \subseteq \mathcal{B} \in \mathcal{RC}} \llbracket S \rrbracket_{\rho',(X_+,X_-)\mapsto(\mathcal{A},\mathcal{B})} = \llbracket \forall X.S \rrbracket_{\rho'}$. The case $\llbracket \forall X.S \rrbracket_{\bar{\rho}'} \subseteq \llbracket \forall X.S \rrbracket_{\bar{\rho}}$ is completely analogous.

- T is not a unit type and $T \equiv \sum_i \alpha_i \cdot U_i$. Then $\llbracket T \rrbracket_\rho = \sum_i \llbracket U_i \rrbracket_\rho$. By the induction hypothesis $\llbracket U_i \rrbracket_\rho \subseteq \llbracket U_i \rrbracket_{\rho'}$. We proceed by induction on the definition of $\sum_i U_i$ to show that $\sum_i \llbracket U_i \rrbracket_\rho \subseteq \sum_i \llbracket U_i \rrbracket_{\rho'}$.
  - Let $\mathbf{t} \in \{\sum_i \alpha_i \cdot \mathbf{r}_i \,|\, \mathbf{r}_i \in \llbracket U_i \rrbracket_\rho\}$, Notice that $\forall \mathbf{r}_i \in \llbracket U_i \rrbracket_\rho$, $\mathbf{r}_i \in \llbracket U_i \rrbracket_{\rho'}$ and so $\sum_i \alpha_i.\mathbf{r}_i \in \sum_i \llbracket U_i \rrbracket_{\rho'} = \llbracket T \rrbracket_{\rho'}$.
  - Let $\mathrm{Red}(\mathbf{t}) \in \llbracket T \rrbracket_\rho$ and $\mathbf{t} \in \mathcal{N}$. By the induction hypothesis $\mathrm{Red}(\mathbf{t}) \subseteq \llbracket T \rrbracket_{\rho'}$ and so, by $\mathbf{RC}_3$, $\mathbf{t} \in \llbracket T \rrbracket_{\rho'}$.
  - Let $\mathbf{t} = \mathbf{0}$. By $\mathbf{RC}_4$, $\mathbf{0}$ is in any reducibility candidate, in particular it is in $\llbracket T \rrbracket_{\rho'}$.

  The case $\llbracket T \rrbracket_{\bar{\rho}'} \subseteq \llbracket T \rrbracket_{\bar{\rho}}$ is completely analogous.

□

Lemma D.15.3. *For all reducibility candidates $\mathcal{A}$, $\mathcal{A} \subseteq 1.\mathcal{A}$. Moreover, if $\mathbf{b} \in 1.\mathcal{A}$ is a base term, then $\mathbf{b} \in \mathcal{A}$.*

Proof. For all $\mathbf{t} \in \mathcal{A}$, the term $1.\mathbf{t} \in 1 \cdot \mathcal{A}$. Since $1.\mathbf{t} \to \mathbf{t}$, we conclude using $\mathbf{RC}_2$.

Now, consider $\mathbf{b} \in 1.\mathcal{A}$. We proceed by structural induction on $1.\mathcal{A}$.

- Base case: the sum is trivial: the term $\mathbf{b}$ is in $\mathcal{A}$.
- Suppose that $\mathbf{t}$ is in $1.\mathcal{A}$ and that $\mathbf{t} \to \mathbf{b}$, with $\mathbf{b}$ a base term. We prove that $\mathbf{b}$ is in $\mathcal{A}$ by induction on the length of the longest reduction sequence starting from $\mathbf{t}$.
- Suppose that $\mathbf{t}$ is neutral, yet a base term, and that $\mathrm{Red}(\mathbf{t}) \subseteq 1.\mathcal{A}$. Then $\mathbf{t}$ would have to be a lambda-abstraction, which it cannot since it is neutral: we do not consider this case.

□

Lemma D.15.4. *For all reducibility candidates $\{\mathcal{A}_{i,1}\}_{i=1\cdots n_1}$, $\{\mathcal{A}_{i,2}\}_{i=1\cdots n_2}$, if $\mathbf{s} \in \sum_{i=1}^{n_1} \mathcal{A}_{i,k}$ and $\mathbf{t} \in \sum_{i=1}^{n_2} \mathcal{A}_{i,2}$, then $\mathbf{s} + \mathbf{t} \in \sum_{k=1,2,i=1\cdots n_k} \mathcal{A}_{i,k}$.*

Proof. By structural induction on $\sum_{i=1}^{n_1} \mathcal{A}_{i,1}$ and $\sum_{i=1}^{n_2} \mathcal{A}_{i,2}$.

- If $\mathbf{s}$ and $\mathbf{t}$ are respectively of the form $\sum_i \alpha_i.\mathbf{s}_i$ and $\sum_j \beta_j.\mathbf{t}_j$, it is trivial.

- If $\mathbf{s}$ is of that form but $\mathbf{t}$ is neutral such that $\mathrm{Red}(t) \subseteq \sum_i \mathcal{A}_{i,2}$, then using the induction hypothesis, $\mathrm{Red}(\mathbf{s}+\mathbf{t})$ is in $\sum_{k=1,2,i=1\cdots n_k} \mathcal{A}_{i,k}$. We conclude with $\mathbf{RC}_3$.

The other cases are similar. □

Lemma D.15.5. *Suppose that* $\mathbf{s} \in \mathcal{A} \to \mathcal{B}$ *and* $\mathbf{b} \in \mathcal{A}$*, then* $(\mathbf{s})\,\mathbf{b} \in \mathcal{B}$*.*

Proof. Induction on the definition of $\mathcal{A} \to \mathcal{B}$.

- If $\mathbf{s}$ is in $\{\mathbf{s} \mid \forall \mathbf{b} \in \mathcal{A}, (\mathbf{s})\,\mathbf{b} \in \mathcal{B}\}$, then it is trivial
- If $\mathbf{s}$ is in $\mathcal{A} \to \mathcal{B}$ because $\mathbf{r}$ is in $\mathcal{A} \to \mathcal{B}$ and $\mathbf{r} \to \mathbf{s}$, then by the induction hypothesis $\mathbf{b}$ in $\mathcal{A}$ implies $(\mathbf{r})\,\mathbf{b}$ in $\mathcal{B}$, so by $\mathbf{RC}_2$, $(\mathbf{s})\,\mathbf{b}$ is in $\mathcal{B}$.
- If $\mathbf{s}$ in $\mathcal{A} \to \mathcal{B}$ because $\mathbf{s}$ is neutral and $\mathrm{Red}(\mathbf{s}) \subseteq \mathcal{A} \to \mathcal{B}$, we do a case by case analysis on $\mathbf{r}$ in $\mathrm{Red}((\mathbf{s})\,\mathbf{b})$ to show that $(\mathbf{s})\,\mathbf{b}$ is in $\mathcal{B}$ (and so is $(\mathbf{s})\,\mathbf{b}$ by $\mathbf{RC}_3$).
  - $\mathbf{r} = (\mathbf{t})\,\mathbf{b}$, with $\mathbf{t}$ in $\mathrm{Red}(\mathbf{s})$, then by the induction hypothesis, for any $\mathbf{t}$ in $\mathrm{Red}(\mathbf{s})$, $(\mathbf{t})\,\mathbf{b}$ is in $\mathcal{B}$.
  - $\mathbf{r} = (\mathbf{s})\,\mathbf{b}'$ with $\mathbf{b} \to \mathbf{b}'$, then $\mathbf{b}'$ is also a base term in $\mathcal{A}$, so the result holds.
  - The case $\mathbf{s} = \lambda x.\mathbf{t}$ and $\mathbf{r} = \mathbf{t}[\mathbf{b}/x]$ cannot occur because $\mathbf{s}$ has to be neutral by hypothesis.

□

Now we can prove the Adequacy Lemma:
Lemma 5.4.3 (Adequacy Lemma)**.** Every derivable typing judgement is valid: for every valid sequent $\Gamma \vdash \mathbf{t} : T$, we have $\Gamma \models \mathbf{t} : T$.

Proof. We proceed by induction on the size of the typing derivation of $\Gamma \vdash \mathbf{t}:T$. We look at the last typing rule that is used, and show in each case that $\Gamma \models \mathbf{t}:T$, *i.e.* if $T \equiv \sum_{i=1}^n \alpha_i.U_i$ in the sense of Corollary 5.4.2, then $\mathbf{t}_\sigma \in \sum_{i=1}^n [\![U_i]\!]_{\rho,\rho_i}$ for every valuation ρ, set of valuations $\{\rho_i\}_n$, and substitution $\sigma \in [\![\Gamma]\!]_\rho$ (*i.e.* substitution σ such that $(x:V) \in \Gamma$ implies $x_\sigma \in [\![V]\!]_{\bar\rho}$).

1. $\dfrac{}{\Gamma, x:U \vdash x:U}\, ax$ — Then for any ρ, $\forall \sigma \in [\![\Gamma, x:U]\!]_\rho$ by definition we have $x_\sigma \in [\![U]\!]_{\bar\rho,\emptyset}$. From Lemma D.15.1, we deduce that $x_\sigma \in [\![U]\!]_{\rho,\emptyset}$, *i.e.* $x_\sigma \in [\![1.U]\!]_{\rho,\emptyset}$ by Lemma D.15.3.
2. $\dfrac{\Gamma \vdash \mathbf{t}:T}{\Gamma \vdash \mathbf{0}:0.T}\, 0_I$ — Note that $\forall \sigma$, $\mathbf{0}_\sigma = \mathbf{0}$, and $\mathbf{0}$ is in any reducibility candidate by $\mathbf{RC}_4$.
3. $\dfrac{\Gamma, x:U \vdash \mathbf{t}:T}{\Gamma \vdash \lambda x.\mathbf{t}:U \to T} \to_I$ — Let $T \equiv \sum_{i=1}^n \alpha_i.U_i$. Then by the induction hypothesis, for any ρ, set $\{\rho_i\}_n$ not acting on $FV(\Gamma)$, and $\forall \sigma \in [\![\Gamma, x:U]\!]_\rho$, we have $\mathbf{t}_\sigma \in \sum_{i=1}^n [\![U_i]\!]_{\rho,\rho_i}$. Then by Lemma D.15.3 it is enough to prove that $\forall \sigma \in [\![\Gamma]\!]_\rho$, $(\lambda x.\mathbf{t})_\sigma \in [\![U \to T]\!]_{\rho,\rho'}$, or what is the same $\lambda x.\mathbf{t}_\sigma \in [\![U]\!]_{\bar\rho,\bar\rho'} \to [\![T]\!]_{\rho,\rho'}$, where ρ' does not act on $FV(\Gamma)$. If we can show that $\mathbf{b} \in [\![U]\!]_{\bar\rho,\bar\rho'}$ implies $(\lambda x.\mathbf{t}_\sigma)\ \mathbf{b} \in [\![T]\!]_{\rho,\rho'}$, then we are done. Take $\mathbf{b} \in [\![U]\!]_{\bar\rho,\bar\rho'}$ and write $\sigma' = \sigma; x \mapsto \mathbf{b}$. Then $\sigma' \in [\![\Gamma, x:U]\!]_{\rho,\rho'}$, thus $\mathbf{t}_{\sigma'} \in \sum_{i=1}^n [\![U_i]\!]_{\rho,\rho',\emptyset}$, so $\mathbf{t}_{\sigma'}$ is strongly normalising, and we show by induction that $(\lambda x.\mathbf{t}_\sigma)\ \mathbf{b} \in [\![T]\!]_{\rho,\rho'} = \sum_{i=1}^n [\![U_i]\!]_{\rho,\rho',\emptyset}$. Since it is a neutral term, we just need to prove that every one-step reduction of it is in $[\![T]\!]_\rho$, which by $\mathbf{RC}_2$ closes the case.

Structural induction on the reduct of $(\lambda x.\mathbf{t}_\sigma)\ \mathbf{b}$:

$(\lambda x.\mathbf{t}_\sigma)\ \mathbf{b} \to (\lambda x.\mathbf{t}_\sigma)\ \mathbf{b}'$ with $\mathbf{b} \to \mathbf{b}'$. Then $\mathbf{b}' \in [\![U]\!]_{\bar{\rho},\bar{\rho}'}$ and we close by induction hypothesis.

$(\lambda x.\mathbf{t}_\sigma)\ \mathbf{b} \to (\lambda x.\mathbf{t}')\ \mathbf{b}$ with $\mathbf{t}_\sigma \to \mathbf{t}$. Since $\mathbf{t}_\sigma \in \sum_{i=1}^n [\![U_i]\!]_{\rho,\rho_i}$ for any $\{\rho_i\}_n$ not acting on $FV(\Gamma)$, take $\forall i,\ \rho_i = \rho'$, so $\mathbf{t}_\sigma \in [\![T]\!]_{\rho,\rho'}$ and so are its reducts, such as $\mathbf{t}'$. We close by induction hypothesis.

$(\lambda x.\mathbf{t}_\sigma)\ \mathbf{b} \to \mathbf{t}_\sigma[\mathbf{b}/x]$ notice that $\mathbf{t}_\sigma[\mathbf{b}/x] = \mathbf{t}_{\sigma'} \in [\![T]\!]_{\rho,\rho_i}$.

4. $$\frac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^n \alpha_i.\forall \vec{X}.(U \to T_i) \qquad \Gamma \vdash \mathbf{r} : \sum_{j=1}^m \beta_j.V_j \qquad \forall V_j,\ \exists \vec{W_j},\ / U[\vec{W_j}/\vec{X}] = V_j}{\Gamma \vdash (\mathbf{t})\ \mathbf{r} : \sum_{i=1}^n \sum_{j=1}^m \alpha_i \times \beta_j.T_i[\vec{W_j}/\vec{X}]} \to_E$$

Without loss of generality, assume each T_i is different from each other, and the same for V_j. By the induction hypothesis, for any ρ, $\{\rho_{i,j}\}_{n,m}$ not acting on $FV(\Gamma)$, and $\forall \sigma \in [\![\Gamma]\!]_\rho$ we have $\mathbf{t}_\sigma \in \sum_{i=1}^n \cap_{\vec{\mathcal{A}} \subseteq \vec{\mathcal{B}} \in \mathcal{RC}} [\![(U \to T_i)]\!]_{\rho,\rho_i,(\vec{X}_+,\vec{X}_-)\mapsto(\vec{\mathcal{A}},\vec{\mathcal{B}})}$ and $\mathbf{r}_\sigma \in \sum_{j=1}^m [\![V_j]\!]_{\rho,\rho_j}$.

For all i,j, let $T_i[\vec{W_j}/\vec{X}] \equiv \sum_{k=1}^{r^{ij}} \delta_k^{ij}.W_k^{ij}$. We want to show that for any ρ, sets $\{\rho'_{i,j,k}\}_{r_{i,j}}$ not acting on $FV(\Gamma)$ and $\forall \sigma \in [\![\Gamma]\!]_\rho$, $((\mathbf{t})\ \mathbf{r})_\sigma \in \sum_{i=1\cdots n,j=1\cdots m,k=1\cdots r^{ij}} [\![W_k^{ij}]\!]_{\rho,\rho_{ijk}}$ Since both $\mathbf{t}_\sigma$ and $\mathbf{r}_\sigma$ are strongly normalising, we proceed by induction on the sum of the lengths of their rewrite sequence. The set $\mathrm{Red}(((\mathbf{t})\ \mathbf{r})_\sigma)$ contains:

- $(\mathbf{t}_\sigma)\ \mathbf{r}'$ or $(\mathbf{t}')\ \mathbf{r}_\sigma$ when $\mathbf{t}_\sigma \to \mathbf{t}'$ or $\mathbf{r}_\sigma \to \mathbf{r}'$. By $\mathbf{RC}_2$, the term $\mathbf{t}'$ is in the reducibility candidate $\sum_{i=1}^n \cap_{\vec{\mathcal{A}} \subseteq \vec{\mathcal{B}} \in \mathcal{RC}} [\![(U \to T_i)]\!]_{\rho,\rho_i,(\vec{X}_+,\vec{X}_-)\mapsto(\vec{\mathcal{A}},\vec{\mathcal{B}})}$ and $\mathbf{r}'$ is in $\sum_{j=1}^m [\![V_j]\!]_{\rho,\rho_j}$. We conclude by induction hypothesis.
- A term coming from one of the rewrite rules of Group A. We conclude by noting that we obtain a linear combination of terms smaller than the original one. We can conclude with the induction hypothesis, Lemma 5.3.15 and the definition of the sum of reducibility candidates.
- The term $\mathbf{t}'_\sigma[\mathbf{r}_\sigma/x]$, when $\mathbf{t}_\sigma = \lambda x.\mathbf{t}'$ and $\mathbf{r}$ is a base term. Note that this term is of the form $\mathbf{t}'_{\sigma'}$ where $\sigma' = \sigma; x \mapsto \mathbf{r}$. We are in the situation where the types of $\mathbf{t}$ and $\mathbf{r}$ are respectively $\forall \vec{X}.(U \to T)$ and V, and there is $\vec{W}$ such that $U[\vec{W}/\vec{X}] = V$, and so $\sum_{i,j,k} [\![W_k^{ij}]\!]_{\rho,\rho_{ijk}} = \sum_{k=1}^r [\![W_k]\!]_{\rho,\rho_k}$, where we omit the index "11". Note that (using Lemma D.15.3)

$$\lambda x.\mathbf{t}'_\sigma \in [\![\forall \vec{X}.(U \to T)]\!]_{\rho,\rho'} = \cap_{\vec{\mathcal{A}} \subseteq \vec{\mathcal{B}} \in \mathcal{RC}} [\![U \to T]\!]_{\rho,\rho',(\vec{X}_+,\vec{X}_-)\mapsto(\vec{A},\vec{B})}$$

for all possible ρ' such that $|\rho'|$ does not intersect $FV(\Gamma)$. Choose $\vec{\mathcal{A}}$ and $\vec{\mathcal{B}}$ equal to $[\![\vec{W}]\!]_{\rho,\rho'}$ and choose ρ'_- to send every X in its domain to $\cap_k \rho_{k-}(X)$ and ρ'_+ to send all the X in its domain to $\sum_k \rho_{k+}(X)$. Then

$$\lambda x.\mathbf{t}'_\sigma \in [\![U \to T]\!]_{\rho,\rho',\vec{X}\mapsto\vec{W}} = [\![V]\!]_{\rho,\rho'} \to [\![T]\!]_{\rho,\rho',\vec{X}\mapsto\vec{W}}.$$

The valuation $\vec{X} \mapsto \vec{W}$ is a shortcut for $(X_+, X_-) \mapsto ([\![W]\!]_{\bar{\rho},\bar{\rho}'}, [\![W]\!]_{\rho,\rho'})$. For all base terms $\mathbf{b} \in 1.[\![V]\!]_{\bar{\rho},\bar{\rho}'}$, by Lemma D.15.3 they are in $[\![V]\!]_{\bar{\rho},\bar{\rho}'}$ and we have by Lemma D.15.5

$$(\lambda x.\mathbf{t}_\sigma)\ \mathbf{b} \in [\![T]\!]_{\rho,\rho',\vec{X}\mapsto\vec{W}} = [\![T[\vec{W}/\vec{X}]]\!]_{\rho,\rho'} = \left[\!\!\left[\sum_{k=1}^r \delta_k.W_k\right]\!\!\right]_{\rho,\rho'} = \sum_{k=1}^n [\![W_k]\!]_{\rho,\rho'}.$$

Now, from Lemma D.15.2, for all k we have $[\![W_k]\!]_{\rho,\rho'} \subseteq [\![W_k]\!]_{\rho,\rho_k}$. Therefore

$$(\lambda x.\mathbf{t}_\sigma)\ \mathbf{b} \in \sum_{k=1}^n [\![W_k]\!]_{\rho,\rho_k}$$

Since $\mathbf{r} \in 1.\llbracket V \rrbracket_{\bar{\rho},\bar{\rho}'}$, we are done.

Since the set $\mathrm{Red}(((\mathbf{t})\ \mathbf{r})_\sigma) \subseteq \sum_{i=1\cdots n, j=1\cdots m, k=1\cdots r^{ij}} \llbracket W_k^{ij} \rrbracket_{\rho,\rho_{ijk}}$, we can conclude by $\mathbf{RC}_3$.

5. $$\frac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.U_i \quad X \notin FV(\Gamma)}{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i}\ \forall_I$$

By the induction hypothesis, for any ρ, set $\{\rho_i\}_n$ not acting on $FV(\Gamma)$, we have $\forall \sigma \in \llbracket \Gamma \rrbracket_\rho$, $\mathbf{t}_\sigma \in \sum_{i=1}^n \llbracket U_i \rrbracket_{\rho,\rho_i}$. Since $X \notin FV(\Gamma)$, we can take $\rho_i = \rho_i', (X_+, X_-) \mapsto (\mathcal{A}, \mathcal{B})$, then for any $\mathcal{A} \subseteq \mathcal{B}$, $\mathbf{t}_\sigma \in \sum_{i=1}^n \llbracket U_i \rrbracket_{\rho,\rho_i',(X_+,X_-)\mapsto(\mathcal{A},\mathcal{B})}$. Since it is valid for any $\mathcal{A} \subseteq \mathcal{B}$, we can take the intersections, thus we have $\mathbf{t}_\sigma \in \sum_{i=1}^n \cap_{\mathcal{A}\subseteq\mathcal{B}\in\mathcal{RC}} \llbracket U_i \rrbracket_{\rho,\rho_i',(X_+,X_-)\mapsto(\mathcal{A},\mathcal{B})} = \sum_{i=1}^n \llbracket \forall X.U_i \rrbracket_{\rho,\rho_i'}$.

6. $$\frac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i}{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.U_i[V/X]}\ \forall_E$$

By the induction hypothesis, for any ρ and $\{\rho_i\}_n$, we have $\forall \sigma \in \llbracket \Gamma \rrbracket_\rho$, $\mathbf{t}_\sigma \in \sum_{i=1}^n \llbracket \forall X.U_i \rrbracket_{\rho,\rho_i} = \sum_{i=1}^n \cap_{\mathcal{A}\subseteq\mathcal{B}\in\mathcal{RC}} \llbracket U_i \rrbracket_{\rho,\rho_i',(X_+,X_-)\mapsto(\mathcal{A},\mathcal{B})}$. Since it is in the intersections, we can chose $\mathcal{A} = \llbracket V \rrbracket_{\bar{\rho},\bar{\rho}_i}$ and $\mathcal{B} = \llbracket V \rrbracket_{\rho,\rho_i}$, and then $\mathbf{t}_\sigma$ will be in those particular sets, so $\mathbf{t}_\sigma \in \sum_{i=1}^n \llbracket U_i \rrbracket_{\rho,\rho_i',X\mapsto V} = \sum_{i=1}^n \llbracket U_i[V/X] \rrbracket_{\rho,\rho_i'}$.

7. $$\frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \alpha.T}\ \alpha_I$$

Let $T \equiv \sum_{i=1}^n \beta_i.U_i$, so $\alpha.T \equiv \sum_{i=1}^n \alpha \times \beta_i.U_i$. By the induction hypothesis, for any ρ, we have $\forall \sigma \in \llbracket \Gamma \rrbracket_\rho$, $\mathbf{t}_\sigma \in \sum_{i=1}^n \llbracket U_i \rrbracket_{\rho,\rho_i}$. Note that $(\alpha.\mathbf{t})_\sigma = \alpha.\mathbf{t}_\sigma \in \sum_{i=1}^n \llbracket U_i \rrbracket_{\rho,\rho_i}$.

8. $$\frac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{r} : R}{\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R}\ +_I$$

Let $T \equiv \sum_{i=1}^n \alpha_i.U_{i,1}$ and $R \equiv \sum_{j=1}^m \beta_j.U_{j,2}$. By the induction hypothesis, for any ρ, $\{\rho_i\}_n$, $\{\rho_j'\}_m$, we have $\forall \sigma \in \llbracket \Gamma \rrbracket_\rho$, $\mathbf{t}_\sigma \in \sum_{i=1}^n \llbracket U_{i,1} \rrbracket_{\rho,\rho_i}$ and $\mathbf{r}_\sigma \in \sum_{j=1}^m \llbracket U_{j,2} \rrbracket_{\rho,\rho'_j}$. Then by Lemma D.15.4, $(\mathbf{t} + \mathbf{r})_\sigma = \mathbf{t}_\sigma + \mathbf{r}_\sigma \in \sum_{i,k} \llbracket U_{i,k} \rrbracket_{\rho,\rho_i}$.

9. $$\frac{\Gamma \vdash \mathbf{t} : T \qquad T \equiv R}{\Gamma \vdash \mathbf{t} : R}\ \equiv$$

Let $T \equiv \sum_{i=1}^n \alpha_i.U_i$ in the sense of Corollary 5.4.2, then since $T \equiv R$, R is also equivalent to $\sum_{i=1}^n \alpha_i.U_i$, so $\Gamma \vDash \mathbf{t} : T \Rightarrow \Gamma \vDash \mathbf{t} : R$.

□

Appendix E

Proofs from Chapter 6

E.1 Proof of Theorem 6.2.2

Theorem 6.2.2 (Subject Reduction)**.** For any terms $\mathbf{t}$ and $\mathbf{t}'$, context Γ and type T, For any terms $\mathbf{t}$ and $\mathbf{t}'$, context Γ and type T, if $\mathbf{t} \to \mathbf{t}'$ and $\Gamma \vdash \mathbf{t} : T$ then there exists some type R such that $\Gamma \vdash \mathbf{t}' : R$ and $T \preccurlyeq R$.

Proof. We proceed by induction on $\mathbf{t} \to \mathbf{t}'$.

Elementary rules

Rule $\mathbf{u} + \mathbf{0} \to \mathbf{u}$. Let $\Gamma \vdash \mathbf{u} + \mathbf{0} : T$. Then by Lemma 6.2.4(3), there exist types R, S such that $\Gamma \vdash \mathbf{u} : R$ and $\Gamma \vdash \mathbf{0} : S$, with $R + S \equiv T$. By Remark 6.2.7, we have $S \equiv \overline{0}$, therefore $R + \overline{0} \equiv R \equiv T$, and, by rule $\equiv$, we conclude $\Gamma \vdash \mathbf{u} : T$.

Rule $0.\mathbf{u} \to \mathbf{0}$. Let $\Gamma \vdash 0.\mathbf{u} : T$, then by Lemma 6.2.4(4) we have $T \equiv \overline{0}$. By rule $ax_{\overline{0}}$ we have $\Gamma \vdash \mathbf{0} : \overline{0}$.

Rule $1.\mathbf{u} \to \mathbf{u}$. Let $\Gamma \vdash 1.\mathbf{u} : T$, then by Lemma 6.2.4(4), there exists R such that $\Gamma \vdash \mathbf{u} : R$ and $R \equiv T$. Therefore, by rule $\equiv$ we have $\Gamma \vdash \mathbf{u} : T$.

Rule $\alpha.\mathbf{0} \to \mathbf{0}$. Let $\Gamma \vdash \alpha.\mathbf{0} : T$, then by Lemma 6.2.4(4) we have $\Gamma \vdash \mathbf{0} : R$ with $\lfloor \alpha \rfloor . R \equiv T$. However, by Remark 6.2.7, we have $R \equiv \overline{0}$, so $\overline{0} \equiv T$. By rule $\equiv$ we conclude $\Gamma \vdash \mathbf{0} : T$.

Rule $\alpha.(\beta.\mathbf{u}) \to (\alpha \times \beta).\mathbf{u}$. True by Lemma 6.2.9.

Rule $\alpha.(\mathbf{u} + \mathbf{t}) \to \alpha.\mathbf{u} + \alpha.\mathbf{t}$. True by Lemma 6.2.10.

Factorisation rules

Rule $\alpha.\mathbf{u} + \beta.\mathbf{u} \to (\alpha + \beta).\mathbf{u}$. True by Lemma 6.2.11.

Rule $\alpha.\mathbf{u} + \mathbf{u} \to (\alpha + 1).\mathbf{u}$. Let $\Gamma \vdash \alpha.\mathbf{u} + \mathbf{u} : T$, then by rule s_I, $\Gamma \vdash 1.(\alpha.\mathbf{u} + \mathbf{u}) : 1.T$. So, by Lemma 6.2.10, $\Gamma \vdash 1.\alpha.\mathbf{u} + 1.\mathbf{u} : R_1$ with $1.T \preccurlyeq R_1$. By Lemma 6.2.4(3), there exist types R, S such that $\Gamma \vdash 1.\alpha.\mathbf{u} : R$ and $\Gamma \vdash 1.\mathbf{u} : S$, with $R + S \equiv R_1$. Then, by Lemma 6.2.9, we have $\Gamma \vdash \alpha.\mathbf{u} : R$, so with rule $+_I$ we can conclude $\Gamma \vdash \alpha.\mathbf{u} + 1.\mathbf{u} : R + S$ and then $\Gamma \vdash \alpha.\mathbf{u} + 1.\mathbf{u} : R_1$ by rule $\equiv$. Finally, by Lemma 6.2.11, we conclude $\Gamma \vdash (\alpha + 1).\mathbf{u} : R_2$, with $R_1 \preccurlyeq R_2$. Notice that $T \equiv 1.T \preccurlyeq R_1 \preccurlyeq R_2$.

Rule $\mathbf{u}+\mathbf{u} \to (1+1).\mathbf{u}$. Let $\Gamma \vdash \mathbf{u}+\mathbf{u} : T$. By rule s_I, $\Gamma \vdash 1.(\mathbf{u}+\mathbf{u}) : 1.T$. So by Lemma 6.2.10, we have $\Gamma \vdash 1.\mathbf{u} + 1.\mathbf{u} : R_1$ with $T \preccurlyeq R_1$. Then by Lemma 6.2.11, $\Gamma \vdash (1+1).\mathbf{u} : R_2$ with $R_1 \preccurlyeq R_2$, hence $T \preccurlyeq R_2$.

Application rules

Rule $(\mathbf{u}+\mathbf{r})\ \mathbf{t} \to (\mathbf{u})\ \mathbf{t} + (\mathbf{r})\ \mathbf{t}$. Let $\Gamma \vdash (\mathbf{u}+\mathbf{r})\ \mathbf{t} : T$. Then, by Lemma 6.2.4(1) there exist natural numbers n and m and types $U, T_1, \ldots, T_n$ such that $\Gamma \vdash \mathbf{u}+\mathbf{r} : \sum_{i=1}^{n}(U \to T_i)$ and $\Gamma \vdash \mathbf{t} : m.U$, with $\sum_{i=1}^{n} m.T_i \equiv T$. Then by Lemma 6.2.4(3), there exist R and S such that $\Gamma \vdash \mathbf{u} : R$ and $\Gamma \vdash \mathbf{r} : S$, with $R + S \equiv \sum_{i=1}^{n}(U \to T_i)$. Furthermore, by Remark 4.1.2, there exist natural numbers k and p and types $V_1, \ldots, V_k, W_1, \ldots, W_p$ such that $R \equiv \sum_{j=1}^{k} V_j$, $S \equiv \sum_{k=1}^{p} W_k$, where these sums are minimal, *i.e.* $W_i \neq \overline{0}$ and $V_i \neq \overline{0}$ for all i. Due to the shape of $\sum_{i=1}^{n}(U \to T_i)$, we have that for each $j = 1, \ldots, k$, there is some i_j such that $V_j \equiv U \to T_{i_j}$. Let $A \subset \{1, \ldots, n\}$ such that $i_j \in A$ for all j. Analogously, for each $k = 1, \ldots, p$, there is some i_k such that $W_k \equiv U \to T_{i_k}$. Therefore, $R \equiv \sum_{j=1}^{k} V_j \equiv \sum_{j=1}^{k} U \to T_{i_j} \equiv \sum_{i\in A} U \to T_i$ and $S \equiv \sum_{k=1}^{p} W_k \equiv \sum_{k=1}^{p} U \to T_{i_k} \equiv \sum_{k\in \bar{A}} U \to T_k$. So by rule $\equiv$, we have $\Gamma \vdash \mathbf{u} : \sum_{i\in A} U \to T_i$ and $\Gamma \vdash \mathbf{r} : \sum_{k\in \bar{A}} U \to T_k$. Hence, using rule $\to_E$ we can derive $\Gamma \vdash (\mathbf{u})\ \mathbf{t} : \sum_{i\in A} m.T_i$ and $\Gamma \vdash (\mathbf{r})\ \mathbf{t} : \sum_{k\in \bar{A}} m.T_k$, from where, using $+_I$, can be derived $\Gamma \vdash (\mathbf{u})\ \mathbf{t} + (\mathbf{r})\ \mathbf{t} : \sum_{i\in A} m.T_i + \sum_{k\in \bar{A}} m.T_k$. Note that $\sum_{i\in A} m.T_i + \sum_{k\in \bar{A}} m.T_k \equiv \sum_{i=1}^{n} m.T_i \equiv T$. Then by rule $\equiv$ we have $\Gamma \vdash (\mathbf{u})\ \mathbf{t} + (\mathbf{r})\ \mathbf{t} : T$.

Rule $\mathbf{t}\ (\mathbf{u}+\mathbf{r}) \to (\mathbf{t})\ \mathbf{u} + (\mathbf{t})\ \mathbf{r}$. Let $\Gamma \vdash \mathbf{t}\ (\mathbf{u}+\mathbf{r}) : T$. Then by Lemma 6.2.4(1), there exist natural numbers n and m and types $U, T_1, \ldots, T_n$ such that $\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n}(U \to T_i)$ and $\Gamma \vdash \mathbf{u}+\mathbf{r} : m.U$, with $\sum_{i=1}^{n}(m.T_i) \equiv T$. Then by Lemma 6.2.4(3), there exist types R, S such that $\Gamma \vdash \mathbf{u} : R$ and $\Gamma \vdash \mathbf{r} : S$, with $R + S \equiv m.U \equiv \sum_{i=1}^{m} U$. We know $m = k + p$, where $R \equiv k.U$ and $S \equiv p.U$. Then, by rule $\equiv$, we have $\Gamma \vdash \mathbf{u} : k.U$ and $\Gamma \vdash \mathbf{r} : p.U$. Therefore, by rules $\to_E$ and $+_I$, we can derive $\Gamma \vdash (\mathbf{t})\ \mathbf{u} + (\mathbf{t})\ \mathbf{r} : \sum_{i=1}^{n} k.T_i + \sum_{i=1}^{n} p.T_i$. Note that $\sum_{i=1}^{n} k.T_i + \sum_{i=1}^{n} p.T_i \equiv \sum_{i=1}^{n}(k.T_i + p.T_i) \equiv \sum_{i=1}^{n} m.T_i \equiv T$. Hence by rule $\equiv$ we have $\Gamma \vdash (\mathbf{t})\ \mathbf{u} + (\mathbf{t})\ \mathbf{r} : T$.

Rule $(\mathbf{0})\ \mathbf{u} \to \mathbf{0}$. True by Lemma 6.2.8 and rule $ax_{\overline{0}}$.

Rule $(\mathbf{u})\ \mathbf{0} \to \mathbf{0}$. True by Lemma 6.2.8 and rule $ax_{\overline{0}}$.

Beta reduction

Rule $(\lambda x : U.\mathbf{t})\ \mathbf{b} \to \mathbf{t}[\mathbf{b}/x]$. Let $\Gamma \vdash (\lambda x : U.\mathbf{t})\ \mathbf{b} : T$. Then, by Lemma 6.2.4(1) there exist natural numbers n, m and types $T_1, \ldots, T_n$ such that $\Gamma \vdash \lambda x : U.\mathbf{t} : \sum_{i=1}^{n} U \to T_i$ and $\Gamma \vdash \mathbf{b} : m.U$, with $\sum_{i=1}^{n} m.T_i \equiv T$. In addition, by Lemma 6.2.12, we have $\Gamma \vdash \lambda x : U.\mathbf{t} : U \to T_1$ and $\Gamma \vdash \mathbf{b} : U$. Then by Corollary 6.2.5, we have $\Gamma \vdash [\Gamma, x : U]\mathbf{t} : T_1$. Therefore, by Lemma 6.2.6, $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T_1$. Finally, since $T_1 \equiv T$, by rule $\equiv$ we have $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T$.

Rule $(\Lambda X.\mathbf{t})@U \to \mathbf{t}[U/X]$. Let $\Gamma \vdash (\Lambda X.\mathbf{t})@U : T$. Then, by Lemma 6.2.4(6) there exists a type R such that $\Gamma \vdash \Lambda X.\mathbf{t} : \forall X.R$ and $R[U/X] \equiv T$. Moreover, by Lemma 6.2.4(5) there exists a type S such that $\Gamma \vdash \mathbf{t} : S$, where $X \notin \mathrm{FV}(\Gamma)$ and $\forall X.S \equiv \forall X.R$. By Lemma 6.2.6, $\Gamma[U/X] \vdash \mathbf{t}[U/X] : S[U/X]$ and since $X \notin \mathrm{FV}(\Gamma)$ we have $\Gamma \vdash \mathbf{t}[U/X] : S[U/X]$. Finally, since $S \equiv R$ and $R[U/X] \equiv T$, we can use rule $\equiv$ to conclude $\Gamma \vdash \mathbf{t}[U/X] : T$.

AC equivalences

Commutativity of $+$. We must prove that $\mathbf{u}+\mathbf{r}$ and $\mathbf{r}+\mathbf{u}$ have equivalent types. Let $\Gamma \vdash \mathbf{u}+\mathbf{r}:T$. Then, by Lemma 6.2.4(3) there exist R, S such that $\Gamma \vdash \mathbf{u}:R$ and $\Gamma \vdash \mathbf{r}:S$, with $R+S \equiv T$. Then using $+_I$, one gets $\Gamma \vdash \mathbf{r}+\mathbf{u}:S+R$. Note that $S+R \equiv R+S \equiv T$, then by rule $\equiv$ we have $\Gamma \vdash \mathbf{r}+\mathbf{u}:T$.

Associativity of $+$ We must prove that $(\mathbf{u}+\mathbf{r})+\mathbf{t}$ and $\mathbf{u}+(\mathbf{r}+\mathbf{t})$ have equivalent types. Let $\Gamma \vdash (\mathbf{u}+\mathbf{r})+\mathbf{t}:T$. Then, by Lemma 6.2.4(3), there exist R_1, R_2 such that $\Gamma \vdash \mathbf{u}+\mathbf{r}:R_1$ and $\Gamma \vdash \mathbf{t}:R_2$, with $R_1+R_2 \equiv T$. Then by Lemma 6.2.4(3) again, there exist S_1, S_2 such that $\Gamma \vdash \mathbf{u}:S_1$ and $\Gamma \vdash \mathbf{r}:S_2$, with $S_1+S_2 \equiv R_1$. Therefore, using $+_I$ twice, in the correct order, we can derive $\Gamma \vdash \mathbf{u}+(\mathbf{r}+\mathbf{t}):S_1+(S_2+R_2)$. Note that $S_1+(S_2+R_2) \equiv (S_1+S_2)+R_2 \equiv R_1+R_2 \equiv T$, so by rule $\equiv$, we have $\Gamma \vdash \mathbf{u}+(\mathbf{r}+\mathbf{t}):T$.

Contextual rules The inductive cases are applications of the context rules, which trivially follows by induction.

□

E.2 Proof of Lemma 6.3.2

Lemma 6.3.2. If $\Gamma \vdash \mathbf{t}:T$, then $\exists \Delta, R$ such that $\Delta \vdash_v |\mathbf{t}|:R$.

Proof. First we define a translation from a subset of types in *Vectorial* to types in λ^{CA}. Consider the subset of types of *Vectorial*, where scalars range over non-negative real numbers. Then the translation has this subset as domain:

$$|X| = X \qquad |U \to T| = |U| \to |T| \qquad |\forall X.U| = \forall X.|U| \qquad |\alpha.T| = \lfloor \alpha \rfloor .|T| \qquad |T+R| = |T|+|R|$$

where $n.T = \sum_{i=1}^{n} T$ with the convention that $0.T = \overline{0}$. We also extend this definition to contexts: $|\Gamma| = \{x : |U| \mid U \in \Gamma\}$.

Then we prove by induction on the length of the type derivation of $\Gamma \vdash \mathbf{t}:T$ that $|\Gamma| \vdash_v |\mathbf{t}|:R$ with $|R| \equiv T$.

1. $\dfrac{}{\Gamma, x:U \vdash x:U}\, ax$ Then by rule ax in *Vectorial*, $|\Gamma|, x:|U| \vdash_v x:|U|$. Notice that $|x| = x$.

2. $\dfrac{}{\Gamma \vdash \mathbf{0}:\overline{0}}\, ax$ Then take any type T and term $\mathbf{t}$ such that $|\Gamma| \vdash_v \mathbf{t}:T$, and so by rule 0_I, $|\Gamma| \vdash_v \mathbf{0}:0.T$. Notice that $|\mathbf{0}| = \mathbf{0}$ and $|0.T| = \overline{0}$.

3. $\dfrac{\Gamma \vdash \mathbf{t}: \sum_{i=1}^{n} U \to T_i \qquad \Gamma \vdash \mathbf{r}:m.U}{\Gamma \vdash (\mathbf{t})\ \mathbf{r}: \sum_{i=1}^{n} m.T_i}\, \to_E$

   Then by the induction hypothesis, $\exists R, S$ such that $|\Gamma| \vdash_v |\mathbf{t}|:R$ and $|\Gamma| \vdash_v |\mathbf{r}|:S$, with $|R| \equiv \sum_{i=1}^{n} U \to T_i$ and $|S| \equiv m.U$. Notice that the translation of types from *Vectorial* to types of λ^{CA} only acts on the scalars by converting them on sums. The rest of the structure of the type remains untouched. Then we can easily prove using structural induction that $R \equiv \sum_{i=1}^{n} \alpha_i.U' \to T_i'$ and $S \equiv \beta.U'$ such that $\lfloor \beta \rfloor = m$, $|U'| \equiv U$ and $\forall i$, $|T_i'| \equiv T_i$. Then using rules $\equiv$ and $\to_E$ in *Vectorial*, we can derive $|\Gamma| \vdash_v (|\mathbf{t}|)\ |\mathbf{r}|: \sum_{i=1}^{n} \beta.T_i'$. Notice that $|(\mathbf{t})\ \mathbf{r}| = (|\mathbf{t}|)\ |\mathbf{r}|$ and $|\sum_{i=1}^{n} \beta.T_i'| \equiv \sum_{i=1}^{n} m.T_i$.

4. $\dfrac{\Gamma, x:U \vdash \mathbf{t}:T}{\Gamma \vdash \lambda x:U.\mathbf{t}:U \to T}\ {\to_I}$ Then by the induction hypothesis, exists R such that $|R| \equiv T$ and $|\Gamma|, x:|U| \vdash_v |\mathbf{t}|:R$. Then using rule $\to_I$ in *Vectorial*, we can derive $|\Gamma| \vdash_v \lambda x.|\mathbf{t}|:|U| \to R$. Notice that $|\lambda x:U.\mathbf{t}| = \lambda x.|\mathbf{t}|$ and $||U| \to R| = ||U|| \to |R|$. It is easy to check that if we take a type in λ^{CA} and do its translation using $|\cdot|$, we obtain the same type. So $||U|| = |U| = U$. Then $||U|| \to |R| \equiv U \to T$.

5. $\dfrac{\Gamma \vdash \mathbf{t}:\forall X.U}{\Gamma \vdash \mathbf{t}@V:U[V/X]}\ {\forall_E}$ Then by the induction hypothesis $\exists R$ such that $|\Gamma| \vdash_v |\mathbf{t}|:R$, with $|R| \equiv \forall X.U$. Then $R \equiv \alpha.\forall X.U'$ where $\lfloor\alpha\rfloor = 1$ and $|U'| = U$. So, using rules $\equiv$ and $\forall_E$ in *Vectorial*, we can derive $|\Gamma| \vdash_v |\mathbf{t}|:\alpha.U'[V/X]$. Notice that $|\mathbf{t}@V| = |\mathbf{t}|$ and since $|V| = V$, $|\alpha.U'[V/X]| \equiv U[V/X]$.

6. $\dfrac{\Gamma \vdash \mathbf{t}:U \quad X \notin FV(\Gamma)}{\Gamma \vdash \Lambda X.\mathbf{t}:\forall X.U}\ {\forall_I}$ Then by the induction hypothesis $\exists R$ such that $|\Gamma| \vdash_v |\mathbf{t}|:R$, with $|R| \equiv U$, so $R \equiv \alpha.U'$ with $\lfloor\alpha\rfloor = 1$ and $|U'| = U$. Notice also that $X \notin FV(|\Gamma|)$. Thus, using rules $\equiv$ and $\forall_I$ in *Vectorial*, we can derive $|\Gamma| \vdash_v |\mathbf{t}|:\alpha.\forall X.U'$. Notice that $|\Lambda X.\mathbf{t}| = |\mathbf{t}|$ and $|\alpha.\forall X.U'| \equiv \forall X.|U'| = \forall X.U$.

7. $\dfrac{\Gamma \vdash \mathbf{t}:T \qquad \Gamma \vdash \mathbf{r}:R}{\Gamma \vdash \mathbf{t}+\mathbf{r}:T+R}\ {+_I}$ Then by the induction hypothesis $\exists T', R'$ such that $|\Gamma| \vdash_v |\mathbf{t}|:T'$ and $|\Gamma| \vdash_v |\mathbf{r}|:R'$, where $|T'| \equiv T$ and $|R'| \equiv R$. Then using rule $+_I$ in *Vectorial*, we can derive $|\Gamma| \vdash_v |\mathbf{t}| + |\mathbf{r}|:T' + R'$. Notice that $|\mathbf{t}+\mathbf{r}| = |\mathbf{t}| + |\mathbf{r}|$ and $|T' + R'| = |T'| + |R'| \equiv T + R$.

8. $\dfrac{\Gamma \vdash \mathbf{t}:T}{\Gamma \vdash \alpha.\mathbf{t}:\lfloor\alpha\rfloor.T}\ {s_I}$ Then by the induction hypothesis $\exists R$ such that $|\Gamma| \vdash_v |\mathbf{t}|:R$ and $|R| \equiv T$. Then using rule s_I in *Vectorial*, we can derive $|\Gamma| \vdash_v \alpha.|\mathbf{t}|:\alpha.R$. Notice that $|\alpha.\mathbf{t}| = \alpha.|\mathbf{t}|$ and $|\alpha.R| = \lfloor\alpha\rfloor.|R| \equiv \lfloor\alpha\rfloor.T$.

9. $\dfrac{\Gamma \vdash \mathbf{t}:T \qquad T \equiv R}{\Gamma \vdash \mathbf{t}:R}\ {\equiv}$ Then by the induction hypothesis $\exists S$ such that $|\Gamma| \vdash_v |\mathbf{t}|:S$ with $|S| \equiv T \equiv R$.

□

E.3 Proof of Lemma 6.3.3

Lemma 6.3.3. There is no an infinite sequence reduction consisting only of type beta rules.

Proof. Consider the following function from terms to natural numbers:

$$\sigma(x:U) = 1 \qquad \sigma(\Lambda X.\mathbf{t}) = 1 + \sigma(\mathbf{t}) \qquad \sigma(\mathbf{t}@U) = \sigma(\mathbf{t}) \qquad \sigma(\alpha.\mathbf{t}) = \sigma(\mathbf{t})$$
$$\sigma(\lambda x:U.\mathbf{t}) = \sigma(\mathbf{t}) \qquad \sigma((\mathbf{t})\ \mathbf{r}) = \sigma(\mathbf{t})\ \sigma(\mathbf{r}) \qquad \sigma(\mathbf{0}) = 1 \qquad \sigma(\mathbf{t}+\mathbf{r}) = \sigma(\mathbf{t}) + \sigma(\mathbf{r})$$

Then we prove by structural induction on $\mathbf{t}$ that $\sigma((\Lambda X.\mathbf{t})@U) > \sigma(\mathbf{t}[U/X])$. Since it is positive a strictly decreasing function on the type beta reduction, the sequence cannot be infinite.

We proceed with the induction. Notice that $\sigma(\mathbf{t}) = \sigma(\mathbf{t}[U/X])$ since it does not depend on the types in $\mathbf{t}$.

- $\mathbf{t} = x:V$, then $\sigma((\Lambda X.(x:V))@U) = 2 > 1 = \sigma(x:V[U/X]) = \sigma((x:V)[U/X])$.
- $\mathbf{t} = \lambda x:V.\mathbf{r}$, then $\sigma((\Lambda X.(\lambda x:V.\mathbf{r}))@U) = 1 + \sigma(\mathbf{r}) > \sigma(\mathbf{r}) = \sigma(\mathbf{r}[U/X])$.
- $\mathbf{t} = \Lambda Y.\mathbf{r}$, then $\sigma((\Lambda X.\Lambda Y.\mathbf{r})@U) = 2 + \sigma(\mathbf{r}) > 1 + \sigma(\mathbf{r}) = \sigma(\Lambda Y.(\mathbf{r}[U/X])) = \sigma((\Lambda Y.\mathbf{r})[U/X])$.
- $\mathbf{t} = (\mathbf{r})\ \mathbf{u}$, then $\sigma((\Lambda X.(\mathbf{r})\ \mathbf{u})@U) = 1 + \sigma(\mathbf{r}) + \sigma(\mathbf{u}) > \sigma(\mathbf{r}) + \sigma(\mathbf{u}) = \sigma((\mathbf{r})\ \mathbf{u}) = \sigma(((\mathbf{r})\ \mathbf{u})[U/X])$.

- $\mathbf{t} = \mathbf{r}@V$, then $\sigma((\Lambda X.(\mathbf{r}@V))@U) = 1 + \sigma(\mathbf{r}) > \sigma(\mathbf{r}) = \sigma(\mathbf{r}@V) = \sigma((\mathbf{r}@V)[U/X])$.
- $\mathbf{t} = \mathbf{0}$, then $\sigma((\Lambda X.\mathbf{0})@U) = 2 > 1 = \sigma(\mathbf{0}) = \sigma(\mathbf{0}[U/X])$.
- $\mathbf{t} = \alpha.\mathbf{r}$, then $\sigma((\Lambda X.\alpha.\mathbf{r})@U) = 1 + \sigma(\mathbf{r}) > \sigma(\mathbf{r}) = \sigma(\alpha.\mathbf{r}) = \sigma((\alpha.\mathbf{r})[U/X])$.
- $\mathbf{t} = \mathbf{r} + \mathbf{u}$, then $\sigma((\Lambda X.(\mathbf{r} + \mathbf{u}))@U) = 1 + \sigma(\mathbf{r}) + \sigma(\mathbf{u}) > \sigma(\mathbf{r}) + \sigma(\mathbf{u}) = \sigma(\mathbf{r} + \mathbf{u}) = \sigma((\mathbf{r} + \mathbf{u})[U/X])$.

□

E.4 Proof of Lemma 6.4.1

Lemma 6.4.1 (Poset)**.**

1. $\sqsubseteq$ is a partial order relation
2. $\lesssim$ is a partial order relation in $\sim$.

Proof.

1. For the purposes of the proof we use an equivalent definition of $\sqsubseteq$ and define it as the least relation satisfying:

$$\begin{array}{lll} \mathbf{0} \sqsubseteq \mathbf{t} & \mathbf{t} \sqsubseteq \mathbf{t} & \mathbf{t} \sqsubseteq \mathbf{t} + \mathbf{t} \\ \mathbf{t} \sqsubseteq \mathbf{t}' \;\Rightarrow\; \lambda x : U.\mathbf{t} \sqsubseteq \lambda x : U.\mathbf{t}' & \mathbf{t} \sqsubseteq \mathbf{t}' \;\wedge\; \mathbf{r} \sqsubseteq \mathbf{r}' \;\Rightarrow\; (\mathbf{t})\ \mathbf{r} \sqsubseteq (\mathbf{t}')\ \mathbf{r}' & \\ \mathbf{t} \sqsubseteq \mathbf{t}' \;\Rightarrow\; \Lambda X.\mathbf{t} \sqsubseteq \Lambda X.\mathbf{t}' & \mathbf{t} \sqsubseteq \mathbf{t}' \;\wedge\; \mathbf{r} \sqsubseteq \mathbf{r}' \;\Rightarrow\; \mathbf{t} + \mathbf{r} \sqsubseteq \mathbf{t}' + \mathbf{r}' & \\ \mathbf{t} \sqsubseteq \mathbf{t}' \;\Rightarrow\; \mathbf{t}@U \sqsubseteq \mathbf{t}'@U & \mathbf{t} \sqsubseteq \mathbf{r} \;\wedge\; \mathbf{r} \sqsubseteq \mathbf{u} \;\Rightarrow\; \mathbf{t} \sqsubseteq \mathbf{u} & \end{array}$$

   where the first three rules replace the first rule in the definition of $\sqsubseteq$ in Section 6.4. The equivalence can be shown by a simple inductive argument in one direction, and in the other direction by noting that replacing (α, β) by $(0, 1)$, $(1, 1)$, and $(1, 2)$ yields the new three rules.

   Reflexivity and transitivity are trivial since they are part of the definition.

   In order to prove antisymmetry, let us assume $\mathbf{t} \sqsubseteq \mathbf{u}$ and $\mathbf{u} \sqsubseteq \mathbf{t}$. We proceed by induction on $\mathbf{t} \sqsubseteq \mathbf{u}$.

   - case $\mathbf{0} \sqsubseteq \mathbf{t}$. Then $\mathbf{t} \sqsubseteq \mathbf{0}$ implies that $\mathbf{t} = \mathbf{0}$.
   - case $\mathbf{t} \sqsubseteq \mathbf{t} + \mathbf{t}$. This case is cannot happen, since then we would need $\mathbf{t} + \mathbf{t} \sqsubseteq \mathbf{t}$, which is not part of the relation.
   - In the rest of the cases the structure of the terms is preserved and no differences are introduced.

2. The relation $\lesssim$ is a partial order on terms quotiented by $\sim$ because $\sqsubseteq$ restricted to normal forms is a partial order. More explicitly:

$$\mathbf{t} \lesssim \mathbf{t} \quad\Leftrightarrow\quad \mathbf{t}{\downarrow_A} \sqsubseteq \mathbf{t}{\downarrow_A}$$

$$(\mathbf{t} \lesssim \mathbf{u} \;\wedge\; \mathbf{u} \lesssim \mathbf{u} \;\Rightarrow\; \mathbf{t} \lesssim \mathbf{u}) \quad\Leftrightarrow\quad (\mathbf{t}{\downarrow_A} \sqsubseteq \mathbf{u}{\downarrow_A} \;\wedge\; \mathbf{u}{\downarrow_A} \sqsubseteq \mathbf{u}{\downarrow_A} \;\Rightarrow\; \mathbf{t}{\downarrow_A} \sqsubseteq \mathbf{u}{\downarrow_A})$$

$$\mathbf{t} \lesssim \mathbf{u} \;\wedge\; \mathbf{u} \lesssim \mathbf{t} \quad\Leftrightarrow\quad \mathbf{t}{\downarrow_A} \sqsubseteq \mathbf{u}{\downarrow_A} \;\wedge\; \mathbf{u}{\downarrow_A} \sqsubseteq \mathbf{t}{\downarrow_A} \quad\Rightarrow\quad \mathbf{t}{\downarrow_A} = \mathbf{u}{\downarrow_A} \quad\Leftrightarrow\quad \mathbf{t} \sim \mathbf{u}$$

□

E.5 Proof of Theorem 6.4.2

We need some auxiliary lemmas in order to prove this theorem.

Lemma E.5.1. *For any term* $\mathbf{t}$ *and base term* $\mathbf{b}$ *in* λ^{CA}, $\sigma(\mathbf{t}[\mathbf{b}/x]) = \sigma(\mathbf{t})[\sigma(\mathbf{b})/x]$

Proof. We proceed by structural induction over $\mathbf{t} \in \lambda^{CA}$.

- $\mathbf{t} = x$. Then $\sigma(x[\mathbf{b}/x]) = \sigma(\mathbf{b}) = x[\sigma(\mathbf{b})/x] = \sigma(x)[\sigma(\mathbf{b})/x]$.
- $\mathbf{t} = y$. Then $\sigma(y[\mathbf{b}/x]) = \sigma(y) = y = y[\sigma(\mathbf{b})/x]$.
- $\mathbf{t} = \lambda y : U.\mathbf{t}'$. Then $\sigma((\lambda y : U.\mathbf{t}')[\mathbf{b}/x]) = \sigma(\lambda y : U.\mathbf{t}'[\mathbf{b}/x])$ and this is equal to $\lambda y : U.\sigma(\mathbf{t}'[\mathbf{b}/x])$ which by the induction hypothesis is equal to $\lambda y : U.\sigma(\mathbf{t}')[\sigma(\mathbf{b})/x] = (\lambda y : U.\sigma(\mathbf{t}'))[\sigma(\mathbf{b})/x] = \sigma(\lambda y : U.\mathbf{t}')[\sigma(\mathbf{b})/x]$.
- $\mathbf{t} = (\mathbf{r})\ \mathbf{u}$. Then $\sigma(((\mathbf{r})\ \mathbf{u})[\mathbf{b}/x]) = \sigma((\mathbf{r}[\mathbf{b}/x])\ \mathbf{u}[\mathbf{b}/x]) = (\sigma(\mathbf{r}[\mathbf{b}/x]))\ \sigma(\mathbf{u}[\mathbf{b}/x])$, which by the induction hypothesis is equal to $(\sigma(\mathbf{r})[\sigma(\mathbf{b})/x])\ \sigma(\mathbf{u})[\sigma(\mathbf{b})/x] = ((\sigma(\mathbf{r}))\ \sigma(\mathbf{u}))[\sigma(\mathbf{b})/x] = \sigma((\mathbf{r})\ \mathbf{u})[\sigma(\mathbf{b})/x]$.
- $\mathbf{t} = \Lambda X.\mathbf{r}$. Then $\sigma((\Lambda X.\mathbf{r})[\mathbf{b}/x]) = \sigma(\Lambda X.\mathbf{r}[\mathbf{b}/x]) = \Lambda X.\sigma(\mathbf{r}[\mathbf{b}/x])$ which by the induction hypothesis is equal to $\Lambda X.\sigma(\mathbf{r})[\sigma(\mathbf{b})/x] = (\Lambda X.\sigma(\mathbf{r}))[\sigma(\mathbf{b})/x]$ and this equal to $\sigma(\Lambda X.\mathbf{r})[\sigma(\mathbf{b})/x]$.
- $\mathbf{t} = \mathbf{r}@U$. Then $\sigma(\mathbf{r}@U[\mathbf{b}/x]) = \sigma(\mathbf{r}[\mathbf{b}/x]@U) = \sigma(\mathbf{r}[\mathbf{b}/x])@U$ which by the induction hypothesis is equal to $\sigma(\mathbf{r})[\sigma(\mathbf{b})/x]@U = \sigma(\mathbf{r})@U[\sigma(\mathbf{b})/x] = \sigma(\mathbf{r}@U)[\sigma(\mathbf{b})/x]$.
- $\mathbf{t} = \alpha.\mathbf{r}$. Then $\sigma((\alpha.\mathbf{r})[\mathbf{b}/x]) = \sigma(\alpha.\mathbf{r}[\mathbf{b}/x]) = \sum_{i=1}^{\lfloor\alpha\rfloor} \sigma(\mathbf{r}[\mathbf{b}/x])$, which by the induction hypothesis is equal to $\sum_{i=1}^{\lfloor\alpha\rfloor} \sigma(\mathbf{r})[\sigma(\mathbf{b})/x] = (\sum_{i=1}^{\lfloor\alpha\rfloor} \sigma(\mathbf{r}))[\sigma(\mathbf{b})/x] = \sigma(\alpha.\mathbf{r})[\sigma(\mathbf{b})/x]$.
- $\mathbf{t} = \mathbf{r}+\mathbf{u}$. Then $\sigma((\mathbf{r}+\mathbf{u})[\mathbf{b}/x]) = \sigma(\mathbf{r}[\mathbf{b}/x]+\mathbf{u}[\mathbf{b}/x]) = \sigma(\mathbf{r}[\mathbf{b}/x])+\sigma(\mathbf{u}[\mathbf{b}/x])$ which by the induction hypothesis is equal to $\sigma(\mathbf{r})[\sigma(\mathbf{b})/x] + \sigma(\mathbf{u})[\sigma(\mathbf{b})/x] = (\sigma(\mathbf{r}) + \sigma(\mathbf{u}))[\sigma(\mathbf{b})/x] = \sigma(\mathbf{r} + \mathbf{u})[\sigma(\mathbf{b})/x]$.

□

Lemma E.5.2. *For any terms* $\mathbf{t}_1$, $\mathbf{t}_2$ *in* λ^{add}, $(\mathbf{t}_1 + \mathbf{t}_2)\!\downarrow_A \lesssim \mathbf{t}_1\!\downarrow_A + \mathbf{t}_2\!\downarrow_A$.

Proof. Notice that the only case where these terms are different is when one of the addends reduces to $\mathbf{0}$, in such case, their normal forms coincides, making it possible to compare in this way. □

Now we are ready to prove the theorem.

Theorem 6.4.2 (Abstract Interpretation)**.** $\downarrow$ is a valid concretization of $\downarrow_A$, that is, $\forall \mathbf{t} \in \lambda^{CA}$, $\sigma(\mathbf{t})\!\downarrow_A \lesssim \sigma(\mathbf{t}\!\downarrow)$.

Proof. We proceed by structural induction over $\mathbf{t} \in \lambda^{CA}$.

1. $\mathbf{t} = x$ or $\mathbf{t} = \mathbf{0}$. Then $\sigma(\mathbf{t})\!\downarrow_A = \mathbf{t} = \sigma(\mathbf{t}\!\downarrow)$.
2. $\mathbf{t} = \lambda x : U.\mathbf{r}$. Then by the induction hypothesis $\sigma(\mathbf{r})\!\downarrow_A \lesssim \sigma(\mathbf{r}\!\downarrow)$ and so $\sigma(\lambda x : U.\mathbf{r})\!\downarrow_A = \lambda x : U.\sigma(\mathbf{r})\!\downarrow_A \lesssim \lambda x : U.\sigma(\mathbf{r}\!\downarrow) = \sigma(\lambda x : U.\mathbf{r}\!\downarrow)$.
3. $\mathbf{t} = \Lambda X.\mathbf{r}$. By the induction hypothesis $\sigma(\mathbf{r})\!\downarrow_A \lesssim \sigma(\mathbf{r}\!\downarrow)$, so $\sigma(\Lambda X.\mathbf{r})\!\downarrow_A = \Lambda X.\sigma(\mathbf{r})\!\downarrow_A \lesssim \Lambda X.\sigma(\mathbf{r}\!\downarrow) = \sigma(\Lambda X.\mathbf{r}\!\downarrow)$.
4. $\mathbf{t} = \mathbf{r}@U$. By the induction hypothesis $\sigma(\mathbf{r})\!\downarrow_A \lesssim \sigma(\mathbf{r}\!\downarrow)$, so $\sigma(\mathbf{r}@U)\!\downarrow_A = \sigma(\mathbf{r})\!\downarrow_A @U \lesssim \sigma(\mathbf{r}\!\downarrow)@U = \sigma(\mathbf{r}@U\!\downarrow)$.

5. $\mathbf{t} = \alpha.\mathbf{r}$. Case by case on the possible one-step $\to$-reductions starting from $\alpha.\mathbf{r}$.

   (a) $\alpha = 0$ and $0.\mathbf{r} \to \mathbf{0}$. Then $\sigma((0.\mathbf{r})\downarrow) = \sigma(\mathbf{0}) = \mathbf{0} = \mathbf{0}\downarrow_{\mathcal{A}} = (\sum_{i=1}^{0} \mathbf{r})\downarrow_{\mathcal{A}} = \sigma(0.\mathbf{r})\downarrow_{\mathcal{A}}$.

   (b) $\alpha = 1$ and $1.\mathbf{r} \to \mathbf{r}$. Then $\sigma((1.\mathbf{r})\downarrow) = \sigma(\mathbf{r}\downarrow)$. By the induction hypothesis $\sigma(\mathbf{r})\downarrow_{\mathcal{A}} \lesssim \sigma(\mathbf{r}\downarrow)$ and notice that $\sigma(\mathbf{r})\downarrow_{\mathcal{A}} = (\sum_{i=1}^{1} \sigma(\mathbf{r}))\downarrow_{\mathcal{A}} = \sigma(1.\mathbf{r})\downarrow_{\mathcal{A}}$.

   (c) $\mathbf{r} = \mathbf{0}$ and $\alpha.\mathbf{0} \to \mathbf{0}$. Then $\sigma((\alpha.\mathbf{0})\downarrow) = \sigma(\mathbf{0}) = \mathbf{0} = (\sum_{i=1}^{\lfloor\alpha\rfloor} \mathbf{0})\downarrow_{\mathcal{A}} = (\sigma(\alpha.\mathbf{0}))\downarrow_{\mathcal{A}}$.

   (d) $\mathbf{r} = \beta.\mathbf{u}$ and $\alpha.\beta.\mathbf{u} \to \alpha \times \beta.\mathbf{u}$. Then $\sigma(\alpha.\beta.\mathbf{u})\downarrow_{\mathcal{A}} = (\sum_{i=1}^{\lfloor\alpha\rfloor\times\lfloor\beta\rfloor} \sigma(\mathbf{u}))\downarrow_{\mathcal{A}} \lesssim (\sum_{i=1}^{\lfloor\alpha\times\beta\rfloor} \sigma(\mathbf{u}))\downarrow_{\mathcal{A}} = \sigma(\alpha \times \beta.\mathbf{u})\downarrow_{\mathcal{A}}$ which by the induction hypothesis is $\lesssim$ than $\sigma(\alpha \times \beta.\mathbf{u}\downarrow) = \sigma((\alpha.\beta.\mathbf{u})\downarrow)$.

   (e) $\mathbf{r} = \mathbf{u}_1 + \mathbf{u}_2$ and $\alpha.(\mathbf{u}_1 + \mathbf{u}_2) \to \alpha.\mathbf{u}_1 + \alpha.\mathbf{u}_2$. Then $\sigma(\alpha.(\mathbf{u}_1 + \mathbf{u}_2))\downarrow_{\mathcal{A}} = (\sum_{i=1}^{\lfloor\alpha\rfloor} \sigma(\mathbf{u}_1 + \mathbf{u}_2))\downarrow_{\mathcal{A}} = (\sum_{i=1}^{\lfloor\alpha\rfloor} \sigma(\mathbf{u}_1) + \sum_{i=1}^{\lfloor\alpha\rfloor} \sigma(\mathbf{u}_2))\downarrow_{\mathcal{A}} = \sigma(\alpha.\mathbf{u}_1 + \alpha.\mathbf{u}_2)\downarrow_{\mathcal{A}}$, which by the induction hypothesis is $\lesssim$ than $\sigma((\alpha.\mathbf{u}_1 + \alpha.\mathbf{u}_2)\downarrow)$ and notice that this term is equal to $\sigma((\alpha.(\mathbf{u}_1 + \mathbf{u}_2))\downarrow)$.

   (f) $\alpha.\mathbf{t}$ is in normal form. Then $\sigma((\alpha.\mathbf{t})\downarrow) = \sigma(\alpha.\mathbf{t}) = \sum_{i=1}^{\lfloor\alpha\rfloor} \sigma(\mathbf{t})$ and notice that $\sigma(\alpha.\mathbf{t})\downarrow_{\mathcal{A}} = (\sum_{i=1}^{\lfloor\alpha\rfloor} \sigma(\mathbf{t}))\downarrow_{\mathcal{A}} \lesssim \sum_{i=1}^{\lfloor\alpha\rfloor} \sigma(\mathbf{t})$.

6. $\mathbf{t} = \mathbf{r} + \mathbf{u}$. Then $\sigma(\mathbf{r} + \mathbf{u})\downarrow_{\mathcal{A}} = (\sigma(\mathbf{r}) + \sigma(\mathbf{u}))\downarrow_{\mathcal{A}}$ which by Lemma E.5.2 is $\lesssim$ than $\sigma(\mathbf{r})\downarrow_{\mathcal{A}} + \sigma(\mathbf{u})\downarrow_{\mathcal{A}}$ which by the induction hypothesis is $\lesssim$ than $\sigma(\mathbf{r}\downarrow) + \sigma(\mathbf{u}\downarrow) = \sigma(\mathbf{r}\downarrow + \mathbf{u}\downarrow)$. If $\mathbf{r}\downarrow + \mathbf{u}\downarrow$ is in normal form, then due to the confluence of λ^{CA}, it is equal to $(\mathbf{r} + \mathbf{u})\downarrow$, and then we are done. In other case, it means that $\mathbf{r}\downarrow + \mathbf{u}\downarrow$ reduces. Cases:

   (a) $\mathbf{u}\downarrow = \mathbf{0}$, then $(\mathbf{r} + \mathbf{u})\downarrow = \mathbf{r}\downarrow$. Then $\sigma(\mathbf{r}\downarrow) + \sigma(\mathbf{u}\downarrow) = \sigma(\mathbf{r}\downarrow) + \mathbf{0} \lesssim \sigma(\mathbf{r}\downarrow) = \sigma((\mathbf{r} + \mathbf{u})\downarrow)$.

   (b) $\mathbf{r}\downarrow = \alpha.\mathbf{u}$ and $\mathbf{u}\downarrow = \beta.\mathbf{u}$, then $(\mathbf{r} + \mathbf{u})\downarrow = (\alpha + \beta).\mathbf{u}$. So $\sigma(\mathbf{r}\downarrow) + \sigma(\mathbf{u}\downarrow) = \sigma(\alpha.\mathbf{u}) + \sigma(\beta.\mathbf{u}) = \sum_{i=1}^{\lfloor\alpha\rfloor} \mathbf{u} + \sum_{i=1}^{\lfloor\beta\rfloor} \mathbf{u} = \sum_{i=1}^{\lfloor\alpha\rfloor+\lfloor\beta\rfloor} \mathbf{u} \lesssim \sum_{i=1}^{\lfloor\alpha+\beta\rfloor} \mathbf{u} = \sigma((\alpha + \beta).\mathbf{u}) = \sigma((\mathbf{r} + \mathbf{u})\downarrow)$.

   (c) The case where $\mathbf{r}\downarrow = \alpha.\mathbf{u}$ and $\mathbf{u}\downarrow = \mathbf{u}$ and the case $\mathbf{r}\downarrow = \mathbf{u}\downarrow$ are analogous to the previous one.

7. $\mathbf{t} = (\mathbf{r})\ \mathbf{u}$. Then $\sigma((\mathbf{r})\ \mathbf{u})\downarrow_{\mathcal{A}} = ((\sigma(\mathbf{r}))\ \sigma(\mathbf{u}))\downarrow_{\mathcal{A}}$. Case by case on the possible one-step $\to$-reductions starting from $(\mathbf{r})\ \mathbf{u}$

   (a) $((\mathbf{r})\ \mathbf{u})\downarrow = (\mathbf{r}\downarrow)\ \mathbf{u}\downarrow$ Then $\sigma(((\mathbf{r})\ \mathbf{u})\downarrow) = (\sigma(\mathbf{r}\downarrow))\ \sigma(\mathbf{u}\downarrow)$. On the other hand, due to the confluence in λ^{add}, $((\sigma(\mathbf{r}))\ \sigma(\mathbf{u}))\downarrow_{\mathcal{A}} = ((\sigma(\mathbf{r})\downarrow_{\mathcal{A}})\ \sigma(\mathbf{u})\downarrow_{\mathcal{A}})\downarrow_{\mathcal{A}}$ and since this is the normal form of $(\sigma(\mathbf{r})\downarrow_{\mathcal{A}})\ \sigma(\mathbf{u})\downarrow_{\mathcal{A}}$, by definition, $((\sigma(\mathbf{r})\downarrow_{\mathcal{A}})\ \sigma(\mathbf{u})\downarrow_{\mathcal{A}})\downarrow_{\mathcal{A}} \lesssim (\sigma(\mathbf{r})\downarrow_{\mathcal{A}})\ \sigma(\mathbf{u})\downarrow_{\mathcal{A}}$, which by the induction hypothesis is $\lesssim$ to $(\sigma(\mathbf{r}\downarrow))\ \sigma(\mathbf{u}\downarrow)$.

   (b) $\mathbf{r} = \mathbf{r}_1 + \mathbf{r}_2$, and $(\mathbf{r})\ \mathbf{u} \to (\mathbf{r}_1)\ \mathbf{u} + (\mathbf{r}_2)\ \mathbf{u}$. Notice that $\sigma((\mathbf{r}_1 + \mathbf{r}_2)\ \mathbf{u})\downarrow_{\mathcal{A}}$ is equal to $((\sigma(\mathbf{r}_1) + \sigma(\mathbf{r}_2))\ \sigma(\mathbf{u}))\downarrow_{\mathcal{A}} = ((\sigma(\mathbf{r}_1))\ \sigma(\mathbf{u}) + (\sigma(\mathbf{r}_2))\ \sigma(\mathbf{u}))\downarrow_{\mathcal{A}}$ which by Lemma E.5.2 is $\lesssim$ than $((\sigma(\mathbf{r}_1))\ \sigma(\mathbf{u}))\downarrow_{\mathcal{A}} + ((\sigma(\mathbf{r}_2))\ \sigma(\mathbf{u}))\downarrow_{\mathcal{A}} = (\sigma((\mathbf{r}_1)\ \mathbf{u}))\downarrow_{\mathcal{A}} + (\sigma((\mathbf{r}_2)\ \mathbf{u}))\downarrow_{\mathcal{A}}$ which by the induction hypothesis is $\lesssim$ than $\sigma(((\mathbf{r}_1)\ \mathbf{u})\downarrow) + \sigma(((\mathbf{r}_2)\ \mathbf{u})\downarrow) = \sigma(((\mathbf{r}_1)\ \mathbf{u})\downarrow + ((\mathbf{r}_2)\ \mathbf{u})\downarrow)$. Cases:

      i. $((\mathbf{r}_1)\ \mathbf{u})\downarrow + ((\mathbf{r}_2)\ \mathbf{u})\downarrow = ((\mathbf{r}_1)\ \mathbf{u} + (\mathbf{r}_2)\ \mathbf{u})\downarrow$. Then the term $\sigma(((\mathbf{r}_1)\ \mathbf{u})\downarrow + ((\mathbf{r}_2)\ \mathbf{u})\downarrow)$ is equal to $\sigma((((\mathbf{r}_1)\ \mathbf{u}) + ((\mathbf{r}_2)\ \mathbf{u}))\downarrow) = \sigma(((\mathbf{r}_1 + \mathbf{r}_2)\ \mathbf{u})\downarrow)$.

      ii. $((\mathbf{r}_1)\ \mathbf{u})\downarrow = \mathbf{0}$. Then $\sigma(((\mathbf{r}_1)\ \mathbf{u})\downarrow + ((\mathbf{r}_2)\ \mathbf{u})\downarrow) = \mathbf{0} + \sigma(((\mathbf{r}_2)\ \mathbf{u})\downarrow)$. Notice that the term $(\mathbf{0} + \sigma(((\mathbf{r}_2)\ \mathbf{u})\downarrow))\downarrow_{\mathcal{A}}$ is equal to $\sigma(((\mathbf{r}_2)\ \mathbf{u})\downarrow)\downarrow_{\mathcal{A}}$, so $\mathbf{0} + \sigma(((\mathbf{r}_2)\ \mathbf{u})\downarrow) \lesssim \sigma(((\mathbf{r}_2)\ \mathbf{u})\downarrow) = \sigma((\mathbf{0} + (\mathbf{r}_2)\ \mathbf{u})\downarrow) = \sigma((((\mathbf{r}_1)\ \mathbf{u})\downarrow + (\mathbf{r}_2)\ \mathbf{u})\downarrow)$, which due to the confluence of λ^{CA} is equal to $\sigma(((\mathbf{r}_1)\ \mathbf{u} + (\mathbf{r}_2)\ \mathbf{u})\downarrow) = \sigma(((\mathbf{r}_1 + \mathbf{r}_2)\ \mathbf{u})\downarrow)$.

      iii. $((\mathbf{r}_1)\ \mathbf{u})\downarrow = \alpha.\mathbf{u}$ and $((\mathbf{r}_2)\ \mathbf{u})\downarrow = \beta.\mathbf{u}$. So $\sigma(((\mathbf{r}_1)\ \mathbf{u})\downarrow + ((\mathbf{r}_2)\ \mathbf{u})\downarrow) = \sigma(\alpha.\mathbf{u} + \beta.\mathbf{u}) = \sum_{i=1}^{\lfloor\alpha\rfloor} \sigma(\mathbf{u}) + \sum_{i=1}^{\lfloor\beta\rfloor} \sigma(\mathbf{u}) = \sum_{i=1}^{\lfloor\alpha\rfloor+\lfloor\beta\rfloor} \sigma(\mathbf{u})$ which is $\lesssim$ than $\sum_{i=1}^{\lfloor\alpha+\beta\rfloor} \sigma(\mathbf{u}) = \sigma((\alpha+\beta).\mathbf{u}) = \sigma(((\mathbf{r}_1)\ \mathbf{u} + (\mathbf{r}_2)\ \mathbf{u})\downarrow) = \sigma(((\mathbf{r}_1 + \mathbf{r}_2)\ \mathbf{u})\downarrow)$.

iv. The cases where $((\mathbf{r}_1)\ \mathbf{u})\downarrow = \alpha.\mathbf{u}$ and $((\mathbf{r}_2)\ \mathbf{u})\downarrow = \mathbf{u}$ or $((\mathbf{r}_1)\ \mathbf{u})\downarrow = \mathbf{u}$ and $((\mathbf{r}_2)\ \mathbf{u})\downarrow = \mathbf{u}$ are analogous to the previous one.

(c) $\mathbf{u} = \mathbf{u}_1 + \mathbf{u}_2$ and $(\mathbf{r})\ \mathbf{u} \to (\mathbf{r})\ \mathbf{u}_1 + (\mathbf{r})\ \mathbf{u}_2$. Analogous to the previous case.

(d) $\mathbf{r} = \alpha.\mathbf{r}'$ and $(\mathbf{r})\ \mathbf{u} \to \alpha.(\mathbf{r}')\ \mathbf{u}$. Then the term $\sigma((\alpha.\mathbf{r}')\ \mathbf{u})\downarrow_A$ is equal to $((\sigma(\alpha.\mathbf{r}'))\ \sigma(\mathbf{u}))\downarrow_A$ which is equal to $((\sum_{i=1}^{\lfloor\alpha\rfloor}\sigma(\mathbf{r}'))\ \sigma(\mathbf{u}))\downarrow_A = ((\sum_{i=1}^{\lfloor\alpha\rfloor}\sigma(\mathbf{r}'))\ \sigma(\mathbf{u}))\downarrow_A$ and this, by Lemma E.5.2, is $\lesssim$ than $\sum_{i=1}^{\lfloor\alpha\rfloor}((\sigma(\mathbf{r}'))\ \sigma(\mathbf{u}))\downarrow_A = \sum_{i=1}^{\lfloor\alpha\rfloor}\sigma((\mathbf{r}')\ \mathbf{u})\downarrow_A$ which by the induction hypothesis is $\lesssim$ than $\sum_{i=1}^{\lfloor\alpha\rfloor}\sigma(((\mathbf{r}')\ \mathbf{u})\downarrow) = \sigma(\alpha.((\mathbf{r}')\ \mathbf{u})\downarrow)$. Cases:

i. $\alpha.((\mathbf{r}')\ \mathbf{u})\downarrow = ((\alpha.\mathbf{r}')\ \mathbf{u})\downarrow$. Then $\sigma(\alpha.((\mathbf{r}')\ \mathbf{u})\downarrow) = \sigma(((\alpha.\mathbf{r}')\ \mathbf{u})\downarrow) = \sigma(((\alpha.\mathbf{r}')\ \mathbf{u})\downarrow)$.

ii. $\alpha = 0$, so $\sigma(0.((\mathbf{r}')\ \mathbf{u})\downarrow) = \sum_{i=1}^{0}\sigma(((\mathbf{r}')\ \mathbf{u})\downarrow) = \mathbf{0} = \sigma(\mathbf{0}) = ((0.\mathbf{r}')\ \mathbf{u})\downarrow$.

iii. $\alpha = 1$ so $\sigma(1.((\mathbf{r}')\ \mathbf{u})\downarrow) = \sum_{i=1}^{1}\sigma(((\mathbf{r}')\ \mathbf{u})\downarrow) = \sigma(((\mathbf{r}')\ \mathbf{u})\downarrow) = \sigma(((1.\mathbf{r}')\ \mathbf{u})\downarrow)$.

iv. $((\mathbf{r}')\ \mathbf{u})\downarrow = \mathbf{0}$, so $\sigma(\alpha.((\mathbf{r}')\ \mathbf{u})\downarrow) = \sigma(\alpha.\mathbf{0}) = \sum_{i=1}^{\lfloor\alpha\rfloor}\mathbf{0}$, since its normal form in λ^{add} is $\mathbf{0}$, it is $\lesssim$ than $\mathbf{0} = \sigma(\mathbf{0}) = \sigma(((\alpha.\mathbf{r}')\ \mathbf{u})\downarrow)$.

v. $((\mathbf{r}')\ \mathbf{u})\downarrow = \beta.\mathbf{u}$, so $\sigma(\alpha.((\mathbf{r}')\ \mathbf{u})\downarrow) = \sigma(\alpha.\beta.\mathbf{u}) = \sum_{i=1}^{\lfloor\alpha\rfloor}\sum_{i=1}^{\lfloor\beta\rfloor}\sigma(\mathbf{u}) = \sum_{i=1}^{\lfloor\alpha\rfloor\times\lfloor\beta\rfloor}\sigma(\mathbf{u})$ which is $\lesssim$ than $\sum_{i=1}^{\lfloor\alpha\times\beta\rfloor}\sigma(\mathbf{u}) = \sigma(\alpha\times\beta.\mathbf{u}) = \sigma(((\alpha.\mathbf{r}')\ \mathbf{u})\downarrow)$.

vi. $((\mathbf{r}')\ \mathbf{u})\downarrow = \mathbf{u}+\mathbf{t}$, so $\sigma(\alpha.((\mathbf{r}')\ \mathbf{u})\downarrow) = \sigma(\alpha.(\mathbf{u}+\mathbf{t})) = \sum_{i=1}^{\lfloor\alpha\rfloor}\sigma(\mathbf{u}+\mathbf{t}) = \sum_{i=1}^{\lfloor\alpha\rfloor}(\sigma(\mathbf{u})+\sigma(\mathbf{t})) = \sum_{i=1}^{\lfloor\alpha\rfloor}\sigma(\mathbf{u}) + \sum_{i=1}^{\lfloor\alpha\rfloor}\sigma(\mathbf{t}) = \sigma(\alpha.\mathbf{u}) + \sigma(\alpha.\mathbf{t}) = \sigma(\alpha.\mathbf{u}+\alpha.\mathbf{t}) = \sigma((\alpha.(\mathbf{r}')\ \mathbf{u})\downarrow)$.

(e) $\mathbf{u} = \alpha.\mathbf{u}'$ and $(\mathbf{r})\ \mathbf{u} \to \alpha.(\mathbf{r})\ \mathbf{u}'$. Analogous to the previous case.

(f) $\mathbf{r} = \mathbf{0}$ and $(\mathbf{r})\ \mathbf{u} \to \mathbf{0}$. $\sigma((\mathbf{0})\ \mathbf{u})\downarrow_A = ((\mathbf{0})\ \sigma(\mathbf{u}))\downarrow_A = \mathbf{0} = \sigma(\mathbf{0}) = \sigma(((\mathbf{0})\ \mathbf{u})\downarrow)$.

(g) $\mathbf{u} = \mathbf{0}$ and $(\mathbf{r})\ \mathbf{u} \to \mathbf{0}$. Analogous to the previous case.

(h) $\mathbf{r} = \lambda x : U.\mathbf{r}'$, $\mathbf{u}$ is a base term and $(\mathbf{r})\ \mathbf{u} \to \mathbf{r}'[\mathbf{u}/x]$. Notice that $\sigma((\lambda x : U.\mathbf{r}')\ \mathbf{u})\downarrow_A$ is equal to $((\lambda x : U.\sigma(\mathbf{r}'))\ \sigma(\mathbf{u}))\downarrow_A = (\sigma(\mathbf{r}')[\sigma(\mathbf{u})/x])\downarrow_A$, which by Lemma E.5.1 is equal to $\sigma(\mathbf{r}'[\mathbf{u}/x])\downarrow_A$. If we prove that this term is $\lesssim$ than $\sigma((\mathbf{r}'[\mathbf{u}/x])\downarrow) = \sigma(((\lambda x : U.\mathbf{r}')\ \mathbf{u})\downarrow)$ then we are done. Let us prove it by structural induction over $\mathbf{r}'$, assuming $\sigma(\mathbf{r}')\downarrow_A \lesssim \sigma(\mathbf{r}'\downarrow)$ and $\sigma(\mathbf{u})\downarrow_A \lesssim \sigma(\mathbf{u}\downarrow)$, which are true by the induction hypothesis.

i. $\mathbf{r}' = x$, then $\sigma(x[\mathbf{u}/x])\downarrow_A = \sigma(\mathbf{u})\downarrow_A \lesssim \sigma(\mathbf{u}\downarrow) = \sigma((x'[\mathbf{u}/x])\downarrow)$.

ii. $\mathbf{r}' = y$ or $\mathbf{r}' = \mathbf{0}$, then $\sigma(\mathbf{r}'[\mathbf{u}/x])\downarrow_A = \sigma(\mathbf{r}')\downarrow_A \lesssim \sigma(\mathbf{r}'\downarrow) = \sigma((\mathbf{r}'[\mathbf{u}/x])\downarrow)$.

iii. $\mathbf{r}' = \lambda y : V.\mathbf{r}''$. Then $\sigma((\lambda y : V.\mathbf{r}'')[\mathbf{u}/x])\downarrow_A = \lambda y : V.\sigma(\mathbf{r}''[\mathbf{u}/x])\downarrow_A$ which by the induction hypothesis is $\lesssim$ than $\lambda y : V.\sigma(\mathbf{r}''[\mathbf{u}/x]\downarrow) = \sigma(\lambda y : V.\mathbf{r}''[\mathbf{u}/x]\downarrow)$.

iv. $\mathbf{r}' = \Lambda Y.\mathbf{r}''$. Then $\sigma((\Lambda Y.\mathbf{r}'')[\mathbf{u}/x])\downarrow_A = \Lambda Y.\sigma(\mathbf{r}''[\mathbf{u}/x])\downarrow_A$ which by the induction hypothesis is $\lesssim$ than $\Lambda Y.\sigma(\mathbf{r}''[\mathbf{u}/x]\downarrow) = \sigma(\Lambda Y.\mathbf{r}''[\mathbf{u}/x]\downarrow)$.

v. $\mathbf{r}' = (\mathbf{t}_1)\ \mathbf{t}_2$. Notice that $\sigma(((\mathbf{t}_1)\ \mathbf{t}_2)[\mathbf{u}/x])\downarrow_A = ((\sigma(\mathbf{t}_1[\mathbf{u}/x]))\ \sigma(\mathbf{t}_2[\mathbf{u}/x]))\downarrow_A$. Cases:

A. $((\mathbf{t}_1[\mathbf{u}/x])\ \mathbf{t}_2[\mathbf{u}/x])\downarrow = (\mathbf{t}_1[\mathbf{u}/x]\downarrow)\ \mathbf{t}_2[\mathbf{u}/x]\downarrow$. Since the terms $((\sigma(\mathbf{t}_1[\mathbf{u}/x]))\,\sigma(\mathbf{t}_2[\mathbf{u}/x]))\downarrow_A$ and $(\sigma(\mathbf{t}_1[\mathbf{u}/x])\downarrow_A\ \sigma(\mathbf{t}_2[\mathbf{u}/x]))\downarrow_A$ have the same normal form, by definition of $\lesssim$, one has $((\sigma(\mathbf{t}_1[\mathbf{u}/x]))\ \sigma(\mathbf{t}_2[\mathbf{u}/x]))\downarrow_A \lesssim (\sigma(\mathbf{t}_1[\mathbf{u}/x])\downarrow_A\ \sigma(\mathbf{t}_2[\mathbf{u}/x]))\downarrow_A$, which by the induction hypothesis is $\lesssim$ than $(\sigma(\mathbf{t}_1[\mathbf{u}/x]\downarrow))\ \sigma(\mathbf{t}_2[\mathbf{u}/x]\downarrow) = \sigma(\mathbf{t}_1[\mathbf{u}/x]\downarrow\ \mathbf{t}_2[\mathbf{u}/x]\downarrow) = \sigma(((\mathbf{t}_1[\mathbf{u}/x])\ \mathbf{t}_2[\mathbf{u}/x])\downarrow)$.

B. $\mathbf{t}_1[\mathbf{u}/x] = \mathbf{t}_{1_1} + \mathbf{t}_{1_2}$, so $((\mathbf{t}_1[\mathbf{u}/x])\ \mathbf{t}_2[\mathbf{u}/x])\downarrow = ((\mathbf{t}_{1_1})\ \mathbf{t}_2[\mathbf{u}/x] + (\mathbf{t}_{1_2})\ \mathbf{t}_2[\mathbf{u}/x])\downarrow$. This case is analogous to case 7b.

C. $\mathbf{t}_2[\mathbf{u}/x] = \mathbf{t}_{2_1} + \mathbf{t}_{2_2}$, so $((\mathbf{t}_1[\mathbf{u}/x])\ \mathbf{t}_2[\mathbf{u}/x])\downarrow = ((\mathbf{t}_1[\mathbf{u}/x])\ \mathbf{t}_{2_1} + (\mathbf{t}_1[\mathbf{u}/x])\ \mathbf{t}_{2_2})\downarrow$. Analogous to previous case.

D. $\mathbf{t}_1[\mathbf{u}/x] = \alpha.\mathbf{t}'_1$ or $\mathbf{t}_2[\mathbf{u}/x] = \alpha.\mathbf{t}'_2$, so $((\mathbf{t}_1[\mathbf{u}/x])\ \mathbf{t}_2[\mathbf{u}/x])\downarrow$ is equal to $(\alpha.(\mathbf{t}'_1)\ \mathbf{t}_2[\mathbf{u}/x])\downarrow$. This cases are analogous to case 7d.

E. $\mathbf{t}_1[\mathbf{u}/x] = \mathbf{0}$ or $\mathbf{t}_2[\mathbf{u}/x] = \mathbf{0}$, so $((\mathbf{t}_1[\mathbf{u}/x])\ \mathbf{t}_2[\mathbf{u}/x])\downarrow = \mathbf{0}$. This cases are analogous to case 7f.

F. $\mathbf{t}_1[\mathbf{u}/x] = \lambda y : V.\mathbf{t}'_1$ and $\mathbf{t}_2[\mathbf{u}/x]$ is a base term, let us call it $\mathbf{b}$, then the term $((\mathbf{t}_1[\mathbf{u}/x])\ \mathbf{t}_2[\mathbf{u}/x])\downarrow = \mathbf{t}'_1[\mathbf{b}/y]\downarrow$. Then $\sigma((\lambda y : V.\mathbf{t}'_1)\ \mathbf{b})\downarrow_A = ((\lambda y : V.\sigma(\mathbf{t}'_1))\ \sigma(\mathbf{b}))\downarrow_A = (\sigma(\mathbf{t}'_1)[\sigma(\mathbf{b})/y])\downarrow_A$, which by Lemma E.5.1 is equal to $\sigma(\mathbf{t}'_1[\mathbf{b}/y])\downarrow_A$ which by the induction hypothesis is $\lesssim$ than $\sigma((\mathbf{t}'_1[\mathbf{b}/y])\downarrow) = \sigma(((\mathbf{t}_1[\mathbf{u}/x])\ \mathbf{t}_2[\mathbf{u}/x])\downarrow)$.

vi. $\mathbf{r}' = \mathbf{r}''@V$. Then $\sigma(\mathbf{r}''@V[\mathbf{u}/x])\downarrow_A = \sigma(\mathbf{r}''[\mathbf{u}/x])\downarrow_A\ @V$ which by the induction hypothesis is $\lesssim$ than $\sigma(\mathbf{r}''[\mathbf{u}/x]\downarrow)@V = \sigma(\mathbf{r}''@V[\mathbf{u}/x]\downarrow)$.

vii. $\mathbf{r}' = \alpha.\mathbf{r}''$. Case by case on the possible one-step $\rightarrow$-reductions starting from $(\alpha.\mathbf{r}'')[\mathbf{u}/x] = \alpha.\mathbf{r}''[\mathbf{u}/x]$.

A. $\alpha = 0$, so $0.\mathbf{r}''[\mathbf{u}/x] \rightarrow \mathbf{0}$. Notice that $\sigma((0.\mathbf{r}''[\mathbf{u}/x])\downarrow) = \sigma(\mathbf{0}) = \mathbf{0} = \mathbf{0}\downarrow_A = (\sum_{i=1}^{0} \mathbf{r}''[\mathbf{u}/x])\downarrow_A = \sigma(0.\mathbf{r}''[\mathbf{u}/x])\downarrow_A$.

B. $\alpha = 1$, so $1.\mathbf{r}''[\mathbf{u}/x] \rightarrow \mathbf{r}''[\mathbf{u}/x]$. Then $\sigma((1.\mathbf{r}''[\mathbf{u}/x])\downarrow)$ is equal to $\sigma(\mathbf{r}''[\mathbf{u}/x]\downarrow)$. On the other hand, by the induction hypothesis $\sigma(\mathbf{r}''[\mathbf{u}/x])\downarrow_A \lesssim \sigma(\mathbf{r}''[\mathbf{u}/x]\downarrow)$ and notice that $\sigma(\mathbf{r}''[\mathbf{u}/x])\downarrow_A = (\sum_{i=1}^{1} \sigma(\mathbf{r}''[\mathbf{u}/x]))\downarrow_A = \sigma(1.\mathbf{r}''[\mathbf{u}/x])\downarrow_A$.

C. $\mathbf{r}''[\mathbf{u}/x] = \mathbf{0}$, so $\alpha.\mathbf{r}''[\mathbf{u}/x] \rightarrow \mathbf{0}$. Then $\sigma(\alpha.\mathbf{r}''[\mathbf{u}/x]\downarrow) = \sigma(\mathbf{0}) = \mathbf{0} = (\sum_{i=1}^{\lfloor\alpha\rfloor} \sigma(\mathbf{0}))\downarrow_A = \sigma(\alpha.\mathbf{r}''[\mathbf{u}/x])\downarrow_A$.

D. $\mathbf{r}''[\mathbf{u}/x] = \beta.\mathbf{t}_1$, so $\alpha.\mathbf{r}''[\mathbf{u}/x] \rightarrow \alpha \times \beta.\mathbf{t}_1$. Then $\sigma(\alpha.\mathbf{r}''[\mathbf{u}/x])\downarrow_A = (\sum_{i=1}^{\lfloor\alpha\rfloor \times \lfloor\beta\rfloor} \sigma(\mathbf{t}_1))\downarrow_A \lesssim (\sum_{i=1}^{\lfloor\alpha\times\beta\rfloor} \sigma(\mathbf{t}_1))\downarrow_A = \sigma(\alpha \times \beta.\mathbf{t}_1)\downarrow_A$ which is $\lesssim$ than $\sigma((\alpha \times \beta.\mathbf{t}_1)\downarrow)$ by the induction hypothesis. Notice that $\sigma((\alpha \times \beta.\mathbf{t}_1)\downarrow) = \sigma((\alpha.\mathbf{r}''[\mathbf{u}/x])\downarrow)$.

E. $\mathbf{r}''[\mathbf{u}/x] = \mathbf{t}_1 + \mathbf{t}_2$, so $\alpha.\mathbf{r}''[\mathbf{u}/x] \rightarrow \alpha.\mathbf{t}_1 + \alpha.\mathbf{t}_2$. Since $\mathbf{u}$ is a base term we can assume $\mathbf{r}' = \mathbf{t}'_1 + \mathbf{t}'_2$ with $\mathbf{t}'_1[\mathbf{u}/x] = \mathbf{t}_1$ and $\mathbf{t}'_2[\mathbf{u}/x] = \mathbf{t}_2$. Then $\sigma(\alpha.\mathbf{r}''[\mathbf{u}/x])\downarrow_A = (\sum_{i=1}^{\lfloor\alpha\rfloor} \sigma(\mathbf{t}_1 + \mathbf{t}_2))\downarrow_A = (\sum_{i=1}^{\lfloor\alpha\rfloor} \sigma(\mathbf{t}_1) + \sum_{i=1}^{\lfloor\alpha\rfloor} \sigma(\mathbf{t}_2))\downarrow_A = \sigma(\alpha.\mathbf{t}_1 + \alpha.\mathbf{t}_2)\downarrow_A$ which is equal to $\sigma((\alpha.\mathbf{t}'_1 + \alpha.\mathbf{t}'_2)[\mathbf{u}/x])\downarrow_A$, and this, by the induction hypothesis is $\lesssim$ than the term $\sigma((\alpha.\mathbf{t}'_1 + \alpha.\mathbf{t}'_2)[\mathbf{u}/x]\downarrow) = \sigma((\alpha.\mathbf{r}''[\mathbf{u}/x])\downarrow)$.

F. $\alpha.\mathbf{r}''[\mathbf{u}/x]$ is in normal form. Then we have $\sigma((\alpha.\mathbf{r}''[\mathbf{u}/x])\downarrow) = \sigma(\alpha.\mathbf{r}''[\mathbf{u}/x]) = \sum_{i=1}^{\lfloor\alpha\rfloor} \sigma(\mathbf{r}''[\mathbf{u}/x])$, and note that $\sigma(\alpha.\mathbf{r}''[\mathbf{u}/x])\downarrow_A = (\sum_{i=1}^{\lfloor\alpha\rfloor} \sigma(\mathbf{r}''[\mathbf{u}/x]))\downarrow_A$ and this is $\lesssim$ than $\sum_{i=1}^{\lfloor\alpha\rfloor} \sigma(\mathbf{r}''[\mathbf{u}/x])$.

viii. $\mathbf{r}' = \mathbf{t}_1 + \mathbf{t}_2$. Then $\sigma((\mathbf{t}_1 + \mathbf{t}_2)[\mathbf{u}/x])\downarrow_A = (\sigma(\mathbf{t}_1[\mathbf{u}/x]) + \sigma(\mathbf{t}_2[\mathbf{u}/x]))\downarrow_A$ which is, by Lemma E.5.2, $\lesssim$ than $\sigma(\mathbf{t}_1[\mathbf{u}/x])\downarrow_A + \sigma(\mathbf{t}_2[\mathbf{u}/x])\downarrow_A$ which by the induction hypothesis is $\lesssim$ than $\sigma(\mathbf{t}_1[\mathbf{u}/x]\downarrow) + \sigma(\mathbf{t}_2[\mathbf{u}/x]\downarrow) = \sigma(\mathbf{t}_1[\mathbf{u}/x]\downarrow + \mathbf{t}_2[\mathbf{u}/x]\downarrow)$. If $\mathbf{t}_1[\mathbf{u}/x]\downarrow + \mathbf{t}_2[\mathbf{u}/x]\downarrow$ is in normal form, then due to the confluence of λ^{CA}, it is equal to $(\mathbf{t}_1 + \mathbf{t}_2)[\mathbf{u}/x]\downarrow$, and then we are done. In other case, it means that $\mathbf{t}_1[\mathbf{u}/x]\downarrow + \mathbf{t}_2[\mathbf{u}/x]\downarrow$ reduces. Cases:

A. $\mathbf{t}_2[\mathbf{u}/x]\downarrow = \mathbf{0}$, then $(\mathbf{t}_1[\mathbf{u}/x] + \mathbf{t}_2[\mathbf{u}/x])\downarrow = \mathbf{t}_1[\mathbf{u}/x]\downarrow$. Then $\sigma(\mathbf{t}_1[\mathbf{u}/x]\downarrow) + \sigma(\mathbf{t}_2[\mathbf{u}/x]\downarrow) = \sigma(\mathbf{t}_1[\mathbf{u}/x]\downarrow) + \mathbf{0} \lesssim \sigma(\mathbf{t}_1[\mathbf{u}/x]\downarrow) = \sigma((\mathbf{t}_1[\mathbf{u}/x] + \mathbf{t}_2[\mathbf{u}/x])\downarrow)$.

B. $\mathbf{t}_1[\mathbf{u}/x]\downarrow = \alpha.\mathbf{t}'_1$ and $\mathbf{t}_2[\mathbf{u}/x]\downarrow = \beta.\mathbf{t}'_1$. Then $(\mathbf{t}_1 + \mathbf{t}_2)[\mathbf{u}/x]\downarrow = (\alpha + \beta).\mathbf{t}'_1$. So $\sigma(\mathbf{t}_1[\mathbf{u}/x]\downarrow) + \sigma(\mathbf{t}_2[\mathbf{u}/x]\downarrow) = \sigma(\alpha.\mathbf{t}'_1) + \sigma(\beta.\mathbf{t}'_1) = \sum_{i=1}^{\lfloor\alpha\rfloor} \mathbf{t}'_1 + \sum_{i=1}^{\lfloor\beta\rfloor} \mathbf{t}'_1 = \sum_{i=1}^{\lfloor\alpha\rfloor + \lfloor\beta\rfloor} \mathbf{t}'_1 \lesssim \sum_{i=1}^{\lfloor\alpha+\beta\rfloor} \mathbf{t}'_1 = \sigma((\alpha + \beta).\mathbf{t}'_1)$ which is equal to $\sigma((\mathbf{t}_1 + \mathbf{t}_2)[\mathbf{u}/x]\downarrow)$.

C. The case where $\mathbf{t}_1[\mathbf{u}/x]\downarrow = \alpha.\mathbf{t}'_1$ and $\mathbf{t}_2[\mathbf{u}/x]\downarrow = \mathbf{t}'_1$ and the case $\mathbf{t}_1[\mathbf{u}/x]\downarrow = \mathbf{t}_2[\mathbf{u}/x]\downarrow$ are analogous to the previous one.

□

E.6 Proof of Lemma 6.4.3

Lemma 6.4.3 (Typing preservation)**.** For any context Γ, term $\mathbf{t}$ and type T, if $\Gamma \vdash \mathbf{t} : T$ then $\Gamma \vdash_A \sigma(\mathbf{t}) : T$

Proof. We proceed by induction on the derivation of $\Gamma \vdash \mathbf{t} : T$.

- $\Gamma, x : U \vdash x : U$ as a consequence of rule ax. Notice that $\sigma(x) = x$ and by rule ax in λ^{add} we have $\Gamma, x : U \vdash_A x : U$.
- $\Gamma \vdash \mathbf{0} : \overline{0}$ as a consequence of rule $ax_{\overline{0}}$. Notice that $\sigma(\mathbf{0}) = \mathbf{0}$ and by rule $ax_{\overline{0}}$ in λ^{add} we have $\Gamma \vdash_A \mathbf{0} : \overline{0}$.
- $\Gamma \vdash (\mathbf{t})\ \mathbf{r} : \sum_{i=1}^{n}(m.T_i)$ as a consequence of $\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n}(U \to T_i)$, $\Gamma \vdash \mathbf{r} : m.U$ and rule $\to_E$. Then by the induction hypothesis $\Gamma \vdash_A \sigma(\mathbf{t}) : \sum_{i=1}^{n}(U \to T_i)$, $\Gamma \vdash_A \sigma(\mathbf{r}) : m.U$ and so, by rule $\to_E$ in λ^{add} we have $\Gamma \vdash_A (\sigma(\mathbf{t}))\ \sigma(\mathbf{r}) : \sum_{i=1}^{n}(m.T_i)$. Notice that $(\sigma(\mathbf{t}))\ \sigma(\mathbf{r}) = \sigma((\mathbf{t})\ \mathbf{r})$.
- $\Gamma \vdash \lambda x : U.\mathbf{t} : U \to T$ as a consequence of $\Gamma, x : U \vdash \mathbf{t} : T$ and rule $\to_I$. Then by the induction hypothesis $\Gamma, x : U \vdash_A \sigma(\mathbf{t}) : T$ and so, by rule $\to_I$ in λ^{add} we have $\Gamma \vdash_A \lambda x : U.\sigma(\mathbf{t}) : U \to T$. Notice that $\lambda x : U.\sigma(\mathbf{t}) = \sigma(\lambda x : U.\mathbf{t})$.
- $\Gamma \vdash \mathbf{t}@V : U[V/X]$ as a consequence of $\Gamma \vdash \mathbf{t} : \forall X.U$ and rule $\forall_E$. Then by the induction hypothesis $\Gamma \vdash_A \sigma(\mathbf{t}) : \forall X.U$ and so, by rule $\forall_E$ in λ^{add} we have $\Gamma \vdash_A \sigma(\mathbf{t})@V : U[V/X]$. Notice that $\sigma(\mathbf{t})@V = \sigma(\mathbf{t}@V)$.
- $\Gamma \vdash \Lambda X.\mathbf{t} : \forall X.U$ where $X \notin FV(\Gamma)$ as a consequence of $\Gamma \vdash \mathbf{t} : U$ and rule $\forall_I$. Then by the induction hypothesis $\Gamma \vdash_A \sigma(\mathbf{t}) : U$ and so, by rule $\forall_I$ in λ^{add} we have $\Gamma \vdash \Lambda X.\sigma(\mathbf{t}) : \forall X.U$. Notice that $\Lambda X.\sigma(\mathbf{t}) = \sigma(\Lambda X.\mathbf{t})$.
- $\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R$ as a consequence of $\Gamma \vdash \mathbf{t} : T$, $\Gamma \vdash \mathbf{r} : R$ and rule $\to_I$. Then by the induction hypothesis $\Gamma \vdash_A \sigma(\mathbf{t}) : T$ and $\Gamma \vdash_A \sigma(\mathbf{r}) : R$ and so, by rule $+_I$ in λ^{add} we have $\Gamma \vdash_A \sigma(\mathbf{t}) + \sigma(\mathbf{r}) : T + R$. Notice that $\sigma(\mathbf{t}) + \sigma(\mathbf{r}) = \sigma(\mathbf{t} + \mathbf{r})$.
- $\Gamma \vdash \alpha.\mathbf{t} : \lfloor\alpha\rfloor.T$ as a consequence of $\Gamma \vdash \mathbf{t} : T$ and rule sI. Then by the induction hypothesis $\Gamma \vdash_A \sigma(\mathbf{t}) : T$. Cases:
  - If $\alpha \geq 2$, then by rule $+_I$ $\lfloor\alpha\rfloor$-times, we have $\Gamma \vdash_A \sum_{i=1}^{\lfloor\alpha\rfloor} \sigma(\mathbf{t}) : T$. Notice that $\sum_{i=1}^{\lfloor\alpha\rfloor} \sigma(\mathbf{t}) = \sigma(\alpha.\mathbf{t})$.
  - If $1 \leq \alpha \leq 2$, then notice that $\sigma(\mathbf{t}) = \sum_{i=1}^{1} \sigma(\mathbf{t}) = \sigma(\alpha.\mathbf{t})$.
  - If $\alpha \leq 1$, then notice that $\sigma(\alpha.\mathbf{t}) = \mathbf{0}$ and by rule $ax_{\overline{0}}$, $\Gamma \vdash_A \mathbf{0} : 0.T$.

□

Appendix F

Proofs from Chapter 7

F.1 Proof of Corollary 7.2.3

Corollary 7.2.3 (Base terms)**.** If $\Gamma \vdash \mathbf{b} : T$, then exists V such that $\Gamma \vdash \mathbf{b} : V$ and $V \preceq T$.

Proof. Structural induction on $\mathbf{b}$.

$\mathbf{b} = x$**:** Then by 7.2.2(1), $\exists V$ and Δ such that $V \preceq T$ and $\Gamma = \Delta \cup \{x : V\}$. Then by rule ax, $\Gamma \vdash x : V$.

$\mathbf{b} = \lambda x : U.\mathbf{t}$**:** Then by 7.2.2(4), $\exists R$ such that $\Gamma, x : U \vdash \mathbf{t} : R$, with $U \to R \preceq T$. Then by rule $\to_I$, $\Gamma \vdash \lambda x : U.\mathbf{t} : U \to T$.

$\mathbf{b} = \Lambda X.\mathbf{b}'$**:** Then by Lemma 7.2.2(7), $X \notin FV(\Gamma)$, and $\exists \langle U \rangle_n$, $\langle \alpha \rangle_n$ such that $\Gamma \vdash \mathbf{b}' : \sum_{i=1}^{n} \alpha_i.U_i$ and $\sum_{i=1}^{n} \alpha_i.\forall X.U \preceq T$. By the induction hypothesis $\exists V$ such that $\Gamma \vdash \mathbf{b}' : V$ and $V \preceq \sum_{i=1}^{n} \alpha_i.U_i$. Then by rule $\forall_I$, $\Gamma \vdash \Lambda X.\mathbf{b}' : \forall X.V$. Notice that $\forall X.V \preceq \sum_{i=1}^{n} \alpha_i.\forall X.U_i \preceq T$.

□

F.2 Proof of Theorem 7.2.1

Theorem 7.2.1 (Subject reduction)**.** For any terms $\mathbf{t}$, $\mathbf{t}'$, any context Γ and any type T , if $\mathbf{t} \to \mathbf{t}'$ then $\Gamma \vdash \mathbf{t} : T \Rightarrow \Gamma \vdash \mathbf{t}' : T$.

Proof. We proceed by checking that every reduction rule preserves the type. Let $\mathbf{t} \to \mathbf{r}$ and $\Gamma \vdash \mathbf{t} : T$. To show that $\Gamma \vdash \mathbf{r} : T$, we proceed by induction on the derivation of $\mathbf{t} \to \mathbf{r}$.

Elementary rules

rule $0.\mathbf{t} \to \mathbf{0}$**:** Let $\Gamma \vdash 0.\mathbf{t} : T$. Then by Lemma 7.2.2(6), $\exists R$ such that $0.R \preceq T$ and $\Gamma \vdash \mathbf{t} : R$. Then by rule $ax_{\overline{0}}$, $\Gamma \vdash \mathbf{0} : 0.R$. We conclude using rule $\preceq$.

rule $1.\mathbf{t} \to \mathbf{t}$**:** Let $\Gamma \vdash 1.\mathbf{t} : T$. Then by Lemma 7.2.2(6), $\exists R$ such that $1.R \preceq T$ and $\Gamma \vdash \mathbf{t} : R$. Since $R \preceq 1.R \preceq T$, we conclude using rule $\preceq$.

rule $\alpha.\mathbf{0} \to \mathbf{0}$**:** Let $\Gamma \vdash \alpha.\mathbf{0} : T$. Then by Lemma 7.2.2(6), $\exists R$ such that $\Gamma \vdash \mathbf{0} : R$ and $\alpha.R \preceq T$. By Lemma 7.2.2(2), $\exists S$ and $\mathbf{t}$ such that $0.S \preceq R$ and $\Gamma \vdash \mathbf{t} : S$. Then $0.S = (\alpha \times 0).S \preceq \alpha.(0.S) \preceq \alpha.R \preceq T$. So, by rule $ax_{\overline{0}}$, $\Gamma \vdash \mathbf{0} : 0.S$, and by rule $\preceq$, $\Gamma \vdash \mathbf{0} : T$.

rule $\alpha.(\beta.\mathbf{t}) \to (\alpha \times \beta).\mathbf{t}$**:** Let $\Gamma \vdash \alpha.(\beta.\mathbf{t}) : T$. Then by Lemma 7.2.2(6), $\exists R$ such that $\alpha.R \preceq T$ and $\Gamma \vdash \beta.\mathbf{t} : R$, and so, by Lemma 7.2.2(6) again, $\exists S$ such that $\beta.S \preceq R$ and $\Gamma \vdash \mathbf{t} : S$. Using rule s_I we can derive $\Gamma \vdash (\alpha \times \beta).\mathbf{t} : (\alpha \times \beta).S$. Notice that $(\alpha \times \beta).S \preceq \alpha.(\beta.S) \preceq \alpha.R \preceq T$, so we conclude using rule $\preceq$.

rule $\alpha.(\mathbf{t} + \mathbf{r}) \to \alpha.\mathbf{t} + \alpha.\mathbf{r}$**:** Let $\Gamma \vdash \alpha.(\mathbf{t} + \mathbf{r}) : T$. Then by Lemma 7.2.2(6), $\exists R$ such that $\alpha.R \preceq T$ and $\Gamma \vdash \mathbf{t} + \mathbf{r} : R$. Then by Lemma 7.2.2(5), $\exists S_1$ and S_2 such that $S_1 + S_2 \preceq R$, $\Gamma \vdash \mathbf{t} : S_1$ and $\Gamma \vdash \mathbf{r} : S_2$. Then using rule s_I, one has $\Gamma \vdash \alpha.\mathbf{t} : \alpha.S_1$ and $\Gamma \vdash \alpha.\mathbf{r} : \alpha.S_2$, from where, using rule $+_I$, one can derive $\Gamma \vdash \alpha.\mathbf{t} + \alpha.\mathbf{r} : \alpha.S_1 + \alpha.S_2$. Notice that $\alpha.S_1 + \alpha.S_2 \preceq \alpha.(S_1 + S_2) \preceq \alpha.R \preceq T$, so we conclude by rule $\preceq$.

rule $\mathbf{t} + \mathbf{0} \to \mathbf{t}$**:** Let $\Gamma \vdash \mathbf{t} + \mathbf{0} : T$. Then by Lemma 7.2.2(5), $\exists R_1$ and R_2 such that $R_1 + R2 \preceq T$, $\Gamma \vdash \mathbf{t} : R_1$ and $\Gamma \vdash \mathbf{0} : R_2$. By Lemma 7.2.2(2), $\exists S$ such that $0.S \preceq R_2$. Notice that $R_1 \preceq R_1 + 0.S \preceq R_1 + R_2 \preceq T$, so we conclude using rule $\preceq$.

Factorisation rules

rule $\alpha.\mathbf{t} + \beta.\mathbf{t} \to (\alpha + \beta).\mathbf{t}$**:** Let $\Gamma \vdash \alpha.\mathbf{t} + \beta.\mathbf{t} : T$. By Lemma 7.2.2(5), there exist T_1 and T_2 such that $\Gamma \vdash \alpha.\mathbf{t} : T_1$ and $\Gamma \vdash \beta.\mathbf{t} : T_2$, where $T_1 + T_2 \preceq T$. Then by Lemma 7.2.2(6), there exist R_1 and R_2 such that $\Gamma \vdash \mathbf{t} : R_1$ and $\Gamma \vdash \mathbf{t} : R_2$, with $\alpha.R_1 \preceq T_1$ and $\beta.R_2 \preceq T_2$. Since R_1 and R_2 are both types for $\mathbf{t}$, by Lemma 7.2.4 $\exists S$ such that $S \preceq R_1$, $S \preceq R_2$, and $\Gamma \vdash \mathbf{t} : S$. Using s_I, we obtain $\Gamma \vdash (\alpha + \beta).\mathbf{t} : (\alpha + \beta).S$. Notice that $(\alpha + \beta).S \preceq \alpha.S + \beta.S \preceq \alpha.R_1 + \beta.R_2 \preceq T_1 + T_2 \preceq T$. We conclude with rule $\preceq$.

rules $\alpha.\mathbf{t} + \mathbf{t} \to (\alpha + 1).\mathbf{t}$ **and rule** $\mathbf{t} + \mathbf{t} \to (1 + 1).\mathbf{t}$**:** Analogous to previous case.

Application rules

rule $(\mathbf{t} + \mathbf{r})\ \mathbf{u} \to (\mathbf{t})\ \mathbf{u} + (\mathbf{r})\ \mathbf{u}$**:** Let $\Gamma \vdash (\mathbf{t} + \mathbf{r})\ \mathbf{u} : T$, then by Lemma 7.2.2(3), $\exists n$, m, δ, k, $\langle X \rangle_k$, $\langle \alpha \rangle_n$, $\langle \beta \rangle_m$, U, $\langle T \rangle_n$, $\langle V \rangle_m$, $\langle \langle W \rangle_{m+\delta} \rangle_m$, where $\forall V_j$, $\exists j_1, \ldots, j_k$ / $U\langle [W_j/X] \rangle_k = V_j$, and such that

$$\Gamma \vdash \mathbf{t} + \mathbf{r} : \sum_{i=1}^{n} \alpha_i.(\langle \forall X \rangle_k.(U \to T_i))@\langle \sum_{j=1}^{m+\delta} W_j \rangle_k \qquad \text{and} \qquad \Gamma \vdash \mathbf{u} : \sum_{j=1}^{m} \beta_j.V_j,$$

$$\text{with} \qquad \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \times \beta_j.T_i\langle [W_j/X] \rangle_k \preceq T.$$

Then by Lemma 7.2.2(5), $\exists R$ and S such that

$$\Gamma \vdash \mathbf{t} : R \qquad \text{and} \qquad \Gamma \vdash \mathbf{r} : S,$$

$$\text{with} \qquad R + S \preceq \sum_{i=1}^{n} \alpha_i.(\langle \forall X \rangle_k.(U \to T_i))@\langle \sum_{j=1}^{m+\delta} W_j \rangle_k.$$

Then $\exists N_1, N_2 \subseteq \{1, \ldots, n\}$, such that

$$R \preceq \sum_{i \in N_1 \setminus N_2} \alpha_i.(\langle \forall X \rangle_k.(U \to T_i))@\langle \sum_{j=1}^{m+\delta} W_j \rangle_k + \sum_{i \in N_1 \cap N_2} \gamma_i.(\langle \forall X \rangle_k.(U \to T_i))@\langle \sum_{j=1}^{m+\delta} W_j \rangle_k$$

and

$$S \preceq \sum_{i \in N_2 \setminus N_1} \alpha_i.(\langle \forall X \rangle_k.(U \to T_i))@\langle \sum_{j=1}^{m+\delta} W_j \rangle_k + \sum_{i \in N_1 \cap N_2} \phi_i.(\langle \forall X \rangle_k.(U \to T_i))@\langle \sum_{j=1}^{m+\delta} W_j \rangle_k,$$

where $\forall i \in N_1 \cap N_2$, $\gamma_i + \phi_i = \alpha_i$.

First we give the bigger types to $\mathbf{t}$ and $\mathbf{r}$, then we use $\rightarrow_E$ to derive

$$\Gamma \vdash (\mathbf{t})\ \mathbf{u}\colon \sum_{i\in N_1\setminus N_2}\sum_{j=1}^{m}\alpha_i\times\beta_j.T_i\langle[W_j/X]\rangle_k + \sum_{i\in N_1\cap N_2}\sum_{j=1}^{m}\gamma_i\times\beta_j.T_i\langle[W_j/X]\rangle_k$$

and

$$\Gamma \vdash (\mathbf{r})\ \mathbf{u}\colon \sum_{i\in N_2\setminus N_1}\sum_{j=1}^{m}\alpha_i\times\beta_j.T_i\langle[W_j/X]\rangle_k + \sum_{i\in N_1\cap N_2}\sum_{j=1}^{m}\phi_i\times\beta_j.T_i\langle[W_j/X]\rangle_k,$$

then we sum them up using rule $+_I$, obtaining

$$\begin{aligned}\Gamma \vdash (\mathbf{t})\ \mathbf{u} + (\mathbf{r})\ \mathbf{u}\colon &\sum_{i\in N_1\setminus N_2}\sum_{j=1}^{m}\alpha_i\times\beta_j.T_i\langle[W_j/X]\rangle_k + \sum_{i\in N_1\cap N_2}\sum_{j=1}^{m}\gamma_i\times\beta_j.T_i\langle[W_j/X]\rangle_k +\\ &\sum_{i\in N_2\setminus N_1}\sum_{j=1}^{m}\alpha_i\times\beta_j.T_i\langle[W_j/X]\rangle_k + \sum_{i\in N_1\cap N_2}\sum_{j=1}^{m}\phi_i\times\beta_j.T_i\langle[W_j/X]\rangle_k\end{aligned}$$

Notice that this type is equivalent to $\sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_i\times\beta_j.T_i\langle[W_j/X]\rangle_k \preceq T$. We conclude by rule $\preceq$.

rule $(\mathbf{u})\ (\mathbf{t}+\mathbf{r}) \rightarrow (\mathbf{u})\ \mathbf{t} + (\mathbf{u})\ \mathbf{r}$**:** Analogous to previous case.

rule $(\alpha.\mathbf{t})\ \mathbf{r} \rightarrow \alpha.(\mathbf{t})\ \mathbf{r}$**:** Let $\Gamma \vdash (\alpha.\mathbf{t})\ \mathbf{r}\colon T$. Then by Lemma 7.2.2(3), $\exists n$, m, δ, k, $\langle X\rangle_k$, $\langle\alpha\rangle_n$, $\langle\beta\rangle_m$, U, $\langle T\rangle_n$, $\langle V\rangle_m$, $\langle\langle W\rangle_{m+\delta}\rangle_m$, where $\forall V_j$, $\exists j_1,\ldots,j_k$ / $U\langle[W_j/X]\rangle_k = V_j$, and such that

$$\Gamma \vdash \alpha.\mathbf{t}\colon \sum_{i=1}^{n}\alpha_i.(\langle\forall X\rangle_k.(U\rightarrow T_i))@\langle\sum_{j=1}^{m+\delta}W_j\rangle_k \qquad \text{and} \qquad \Gamma\vdash\mathbf{u}\colon\sum_{j=1}^{m}\beta_j.V_j,$$

$$\text{with} \qquad \sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_i\times\beta_j.T_i\langle[W_j/X]\rangle_k \preceq T.$$

Then by Lemma 7.2.2(6), $\exists R$ such that

$$\Gamma\vdash\mathbf{t}\colon R \qquad \text{with} \qquad \alpha.R \preceq \sum_{i=1}^{n}\alpha_i.(\langle\forall X\rangle_k.(U\rightarrow T_i))@\langle\sum_{j=1}^{m+\delta}W_j\rangle_k.$$

Then, $\exists\langle\gamma\rangle_m$ such that $R \preceq \sum_{i=1}^{n}\gamma_i.(\langle\forall X\rangle_k.(U\rightarrow T_i))@\langle\sum_{j=1}^{m+\delta}W_j\rangle_k$, where $\forall i$, $\alpha\times\gamma_i = \alpha_i$. Taking the bigger type for $\mathbf{t}$ (by rule $\preceq$), we can use $\rightarrow_E$ to derive

$$\Gamma\vdash(\mathbf{t})\ \mathbf{r}\colon\sum_{i=1}^{n}\sum_{j=1}^{m}\gamma_i\times\beta_j.T_i\langle[W_j/X]\rangle_k.$$

And then, using s_I,

$$\Gamma\vdash\alpha.(\mathbf{t})\ \mathbf{t}\colon\alpha.\sum_{i=1}^{n}\sum_{j=1}^{m}\gamma_i\times\beta_j.T_i\langle[W_j/X]\rangle_k.$$

Notice that $\alpha.\sum_{i=1}^{n}\sum_{j=1}^{m}\gamma_i\times\beta_j.T_i\langle[W_j/X]\rangle_k \equiv \sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_i\times\beta_j.T_i\langle[W_j/X]\rangle_k \preceq T$. We conclude by rule $\preceq$.

rule $(\mathbf{r})\ (\alpha.\mathbf{t}) \rightarrow \alpha.(\mathbf{r})\ \mathbf{t}$**:** Analogous to previous case.

rule $(\mathbf{0})\ \mathbf{t} \rightarrow \mathbf{0}$**:** Let $\Gamma\vdash(\mathbf{0})\ \mathbf{t}\colon T$. Then by Lemma 7.2.2(3), $\exists n$, m, δ, k, $\langle X\rangle_k$, $\langle\alpha\rangle_n$, $\langle\beta\rangle_m$, U, $\langle T\rangle_n$, $\langle V\rangle_m$, $\langle\langle W\rangle_{m+\delta}\rangle_m$, where $\forall V_j$, $\exists j_1,\ldots,j_k$ / $U\langle[W_j/X]\rangle_k = V_j$, and such that

$$\Gamma\vdash\mathbf{0}\colon\sum_{i=1}^{n}\alpha_i.(\langle\forall X\rangle_k.(U\rightarrow T_i))@\langle\sum_{j=1}^{m+\delta}W_j\rangle_k \qquad \text{and} \qquad \Gamma\vdash\mathbf{u}\colon\sum_{j=1}^{m}\beta_j.V_j,$$

$$\text{with} \qquad \sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_i \times \beta_j.T_i\langle[W_j/X]\rangle_k \preceq T.$$

By Lemma 7.2.2(2), $\exists R$ such that

$$0.R \preceq \sum_{i=1}^{n}\alpha_i.(\langle\forall X\rangle_k.(U \to T_i))@\langle\sum_{j=1}^{m+\delta} W_j\rangle_k.$$

Then $\exists A \subseteq \{1,\ldots,n\}$ such that

$$\sum_{i=1}^{n}\alpha_i.(\langle\forall X\rangle_k.(U \to T_i))@\langle\sum_{j=1}^{m+\delta} W_j\rangle_k \equiv \sum_{i\in A} 0.(\langle\forall X\rangle_k.(U \to T_i))@\langle\sum_{j=1}^{m+\delta} W_j\rangle_k$$

Then by rule $\preceq$,

$$\Gamma \vdash \mathbf{0}\colon \sum_{i\in A} 0.(\langle\forall X\rangle_k.(U \to T_i))@\langle\sum_{j=1}^{m+\delta} W_j\rangle_k$$

and so by rule $\to_E$,

$$\Gamma \vdash (\mathbf{0})\ \mathbf{t}\colon \sum_{i\in A}\sum_{j=1}^{m} 0 \times \beta_j.T_i\langle[W_j/X]\rangle_k$$

Then by rule $ax_{\overline{0}}$, $\Gamma \vdash \mathbf{0}\colon 0.\sum_{i\in A}\sum_{j=1}^{m} 0 \times \beta_j.T_i\langle[W_j/X]\rangle_k$. Notice that $0.\sum_{i\in A}\sum_{j=1}^{m} 0 \times \beta_j.T_i\langle[W_j/X]\rangle_k \equiv \sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_i \times \beta_j.T_i\langle[W_j/X]\rangle_k \preceq T$. We conclude by rule $\preceq$.

rule $(\mathbf{t})$ $\mathbf{0} \to \mathbf{0}$**:** Analogous to previous case.

Type-linearity rules

rule $(\alpha.\mathbf{t})@(\sum_{i=1}^{n} U_i) \to \alpha.\mathbf{t}@(\sum_{i=1}^{n} U_i)$**:** Let $\Gamma \vdash (\alpha.\mathbf{t})@(\sum_{i=1}^{n} U_i)\colon T$. Then by Lemma 7.2.2(8), $\exists\langle V\rangle_m$, X, $\langle\beta\rangle_m$ such that

$$\Gamma \vdash \alpha.\mathbf{t}\colon \sum_{j=1}^{m}\beta_j.\forall X.V_j \qquad \text{and} \qquad \sum_{j=1}^{m}\beta_j.(\forall X.V_j)@(\sum_{i=1}^{n} U_i) \preceq T.$$

Then by Lemma 7.2.2(6), $\exists S$ such that

$$\Gamma \vdash \mathbf{t}\colon S \qquad \text{and} \qquad \alpha.S \preceq \sum_{j=1}^{m}\beta_j.\forall X.V_j.$$

By Lemma 7.1.2, $S \equiv \sum_{o=1}^{h}\delta_o.\forall X.W_o$, so by rule $\preceq$,

$$\Gamma \vdash \mathbf{t}\colon \sum_{o=1}^{h}\delta_o.W_o.$$

Thus, by Lemma 7.1.3, $\forall o$, $\exists W'_o$ such that $W_o \equiv \forall X.W'_o$. So, using rule $@_I$, we can derive

$$\Gamma \vdash \mathbf{t}@(\sum_{i=1}^{n} U_i)\colon \sum_{o=1}^{h}\delta_o.(\forall X.W'_o)@(\sum_{i=1}^{n} U_i),$$

and using the rule s_I on this sequent, we derive

$$\Gamma \vdash \alpha.\mathbf{t}@(\sum_{i=1}^{n} U_i)\colon \alpha.\sum_{o=1}^{h}\delta_o.(\forall X.W'_o)@(\sum_{i=1}^{n} U_i).$$

Note that

$$\alpha.\sum_{o=1}^{h}\delta_o.(\forall X.W'_o)@(\sum_{i=1}^{n}U_i) \equiv \sum_{o=1}^{h}\alpha\times\delta_o.(\forall X.W'_o)@(\sum_{i=1}^{n}U_i)$$

and since $\sum_{o=1}^{h}\alpha\times\delta_o.\forall X.W'_o \equiv \alpha.S \preceq \sum_{j=1}^{m}\beta_j.\forall X.V_j$, by subtyping rules Cx_+, Cx_s, $Cx_@$ and Tr,

$$\sum_{o=1}^{h}\alpha\times\delta_o.(\forall X.W'_o)@(\sum_{i=1}^{n}U_i) \preceq \sum_{j=1}^{m}\beta_j.(\forall X.V_j)@(\sum_{i=1}^{n}U_i) \preceq T.$$

We conclude by rule $\preceq$.

rule $(\mathbf{t}+\mathbf{r})@(\sum_{i=1}^{n}U_i)\to\mathbf{t}@(\sum_{i=1}^{n}U_i)+\mathbf{r}@(\sum_{i=1}^{n}U_i)$**:** Let $\Gamma\vdash(\mathbf{t}+\mathbf{r})@(\sum_{i=1}^{n}U_i):T$. Then by Lemma 7.2.2(8), $\exists\langle V\rangle_m$, X, $\langle\beta\rangle_m$ such that

$$\Gamma\vdash\mathbf{t}+\mathbf{r}:\sum_{j=1}^{m}\beta_j.\forall X.V_j \qquad\text{and}\qquad \sum_{j=1}^{m}\beta_j.(\forall X.V_j)@(\sum_{i=1}^{n}U_i)\preceq T.$$

Then by Lemma 7.2.2(5), $\exists S_1,S_2$ such that

$$\Gamma\vdash\mathbf{t}:S_1,\qquad \Gamma\vdash\mathbf{r}:S_2\qquad\text{and}\qquad S_1+S_2\preceq\sum_{j=1}^{m}\beta_j.\forall X.V_j.$$

By Lemma 7.1.2, $S_1\equiv\sum_{o=1}^{h^1}\delta_o^1.W_o^1$ and $S_2\equiv\sum_{o=1}^{h^2}\delta_o^2.W_o^2$. So, by Lemma 7.1.3, for $i=1,2$, $\forall o$, $\exists W_o'^i$ such that $W_o^i\equiv\forall X.W_o'^i$. Then, using rule $@_I$,

$$\Gamma\vdash\mathbf{t}@(\sum_{i=1}^{n}U_i):\sum_{o=1}^{h^1}\delta_o^1.(\forall X.W_o'^1)@(\sum_{i=1}^{n}U_i)\quad\text{and}\quad\Gamma\vdash\mathbf{r}@(\sum_{i=1}^{n}U_i):\sum_{o=1}^{h^2}\delta_o^2.(\forall X.W_o'^2)@(\sum_{i=1}^{n}U_i).$$

By rule $+_I$,

$$\Gamma\vdash\mathbf{t}@(\sum_{i=1}^{n}U_i)+\mathbf{r}@(\sum_{i=1}^{n}U_i):\sum_{o=1}^{h^1}\delta_o^1.(\forall X.W_o'^1)@(\sum_{i=1}^{n}U_i)+\sum_{o=1}^{h^2}\delta_o^2.(\forall X.W_o'^2)@(\sum_{i=1}^{n}U_i).$$

Notice that, since

$$\sum_{o=1}^{h^1}\delta_o^1.\forall X.W_o'^1+\sum_{o=1}^{h^2}\delta_o^2.\forall X.W_o'^2\equiv S_1+S_2\preceq\sum_{j=1}^{m}\beta_j.\forall X.V_j,$$

using Tr and Cx subtyping rules,

$$\sum_{o=1}^{h^1}\delta_o^1.(\forall X.W_o'^1)@(\sum_{i=1}^{n}U_i)+\sum_{o=1}^{h^2}\delta_o^2.(\forall X.W_o'^2)@(\sum_{i=1}^{n}U_i)\preceq\sum_{j=1}^{m}\beta_j.(\forall X.V_j)@(\sum_{i=1}^{n}U_i)\preceq T.$$

We conclude by rule $\preceq$.

Type-distributivity rule

rule $((\langle\Lambda X\rangle_k.\langle\lambda x:U\rangle_n.\mathbf{t})@\langle\sum_{i=1}^{m}W_i\rangle_k)\ \langle\mathbf{b}^V\rangle_n\to(\langle\lambda x:V\rangle_n.\mathbf{t}\langle[W_j/X]\rangle_k)\ \langle\mathbf{b}\rangle_n$**:**

Let $\Gamma\vdash((\langle\Lambda X\rangle_k.\langle\lambda x:U\rangle_n.\mathbf{t})@\langle\sum_{i=1}^{m}W_i\rangle_k)\ \langle\mathbf{b}^V\rangle_n:T$, where $U\langle[W_j/X]\rangle_k=V$. Then, by Lemma 7.2.2(3), $\exists p^n$, m'^n, k'^n, $\langle\delta^n\rangle_{k'^n}$, $\langle Y^n\rangle_{k'^n}$, $\langle\alpha^n\rangle_{p^n}$, $\langle\beta^n\rangle_{m'^n}$, U'^n, $\langle T^n\rangle_{p^n}$, $\langle V'^n\rangle_{m'^n}$, $\langle\langle W'^n\rangle_{m'^n+\delta^n}\rangle_{m'^n}$, where $\forall V_o'^n$, $\exists o_1,\ldots,o_{k'^n}\ /\ U'^n\langle[W_o'^n/Y^n]\rangle_{k'^n}=V_o'^n$, and such that

$$\Gamma\vdash((\langle\Lambda X\rangle_k.\langle\lambda x:U\rangle_n.\mathbf{t})@\langle\sum_{i=1}^{m}W_i\rangle_k)\ \langle\mathbf{b}\rangle_{n-1}:\sum_{j=1}^{p^n}\alpha_j^n.(\langle\forall Y^n\rangle_{k'^n}.(U'^n\to T_i^n))@\langle\sum_{o=1}^{m'^n+\delta^n}W_o'^n\rangle_{k'^n},$$

$$\Gamma \vdash \mathbf{b}_n : \sum_{o=1}^{m'^n} \beta_o^n.V_o'^n \qquad \text{and} \qquad \sum_{j=1}^{p^n}\sum_{o=1}^{m'^n} \alpha_j^n \times \beta_o^n.T_i^n\langle [W_o'^n/Y^n]\rangle_{k'^n} \preceq T$$

Using the same Lemma, we get for $i = 2, \ldots, n-1$, $\exists p^i$, m'^i, k'^i, $\langle \delta^i \rangle_{k'^i}$, $\langle Y^i \rangle_{k'^i}$, $\langle \alpha^i \rangle_{p^i}$, $\langle \beta^i \rangle_{m'^i}$, U'^i, $\langle T^i \rangle_{p^i}$, $\langle V'^i \rangle_{m'^i}$, $\langle \langle W'^i \rangle_{m'^i+\delta^i} \rangle_{m'^i}$, where $\forall V_o'^i$, $\exists o_1, \ldots, o_{k'^i}$ / $U'^i\langle [W_o'^i/Y^i]\rangle_{k'^i} = V_o'^i$, and such that

$$\Gamma \vdash ((\langle \Lambda X \rangle_k.\langle \lambda x : U \rangle_n.\mathbf{t})@\langle \sum_{i=1}^{m} W_i \rangle_k)\ \langle \mathbf{b} \rangle_{n-1} : \sum_{j=1}^{p^i} \alpha_j^i.(\langle \forall Y^i \rangle_{k'^i}.(U'^i \to T_i^i))@\langle \sum_{o=1}^{m'^i+\delta^i} W_o'^i \rangle_{k'^i},$$

$$\Gamma \vdash \mathbf{b}_i : \sum_{o=1}^{m'^i} \beta_o^i.V_o'^i \qquad \text{and}$$

$$\sum_{j=1}^{p^i}\sum_{o=1}^{m'^i} \alpha_j^i \times \beta_o^i.T_i^i\langle [W_o'^i/Y^i]\rangle_{k'^i} \preceq \sum_{j=1}^{p^{i+1}} \alpha_j^{i+1}.(\langle \forall Y^{i+1} \rangle_{k'^{i+1}}.(U'^{i+1} \to T_i^{i+1}))@\langle \sum_{o=1}^{m'^{i+1}+\delta^{i+1}} W_o'^{i+1} \rangle_{k'^{i+1}}.$$

Then, one more time the same Lemma from $i = 2$, we end with $\exists p$, m', k', $\langle \delta \rangle_{k'}$, $\langle Y \rangle_{k'}$, $\langle \alpha \rangle_p$, $\langle \beta \rangle_{m'}$, U', $\langle T \rangle_p$, $\langle V' \rangle_{m'}$, $\langle \langle W' \rangle_{m'+\delta} \rangle_{m'}$, where $\forall V_o'$, $\exists o_1, \ldots, o_{k'}$ / $U'\langle [W_o'/Y]\rangle_{k'} = V_o'$, and such that

$$\Gamma \vdash (\langle \Lambda X \rangle_k.\langle \lambda x : U \rangle_n.\mathbf{t})@\langle \sum_{i=1}^{m} W_i \rangle_k : \sum_{j=1}^{p} \alpha_j.(\langle \forall Y \rangle_{k'}.(U' \to T_i))@\langle \sum_{o=1}^{m'+\delta} W_o' \rangle_{k'},$$

$$\Gamma \vdash \mathbf{b}_1 : \sum_{o=1}^{m'} \beta_o.V_o' \qquad \text{and}$$

$$\sum_{j=1}^{p}\sum_{o=1}^{m'} \alpha_j \times \beta_o.T_i\langle [W_o'/Y]\rangle_{k'} \preceq \sum_{j=1}^{p^2} \alpha_j^2.(\langle \forall Y^2 \rangle_{k'^2}.(U'^2 \to T_i^2))@\langle \sum_{o=1}^{m'^2+\delta^2} W_o'^2 \rangle_{k'^2}.$$

By Lemma 7.2.2(4) k-times, and using rule $\preceq$ and Lemma 7.1.3, we have that $\exists \langle \gamma \rangle_r$, $\langle Z \rangle_k$ and $\langle U'' \rangle_r$ such that

$$\Gamma \vdash \langle \Lambda X \rangle_k.\langle \lambda x : U \rangle_n.\mathbf{t} : \sum_{g=1}^{r} \gamma_g.\langle \forall Z \rangle_k.U_g'' \qquad \text{and}$$

$$\sum_{g=1}^{r} \gamma_g.(\langle \forall Z \rangle_k.U_g'')@\langle \sum_{j=1}^{m} W_i \rangle_k \preceq \sum_{j=1}^{p} \alpha_j.(\langle \forall Y \rangle_{k'}.(U' \to T_i))@\langle \sum_{o=1}^{m'+\delta} W_o' \rangle_{k'}.$$

By Corollary 7.2.3, $\exists V''$ such that

$$\Gamma \vdash \langle \Lambda X \rangle_k.\langle \lambda x : U \rangle_n.\mathbf{t} : V'' \qquad \text{and} \qquad V'' \preceq \sum_{g=1}^{r} \gamma_g.\langle \forall Z \rangle_k.U_g''.$$

Then, by Lemma 7.1.3, $\exists V'''$ such that $V'' \equiv \langle \forall Z \rangle_k.V'''$. Now, by Lemma 7.2.2(7), k-times, and using rule $\preceq$ and Lemma 7.1.3, we have that $\exists \langle U''' \rangle_{n'}$, $\langle \alpha' \rangle_{n'}$ such that $\langle X \rangle_k \notin FV(\Gamma)$, and

$$\Gamma \vdash \langle \lambda x : U \rangle_n.\mathbf{t} : \sum_{f=1}^{n'} \alpha_f'.U_f''' \qquad \text{and} \qquad \sum_{f=1}^{n'} \alpha_f'.\langle \forall X \rangle_k.U_f''' \preceq \langle \forall Z \rangle_k.V'''.$$

By Corollary 7.2.3, $\exists U^{iv}$ such that

$$\Gamma \vdash \langle \lambda x : U \rangle_n.\mathbf{t} : U^{iv} \qquad \text{and} \qquad U^{iv} \preceq \sum_{f=1}^{n'} \alpha_f'.U_f'''.$$

By Lemma 7.2.2(4), n-times, $\exists R$ such that

$$\Gamma, \langle x : U \rangle_n \vdash \mathbf{t} : R \qquad \text{and} \qquad \langle U \rangle_n \to R \preceq U^{iv}.$$

By Lemma 7.2.4, there is a principal type for $\mathbf{t}$ in the context $\Gamma, \langle x : U \rangle_n$. Chose R to be the principal type (if it is not, just take $R' \preceq R$).

By rule $\to_I$, n-times and Lemma 7.2.5,

$$\Gamma\langle [W_j/X] \rangle_k \vdash \langle \lambda x : U \rangle_n \langle [W_j/X] \rangle_k . \mathbf{t} \langle [W_j/X] \rangle_k : (\langle U \rangle_n \to R) \langle [W_j/X] \rangle_k$$

However notice that since $\langle X \rangle_k \notin FV(\Gamma)$, $\Gamma\langle [W_j/X] \rangle_k = \Gamma$. In addition $\forall h$, $U_h \langle [W_j/X] \rangle_k = V_h$, so we can write this sequent as

$$\Gamma \vdash \langle \lambda x : V \rangle_n . \mathbf{t} \langle [W_j/X] \rangle_k : \langle V \rangle_n \to R \langle [W_j/X] \rangle_k .$$

By hypothesis of the reduction rule, $\forall h$, $\Gamma \vdash \mathbf{b}_h : V_h$. So, using rule $\to_E$, n-times,

$$\Gamma \vdash (\langle \lambda x : V \rangle_n . \mathbf{t} \langle [W_j/X] \rangle_k) \ \langle \mathbf{b} \rangle_n : R \langle [W_j/X] \rangle_k .$$

We must show that $R\langle [W_j/X] \rangle_k \preceq T$.

Notice that using $\to_I$, $\forall_I$ and $@_I$ it is possible to derive

$$\Gamma \vdash (\langle \Lambda X \rangle_k . \langle \lambda x : U \rangle_n . \mathbf{t}) @ \langle \sum_{j=1}^{m} W_i \rangle_k : (\langle \forall X \rangle_k . (U \to R)) @ \langle \sum_{j=1}^{m} W_i \rangle_k .$$

Then, since $\forall h, \Gamma \vdash \mathbf{b}_h : V_h$, using $\to_E$,

$$\Gamma \vdash ((\langle \Lambda X \rangle_k . \langle \lambda x : U \rangle_n . \mathbf{t}) @ \langle \sum_{j=1}^{m} W_i \rangle_k) \ \langle \mathbf{b} \rangle_n : R \langle [W_j/X] \rangle_k .$$

Then by Lemma 7.2.4, $\exists S$ such that $S \preceq R\langle [W_j/X] \rangle_k$ and also $S \preceq T$. Notice that if we show that the only possible S is $R\langle [W_j/X] \rangle_k$, we are done since it would be $R\langle [W_j/X] \rangle_k \preceq T$. Consider S is not $R\langle [W_j/X] \rangle_k$, then $R\langle [W_j/X] \rangle_k$ cannot be the principal type of $\mathbf{t}\langle [W_j/X] \rangle_k$, which is an absurd since we chose R to be the principal type.

Beta reduction

rule $(\lambda x : U.\mathbf{t})\ \mathbf{b} \to \mathbf{t}[\mathbf{b}/x]$**:** Let $\Gamma \vdash (\lambda x : U.\mathbf{t})\ \mathbf{b} : T$. Then by Lemma 7.2.2(3), $\exists n$, m, δ, k, $\langle X \rangle_k$, $\langle \alpha \rangle_n$, $\langle \beta \rangle_m$, U', $\langle T \rangle_n$, $\langle V \rangle_m$, $\langle \langle W \rangle_{m+\delta} \rangle_m$, where $\forall V_j$, $\exists j_1, \ldots, j_k$ / $U'\langle [W_j/X] \rangle_k = V_j$, and such that

$$\Gamma \vdash \lambda x : U.\mathbf{t} : \sum_{i=1}^{n} \alpha_i . (\langle \forall X \rangle_k . (U' \to T_i)) @ \langle \sum_{j=1}^{m+\delta} W_j \rangle_k ,$$

$$\Gamma \vdash \mathbf{b} : \sum_{j=1}^{m} \beta_j . V_j \qquad \text{and} \qquad \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j . T_i \langle [W_j/X] \rangle_k \preceq T.$$

By Corollary 7.2.3, $\exists V'$ such that

$$\Gamma \vdash \lambda x : U.\mathbf{t} : V' \qquad \text{and} \qquad V' \preceq \sum_{i=1}^{n} \alpha_i . (\langle \forall X \rangle_k . (U' \to T_i)) @ \langle \sum_{j=1}^{m+\delta} W_j \rangle_k$$

Then there exists a partition $\{N_r\}$ of $\{1,\dots,n\}$ such that $\forall r$, if $r_1, r_2 \in N_r$, $T_{r_1} \equiv T_{r_2}$ and $\exists r' \,/\, \forall i \in N_{r'}$, $V' \equiv (\langle\forall X\rangle_k.(U' \to T_i))@\langle\sum_{j=1}^{m+\delta} W_j\rangle_k$, $\sum_{i\in N_{r'}} \alpha_i = 1$ and $\forall r \neq r'$, $\sum_{i\in N_r} \alpha_1 = 0$. Then for any $i \in N_{r'}$, by rule $\preceq$ one has

$$\Gamma \vdash \lambda x : U.\mathbf{t} : (\langle\forall X\rangle_k.(U' \to T_i))@\langle\sum_{j=1}^{m+\delta} W_j\rangle_k.$$

Then by Lemma 7.2.2(4), $\exists R$ such that

$$\Gamma, x : U \vdash \mathbf{t} : R \qquad \text{and} \qquad U \to R \preceq (\langle\forall X\rangle_k.(U' \to T_i))@\langle\sum_{j=1}^{m+\delta} W_j\rangle_k.$$

Then notice that $\forall i$, $m + \delta_i = 1$, and since $\Gamma \vdash \mathbf{b} : \sum_{j=1}^{m} \beta_j.V_j$, then $m = 1$ and $\delta_i = 0$. So, by Corollary 7.2.3, $\exists W'$ such that $W' \preceq \beta_1.V_1$, and so $\beta_1 = 1$. Then using rule $\preceq$ we can derive $\Gamma \vdash \mathbf{b} : V_1$. Notice that for any $i' \in N_{r'}$, $(\sum_{i\in N_{r'}} \alpha_i \times 1).T_{i'}\langle[W_1/X]\rangle_k \preceq \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \times \beta_j.T_i\langle[W_j/X]\rangle_k$. Thus, in order to simplify notation, we write $W_{1_i} = W_i$ and $V_1 = V$. Then, since $\sum_{i\in N_{r'}} \alpha_i \times 1 = 1$, we have

$$\forall i \in N_{r'},\ T_i\langle[W/X]\rangle_k \preceq T.$$

Also, we have

$$U \to R \preceq (\langle\forall X\rangle_k.(U' \to T_i))@\langle W\rangle_k \equiv V \to T_i\langle[W/X]\rangle_k.$$

Then notice that the only possibility is $U \equiv V$ and $R \preceq T_i\langle[W/X]\rangle_k$. Then, using $\preceq$,

$$\Gamma, x : U \vdash \mathbf{t} : T_i\langle[W/X]\rangle_k \qquad \text{and} \qquad \Gamma \vdash \mathbf{b} : U.$$

So by Lemma 7.2.5, $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T_i\langle[W/X]\rangle_k$. We conclude by rule $\preceq$.

rule $(\Lambda X.\mathbf{t})@U \to \mathbf{t}[U/X]$**:** Let $\Gamma \vdash (\Lambda X.\mathbf{t})@U : T$. Then by Lemma 7.2.2(8), $\exists\langle V\rangle_n$, Y and $\langle\alpha\rangle_n$ such that $\Gamma \vdash \Lambda X.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall Y.V_i$, and $\sum_{i=1}^{n} \alpha_i.(\forall Y.V_i)@U \preceq T$. Then by Lemma 7.2.2(7), $X \notin FV(\Gamma)$ and $\exists\langle W\rangle_m$ and $\langle\beta\rangle_m$ such that $\Gamma \vdash \mathbf{t} : \sum_{j=1}^{m} \beta_j.W_j$ and $\sum_{j=1}^{m} \beta_j.\forall X.W_j \preceq \sum_{i=1}^{n} \alpha_i.\forall Y.V_i$. Then by Lemma 7.2.5, $\Gamma[U/X] \vdash \mathbf{t}[U/X] : \sum_{j=1}^{m} \beta_j.W_j[U/X]$. Since $X \notin FV(\Gamma)$, $\Gamma[U/X] = \Gamma$. In addition, notice that $\sum_{j=1}^{m} \beta_j.W_j[U/X] \equiv \sum_{j=1}^{m} \beta_j.(\forall X.W_j)@U \preceq \sum_{i=1}^{n} \alpha_i.(\forall Y.V_i)@U \preceq T$. Then, by rule $\preceq$, $\Gamma \vdash \mathbf{t}[U/X] : T$.

AC equivalences

Commutativity: Let $\Gamma \vdash \mathbf{t} + \mathbf{r} : T$. Then by Lemma 7.2.2(5), $\exists R, S$ such that $\Gamma \vdash \mathbf{t} : R$, $\Gamma \vdash \mathbf{r} : S$ and $R + S \preceq T$. Then by rule $+_I$, $\Gamma \vdash \mathbf{r} + \mathbf{t} : S + R$. Notice that $S + R \equiv R + S \preceq T$. We conclude by rule $\preceq$.

Associativity: Let $\Gamma \vdash (\mathbf{t} + \mathbf{r}) + \mathbf{u} : T$. Then by Lemma 7.2.2(5), $\exists R, S$ such that $\Gamma \vdash \mathbf{t} + \mathbf{r} : R$ and $\Gamma \vdash \mathbf{u} : S$, with $R + S \preceq T$. Then by Lemma 7.2.2(5), $\exists R_1, R_2$ such that $\Gamma \vdash \mathbf{t} : R_1$, $\Gamma \vdash \mathbf{r} : R_2$ and $R_1 + R_2 \preceq R$. So, by using rule $+_I$ twice, we get $\Gamma \vdash \mathbf{t} + (\mathbf{r} + \mathbf{u}) : R_1 + (R_2 + S)$. Notice that $R_1 + (R_2 + S) \equiv (R_1 + R_2) + S \preceq R + S \preceq T$. We conclude by rule $\preceq$.

Contextual rules Let $\mathbf{t} \to \mathbf{r}$ and assume as induction hypothesis that for any context Γ and type T, if $\Gamma \vdash \mathbf{t} : T$ then $\Gamma \vdash \mathbf{r} : T$.

(**t**) $\mathbf{u} \to$ (**r**) **u:** Let $\Gamma \vdash$ (**t**) $\mathbf{u} : T$. Then by Lemma 7.2.2(3), $\exists n,\ m,\ \delta,\ k,\ \langle X\rangle_k,\ \langle\alpha\rangle_n,\ \langle\beta\rangle_m,\ U',\ \langle T\rangle_n,\ \langle V\rangle_m,\ \langle\langle W\rangle_{m+\delta}\rangle_m$, where $\forall V_j,\ \exists j_1, \ldots, j_k\ /\ U'\langle [W_j/X]\rangle_k = V_j$, and such that $\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.(\langle\forall X\rangle_k.(U' \to T_i))@\langle\sum_{j=1}^{m+\delta} W_j\rangle_k$, $\Gamma \vdash \mathbf{u} : \sum_{j=1}^{m} \beta_j.V_j$ and $\sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \times \beta_j.T_i\langle [W_j/X]\rangle_k \preceq T$. Then by the induction hypothesis one has $\Gamma \vdash \mathbf{r} : \sum_{i=1}^{n} \alpha_i.(\langle\forall X\rangle_k.(U' \to T_i))@\langle\sum_{j=1}^{m+\delta} W_j\rangle_k$, so using rules $\to_E$ and $\preceq$, we get $\Gamma \vdash$ (**r**) $\mathbf{u} : T$.

(**u**) $\mathbf{t} \to$ (**u**) **r:** Analogous to previous case.

$\mathbf{t} + \mathbf{u} \to \mathbf{r} + \mathbf{u}$**:** Let $\Gamma \vdash \mathbf{t} + \mathbf{u} : T$. Then by Lemma 7.2.2(5), $\exists R, S$ such that $\Gamma \vdash \mathbf{t} : R$ and $\Gamma \vdash \mathbf{u} : S$, with $R + S \preceq T$. Then by the induction hypothesis $\Gamma \vdash \mathbf{r} : R$, so using rule $+_I$, $\Gamma \vdash \mathbf{r} + \mathbf{u} : R + S$. We conclude by rule $\preceq$.

$\mathbf{u} + \mathbf{t} \to \mathbf{u} + \mathbf{r}$**:** Analogous to previous case.

$\alpha.\mathbf{t} \to \alpha.\mathbf{r}$**:** Let $\Gamma \vdash \alpha.\mathbf{t} : T$. Then by Lemma 7.2.2(6), $\exists R$ such that $\Gamma \vdash \mathbf{t} : R$ and $\alpha.R \preceq T$. Then by the induction hypothesis $\Gamma \vdash \mathbf{r} : R$. So using rule s_I, $\Gamma \vdash \alpha.\mathbf{r} : \alpha.R$. We conclude by rule $\preceq$.

$\lambda x.\mathbf{t} \to \lambda x.\mathbf{r}$**:** Let $\Gamma \vdash \lambda x.\mathbf{t} : T$. Then by Lemma 7.2.2(4), $\Gamma, x : U \vdash \mathbf{t} : R$, with $U \to R \preceq T$. So, by the induction hypothesis $\Gamma, x : U \vdash \mathbf{r} : R$. Using rule $\to_I$, $\Gamma \vdash \lambda x.\mathbf{r} : U \to R$. We conclude by rule $\preceq$.

$\Lambda X.\mathbf{t} \to \Lambda X.\mathbf{r}$**:** Let $\Gamma \vdash \Lambda X.\mathbf{t} : T$. Then by Lemma 7.2.2(7), $\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.U_i$ and $\sum_{i=1}^{n} \alpha_i.\forall X.U \preceq T$. So, by the induction hypothesis $\Gamma \vdash \mathbf{r} : \sum_{i=1}^{n} \alpha_i.U_i$. By rule $\forall_I$, $\Gamma \vdash \Lambda X.\mathbf{r} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i$. We conclude by rule $\preceq$.

$\mathbf{t}@(\sum_{i=1}^{n} U_i) \to \mathbf{r}@(\sum_{i=1}^{n} U_i)$**:** Let $\Gamma \vdash \mathbf{t}@(\sum_{i=1}^{n} U_i) : T$. By Lemma 7.2.2(8), $\exists\langle V\rangle_m,\ X\ \langle\alpha\rangle_m$ such that $\Gamma \vdash \mathbf{t} : \sum_{j=1}^{m} \alpha_j.\forall X.V_j$ and $\sum_{j=1}^{m} \alpha_j.(\forall X.V_j)@(\sum_{i=1}^{n} U_i) \preceq T$. The by the induction hypothesis $\Gamma \vdash \mathbf{r} : \sum_{j=1}^{m} \alpha_j.\forall X.V_j$. By rule $@_I$, $\Gamma \vdash \mathbf{r}@(\sum_{i=1}^{n} U_i) : \sum_{j=1}^{m} \alpha_j.(\forall X.V_j)@(\sum_{i=1}^{n} U_i)$. We conclude by rule $\preceq$.

□

F.3 Proof of Lemma 7.3.1

Lemma 7.3.1. If $T \equiv R$, then $\|T\| \equiv \|R\|$.

Proof. Case by case

- $1.T \equiv T$. $\|1.T\| = 1.\|T\| \equiv \|T\|$.
- $\alpha.(\beta.T) \equiv (\alpha \times \beta).T$. $\|\alpha.(\beta.T)\| = \alpha.(\beta.\|T\|) \equiv (\alpha \times \beta).\|T\| = \|(\alpha \times \beta).T\|$.
- $\alpha.T + \alpha.R \equiv \alpha.(T + R)$. $\|\alpha.T + \alpha.R\| = \alpha.\|T\| + \alpha.\|R\| \equiv \alpha.(\|T\| + \|R\|) = \|\alpha.(T + R)\|$.
- $\alpha.T + \beta.T \equiv (\alpha + \beta).T$. $\|\alpha.T + \beta.T\| = \alpha.\|T\| + \beta.\|T\| \equiv (\alpha + \beta).\|T\| = \|(\alpha + \beta).T\|$.
- $T + R \equiv R + T$. $\|T + R\| = \|T\| + \|R\| \equiv \|R\| + \|T\| = \|R + T\|$.
- $T + (R + S) \equiv (T + R) + S$. $\|T + (R + S)\| = \|T\| + (\|R\| + \|S\|) \equiv (\|T\| + \|R\|) + \|S\| = \|(T + R) + S\|$.
- $(\forall X.U)@V \equiv U[V/X]$. $\|(\forall X.U)@V\| = \|U[V/X]\|$.

□

F.4 Proof of Lemma 7.3.2

Lemma 7.3.2.

1. $\|\mathbf{t}[\mathbf{b}/x]\| = \|\mathbf{t}\|[\|\mathbf{b}\|/x]$.
2. $\|T[W/X]\| \equiv \|T\|[\|W\|/X]$.

Proof.

1. Structural induction on $\mathbf{t}$.
   - $\mathbf{t} = x : U$, then $\|(x : U)[\mathbf{b}/x]\| = \|\mathbf{b}\| = x[\|\mathbf{b}\|/x] = \|x : U\|[\|\mathbf{b}\|/x]$.
   - $\mathbf{t} = y : U$, then $\|(y : U)[\mathbf{b}/x]\| = \|y : U\| = y = y[\|\mathbf{b}\|/x] = \|y : U\|[\|\mathbf{b}\|/x]$.
   - $\mathbf{t} = \mathbf{0}$, analogous to previous case.
   - $\mathbf{t} = \lambda y : U.\mathbf{r}$, then $\|(\lambda y : U.\mathbf{r})[\mathbf{b}/x]\| = \|\lambda y : U.\mathbf{r}[\mathbf{b}/x]\| = \lambda y.\|\mathbf{r}[\mathbf{b}/x]\|$, which by the induction hypothesis is equal to $\lambda y.\|\mathbf{r}\|[\|\mathbf{b}\|/x] = \|\lambda y : U.\mathbf{r}\|[\|\mathbf{b}\|/x]$.
   - $\mathbf{t} = \Lambda X.\mathbf{r}$, then $\|(\Lambda X.\mathbf{r})[\mathbf{b}/x]\| = \|\Lambda X.\mathbf{r}[\mathbf{b}/x]\| = \|\mathbf{r}[\mathbf{b}/x]\|$, which by the induction hypothesis is equal to $\|\mathbf{r}\|[\|\mathbf{b}\|/x] = \|\Lambda X.\mathbf{r}\|[\|\mathbf{b}\|/x]$.
   - $\mathbf{t} = (\mathbf{r})\ \mathbf{u}$, then $\|((\mathbf{r})\ \mathbf{u})[\mathbf{b}/x]\| = \|(\mathbf{r}[\mathbf{b}/x])\ \mathbf{u}[\mathbf{b}/x]\| = (\|\mathbf{r}[\mathbf{b}/x]\|)\ \|\mathbf{u}[\mathbf{b}/x]\|$, which by the induction hypothesis is equal to $(\|\mathbf{r}\|[\|\mathbf{b}\|/x])\ \|\mathbf{u}\|[\|\mathbf{b}\|/x] = ((\|\mathbf{r}\|)\ \|\mathbf{u}\|)[\|\mathbf{b}\|/x] = \|(\mathbf{r})\ \mathbf{u}\|[\|\mathbf{b}\|/x]$.
   - $\mathbf{t} = \mathbf{r}@(\sum_{i=1}^{n} U_i)$, then $\|(\mathbf{r}@(\sum_{i=1}^{n} U_i))[\mathbf{b}/x]\| = \|(\mathbf{r}[\mathbf{b}/x])@(\sum_{i=1}^{n} U_i)\| = \|(\mathbf{r}[\mathbf{b}/x])\|$, which by the induction hypothesis is equal to $\|\mathbf{r}\|[\|\mathbf{b}\|/x] = \|\mathbf{r}@(\sum_{i=1}^{n} U_i)\|[\|\mathbf{b}\|/x]$.
   - $\mathbf{t} = \alpha.\mathbf{r}$, then $\|(\alpha.\mathbf{r})[\mathbf{b}/x]\| = \|\alpha.\mathbf{r}[\mathbf{b}/x]\| = \alpha.\|\mathbf{r}[\mathbf{b}/x]\|$, which by the induction hypothesis is equal to $\alpha.\|\mathbf{r}\|[\|\mathbf{b}\|/x] = (\alpha.\|\mathbf{r}\|)[\|\mathbf{b}\|/x] = \|\alpha.\mathbf{r}\|[\|\mathbf{b}\|/x]$.
   - $\mathbf{t} = \mathbf{r}+\mathbf{u}$, then $\|(\mathbf{r} + \mathbf{u})[\mathbf{b}/x]\| = \|\mathbf{r}[\mathbf{b}/x] + \mathbf{u}[\mathbf{b}/x]\| = \|\mathbf{r}[\mathbf{b}/x]\|+\|\mathbf{u}[\mathbf{b}/x]\|$, which by the induction hypothesis is equal to $\|\mathbf{r}\|[\|\mathbf{b}\|/x] + \|\mathbf{u}\|[\|\mathbf{b}\|/x] = (\|\mathbf{r}\| + \|\mathbf{u}\|)[\|\mathbf{b}\|/x] = \|\mathbf{r} + \mathbf{u}\|[\|\mathbf{b}\|/x]$.
2. Structural induction on T
   - $T = X$, then $\|X[W/X]\| = \|W\| = X[\|W\|/X] = \|X\|[\|W\|/X]$.
   - $T = Y$, then $\|Y[W/X]\| = \|Y\| = Y = Y[\|W\|/X] = \|Y\|[\|Y\|/X]$.
   - $T = U \to R$, then $\|(U \to R)[W/X]\| = \|U[W/X] \to R[W/X]\| = \|U[W/X]\| \to \|R[W/X]\|$, which by the induction hypothesis is equivalent to $\|U\|[\|W\|/X] \to \|R\|[\|W\|/X] = (\|U\| \to \|R\|)[\|W\|/X] = \|U \to R\|[\|W\|/X]$.
   - $T = \forall Y.U$, then $\|(\forall Y.U)[W/X]\| = \|\forall Y.U[W/X]\| = \forall Y.\|U[W/X]\|$, which by the induction hypothesis is equivalent to $\forall Y.\|U\|[\|W\|/X] = (\forall Y.\|U\|)[\|W\|/X] = \|\forall Y.U\|[\|W\|/X]$.
   - $T = (\forall X.U)@V$, then by Lemma 7.3.1, $\|((\forall Y.U)@V)[W/X]\| \equiv \|U[V/Y][W/X]\|$, which by the induction hypothesis is equivalent to $\|U[V/Y]\|[\|W\|/X] \equiv \|\forall X.U@V\|[\|W\|/X]$.
   - $T = U@(\sum_{i=1}^{n} V_i)$, with $U \not\equiv \forall X.W$ or $n > 1$, then $\|U@(\sum_{i=1}^{n} V_i)[W/X]\|$ is equal (after doing the needed variable renaming) to $\|U[W/X]@(\sum_{i=1}^{n} V_i[W/X])\| = \|U[W/X]\|$, which by the induction hypothesis, is equivalent to $\|U\|[\|W\|/X] = \|U@(\sum_{i=1}^{n} V_i)\|[\|W\|/X]$.
   - $T = \alpha.R$, then $\|(\alpha.R)[U/X]\| = \|\alpha.R[U/X]\| = \alpha.\|R[U/X]\|$, which by the induction hypothesis, is equivalent to $\alpha.\|R\|[\|W\|/X] = (\alpha.\|R\|)[\|W\|/X] = \|\alpha.R\|[\|W\|/X]$.

- $T = R + S$, then $\|(R+S)[W/X]\| = \|R[W/X] + S[W/X]\| = \|R[W/X]\| + \|S[W/X]\|$, which by the induction hypothesis is equivalent to $\|R\|[\|W\|/X] + \|S\|[\|W\|/X] = (\|R\| + \|S\|)[\|W\|/X] = \|R+S\|[\|W\|/X]$.

□

F.5 Proof of Lemma 7.3.4

Lemma 7.3.4 (Reducibility preservation)**.** If $\mathbf{t} \to \mathbf{r}$, then $\|\mathbf{t}\| \to_v^= \|\mathbf{r}\|$. Moreover, if the reduction $\mathbf{t} \to \mathbf{r}$ is not the type application beta-reduction, the type-distributivity rule nor the type linearity rules, then $\|\mathbf{t}\| \to_v \|\mathbf{r}\|$.

Proof. Rule by rule analysis.

Elementary rules

- Rule $0.\mathbf{t} \to \mathbf{0}$. $\|0.\mathbf{t}\| = 0.\|\mathbf{t}\| \to_v \mathbf{0} = \|\mathbf{0}\|$.
- Rule $1.\mathbf{t} \to \mathbf{t}$. $\|1.\mathbf{t}\| = 1.\|\mathbf{t}\| \to_v \|\mathbf{t}\|$.
- Rule $\alpha.\mathbf{0} \to \mathbf{0}$. $\|\alpha.\mathbf{0}\| = \alpha.\|\mathbf{0}\| = \alpha.\mathbf{0} \to_v \mathbf{0} = \|\mathbf{0}\|$.
- Rule $\alpha.(\beta.\mathbf{t}) \to (\alpha \times \beta).\mathbf{t}$. $\|\alpha.(\beta.\mathbf{t})\| = \alpha.(\beta.\|\mathbf{t}\|) \to_v (\alpha \times \beta).\|\mathbf{t}\| = \|(\alpha \times \beta).\mathbf{t}\|$.
- Rule $\alpha.(\mathbf{t} + \mathbf{r}) \to \alpha.\mathbf{t} + \alpha.\mathbf{r}$. $\|\alpha.(\mathbf{t} + \mathbf{r})\| = \alpha.(\|\mathbf{t}\| + \|\mathbf{r}\|) \to_v \alpha.\|\mathbf{t}\| + \alpha.\|\mathbf{r}\| = \|\alpha.\mathbf{t} + \alpha.\mathbf{r}\|$.
- Rule $\mathbf{t} + \mathbf{0} \to \mathbf{t}$. $\|\mathbf{t} + \mathbf{0}\| = \|\mathbf{t}\| + \mathbf{0} \to_v \|\mathbf{t}\|$.

Factorisation rules

- Rule $\alpha.\mathbf{t} + \beta.\mathbf{t} \to (\alpha + \beta).\mathbf{t}$. $\|\alpha.\mathbf{t} + \beta.\mathbf{t}\| = \alpha.\|\mathbf{t}\| + \beta.\|\mathbf{t}\| \to_v (\alpha + \beta).\|\mathbf{t}\| = \|(\alpha + \beta).\mathbf{t}\|$.
- Rules $\alpha.\mathbf{t} + \mathbf{t} \to (\alpha + 1).\mathbf{t}$ and $\mathbf{t} + \mathbf{t} \to (1 + 1).\mathbf{t}$. Analogous to previous case.

Application rules

- Rule $(\mathbf{t} + \mathbf{r})\ \mathbf{u} \to (\mathbf{t})\ \mathbf{u} + (\mathbf{r})\ \mathbf{u}$. $\|(\mathbf{t} + \mathbf{r})\ \mathbf{u}\| = (\|\mathbf{t}\| + \|\mathbf{r}\|)\ \|\mathbf{u}\| \to_v (\|\mathbf{t}\|)\ \|\mathbf{u}\| + (\|\mathbf{r}\|)\ \|\mathbf{u}\| = \|(\mathbf{t})\ \mathbf{u} + (\mathbf{r})\ \mathbf{u}\|$.
- Rule $(\mathbf{u})\ (\mathbf{t} + \mathbf{r}) \to (\mathbf{u})\ \mathbf{t} + (\mathbf{u})\ \mathbf{r}$. Analogous to previous case
- Rule $(\alpha.\mathbf{t})\ \mathbf{r} \to \alpha.(\mathbf{t})\ \mathbf{r}$. $\|(\alpha.\mathbf{t})\ \mathbf{r}\| = (\alpha.\|\mathbf{t}\|)\ \|\mathbf{r}\| \to_v \alpha.(\|\mathbf{t}\|)\ \|\mathbf{r}\| = \|\alpha.(\mathbf{t})\ \mathbf{r}\|$.
- Rule $(\mathbf{r})\ (\alpha.\mathbf{t}) \to \alpha.(\mathbf{r})\ \mathbf{t}$. Analogous to previous case
- Rule $(\mathbf{0})\ \mathbf{t} \to \mathbf{0}$. $\|(\mathbf{0})\ \mathbf{t}\| = (\mathbf{0})\ \|\mathbf{t}\| \to_v \mathbf{0} = \|\mathbf{0}\|$.
- Rule $(\mathbf{t})\ \mathbf{0} \to \mathbf{0}$. Analogous to previous case

Beta reductions

- Rule $(\lambda x : U.\mathbf{t})\ \mathbf{b} \to \mathbf{t}[\mathbf{b}/x]$. $\|(\lambda x : U.\mathbf{t})\ \mathbf{b}\| = (\lambda x.\|\mathbf{t}\|)\ \|\mathbf{b}\|$. Since base vectors are translated into base vectors, $\|\mathbf{b}\|$ is a base vector, and so the previous term $\to_v^=$-reduces to $\|\mathbf{t}\|[\|\mathbf{b}\|/x]$. By Lemma 7.3.2, this is equal to $\|\mathbf{t}[\mathbf{b}/x]\|$.
- Rule $(\Lambda X.\mathbf{t})@U \to \mathbf{t}[U/X]$. $\|(\Lambda X.\mathbf{t})@U\| = \|\Lambda X.\mathbf{t}\| = \|\mathbf{t}\|$, which by Lemma 7.3.3 is equal to $\|\mathbf{t}[U/X]\|$.

Type-linearity rules

- Rule $(\alpha.\mathbf{t})@(\sum_{i=1}^{n} U_i) \to \alpha.\mathbf{t}@(\sum_{i=1}^{n} U_i)$. $\|(\alpha.\mathbf{t})@(\sum_{i=1}^{n} U_i)\| = \|\alpha.\mathbf{t}\| = \alpha.\|\mathbf{t}\| = \alpha.\|\mathbf{t}@(\sum_{i=1}^{n} U_i)\| = \|\alpha.\mathbf{t}@(\sum_{i=1}^{n} U_i)\|$.
- Rule $(\mathbf{t}+\mathbf{r})@(\sum_{i=1}^{n} U_i) \to \mathbf{t}@(\sum_{i=1}^{n} U_i) + \mathbf{r}@(\sum_{i=1}^{n} U_i)$. $\|(\mathbf{t}+\mathbf{r})@(\sum_{i=1}^{n} U_i)\| = \|\mathbf{t}+\mathbf{r}\| = \|\mathbf{t}\| + \|\mathbf{r}\| = \|\mathbf{t}@(\sum_{i=1}^{n} U_i)\| + \|\mathbf{r}@(\sum_{i=1}^{n} U_i)\| = \|\mathbf{t}@(\sum_{i=1}^{n} U_i) + \mathbf{r}@(\sum_{i=1}^{n} U_i)\|$.

Type-distributivity rule

- Rule $((\langle \Lambda X\rangle_k.\langle \lambda x : U\rangle_n.\mathbf{t})@\langle\sum_{i=1}^{m} W_i\rangle_k)\ \langle\mathbf{b}\rangle_n \to (\langle \lambda x : V\rangle_n.\mathbf{t}\langle[W_j/X]\rangle_k)\ \langle\mathbf{b}\rangle_n$. $\|((\langle \Lambda X\rangle_k.\langle \lambda x : U\rangle_n.\mathbf{t})@\langle\sum_{i=1}^{m} W_i\rangle_k)\ \langle\mathbf{b}\rangle_n\| = (\|(\langle \Lambda X\rangle_k.\langle \lambda x : U\rangle_n.\mathbf{t})@\langle\sum_{i=1}^{m} W_i\rangle_k\|)\ \langle\|\mathbf{b}\|\rangle_n = (\langle \lambda x\rangle_n.\|\mathbf{t}\|)\ \langle\|\mathbf{b}\|\rangle_n$ and using Lemma 7.3.3, this is equal to $(\|\langle \lambda x : V\rangle_n.\mathbf{t}\langle[W_j/X]\rangle_k\|)\ \langle\|\mathbf{b}\|\rangle_n = \|(\langle \lambda x : V\rangle_n.\mathbf{t}\langle[W_j/X]\rangle_k)\ \langle\mathbf{b}\rangle_n\|$.

Contextual rules Let $\mathbf{t} \to \mathbf{r}$ and assume $\|\mathbf{t}\| \to_v^= \|\mathbf{r}\|$. All the rules follow by translating and then using the equivalent rule in *Vectorial*, analogously as the previous cases.

□

F.6 Proof of Lemma 7.3.5

Lemma 7.3.5 (Typability preservation)**.** If $\Gamma \vdash \mathbf{t} : T$, then $\exists R \preceq T$ such that $\|\Gamma\| \vdash_v \|\mathbf{t}\| : \|R\|$.

Proof. Induction on the last rule applied to derive $\Gamma \vdash \mathbf{t} : T$.

1. $\dfrac{}{\Gamma, x : U \vdash x : U}\, ax$ Then, since unit types translate into unit types, by rule ax in *Vectorial*, we have $\|\Gamma\|, x : \|U\| \vdash_v x : \|U\|$. Note that $\|x : U\| = x$.

2. $\dfrac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \mathbf{0} : 0.T}\, 0_I$ By the induction hypothesis $\exists R \preceq T$ such that $\|\Gamma\| \vdash_v \|\mathbf{t}\| : \|R\|$, so by rule 0_I in *Vectorial*, we have $\|\Gamma\| \vdash_v \mathbf{0} : 0.\|R\| = \|0.R\|$. Notice that $0.R \preceq 0.T$. Also, $\|\mathbf{0}\| = \mathbf{0}$.

3. $\dfrac{\Gamma, x : U \vdash \mathbf{t} : T}{\Gamma \vdash \lambda x : U.\mathbf{t} : U \to T}\, \to_I$ By the induction hypothesis $\exists R \preceq T$ such that $\|\Gamma\|, x : \|U\| \vdash_v \|\mathbf{t}\| : \|R\|$, then by rule $\to_I$ in *Vectorial*, $\|\Gamma\| \vdash_v \lambda x.\|\mathbf{t}\| : \|U\| \to \|R\|$. Notice that $\|U\| \to \|R\| = \|U \to R\|$ and $R \preceq T \Rightarrow U \to R \preceq U \to T$. Also, $\|\lambda x : U.\mathbf{t}\| = \lambda x.\|\mathbf{t}\|$.

4. $$\dfrac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.(\langle \forall X\rangle_k.(U \to T_i))@\langle\sum_{j=1}^{m+\delta} W_j\rangle_k \qquad \Gamma \vdash \mathbf{r} : \sum_{j=1}^{m} \beta_j.V_j \qquad \begin{array}{c}\forall V_j, \exists j_1, \ldots, j_k\ /\\ U\langle[W_j/X]\rangle_k = V_j\end{array}}{\Gamma \vdash (\mathbf{t})\ \mathbf{r} : \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \times \beta_j.T_i\langle[W_j/X]\rangle_k}\ \to_E$$

   Then by the induction hypothesis there exists $R \preceq \sum_{i=1}^{n} \alpha_i.(\langle \forall X\rangle_k.(U \to T_i))@\langle\sum_{j=1}^{m+\delta} W_j\rangle_k$ and $S \preceq \sum_{j=1}^{m} \beta_j.V_j$ such that $\|\Gamma\| \vdash_v \|\mathbf{t}\| : \|R\|$ and $\|\Gamma\| \vdash_v \|\mathbf{r}\| : \|S\|$. Using Lemmas 7.1.2 and 7.1.3, we can consider, without lost of generality, $R \equiv \sum_{i=1}^{n} \alpha_i.(\langle \forall X\rangle_k.(U \to T_i))@\langle\sum_{j=1}^{m+\delta} W_j\rangle_k$ and $S \equiv \sum_{j=1}^{m} \beta_j.V_j$. Then, using Lemma 7.3.1, $\|R\| \equiv \sum_{i=1}^{n} \alpha_i.\|(\langle \forall X\rangle_k.(U \to T_i))@\langle\sum_{j=1}^{m+\delta} W_j\rangle_k\|$. Cases:

- If $\forall i,\ m+\delta_i = 1$, then $\delta_i = 0$ and $m = 1$ and so $\|R\| \equiv \sum_{i=1}^{n} \alpha_i.\|V\| \to \|T_i\|\langle[\|W_1\|/X]\rangle_k$ and $\|S\| \equiv \|\beta_1.V_1\|$, so using rule $\to_E$ in *Vectorial*, we obtain $\|\Gamma\| \vdash_v (\|\mathbf{t}\|)\ \|\mathbf{r}\| : \sum_{i=1}^{n} \alpha_i \times \beta_1.\|T_i\|\langle[\|W_1\|/X]\rangle_k$.
- In other case, $\|R\| \equiv \sum_{i=1}^{n} \alpha_i.(\langle\forall X\rangle_k.(\|U\| \to \|T_i\|))$ and $\|S\| \equiv \sum_{j=1}^{m} \beta_j.\|V_j\|$. Since $\forall V_j, \exists j_1, \ldots, j_k \ /\ U\langle[W_j/X]\rangle_k = V_j$, $\|V_j\| = \|U\langle[W_j/X]\rangle_k\|$, which by Lemma 7.3.2, is equal to $\|U\|\langle[\|W_j\|/X]\rangle_k$. So, using rule $\to_E$ in *Vectorial*, we obtain $\|\Gamma\| \vdash_v (\|\mathbf{t}\|)\ \|\mathbf{r}\| : \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \times \beta_j.\|T_i\|\langle[W_j/X]\rangle_k$.

$(\|\mathbf{t}\|)\ \|\mathbf{r}\| = \|(\mathbf{t})\ \mathbf{r}\|$ and $\sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \times \beta_j.\|T_i\|\langle[W_j/X]\rangle_k = \|\sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \times \beta_j.T_i\langle[W_j/X]\rangle_k\|$.

5. $$\frac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.U_i \quad X \notin FV(\Gamma)}{\Gamma \vdash \Lambda X.\mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i}\ \forall_I$$

By the induction hypothesis $\exists R \preceq \sum_{i=1}^{n} \alpha_i.U_i$ such that $\|\Gamma\| \vdash_v \|\mathbf{t}\| : \|R\|$. Notice that since the translation does not introduces any type variable, $X \notin FV(\|\Gamma\|)$. By Lemma 7.1.2, $R \equiv \sum_{j=1}^{m} \beta_j.V_j$, so by Lemma 7.3.1, $\|R\| \equiv \|\sum_{j=1}^{m} \beta_j.V_j\| = \sum_{j=1}^{m} \beta_j.\|V_j\|$. Then, using rule $\forall_I$ in *Vectorial*, we obtain $\|\Gamma\| \vdash_v \|\mathbf{t}\| : \sum_{j=1}^{m} \beta_j.\forall X.\|V_j\|$. Note that $\|\Lambda X.\mathbf{t}\| = \|\mathbf{t}\|$, $\|\sum_{j=1}^{m} \beta_j.\forall X.V_j\| = \sum_{j=1}^{m} \beta_j.\forall X.\|V_j\|$, and $\sum_{j=1}^{m} \beta_j.\forall X.V_j \preceq \sum_{i=1}^{n} \alpha_i.\forall X.U_i$.

6. $$\frac{\Gamma \vdash \mathbf{t} : \sum_{i=1}^{n} \alpha_i.\forall X.U_i}{\Gamma \vdash \mathbf{t}@(\sum_{j=1}^{m} V_j) : \sum_{i=1}^{n} \alpha_i.(\forall X.U_i)@(\sum_{j=1}^{m} V_j)}\ @_I$$

By the induction hypothesis, $\exists R \preceq \sum_{i=1}^{n} \alpha_i.\forall X.U_i$ such that $\|\Gamma\| \vdash_v \|\mathbf{t}\| : \|R\|$. Notice that $\|\mathbf{t}@(\sum_{j=1}^{m} V_j)\| = \|\mathbf{t}\|$. If $m > 1$, we are done, since $\|\sum_{i=1}^{n} \alpha_i.(\forall X.U_i)@(\sum_{j=1}^{m} V_j)\|$ is equal to $\|\sum_{i=1}^{n} \alpha_i.\forall X.U_i\|$. On the other hand, if $m = 1$, then $\|\sum_{i=1}^{n} \alpha_i.(\forall X.U_i)@(V_1)\|$ is equal to $\sum_{i=1}^{n} \alpha_i.\|U_i\|[\|V_1\|/X]$. By Lemmas 7.1.2 and 7.1.3, $R \equiv \sum_{k=1}^{h} \beta_k.\forall X.W_k$. Then by Lemma 7.3.1, $\|R\| \equiv \sum_{k=1}^{h} \beta_k.\forall X.\|W_k\|$. So, using rule $\forall_E$ in *Vectorial*, we can derive the following sequent $\|\Gamma\| \vdash_v \|\mathbf{t}\| : \sum_{k=1}^{h} \beta_k.\|W_k\|[\|V_1\|/X] = \|\sum_{k=1}^{h} \beta_k.(\forall X.W_k)@V_1\|$. Since $R \preceq \sum_{i=1}^{n} \alpha_i.\forall X.U_i$, then $\sum_{k=1}^{h} \beta_k.\forall X.W_k \preceq \sum_{i=1}^{n} \alpha_i.\forall X.U_i$, and so we have $\sum_{k=1}^{h} \beta_k.(\forall X.W_k)@V_1 \preceq \sum_{i=1}^{n} \alpha_i.(\forall X.U_i)@V_1$.

7. $$\frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \alpha.T}\ s_I$$

By the induction hypothesis, $\exists R \preceq T$ such that $\|\Gamma\| \vdash_v \|\mathbf{t}\| : \|R\|$. Then by rule s_I in *Vectorial*, $\|\Gamma\| \vdash_v \alpha.\|\mathbf{t}\| : \alpha.\|R\|$. Notice that $\|\alpha.\mathbf{t}\| = \alpha.\|\mathbf{t}\|$, $\alpha.R \preceq \alpha.T$ and $\|\alpha.R\| = \alpha.\|R\|$.

8. $$\frac{\Gamma \vdash \mathbf{t} : T \qquad \Gamma \vdash \mathbf{r} : R}{\Gamma \vdash \mathbf{t} + \mathbf{r} : T + R}\ +_I$$

By the induction hypothesis, $\exists T' \preceq T$ and $R' \preceq R$ such that $\|\Gamma\| \vdash_v \|\mathbf{t}\| : \|T'\|$ and $\|\Gamma\| \vdash_v \|\mathbf{r}\| : \|R'\|$. Then by rule $+_I$ in *Vectorial*, $\|\Gamma\| \vdash_v \|\mathbf{t}\| + \|\mathbf{r}\| : \|T'\| + \|R'\|$. Notice that $\|\mathbf{t} + \mathbf{r}\| = \|\mathbf{t}\| + \|\mathbf{r}\|$, $\|T' + R'\| = \|T'\| + \|R'\|$ and $T' + R' \preceq T + R$.

9. $$\frac{\Gamma \vdash \mathbf{t} : T \qquad T \preceq R}{\Gamma \vdash \mathbf{t} : R}\ \preceq$$

Then by the induction hypothesis $\exists S \preceq T$ such that $\|\Gamma\| \vdash_v \|\mathbf{t}\| : \|S\|$. Notice that $S \preceq T \preceq R$.

□

F.7 Proof of Theorem 7.3.6

Theorem 7.3.6 (Strong normalisation)**.** If $\Gamma \vdash \mathbf{t} : T$ is derivable in *Lineal*, then $\mathbf{t}$ is strongly normalising.

Proof. Let $\Gamma \vdash \mathbf{t} : T$, then by Lemma 7.3.5, $\exists R \preceq T$ such that $\|\Gamma\| \vdash_v \|\mathbf{t}\| : \|R\|$, and so by Theorem 5.4.4, $\|\mathbf{t}\|$ is strongly normalising. Assume $\mathbf{t}$ is not strongly normalising, say $\mathbf{t} \to \mathbf{t}_1 \to \mathbf{t}_2 \cdots$. Then by Lemma 7.3.4, $\|\mathbf{t}\| \to_v^= \|\mathbf{t}_1\| \to_v^= \|\mathbf{t}_2\| \to_v^= \cdots$. Since $\|\mathbf{t}\|$ is strongly normalising, there exists n such that $\forall i \geq n$, $\|\mathbf{t}_i\| = \|\mathbf{t}_{i+1}\|$. By Lemma 7.3.4 it means that $\forall i \geq n$, the reduction $\mathbf{t}_i \to \mathbf{t}_{i+1}$ can be only one of the type application beta-reduction, the type-distributivity rule or the type linearity rules. We define a positive measure on terms and show that these rules are strictly decreasing with respect to the measure, so $\mathbf{t}$ has to be strongly normalising.

Consider the following *measure*:

$$\begin{array}{rclrcl} |x:U| = |\mathbf{0}| = |\lambda x:U.\mathbf{t}| & = & 1 & |\mathbf{t}+\mathbf{r}| & = & 2+|\mathbf{t}|+|\mathbf{r}| \\ |(\mathbf{t})\ \mathbf{r}| & = & |\mathbf{t}|+|\mathbf{r}| & |\Lambda X.\mathbf{t}| & = & |\mathbf{t}| \\ |\alpha.\mathbf{t}| & = & 1+|\mathbf{t}| & |\mathbf{t}@(\sum_{i=1}^n U_i)| & = & 1+2|\mathbf{t}| \end{array}$$

Notice that $|\mathbf{t}[U/X]| = |\mathbf{t}|$, since the measure does not depend on the free variables. We proceed by checking case by case to show that the mentioned rules are strictly decreasing on this measure.

1. $|(\Lambda X.\mathbf{t})@U| = 1+2|\Lambda X.\mathbf{t}| = 1+2|\mathbf{t}| > |\mathbf{t}| = |\mathbf{t}[U/X]|$.
2. $|(\alpha.\mathbf{t})@(\sum_{i=1}^n U_i)| = 1+2|\alpha.\mathbf{t}| = 1+2(1+|\mathbf{t}|) = 3+2|\mathbf{t}| > 2+2|\mathbf{t}| = 1+1+2|\mathbf{t}| = 1+|\mathbf{t}@(\sum_{i=1}^n U_i)| = |\alpha.\mathbf{t}@(\sum_{i=1}^n U_i)|$
3. $|(\mathbf{t}+\mathbf{r})@(\sum_{i=1}^n U_i)| = 1+2|\mathbf{t}+\mathbf{r}| = 5+2|\mathbf{t}|+2|\mathbf{r}| > 4+2|\mathbf{t}|+2|\mathbf{r}| = 2+1+2|\mathbf{t}|+1+2|\mathbf{r}| = 2+|\mathbf{t}@(\sum_{i=1}^n U_i)|+|\mathbf{r}@(\sum_{i=1}^n U_i)| = |\mathbf{t}@(\sum_{i=1}^n U_i)+\mathbf{r}@(\sum_{i=1}^n U_i)|$.
4. $|((\langle \Lambda X\rangle_k.\langle \lambda x:U\rangle_n.\mathbf{t})@\langle \sum_{i=1}^m W_i\rangle_k)\ \langle \mathbf{b}\rangle_n| = 1+2|\langle \lambda x:U\rangle_n.\mathbf{t}|+n = 1+2+n = 3+n > 1+n = |\langle \lambda x:V\rangle_n.\mathbf{t}\langle [W_j/X]\rangle_k|+n = |(\langle \lambda x:V\rangle_n.\mathbf{t}\langle [W_j/X]\rangle_k)\ \langle \mathbf{b}\rangle_n|$.

□

Appendix G

Proofs from Chapter 8

G.1 Proof of Theorem 8.2.3

Theorem 8.2.3 (No-cloning of scalars)**.** $\nexists \Pi_n$ such that $\forall \alpha, C(\Pi_n(\Gamma \vdash \alpha.U)) = \Delta \vdash (\delta \times \alpha^s + \gamma).V$ with $\delta \neq 0$ and γ constants in $\mathcal{S}$, $s \in \mathbb{N}^{>1}$ and U, V constants in $\mathcal{U}$.

Notice that α is a member of a ring and s is a natural number, so α^s is just the multiplication of α by itself s times.

Proof. Induction over n.

Basic case. $n = 0$. Trivial, as $\Pi_0(\Gamma \vdash \alpha.U) = \Gamma \vdash \alpha.U$ for all Π.

Inductive cases.

- $\Pi_n(\Gamma \vdash \alpha.U) = \dfrac{\Pi_{n-1}(\Gamma \vdash \alpha.U)}{P}\, R$

  Assume $P = \Delta \vdash (\delta \times \alpha^s + \gamma).V$ and let us do an analysis case by case on the possible rules R:

  1. $R =\!\!\rightarrow I[W]$. Because the denominator must be unit, $\forall \alpha,\ \delta \times \alpha^s + \gamma = 1$, which is a contradiction.
  2. $R = \forall E[X := W]$. Then $(\delta \times \alpha^s + \gamma).V = T[X/W]$, and $C(\Pi_{n-1}(\Gamma \vdash \alpha.U)) = \forall X.T$. By Lemma 3.2.2, $\exists Z \in \mathcal{U}, \beta \in \mathcal{S}$ such that $T \equiv \beta.Z$, so by Lemma 3.2.3, $\delta \times \alpha^s + \gamma = \beta$, then $C(\Pi_{n-1}(\Gamma \vdash \alpha.U)) = \forall X.\beta.Z = \forall X.(\delta \times \alpha^s + \gamma).Z \equiv (\delta \times \alpha^s + \gamma).\forall X.Z$, which is a contradiction by the induction hypothesis.
  3. $R = \forall I[X]$. Then $(\delta \times \alpha^s + \gamma).V \equiv \forall X.T$. Analogous to 2.
  4. $R = sI[\beta]$. Then $\delta \times \alpha^s + \gamma.V \equiv \beta.T$. By Lemma 3.2.2, $T \equiv \sigma.W$, then by Lemma 3.2.3, $\delta \times \alpha^s + \gamma = \beta \times \sigma$. Notice that β cannot depend on α as the rule is constant, so it must be σ depending on α^s, which is a contradiction by the induction hypothesis.

- $\Pi_n(\Gamma \vdash \alpha.U) = \dfrac{\Pi_k(\Gamma \vdash \alpha.U) \quad \pi_h}{P}\, R$

  Assume $P = \Delta \vdash (\delta \times \alpha^s + \gamma).V$ and let us do an analysis case by case on the possible rules R:

  1. $R =\!\!\rightarrow E$. Then $C(\pi_h) = \Delta \vdash \beta.W$ and $C(\Pi_k(\Gamma \vdash \alpha.U)) = \Delta \vdash \phi.W \rightarrow \sigma.V$ where $\forall \alpha,\ \beta \times \phi \times \sigma = \delta \times \alpha^s + \gamma$. β cannot depend on α, as π_h is constant, so:
     - Assume ϕ depend on α, and σ do not, then it depend linearly on α by the induction hypothesis.

– Assume σ depend on α, then there are two possibilities:

(a) U is an arrow with the last term of the arrow being $\sigma.V$, which is a contradiction as σ depend on α and U is fixed.

(b) The arrow is set up through the derivation, so at some point we must had to use $\to I$ rule in the following way

$$\frac{\Theta, Z \vdash \sigma.V}{\Theta \vdash Z \to \sigma V} \to I$$

so by the induction hypothesis σ depends linearly on α. Once we reach this point, the only possibility to add something depending on α and multiplying the whole type is with $sI[\alpha]$ as it cannot come from any other branch (all other branches are constants). However, it is not possible either, as all the rules must to be constants.

2. $R = +I$. Then $C(\Pi_k(\Gamma \vdash \alpha.U)) = \Delta \vdash \sigma.V$ and $C(\pi_h) = \Delta \vdash \phi.V$ where $\sigma + \phi = \delta \times \alpha^2 + \gamma$. So, as ϕ is constant, $\sigma = \delta \times \alpha^2 + \gamma - \phi$, which is a contradiction by the induction hypothesis.

- $\Pi_n(\Gamma \vdash \alpha.U) = \dfrac{\pi_k \quad \Pi_h(\Gamma \vdash \alpha.U)}{P} R$

  Assume $P = \Delta \vdash (\delta \times \alpha^s + \gamma).V$ and let us do an analysis case by case on the possible rules R:

  1. $R = \to E$. Then $C(\pi_k) = \Delta \vdash \phi.W \to \sigma.V$ and $C(\Pi_h(\Gamma \vdash \alpha.U)) = \beta.W$ where $\beta \times \phi \times \sigma = \delta \times \alpha^s + \gamma$.

     Notice that nor ϕ nor σ can depend on α, so the only possibility is to β to depend on α^s, which is a contradiction by the induction hypothesis.

  2. $R = +I$. Analogous 2 of the previous case.

□

Bibliography

T. Altenkirch and J. J. Grattage. A functional quantum programming language. In *Proceedings of LICS-2005*, pages 249–258. IEEE Computer Society, 2005.

P. Arrighi and A. Díaz-Caro. A system F accounting for scalars. Preprint at arXiv:0903.3741, Apr. 2011a.

P. Arrighi and A. Díaz-Caro. Scalar system F for linear-algebraic λ-calculus: Towards a quantum physical logic. In B. Coecke, P. Panangaden, and P. Selinger, editors, *Proceedings of QPL-2009*, volume 270/2 of *Electronic Notes in Theoretical Computer Science*, pages 219–229. Elsevier, 2011b.

P. Arrighi and G. Dowek. A computational definition of the notion of vectorial space. In N. Martí-Oliet, editor, *Proceedings of WRLA-2004*, volume 117 of *Electronic Notes in Theoretical Computer Science*, pages 249–261. Elsevier, 2004.

P. Arrighi and G. Dowek. Linear-algebraic lambda-calculus: higher-order, encodings, and confluence. In A. Voronkov, editor, *Proceedings of RTA-2008*, volume 5117 of *Lecture Notes in Computer Science*, pages 17–31. Springer, 2008.

P. Arrighi, A. Díaz-Caro, M. Gadella, and J. J. Grattage. Measurements and confluence in quantum lambda calculi with explicit qubits. In B. Coecke, I. Mackie, P. Panangaden, and P. Selinger, editors, *Proceedings of QPL/DCM-2008*, volume 270/1 of *Electronic Notes in Theoretical Computer Science*, pages 59–74. Elsevier, 2011a.

P. Arrighi, A. Díaz-Caro, and B. Valiron. A type system for the vectorial aspects of the linear-algebraic lambda-calculus. In *Proceedings of the 7th International Workshop on Developments of Computational Methods (DCM 2011)*, Zurich, Switzerland, 2011b. To appear in EPTCS. Draft at `http://membres-liglab.imag.fr/diazcaro/vectorial.pdf`.

A. Assaf and S. Perdrix. Completeness of algebraic CPS simulations. In *Proceedings of the 7th International Workshop on Developments of Computational Methods (DCM 2011)*, Zurich, Switzerland, 2011. To appear in EPTCS. Draft at `http://membres-lig.imag.fr/perdrix/publi/cps-completeness.pdf`.

H. P. Barendregt. *Lambda calculi with types*, volume II of *Handbook of logic in computer science*. Oxford University Press, 1992.

G. Boudol. Lambda-calculi for (strict) parallel functions. *Information and Computation*, 108(1):51–127, 1994.

O. Bournez and M. Hoyrup. Rewriting logic and probabilities. In R. Nieuwenhuis, editor, *Proceedings of RTA-2003*, volume 2706 of *Lecture Notes in Computer Science*, pages 61–75. Springer, 2003.

P. Buiras, A. Díaz-Caro, and M. Jaskelioff. Confluence via strong normalisation in an algebraic λ-calculus with rewriting. In *Proceedings of the 6th Workshop on Logical and Semantic Frameworks, with Applications (LSFA 2011)*, Belo Horizonte, Brazil, 2011. To appear in EPTCS. Draft at `http://membres-liglab.imag.fr/diazcaro/CA.pdf`.

A. Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58 (2):345–363, 1936.

A. Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68, 1940.

Coq Dev. Team. *The Coq proof assistant reference manual*. INRIA, 8.2 edition, Feb. 2009.

U. de'Liguoro and A. Piperno. Non deterministic extensions of untyped λ-calculus. *Information and Computation*, 122(2):149–177, 1995.

R. Di Cosmo. *Isomorphisms of Types: From Lambda-Calculus to Information Retrieval and Language Design*. Birkhauser Boston, 1995.

A. Di Pierro, C. Hankin, and H. Wiklicky. Probabilistic λ-calculus and quantitative program analysis. *Journal of Logic and Computation*, 15(2):159–179, 2005.

A. Díaz-Caro. Agregando medición al cálculo de van tonder. Master's thesis, Universidad Nacional de Rosario, Argentina, Dec. 21, 2007.

A. Díaz-Caro and B. Petit. Sums in linear algebraic lambda-calculus. Preprint at `arXiv:1011.3542`, Nov. 2010.

A. Díaz-Caro, S. Perdrix, C. Tasson, and B. Valiron. Equivalence of algebraic λ-calculi. In *Informal proceedings of HOR-2010*, pages 6–11, Edinburgh, UK, July 14, 2010.

A. Díaz-Caro, S. Perdrix, C. Tasson, and B. Valiron. Call by value, call by name and the vectorial behaviour of algebraic λ-calculus. Preprint at `arXiv:1005.2897`, June 2011.

D. J. Dougherty. Adding algebraic rewriting to the untyped lambda calculus. *Information and Computation*, 101(2):251–267, 1992.

T. Ehrhard. On Köthe sequence spaces and linear logic. *Mathematical Structures in Computer Science*, 12(5):579–623, 2003.

T. Ehrhard. Finiteness spaces. *Mathematical Structures in Computer Science*, 15(4):615–646, 2005.

T. Ehrhard. A finiteness structure on resource terms. In *Proceedings of LICS-2010*, pages 402–410. IEEE Computer Society, 2010.

T. Ehrhard and L. Regnier. The differential lambda-calculus. *Theoretical Computer Science*, 309(1):1–41, 2003.

M. J. Fischer. Lambda calculus schemata. In *Proceedings of ACM conference on Proving assertions about programs*, pages 104–109. ACM, 1972.

J.-Y. Girard. *Interprétation fonctionnelle et élimination des coupures dans l'arith-métique d'ordre supérieure*. PhD thesis, Université Paris Diderot, Paris, France, 1972.

J.-Y. Girard. Linear logic. *Theoretical Compututer Science*, 50:1–102, 1987.

J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*, volume 7 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1989.

O. M. Herescu and C. Palamidessi. Probabilistic asynchronous π-calculus. In J. Tiuryn, editor, *Proceedings of FOSSACS-2000*, volume 1784 of *Lecture Notes in Computer Science*, pages 146–160. Springer, 2000.

G. Jaeger. *Quantum information: An overview*. Springer, 2007.

J.-P. Jouannaud and H. Kirchner. Completion of a set of rules modulo a set of equations. *SIAM Journal on Compututing*, 15(4):1155–1194, 1986.

J.-L. Krivine. *Lambda-calcul: types et modèles*. Études et recherches en informatique. Masson, 1990.

M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press., 2000.

M. Pagani and S. R. D. Rocca. Solvability in resource lambda calculus. In L. Ong, editor, *Proceedings of FOSSACS-2010*, volume 6014 of *Lecture Notes in Computer Science*, pages 358–373. Springer, 2010.

M. Pagani and P. Tranquilli. Parallel reduction in resource lambda-calculus. In Z. Hu, editor, *Proceedings of APLAS-2009*, volume 5904 of *Lecture Notes in Computer Science*, pages 226–242. Springer, 2009.

B. Petit. A polymorphic type system for the lambda-calculus with constructors. In P.-L. Curien, editor, *Proceedings of TLCA-2009*, volume 5608 of *Lecture Notes in Computer Science*, pages 234–248. Springer, 2009.

G. D. Plotkin. Call-by-name, call-by-value and the λ-calculus. *Theoretical Computer Science*, 1(2): 125–159, 1975.

J. C. Reynolds. Towards a theory of type structure. In B. Robinet, editor, *Programming Symposium: Proceedings of the Colloque sur la Programmation*, volume 19 of *Lecture Notes in Computer Science*, pages 408–425. Springer, 1974.

A. Sabry and P. Wadler. A reflection on call-by-value. *ACM Transactions on Programming Languages and Systems*, 19(6):916–941, 1997.

M. H. Sørensen and P. Urzyczyn. *Lectures on the Curry-Howard Isomorphism*, volume 149 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 2006.

W. W. Tait. Intensional interpretations of functionals of finite type I. *The Journal of Symbolic Logic*, 32 (2):198–212, 1967.

C. Tasson. Algebraic totality, towards completeness. In P.-L. Curien, editor, *Proceedings of TLCA-2009*, volume 5608 of *Lecture Notes in Computer Science*, pages 325–340. Springer, 2009.

TeReSe. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

B. Valiron. Semantics of a typed algebraic lambda-calculus. In S. B. Cooper, P. Panangaden, and E. Kashefi, editors, *Proceedings DCM-2010*, volume 26 of *Electronic Proceedings in Theoretical Computer Science*, pages 147–158. Open Publishing Association, 2010.

B. Valiron. Coq proof. `http://www.monoidal.net/vectorial-alglin-coqproof-v2.tgz`, 2011a.

B. Valiron. Coq proof. `http://www.monoidal.net/vectorial-lvec-coqproof.tar.bz2`, 2011b.

A. van Tonder. A lambda calculus for quantum computation. *SIAM Journal of Computing*, 33:1109–1135, 2004.

L. Vaux. On linear combinations of lambda-terms. In F. Baader, editor, *Proceedings of RTA-2007*, volume 4533 of *Lecture Notes in Computer Science*, pages 374–388. Springer, 2007.

L. Vaux. The algebraic lambda calculus. *Mathematical Structures in Computer Science*, 19(5):1029–1059, 2009.

W. K. Wootters and W. H. Zurek. A single quantum cannot be cloned. *Nature*, 299:802–803, 1982.

Résumé

L'objectif de cette thèse est de développer une théorie de types pour le λ-calcul linéaire-algébrique, une extension du λ-calcul motivé par l'informatique quantique. Cette extension algébrique comprend tous les termes du λ-calcul plus leurs combinaisons linéaires, donc si t et r sont des termes, $\alpha.\mathrm{t} + \beta.\mathrm{r}$ est aussi un terme, avec α et β des scalaires pris dans un anneau. L'idée principale et le défi de cette thèse était d'introduire un système de types où les types, de la même façon que les termes, constituent un espace vectoriel, permettant la mise en évidence de la structure de la forme normale d'un terme. Cette thèse présente le système *Lineal*, ainsi que trois systèmes intermédiaires, également intéressants en eux-même : *Scalar*, *Additive* et λ^{CA}, chacun avec leurs preuves de préservation de type et de normalisation forte.

Abstract

The objective of this thesis is to develop a type theory for the linear-algebraic λ-calculus, an extension of λ-calculus motivated by quantum computing. This algebraic extension encompasses all the terms of λ-calculus together with their linear combinations, so if t and r are two terms, so is $\alpha.\mathrm{t} + \beta.\mathrm{r}$, with α and β being scalars from a given ring. The key idea and challenge of this thesis was to introduce a type system where the types, in the same way as the terms, form a vectorial space, providing the information about the structure of the normal form of the terms. This thesis presents the system *Lineal*, and also three intermediate systems, however interesting by themselves: *Scalar*, *Additive* and λ^{CA}, all of them with their subject reduction and strong normalisation proofs.