A Type System for the Vectorial Aspects of the Linear-Algebraic Lambda-Calculus.

Pablo Arrighi¹,², Alejandro Díaz-Caro³, and Benoît Valiron³

¹ Université de Grenoble, LIG, 220 rue de la chimie, 38400 Saint-Martin-d’Hères, France. Email: {pablo.arrighi|alejandro.diaz-caro}@imag.fr
² LIP (CNRS/ENSL/INRIA/UCBL), École Normale Supérieure de Lyon, 69364 Lyon Cedex 07, France.
³ LIPN UMR7030 CNRS, Université Paris 13, 99 av. J-B Clément, 93430 Villetaneuse, France. Email: benoit.valiron@lipn.univ-paris13.fr

Abstract. In this paper we describe a type system for the linear-algebraic lambda-calculus. The type system accounts for the part of the language emulating linear operators and vectors. The type system is able to statically describe the linear combinations of terms resulting from the reduction of programs. This gives rise to an original type theory where types, in the same way as terms, can be superposed into linear combinations. We show that the resulting typed lambda-calculus is strongly normalizing and features a weak subject-reduction.

1 Introduction

A number of recent works seek to endow the \( \lambda \)-calculus with a structure of a vector space; this agenda has emerged simultaneously in two different contexts. A first line of work forked from the study of relational models of linear logic. In [9, 13, 18], various algebraic lambda-calculi, that is, languages with vectorial structures, are considered. These languages are based on an interpretation of intuitionistic logic by linear logic. A second line of work [2, 3, 8] considers linear combinations of terms as a sort of “quantum superposition”. This paper stems from this second approach.

In quantum computation, data is encoded on normalized vectors in Hilbert spaces. For our purpose, this means that the vector spaces are 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 normalized vector \( \alpha|0\rangle + \beta|1\rangle \), where \( |\alpha|^2 + |\beta|^2 = 1 \). The operations on qubits that we consider are the quantum gates, that is, unitary operations. A unitary operator is a function preserving the norm and the orthogonality of vectors.

The language we consider in this paper will be called the vectorial lambda-calculus, denoted with \( \lambda^{vec} \). It is inspired from Lineal [3]. This language admits the regular constructs of lambda-calculus: variables \( x, y, \ldots \), lambda-abstractions \( \lambda x.s \) and application \( (s)t \). It also admits linear combinations of terms: \( 0, s+t \) and
$\alpha \cdot s$ are terms. The scalar $\alpha$ ranges over the ring of complex numbers. Inspired from [3], it behaves in a call-by-value manner, in the sense that arguments are first reduced before getting substituted. Thus $(\lambda x. r)(s + t)$ reduces to $(\lambda x. r)s + (\lambda x. r)t$ and the lambda-abstraction is not linear with respect to the vectorial structure: $\lambda x.(s + t)$ is not the same thing as $\lambda x.s + \lambda x.t$. The set of terms can then be interpreted as a vector space and the term $(\lambda x. r)s$ can be seen as the application of the linear operator $(\lambda x. r)$ to the vector $s$.

**Booleans in the vectorial lambda-calculus.** Both in $\lambda^{vec}$ 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 $\lambda^{vec}$, a representation of $if \; r \; then \; s \; else \; t$ needs to take into account the special relation between sums and applications. We cannot directly encode this test as the usual $((r)\; s)\; t$. Indeed, if $r$, $s$, and $t$ were respectively the terms $true$, $s_1 + s_2$ and $t_1 + t_2$, the term $((r)\; s)\; t$ would reduce to $((true)\; s_1)\; t_1 + ((true)\; s_1)\; t_2 + ((true)\; s_2)\; t_1 + ((true)\; s_2)\; t_2$, then to $2 \cdot s_1 + 2 \cdot s_2$ instead of $s_1 + s_2$. 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 thanks: we encode the test as $\{((r)\; s)\; t\}$, where $[\cdot]$ is the term $\lambda f.\; -$ with $f$ a fresh, unused term variable and where $\{\cdot\}$ is the term $(\lambda x.\; \lambda)$. The former “freezes” the computation while the latter “releases” it. Now, the term $if \; true \; then \; (s_1 + s_2) \; else \; (t_1 + t_2)$ reduces to the terms $s_1 + s_2$ as one could expect. Note that this test is linear in the sense that $if \; (\alpha \cdot true + \beta \cdot false) \; then \; s \; else \; t$ reduces to the term $\alpha \cdot s + \beta \cdot t$.

This has a striking similarity with the quantum test that can be found e.g. [15, 1, 3]. For example, the Hadamard gate $H$ is sending $|0\rangle$ to $\frac{\sqrt{2}}{2}(|0\rangle + |1\rangle)$ and $|1\rangle$ to $\frac{\sqrt{2}}{2}(|0\rangle - |1\rangle)$. If $x$ is a quantum bit, the value $(H)x$ 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 [3], one can simulate this operation in $\lambda^{vec}$ using the test construction we just described: $(H)\; x = \{(x)\; (\frac{\sqrt{2}}{2} \cdot true + \frac{\sqrt{2}}{2} \cdot false)\; (\frac{\sqrt{2}}{2} \cdot true - \frac{\sqrt{2}}{2} \cdot false)\}$. Note that the thunks are necessary: the term $((x)\; (\frac{\sqrt{2}}{2} \cdot true + \frac{\sqrt{2}}{2} \cdot false))\; (\frac{\sqrt{2}}{2} \cdot true - \frac{\sqrt{2}}{2} \cdot false)$ reduces to the term $\frac{1}{2}((x)\; true\; true + ((x)\; true)\; false + ((x)\; false)\; true + ((x)\; false)\; false)$, which is fundamentally different from the term $H$ we are trying to emulate.

**Summary of results.** This paper is part of a general research framework aiming at understanding the relationship between quantum computation and algebraic lambda-calculi. In particular, the question is to design a typed language whose terms can be interpreted both as quantum data and descriptions of quantum algorithms. The type system would then provide a “quantum logic” and the language a Curry-Howard isomorphism for quantum computation.

Various attempts for drawing this correspondence include [15, 17]. The latter shows that it is possible to encode a tiny, non-trivial subset of $\lambda^{vec}$ into quantum...
circuits. To be generalized, this analysis requires more terms identified as “quantum terms”. Note that this is not immediate from looking directly at the terms. For example, the term $(\lambda x. \text{true}) (\sqrt{2} \cdot \text{true} + \sqrt{2} \cdot \text{false})$ reduces to the term $\sqrt{2} \cdot \text{true}$. Although the former is arguably only written with terms of norm 1, therefore potentially “quantum-like”, the latter is of norm $\sqrt{2}$, therefore surely not quantum.

The goal of this paper is to precisely explicit what is a vector in the space of terms of $\lambda \text{vec}$. We want a characterization 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.

The main contribution of this paper is the proposal of a type system with vectorial features, and the proofs of subject-reduction, strong normalization and confluence of the typed language.

Building the type system. Since we are considering a lambda-calculus, we need at least an arrow type $A \to B$. The terms $\text{true}$ and $\text{false}$ can therefore be typed in the usual way with $B = X \to (X \to X)$, for a fixed term $X$. Since the sum $\sqrt{2} \cdot \text{true} + \sqrt{2} \cdot \text{false}$ is a superposition of terms of type $B$, one could decide to also type it with the type $B$; in general, a linear combination of terms of type $A$ would be of type $A$. But then the terms $\lambda x. (1 \cdot x)$ and $\lambda x. (2 \cdot x)$ 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 $\alpha \cdot A$ is also a valid type and if the terms $s$ and $t$ are of type $A$, the term $\alpha \cdot s + \beta \cdot t$ is of type $(\alpha + \beta) \cdot A$. This trick was first presented in [2] and it allows us to distinguish between the two functions $\lambda x.(1 \cdot x)$ and $\lambda x.(2 \cdot x)$: the former is of type $A \to A$ whereas the latter is of type $A \to (2 \cdot A)$.

Let us now consider the term $\sqrt{2} \cdot (\text{true} - \text{false})$. Using the above addition to the type system, this term should be of type $0 \cdot B$, which is a bit odd in the light of the use we want to make of it. Applying the Hadamard gate to this term produces the term $\text{false}$ of type $B$: the “amplitude” of the type 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 to sum types. Provided that $T = X \to (Y \to X)$ and $F = X \to (Y \to Y)$ (with $Y$ another fixed type), we can type the term $\sqrt{2} \cdot (\text{true} - \text{false})$ with $\sqrt{2} \cdot (T - F)$, of “amplitude” 1, the same as the one of $\text{false}$.

This type system is also able to type the term $H$, with $((I \to \frac{\sqrt{2}}{2}(T + F)) \to (I \to \frac{\sqrt{2}}{2}(T - F)) \to (I \to T)) \to T$ provided that $I$ is an identity type of the form $Z \to Z$, for $T$ and $Z$ fixed types.

Let us try to type the term $(H)_\text{true}$. This is possible provided that the fixed type $T$ is equal to $I \to \frac{\sqrt{2}}{2}(T + F)$. If we now want to type the term $(H)_\text{false}$, the fixed type $T$ needs to be equal to $I \to \frac{\sqrt{2}}{2}(T - F)$: we cannot type the term
\((H) (\frac{\sqrt{2}}{2} \cdot \text{true} + \frac{\sqrt{2}}{2} \cdot \text{false})\) 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, making it system-F alike. The term \(H\) can now be typed with \(\forall T.((I \to \sqrt{2}/2 \cdot (T + F)) \to (I \to \sqrt{2}/2 \cdot (T - F)) \to T)\) and the types \(T\) and \(F\) are updated to be respectively \(\forall XY.X \to (Y \to X)\) and \(\forall XY.X \to (Y \to Y)\). The terms \((H) \text{true}\) and \((H) \text{false}\) can both be well-typed with respective types \(\sqrt{2}/2 \cdot (T + F)\) and \(\sqrt{2}/2 \cdot (T - F)\), as expected.

**Related works.** This avenue of research finds its origin in [3], where the authors present an algebraic lambda-calculus (Lineal). Two papers propose typed versions of this language with type systems related to the one we present.

[2] is uniquely concerned with the addition of scalars in the type system. If \(\alpha\) is a scalar and \(\Gamma \vdash t : T\) is a sequent, \(\alpha \cdot t\) is of type \(\alpha \cdot T\). The developed language provides a static analysis tool for probabilistic computation, when the scalars are taken into positive reals. It however fails to address the issue in this paper: without sums but with negative numbers, the term \(\text{true} - \text{false}\) can only be typed with \(0 \cdot B\), as discussed in the previous paragraph.

[8] is concerned with the addition of sums to a regular type system. The language considered is only the additive fragment of Lineal. In this case, if \(\Gamma \vdash s : S\) and \(\Gamma \vdash t : T\) are two valid sequents, \(s + t\) is of type \(S + T\).

The paper we present here is a merge of these two approaches, for the precise goal of characterizing vectors in complex vector spaces. Because of the possible negative coefficients, this requires to keep track of the ‘direction’ as well as the ‘amplitude’ of a term. More than a mere concatenation of type systems, we think that the typed lambda-calculus proposed in this paper is a novel contribution, with non-trivial choices and results, and that it provides interesting insights on algebraic lambda-calculi.

**Plan of the paper.** In Section 2, we present the language. We discuss the differences with the original language Lineal. In Section 3, we expose the type system and the problem arising from the possibility of linear combination of types. Section 4 is devoted to the subject reduction. We first say why the usual result is not valid, then we provide a solution and a candidate subject reduction theorem; the remaining of the section is concerned with the proof of the result. In Section 5, we briefly sketch the proofs for confluence and strong normalization. Finally we close the paper with some examples in Section 6 and conclusions in Section 7.

In the paper, the proofs may be a bit short. Most of them are fully described in the appendix.

## 2 The Terms

We consider the untyped language \(\lambda^{vec}\) described in Figure 1. It is based on Lineal [3]: terms come in two flavours, basis terms which are only ones that will substitute a variable in a \(\beta\)-reduction step, and general terms.
Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$

Terms are considered modulo associativity and commutativity of the operator $\oplus$, making the reduction into an AC-rewrite system $\lambda$
and consider a more canonical set of rewrite rules. Working with a type system enforcing strong normalization (as shown in Section 5), we follow this approach.

3 The Type System

Let us try to type the term $0$. 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 for example in [2]. In this naive interpretation, we would have $0 \cdot S$ equal to $\overline{0}$ and $\overline{0}$ would be the unit for the addition on types.

Consider the following example. let $\lambda x.x$ be of type $U \rightarrow U$ and let $r$ be of type $R$. The term $\lambda x.x + r - r$ is of type $(U \rightarrow U) + 0 \cdot R$, that is, $(U \rightarrow U)$.

Now choose $b$ of type $U$: we are allowed to say that $(\lambda x.x + r - r)b$ is of type $U$.

This term reduces to $b + (r)b - (r)b$. If the type system is reasonable enough, we should at least be able to type $(r)b$. 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 $r - r$, the type of $r$ is lost in the equivalence $0 \equiv 0 \cdot R$. The only solution is to distinguish $0$ from $0 \cdot R$. We can as well remove it altogether and this is the choice we make for $\lambda vec$.

Without type $\overline{0}$, we do not equate $T + 0 \cdot R$ and $T$.

This means that the term $0$ can be typed with any type $0 \cdot T$, as long as $T$ is inhabited (i.e. $0$ can come from a reduction of $r - r$ for some term $r$ of type $T$).

3.1 Types

Types are defined in Figure 2. They come in two flavours: unit types and general types, that is, linear combinations of types. Unit types include all types of System F [10, Chapter 11] and they will intuitively be used to type basis terms. The arrow type admits only a unit type in its domain. This is due to the fact that the argument of a lambda-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. The substitution of $X$ by $U$ in $T$ is defined as usual and is written $T[U/X]$. For a linear combination, the substitution is defined as follows: $(\alpha \cdot T + \beta \cdot R)[U/X] = \alpha \cdot T[U/X] + \beta \cdot R[U/X]$. We may also use the vectorial notation $T[\vec{U}/\vec{X}]$ for $T[U_1/X_1] \cdots [U_n/X_n]$ if $\vec{X} = X_1, \ldots, X_n$ and $\vec{U} = U_1, \ldots, U_n$, and also $\forall \vec{X}$ for $\forall X_1 \ldots X_n = \forall X_1, \ldots, \forall X_n$.

We define an equivalence relation $\equiv$ on types as the least congruence such that

- $1 \cdot T \equiv T$,
- $\alpha \cdot (\beta \cdot T) \equiv (\alpha \times \beta) \cdot T$,
- $\alpha \cdot T + \alpha \cdot R \equiv \alpha \cdot (T + R)$,
- $\alpha \cdot T + \beta \cdot T \equiv (\alpha + \beta) \cdot T$,
- $T + R \equiv R + T$,
- $T + (R + S) \equiv (T + R) + S$.

The following lemmas give some properties of the equivalence relation. Types are linear combinations of unit types (Lemma 3.1). Note that although we do not have any special type $\overline{0}$ (as discussed in the header of the section), we do have $0 \cdot T$; however $0 \cdot T$ is not the unit of the addition on types. Finally, the equivalence is well-behaved with respect to type constructs (Lemma 3.2).
Lemma 3.1 (Types characterization). For any type $T$, there exist $n \in \mathbb{N}$, $\alpha_1, \ldots, \alpha_n \in S$ and unit types $U_1, \ldots, U_n$ such that $T$ is $\bigsum_{i=1}^{n} \alpha_i \cdot U_i$.

Lemma 3.2 (Equivalence forall-introduction).
1. $\bigsum_{i=1}^{n} \alpha_i \cdot U_i \equiv \bigsum_{j=1}^{m} \beta_j \cdot V_j \iff \bigsum_{i=1}^{n} \alpha_i \cdot \forall X. U_i \equiv \bigsum_{j=1}^{m} \beta_j \cdot \forall X. V_j$.
2. $\bigsum_{i=1}^{n} \alpha_i \cdot \forall X. U_i \equiv \bigsum_{j=1}^{m} \beta_j \cdot V_j \Rightarrow \forall V_j, \exists W_j, V_j \equiv \forall X. W_j$.
3. $T \equiv R \Rightarrow T[U/X] \equiv R[U/X]$.

Proof. Straightforward case by case analysis over the equivalence rules.

3.2 Typing Rules

The typing rules are described in Figure 2. The axiom (ax) and the arrow introduction rule ($\rightarrow_I$) are the usual ones. The rule to type the term $0$ ($\rightarrow_I$) takes into account the discussion of the header of Section 3. This rule also ensures that the type of $0$ is inhabited, discarding problematic types like $0 \cdot \forall X. X$. Any sum of typed terms can be typed using rule $\rightarrow_I$. Similarly, any scaled typed term can be typed with $\alpha_I$. The rule $\equiv$ ensures that equivalent types can be used to type the same terms. Finally, the particular form of the arrow-elimination rule ($\rightarrow_E$) is due to the rewrite rules in group $A$ that distribute sums and scalars over application.

The need and use of this complicated forall elimination can be illustrated by three examples.
Example 3.3. The rule \( \rightarrow_F \) is easier to read for trivial linear combinations. It states that provided that \( \Gamma \vdash s : \forall X.U \rightarrow S \) and \( \Gamma \vdash t : V \), if there exists some type \( W \) such that \( V = U[W/X] \), then since the sequent \( \Gamma \vdash s : V \rightarrow S[W/X] \) is valid, we also have \( \Gamma \vdash (s + t) : S[W/X] \).

Example 3.4. Consider the terms \( b_1 \) and \( b_2 \), of respective types \( U_1 \) and \( U_2 \). The term \( b_1 + b_2 \) is of type \( U_1 + U_2 \). We would reasonably expect the term \( (\lambda x.x)(b_1 + b_2) \) to be also of type \( U_1 + U_2 \). This is the case thanks to Rule \( \rightarrow_F \).

Indeed, type the term \( \lambda x.x \) with the type \( \forall X.X \rightarrow X \) and we can now apply the rule.

Example 3.5. A slightly more evolved example is the projection of a pair of elements. It is possible to encode in System F the notion of pairs and projections: \( (b, c) = \lambda x.((x) b, c) \). \( (b', c') = \lambda x.((x) b', c') \). \( \pi_1 = \lambda x.((\lambda y.\lambda z.y) \alpha \cdot (\lambda x.\gamma.\lambda z.y) \beta \cdot (\lambda x.\gamma.\beta \cdot (\lambda x.\gamma.\lambda z.y) \pi_2) = \lambda x.((\lambda y.\lambda z.y) \alpha \cdot (\lambda x.\gamma.\lambda z.y) \beta \cdot (\lambda x.\gamma.\beta \cdot (\lambda x.\gamma.\lambda z.y) \end{proof}

The term \( (\pi_1 + \pi_2)((b, c) + (b', c')) \) is then typable of type \( U + U' + V + V' \), thanks to Rule \( \rightarrow_F \). Note that this is consistent with the rewrite system, since it reduces to \( b + c + b' + c' \).

4 Subject Reduction

Since the terms of \( \lambda^{\forall \exists} \) are not implicitly typed, we are bound to have sequents such as \( \Gamma \vdash t : T_1 \) and \( \Gamma \vdash t : T_2 \) for the same term \( t \). Using rules \( +_I \) and \( \alpha_I \), we get the valid typing judgment \( \Gamma \vdash (\alpha + \beta) \cdot t : \alpha \cdot T_1 + \beta \cdot T_2 \). Given that \( \alpha \cdot t + \beta \cdot t \) reduces to \( (\alpha + \beta) \cdot t \), a regular subject reduction would ask for the valid sequent \( \Gamma \vdash (\alpha + \beta) \cdot t : \alpha \cdot T_1 + \beta \cdot T_2 \). Since in general we do not have \( \alpha \cdot T_1 + \beta \cdot T_2 \equiv (\alpha + \beta) \cdot T_1 \equiv (\alpha + \beta) \cdot T_2 \), we need to find a way around this.

A first natural solution could be by using the notion of principal types. 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 \( \Gamma \vdash (\alpha + \beta) \cdot t : \alpha \cdot T_1 + \beta \cdot T_2 \) 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 \( \Gamma \), the term \( (\alpha + \beta) \cdot t \) could be typed with any combination \( \gamma \cdot T_1 + \delta \cdot T_2 \), when \( \alpha + \beta = \gamma + \delta \).

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

This approach has the benefit to solve a second pitfall coming the rule \( t + 0 \rightarrow t \). Indeed, although \( x : X \vdash x + 0 : X + 0 \cdot T \) is well-typed for any inhabited \( T \),
the sequent \( x : X \vdash x : X + 0 \cdot T \) is not valid in general. The ordering is extended to state \( X \sqsubseteq X + 0 \cdot T \).

### 4.1 An Ordering Relation on Types.

We start with another relation \( \prec \) inspired from [4]. This relation can be deduced from the rules \( \forall_{L} \) and \( \forall_{R} \) as follows: write \( T \prec R \) if either \( T \equiv \sum_{i=1}^{n} \alpha_{i} \cdot U_{i} \) and \( R \equiv \sum_{i=1}^{n} \alpha_{i} \cdot \forall X.U_{i} \) or \( T \equiv \sum_{i=1}^{n} \alpha_{i} \cdot \forall X.U_{i} \) and \( R \equiv \sum_{i=1}^{n} \alpha_{i} \cdot U_{i}[V/X] \). We denote the reflexive (with respect to \( \equiv \)) and transitive closure of \( \prec \) with \( \preceq \). The relation \( \preceq \) admits a subsumption lemma.

**Lemma 4.1 (\( \preceq \)-subsumption).** For any context \( \Gamma \), any term \( t \) and any types \( T, R \) such that \( T \preceq R \) and no free type variable in \( T \) occurs in \( \Gamma \). Then \( \Gamma \vdash t : T \) implies \( \Gamma \vdash t : R \). \( \square \)

We can now define the ordering relation \( \sqsubseteq \) on types discussed above as the smallest reflexive transitive relation satisfying the rules:

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

Note that the fact that \( \Gamma \vdash t : T \) and \( \Gamma \vdash t : T' \) does not imply that \( \beta \cdot T \sqsubseteq \beta \cdot T' \).
Indeed, although \( \beta \cdot T \sqsubseteq 0 \cdot T + \beta \cdot T' \), we do not have \( 0 \cdot T + \beta \cdot T' \equiv \beta \cdot T' \).

Note also that this ordering is not a subtyping relation. Indeed, although \( \vdash (\alpha + \beta) \cdot \lambda x.\lambda y.x : (\alpha + \beta) \cdot \forall X.X \to (X \to X) \) is valid and \( (\alpha + \beta) \cdot \forall X.X \to (X \to X) \sqsubseteq \alpha \cdot \forall X.X \to (X \to X) + \beta \cdot \forall X.Y \to (Y \to Y) \), the sequent \( \vdash (\alpha + \beta) \cdot \lambda x.\lambda y.x : \alpha \cdot \forall X.X \to (X \to X) + \beta \cdot \forall X.Y \to (Y \to Y) \) is not valid.

### 4.2 Weak Subject Reduction

A weak version of the subject reduction theorem can be stated as follows.

**Theorem 4.2 (Weak subject reduction).** For any terms \( t, t' \), any context \( \Gamma \) and any type \( T \), if \( t \to_R t' \) and \( \Gamma \vdash t : T \), then

- If \( R \notin \text{Group F} \), then \( \Gamma \vdash t' : T \).
- If \( R \in \text{Group F} \) or \( R \) is the rule \( t + 0 \to t \), then \( \exists S \sqsubseteq T \) such that \( \Gamma \vdash t' : S \) and \( \Gamma \vdash t : S \).

How weak is this weak subject reduction? The usual subject reduction result holds for most of the rules. Theorem 4.2 ensures that a term \( t \) of a given type, when reduced, can be typed with a type that is also valid for the term \( t \). In the remainder of this section we prove this theorem. A few definitions and lemmas are required beforehand.

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

Proof. By induction on the derivation of \( \Gamma \vdash t : T \).

Lemma 4.4 is precursor of the generation lemma for scalars (Lemma 4.12). 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.

Lemma 4.4 (Scalars, scaling). For any context \( \Gamma \), term \( t \), type \( T \) and scalar \( \alpha \), if \( \Gamma \vdash \alpha \cdot t : T \), then there exists a type \( R \) such that \( T \equiv \alpha \cdot R \) and if \( \alpha \not= 0 \), \( \Gamma \vdash t : R \). In particular, if \( \Gamma \vdash \alpha \cdot t : \alpha \cdot T \), then \( \Gamma \vdash \overline{t} : T \).

Lemma 4.4 exclude the case of scaling by 0. It is covered by the following.

Lemma 4.5 (Zeros). For any context \( \Gamma \), term \( t \), unit types \( U_1, \ldots, U_n \) and scalars \( \alpha_1, \ldots, \alpha_n \), if \( \Gamma \vdash 0 : \sum \alpha_i \cdot U_i \), then \( \Gamma \vdash t : \sum \delta_i \cdot U_i \) and \( \forall i, \alpha_i = 0 \).

A basis term can always be given a unit type.

Lemma 4.6 (Basis terms). For any context \( \Gamma \), type \( T \) and basis term \( b \), if \( \Gamma \vdash b : T \) then there exists a unit type \( U \) such that \( T \equiv U \).

The following lemma is standard in proofs of subject reduction for System F-like systems, and can be found, e.g. in [4, Ch. 4]. It ensures that by substituting type variables for types or term variables for terms in an adequate manner, the derived type is still valid.

Lemma 4.7 (Substitution lemma). For any term \( t \), basis term \( b \), term variable \( x \), context \( \Gamma \), types \( T \), \( U \), \( \overline{W} \) and type variables \( \overline{X} \),

1. if \( \Gamma \vdash t : T \), then \( \Gamma[U/X] \vdash \overline{t} : T[U/X] \);
2. if \( \Gamma, x : U \vdash t : T, \Gamma \vdash b : U[\overline{W}/\overline{X}] \) and \( \overline{X} \not\in \text{FV}(\Gamma) \), then \( \Gamma \vdash t[b/x] : T[\overline{W}/\overline{X}] \).

Proving subject reduction requires the proof that each reduction rule preserves types. Thus three generation lemmas are required: two classical ones, for applications (Lemma 4.8) and for abstractions (Lemma 4.9 and Corollary 4.11) and one for linear combinations: sums, scalars and zero (Lemma 4.12).

Lemma 4.8 (Generation lemma (application)). For any terms \( t, r \), any context \( \Gamma \) and any type \( T \), if \( \Gamma \vdash (t \ r) : T \), 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 \( \Gamma \vdash \overline{t} : \sum \alpha_i \cdot \forall \overline{X} (U \rightarrow T_i) \), \( \Gamma \vdash r : \sum \beta_j \cdot V_j \), for all \( V_j \), there exists \( \overline{W}_j \) such that \( U[\overline{W}_j/\overline{X}] = V_j \) and \( \sum \alpha_i \cdot \forall \overline{X} (U \rightarrow T) \).

Lemma 4.9 (Generation lemma (abstraction)). For any term variable \( x \), term \( t \), context \( \Gamma \) and type \( T \), if \( \Gamma \vdash \lambda x . t : R \), there exist types \( U \) and \( T \) such that \( U \rightarrow T \geq R \) and \( \Gamma, x : U \vdash t : T \).
Finally, the following lemma is needed for the proof of Corollary 4.11.

**Lemma 4.10 (Arrows comparison).** For any types $T, R$ and any unit types $U, V$, if $V \to R \leq U \to T$, then there exist $\vec{W}, \vec{X}$ such that $U \to T \equiv (V \to R)[\vec{W} / \vec{X}]$. □

**Corollary 4.11 (of Lemma 4.9).** For any context $\Gamma$, term variable $x$, term $t$, type variables $\vec{X}$ and types $U$ and $T$, if $\Gamma \vdash \lambda x. t : \forall \vec{X}. (U \to T)$ then the typing judgment $\Gamma, x : U \vdash t : T$ is valid.

**Proof.** By Lemma 4.9, $\exists \vec{V}, \vec{R}, V \to R \leq \forall \vec{X}. (U \to T)$ and $\Gamma, x : V \vdash t : R$. Note that $V \to R \leq \forall \vec{X}. (U \to T) \leq U \to T$, so by Lemma 4.10, $\exists \vec{W}, \vec{Y}$ 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 4.7, $\Gamma[\vec{W} / \vec{Y}], x : V[\vec{W} / \vec{Y}] \vdash t : R[\vec{W} / \vec{Y}]$. By Lemma 4.3 and rule $\equiv$, $\Gamma[\vec{W} / \vec{Y}], x : U \vdash t : T$. If $\Gamma[\vec{W} / \vec{Y}] \equiv \Gamma$, then we are finished. In the other case, $\vec{Y}$ appears free on $\Gamma$, however, to get $U \to T$ from $V \to R$ as a type for $\lambda x. t$ by substitutions, we would need to use the rule $\forall_1$, so $\vec{Y}$ cannot appear free in $\Gamma$, contradiction. So, $\Gamma, x : U \vdash t : T$. □

**Lemma 4.12 (Generation lemma (linear combinations)).** For any context $\Gamma$, scalar $\alpha$, terms $t$ and $r$ and types $S$ and $T$, if $\Gamma \vdash t \vdash \alpha : R$ then there exists types $R$ and $R'$ such that $\Gamma \vdash \alpha \cdot t : R$, $\Gamma \vdash t : R'$ and $R + R' \leq S$; if $\Gamma \vdash \alpha \cdot t : T$, then there exists a type $R$ such that $\alpha \cdot R \leq T$ and $\Gamma \vdash \alpha \cdot t : \alpha : R$; if $\Gamma \vdash \alpha : 0 : T$, then there exists a type $R$ such that $T \equiv \alpha \cdot 0 : R$. □

One last lemma characterizes of the ordering relation $\sqsubseteq$.

**Lemma 4.13 (Order characterization).** For any type $T$, unit types $V_1, \ldots, V_m$ and scalars $\beta_1, \ldots, \beta_m$, if $R \sqsubseteq \sum_{j=1}^{m} \beta_j \cdot V_j$, then there exist a scalar $\delta$, a natural number $k$, a set $N \subseteq \{1, \ldots, m\}$ and a unit type $W \leq V_k$ such that $R \equiv \delta \cdot W + \sum_{j \in N} \beta_j \cdot V_j$ and $\sum_{j=1}^{m} \beta_j = \delta + \sum_{j \in N} \beta_j$. □

### 4.3 Proof of Theorem 4.2

Let $t \to_R t'$ and $\Gamma \vdash t : T$. We proceed by induction, and only list two interesting cases. **Rule** $\alpha \cdot t + \beta \cdot t \to (\alpha + \beta) \cdot t$. Let $\Gamma \vdash \alpha \cdot t + \beta \cdot t : T$. Then by Lemma 4.12, $\exists R, S$ such that $\Gamma \vdash \alpha \cdot t : R$ and $\Gamma \vdash \beta \cdot t : S$ with $R + S \leq T$. Then by Lemma 4.12, $\exists R', \alpha \cdot R' \leq R$ and $\Gamma \vdash \alpha \cdot t : \alpha \cdot R'$, also $\exists S', \beta \cdot S' \leq S$ and $\Gamma \vdash \beta \cdot t : \beta \cdot S'$. If $\alpha = 0$ (or analogously $\beta = 0$), then by Lemma 4.4, $\Gamma \vdash t : R'$ and so by $\alpha_1$ we conclude $\Gamma \vdash (\alpha + \beta) \cdot t : (\alpha + \beta) \cdot R'$. Notice that $(\alpha + \beta) \cdot R' \equiv \alpha \cdot R' + \beta \cdot S' \equiv R + S \equiv T$. Also using rules $+$ and $\equiv$, we conclude $\Gamma \vdash \alpha \cdot t + \beta \cdot t : (\alpha + \beta) \cdot R'$. If $\alpha = \beta = 0$, then notice that $\Gamma \vdash 0 \cdot t : 0 \cdot R'$ and $0 \cdot R' \equiv 0 \cdot R' + 0 \cdot S' \equiv R + S \equiv T$. Also, using rules $+$ and $\equiv$, we conclude $\Gamma \vdash 0 \cdot t + 0 \cdot t : 0 \cdot R'$. **Rule** $(\lambda x. t) b \to t[b/x]$. Let $\Gamma \vdash (\lambda x. t) b : T$. Then by Lemma 4.8, there exist $n, m, \alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_m, U, T_1, \ldots, T_n$ such that $\Gamma \vdash \lambda x. t : \sum_{i=1}^{n} \alpha_i \cdot \forall \vec{X}. (U \to T_i)$ and $\Gamma \vdash b : \sum_{j=1}^{m} \beta_j \cdot V_j$ with $\sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j$. Then $T_i[\vec{W}_j / \vec{X}] \leq T$, where $\forall V_j, \vec{W}_j$ is such that $U[\vec{W}_j / \vec{X}] \equiv V_j$. By Lemma 4.6,
\[ \sum_{i=1}^{n} \alpha_i \cdot \forall \vec{X}. (U \rightarrow T_i) \equiv \forall \vec{X}. (U \rightarrow T_i) \quad \text{and} \quad \forall i, k, \ T_i = T_k, \ \text{analogously} \]
\[ \sum_{j=1}^{m} \beta_j \cdot V_j \equiv V_j \ \text{where} \ \forall j, h, \ V_j = V_k. \ \text{So} \ \sum_{i=1}^{n} \alpha_i = 1 \ \text{and} \ \sum_{j=1}^{m} \beta_j = 1. \ \text{Then} \]
by rule \( \equiv \), \( \Gamma \vdash \lambda x.t : \forall \vec{X}. (U \rightarrow T_i) \), and \( \Gamma \vdash b : V_i \). Thus, by Corollary 4.11, \( \Gamma, x : U \vdash t : T_i \). Notice that \( V_i \equiv U[\vec{W}_i/\vec{X}] \), then, by Lemma 4.7[2], we have \( \Gamma \vdash t[b/x] : T_i[\vec{W}_j/\vec{X}] \). Since \( T_i[\vec{W}_j/\vec{X}] \equiv (1 \times 1) \cdot T_i[\vec{W}_j/\vec{X}] = (\sum_{i=1}^{n} \alpha_i) \times (\sum_{j=1}^{m} \beta_j) \).
\[ T_i[\vec{W}_j/\vec{X}] = (\sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j) \cdot T_i[\vec{W}_j/\vec{X}] \]
and as all the \( T_i \) are equivalents between them, this type is equivalent to \( \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j \cdot T_i[\vec{W}_j/\vec{X}] \leq T \). By Lemma 4.1, \( \Gamma \vdash t[b/x] : T \).}

\[ \sum_{n=1}^{n} \alpha_i \cdot \forall \vec{X}. (U \rightarrow T_i) \equiv \forall \vec{X}. (U \rightarrow T_i) \quad \text{and} \quad \forall i, k, \ T_i = T_k, \ \text{analogously} \]
\[ \sum_{j=1}^{m} \beta_j \cdot V_j \equiv V_j \ \text{where} \ \forall j, h, \ V_j = V_k. \ \text{So} \ \sum_{i=1}^{n} \alpha_i = 1 \ \text{and} \ \sum_{j=1}^{m} \beta_j = 1. \ \text{Then} \]
by rule \( \equiv \), \( \Gamma \vdash \lambda x.t : \forall \vec{X}. (U \rightarrow T_i) \), and \( \Gamma \vdash b : V_i \). Thus, by Corollary 4.11, \( \Gamma, x : U \vdash t : T_i \). Notice that \( V_i \equiv U[\vec{W}_i/\vec{X}] \), then, by Lemma 4.7[2], we have \( \Gamma \vdash t[b/x] : T_i[\vec{W}_j/\vec{X}] \). Since \( T_i[\vec{W}_j/\vec{X}] \equiv (1 \times 1) \cdot T_i[\vec{W}_j/\vec{X}] = (\sum_{i=1}^{n} \alpha_i) \times (\sum_{j=1}^{m} \beta_j) \).
\[ T_i[\vec{W}_j/\vec{X}] = (\sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j) \cdot T_i[\vec{W}_j/\vec{X}] \]
and as all the \( T_i \) are equivalents between them, this type is equivalent to \( \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j \cdot T_i[\vec{W}_j/\vec{X}] \leq T \). By Lemma 4.1, \( \Gamma \vdash t[b/x] : T \).}

5 Confluence and Strong Normalization.

The language has the usual properties for a typed lambda-calculus: the reduction is locally confluent and the type system enforces strong normalisation (modulo associativity and commutativity). From these two results, we infer the confluence of the rewrite system.

Theorem 5.1 (Local confluence). For any terms \( t, r_1 \) and \( r_2 \), if \( t \rightarrow r_1 \) and \( t \rightarrow r_2 \), then there exists a term \( u \) such that \( r_1 \rightarrow^* u \) and \( r_2 \rightarrow^* u \).

Proof. First, one proves the local confluence of the algebraic fragment of the rewrite system (that is, all the rules minus the beta-reduction). This has been automatized [16] using COQ [6]. The proof of confluence of the beta-reduction alone is a straightforward extension of the proof of confluence of the usual untyped \( \lambda \)-calculus which can be found in many textbooks, e.g. [12, Sec. 1.3]. Finally, a straightforward induction entails that the two fragments commute: this entails the local confluence of the whole rewrite system.

For proving strong normalization of well-typed terms, we use reducibility candidates, a well-known method described for example in [10, Ch. 14] The technique is adapted to linear combinations of terms, following [8, 5]. A neutral term is a term admitting a reduction. We write \( \Lambda_0 \) for the set of closed terms and \( SN_0 \) for the set of closed, strongly normalizing terms. If \( t \) is any term, Red(\( t \)) is the set of all terms \( t' \) such that \( t \rightarrow t' \). The definition naturally extends to sets of terms. We say that a set \( S \) of closed terms is a reducibility candidate, denoted with \( S \in RC \) if the following conditions are verified: \( RC_1 \). Strong normalization: \( S \subseteq SN_0 \). \( RC_2 \). Stability under reduction: if \( t \in S \) implies Red(\( t \)) \( \subseteq S \). \( RC_3 \). Stability under neutral expansion: If \( t \) is closed neutral and Red(\( t \)) \( \subseteq S \) then \( t \in S \). \( RC_4 \). Closure under linear combinations: If \( s \) and \( t \) are members of \( S \) then so are \( 0, s \cdot s, \) and \( s + t \). If \( S \) is a set of terms, we define \( \overline{S} \) to be the closure of \( S \) under \( RC_3 \) and \( RC_4 \). If \( A \) and \( B \) are in \( RC \), \( A \rightarrow B \) is the set \( \{ t \in \Lambda_0 | \forall u \in A, (t \ u) \in B \} \) and \( A \oplus B \) is the set \( \overline{A \cup B} \).

Lemma 5.2. If \( A, B \) and the \( A_i \)’s are in \( RC \), so are \( A \rightarrow B, A \oplus B \) and \( \cap_i A_i \).

Proof. The main difficulty of the proof is to show that linear combinations of strongly normalizing terms are strongly normalizing. It is done by using a measure on terms decreasing on algebraic rewrites [3].
Lemma 5.3. The operation $\oplus$ on $\mathcal{RC}$ is commutative and associative, and it commutes with the intersection. \qed

A valuation $\rho$ is a partial function from types variables to reducibility candidates. The interpretation $[T]_\rho$ of a type $T$ is defined inductively as follows:

$[X]_\rho = \rho(X)$, $[U \rightarrow T]_\rho = [U]_\rho \rightarrow [T]_\rho$, $[\forall X.S]_\rho = \bigcap_{A \in \mathcal{RC}} [S]_{\rho,X \rightarrow A}$, $[\alpha \cdot S]_\rho = [S]_\rho$, $[S + T]_\rho = [S]_\rho \oplus [T]_\rho$. From Lemma 5.2, the interpretation of any type is a reducibility candidate. We extend the definition of interpretation to typing contexts: If $\Gamma$ is $(x_i : T_i)_i$ and if $\rho$ is a valuation, then $[\Gamma]_\rho$ is the set of substitutions sending any $x_i$ in $\Gamma$ to $[T_i]_\rho$. We now write $\Gamma \models t : T$ if for every valuation $\rho$, every substitution $\sigma$ in $[\Gamma]_\rho$ we have $\sigma(t) \in [T]_\rho$.

Lemma 5.4. For every valid sequent $\Gamma \vdash t : T$ we also have $\Gamma \models t : T$.

Proof. The proof is done by induction on the size of $t$. Choose a typing derivation for $\Gamma \vdash t : T$, we proceed by case distinction on the first rule used. We prove three cases. For the case $(\alpha T)$, we have $T = \alpha \cdot T'$ and $t = \alpha \cdot t'$. Then $\sigma(t) = \alpha \cdot \sigma(t')$. Note that $t'$ is smaller than $t$. The sequent $\Gamma \vdash t' : T'$ is valid: $\sigma(t') \in [T']_{\rho'}$. From $\mathcal{RC}_4$ we conclude that $\sigma(t)$ belongs to $[T]_\rho = [T]_{\rho'}$. The case $(+T)$ is similar: the set $[R]_\rho \oplus [S]_\rho$ is closed under sums. Now, we show the case $(\rightarrow_E)$. The term $t$ is of the form $(s) r$, and for induction hypothesis we have $\Gamma \models s : \sum_i \alpha_i \cdot \forall X_i (U \rightarrow T_i)$ and $\Gamma \models r : \sum_j \beta_j \cdot V_j$. We want to show that $\sigma((s) r) \in \oplus_{i,j} [T_i]_{\rho,X_i \rightarrow \bar{W}_i}$, that is, $\sigma((s) r)$ belongs to $\oplus_{i,j} [T_i]_{\rho,X_i \rightarrow \bar{W}_i}$.

Since both $\sigma(s)$ and $\sigma(r)$ are strongly normalizing, we proceed by induction on the sum of the lengths of their longest rewrite sequence. The set $\text{Red}(\sigma((s) r))$ contains:

- $(\sigma(s')) r'$ and $(s') \sigma(r)$ when $\sigma(s) \rightarrow s'$ or $\sigma(r) \rightarrow r'$. They both belong to $\oplus_{i,j} [T_i]_{\rho,X_i \rightarrow \bar{W}_i}$ by induction hypothesis.
- A term coming from one of the rewrite of the group $A$. We conclude by noting that we obtain a linear combination of terms smaller that the original one.
- We can conclude with the induction hypothesis, Lemma 4.12 and $\mathcal{RC}_1$.
- The term $\sigma(s') \sigma(r) / x$, when $s = \lambda x . s'$ and $r$ is a base term. Note that this term is of the form $\sigma'(s')$, where $\sigma' \in [T, x : U]_{\rho'}$. We are in the situation where the types of $s$ and $r$ are respectively $\forall X_i (U \rightarrow T)$ and $V$. There is $\bar{W}$ such that $V = U[\bar{W}/X]$, so we can conclude by noting that a set $[S[U/X]]$ is larger than the intersection $\bigcap_{A \in \mathcal{RC}} [S]_{\rho,X \rightarrow A}$.

Since the set $\text{Red}(\sigma((s) r))$ is contained in $\oplus_{i,j} [T_i]_{\rho,X_i \rightarrow \bar{W}_i}$, we can conclude by applying $\mathcal{RC}_3$. \qed

Theorem 5.5 (Strong normalization). Suppose that $\Gamma \vdash t : T$ is a valid typing judgment, then $t$ is strongly normalizing.

Proof. We use Lemma 5.4 and $\mathcal{RC}_1$ on the term $\lambda \bar{x} . t$, where $\bar{x}$ contains all the term variables in $\Gamma$. We deduce that $t$ is strongly normalizing by contradiction: if it were not, so would be $\lambda \bar{x} . t$. \qed
Corollary 5.6 (Confluence). If \( \Gamma \vdash s : S \) is a valid typing judgment and if \( s \rightarrow^* r \) and \( s \rightarrow^* t \), then there exists \( s' \) such that \( r \rightarrow^* s' \) and \( t \rightarrow^* s' \).

Proof. A rewrite system that is both locally confluent and strongly normalizing is confluent \([14, \text{Ch. 10}]\).

6 Expressing Matrices and Vectors

In this section we come back to the motivating example introducing the type system and we show how \( \lambda^\text{vec} \) handles the Hadamard gate, and how to encode matrices and vectors.

With an empty typing context, the booleans \( \text{true} = \lambda x.\lambda y.x \) and \( \text{false} = \lambda x.\lambda y.y \) can be respectively typed with the types \( \alpha \cdot \beta \cdot T = \forall X.Y. X \rightarrow (Y \rightarrow X) \) and \( \mathbb{F} = \forall X.Y. (Y \rightarrow Y) \). The superposition \( \alpha \cdot \text{true} + \beta \cdot \text{false} \) is of type \( \alpha \cdot T + \beta \cdot \mathbb{F} \). (Note that it can also be typed with \((\alpha + \beta) \cdot \forall X.Y. X \rightarrow X \rightarrow X\).

Proof. With an empty typing context, \( \lambda x.\lambda y.x \) can be respectively typed with the types \( \alpha \cdot \beta \cdot T = \forall X.Y. X \rightarrow (Y \rightarrow X) \) and \( \mathbb{F} = \forall X.Y. (Y \rightarrow Y) \). The superposition \( \alpha \cdot \text{true} + \beta \cdot \text{false} \) is of type \( \alpha \cdot T + \beta \cdot \mathbb{F} \). (Note that it can also be typed with \((\alpha + \beta) \cdot \forall X.Y. X \rightarrow X \rightarrow X\).

In this paper we define a strongly normalizing, confluent, typed, algebraic \( \lambda \)-calculus satisfying a weak subject reduction. The language allows making arbitrary linear combinations of \( \lambda \)-terms \( \alpha \cdot t + \beta \cdot u \). Its \textit{vectorial} type system is a fine-gained analysis tool describing the “vectorial” properties of typed terms. First, it keeps track of the ‘amplitude of a term’, \( i.e. \) if \( t \) and \( u \) both have the same type \( U \), then \( \alpha \cdot t + \beta \cdot u \) has type \((\alpha + \beta) \cdot U\). Then it keeps track of the ‘direction of a term’, \( i.e. \) if \( t \) and \( u \) have types \( U \) and \( V \) respectively, then \( \alpha \cdot t + \beta \cdot u \) has type \( \alpha \cdot U + \beta \cdot V \). It is expressive enough to be able to type the encoding of matrices and vectors.

7 Conclusion

In this paper we define a strongly normalizing, confluent, typed, algebraic \( \lambda \)-calculus satisfying a weak subject reduction. The language allows making arbitrary linear combinations of \( \lambda \)-terms \( \alpha \cdot t + \beta \cdot u \). Its \textit{vectorial} type system is a fine-gained analysis tool describing the “vectorial” properties of typed terms: First, it keeps track of the ‘amplitude of a term’, \( i.e. \) if \( t \) and \( u \) both have the same type \( U \), then \( \alpha \cdot t + \beta \cdot u \) has type \((\alpha + \beta) \cdot U\). Then it keeps track of the ‘direction of a term’, \( i.e. \) if \( t \) and \( u \) have types \( U \) and \( V \) respectively, then \( \alpha \cdot t + \beta \cdot u \) has type \( \alpha \cdot U + \beta \cdot V \). It is expressive enough to be able to type the encoding of matrices and vectors.

The tool we propose in this paper is a first step towards the analysis of the “quantumness” of algebraic lambda-calculi. It is a step towards a “quantum logic” coming readily with a Curry-Howard isomorphism. The logic we are sketching merges intuitionistic logic and vectorial structure. It results into a novel and intriguing tool.
The next step in the study of the quantumness of the linear algebraic lambda-calculus is the exploration of the notion of orthogonality between terms and to valid this notion by the mean of a compilation into quantum circuits. The work in [17] may be showing that it is worthwhile pursuing in this direction.

Acknowledgements

We would like to thank to Gilles Dowek, Thomas Ehrhard, Jonathan Grattage, Philippe Jorrand, Michele Pagani, Simon Perdrix, Barbara Petit, Frédéric Prost and Lionel Vaux for enlightening discussions.

References

6. Coq Dev. Team: The Coq proof assistant ref. man.. INRIA, 8.2ed. (Feb 2009)
A Proofs of lemmas

Lemma 3.1 (Types characterization) For any type $T$, $\exists n \in \mathbb{N}, \alpha_1, \ldots, \alpha_n \in S$ and unit types $U_1, \ldots, U_n$ such that $T \equiv \sum_{i=1}^{n} \alpha_i \cdot U_i$.

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

Lemma 4.10 (Arrows comparison) For any types $T, R$ and any unit types $U, V$, if $V \rightarrow R \preceq U \rightarrow T$, then there exists $\bar{W}, \bar{X}$ such that $U \rightarrow T \equiv (V \rightarrow R)[\bar{W}/\bar{X}]$.

*Proof.* A map $(\cdot)^\circ$ from types to types is defined by

\[
\begin{align*}
X^\circ &= X \\
(\alpha \cdot T)^\circ &= \alpha \cdot T^\circ \\
(U \rightarrow T)^\circ &= U \rightarrow T \\
(T + R)^\circ &= T^\circ + R^\circ \\
(\forall X. U)^\circ &= U^\circ
\end{align*}
\]

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 $\exists \bar{U}, \bar{X} / R^\circ \equiv T^\circ[\bar{U}/\bar{X}]$

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 $T = V \rightarrow R$, then $((V \rightarrow R)(U/X))^\circ = (V[U/X] \rightarrow R(U/X))^\circ = V[U/X] \rightarrow R[U/X] = (V \rightarrow R)[U/X] = (V \rightarrow R)^\circ[U/X]$.
   - 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 $T = \alpha \cdot R$, then $(\alpha \cdot R(U/X))^\circ = \alpha \cdot (R(U/X))^\circ$, which, by the induction hypothesis, is equivalent to $\alpha \cdot (R^\circ(U/X)] = (\alpha \cdot R)^\circ[U/X]$.
   - 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 $T \preceq R$. Cases:
   - Let $T = \sum_{i=1}^{n} \alpha_i \cdot U_i$ and $R = \sum_{i=1}^{n} \alpha_i \cdot \forall X.U_i$. Then $T^\circ = (\sum_{i=1}^{n} \alpha_i \cdot U_i)^\circ$ which is equal to $\sum_{i=1}^{n} \alpha_i \cdot U_i^\circ = \sum_{i=1}^{n} \alpha_i \cdot (\forall X.U_i)^\circ = (\sum_{i=1}^{n} \alpha_i \cdot \forall X.U_i)^\circ$ which is just $R^\circ$. 

Induction on the typing derivation.

Proof.

Lemma 4.4 (Scalars)

\[ \Gamma \vdash R \]

Proof of the lemma:

One can assume

\[ \Gamma \vdash \alpha \cdot \top \]

Let \( \Gamma \vdash \alpha \cdot \top \), then \( \Gamma \vdash \alpha \cdot \top \cdot \top \). By the induction hypothesis, \( \exists R \) such that \( \Gamma \vdash \alpha \cdot \top \cdot \top \rightarrow R \) and \( \alpha \neq 0 \). Then by lemma 3.1, \( R \equiv \sum_{i=1}^{n} \alpha_i \cdot \top \cdot \top \vdash \top \cdot \top \). By rule \( \forall \), we get \( \Gamma \vdash \alpha \cdot \top \cdot \top \vdash \top \cdot \top \). Then by lemma 3.2[1], we have \( \sum_{i=1}^{n} \alpha_i \cdot \forall X.\top \equiv \sum_{i=1}^{n} \alpha_i \cdot \top \cdot \forall X.\top \). Also, if \( \alpha \neq 0 \), then \( \Gamma \vdash \alpha \cdot \top \cdot \top \vdash \top \cdot \top \). By rule \( \forall \), we get \( \Gamma \vdash \alpha \cdot \top \cdot \top \vdash \top \cdot \top \). Then by rule \( \forall \), we conclude with \( \Gamma \vdash \alpha \cdot \top \cdot \top \vdash \top \cdot \top \) and this

Lemma 4.4 (Scalars) For any context \( \Gamma \), term \( t \), type \( T \) and scalar \( \alpha \), if \( \Gamma \vdash \alpha \cdot t : T \), then there exists a type \( R \) such that \( T \equiv \alpha \cdot R \) and if \( \alpha \neq 0 \), \( \Gamma \vdash t : R \).

Proof. Induction on the typing derivation.

– Let \( \Gamma \vdash \alpha \cdot t : \alpha \cdot T \) as a consequence of \( \Gamma \vdash t : T \) and rule \( \alpha \). Trivial.

– Let \( \Gamma \vdash \alpha \cdot t : \sum_{i=1}^{n} \alpha_i \cdot \forall X.\top \) as a consequence of \( \Gamma \vdash \alpha \cdot t : \sum_{i=1}^{n} \alpha_i \cdot \top \cdot \top \). By the induction hypothesis, \( \exists R \) such that \( \sum_{i=1}^{n} \alpha_i \cdot \top \cdot \top \vdash \top \cdot \top \) and \( \alpha \neq 0 \). Then by lemma 3.1, \( \Gamma \vdash \alpha \cdot t : \top \cdot \top \). By rule \( \forall \), we get \( \Gamma \vdash \alpha \cdot t : \top \cdot \top \). Then by lemma 3.2[1], we have \( \sum_{i=1}^{n} \alpha_i \cdot \forall X.\top \equiv \sum_{i=1}^{n} \alpha_i \cdot \top \cdot \forall X.\top \). Also, if \( \alpha \neq 0 \), then \( \Gamma \vdash \sum_{i=1}^{n} \alpha_i \cdot \top \cdot \top \vdash \top \cdot \top \). By rule \( \forall \), we get \( \Gamma \vdash \sum_{i=1}^{n} \alpha_i \cdot \top \cdot \top \vdash \top \cdot \top \). Then by rule \( \forall \), we conclude with \( \Gamma \vdash \sum_{i=1}^{n} \alpha_i \cdot \top \cdot \top \vdash \top \cdot \top \) and this
Lemma 4.6 (Basis terms) For any term \( t \), basis term \( b \), term variable \( x \), context \( \Gamma \), types \( T, U, W \) and type variables \( X \),

1. If \( \Gamma \vdash t : T \), then \( \Gamma[U/X] \vdash t[U/X] \).
2. If \( \Gamma, x : U \vdash t : T, \Gamma \vdash b : U[\vec{W} / \vec{X}] \) and \( \vec{X} \notin FV(\Gamma) \), then \( \Gamma \vdash t[b/x] : T[\vec{W} / \vec{X}] \).

**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 \( \Gamma \vdash 0 : 0 \cdot T : T \) and rule \( 0_I \). Then by the induction hypothesis \( \Gamma[U/X] \vdash t : T[U/X] \), so by rule \( 0_I \), \( \Gamma[U/X] \vdash 0 : 0 \cdot T[U/X] \). Notice that \( 0 \cdot T[U/X] = (0 \cdot T)[U/X] \).
   - Let \( \Gamma \vdash (t \cdot (\sum_{i=1}^{n'} \alpha_i \times \beta_j \cdot T_i)[\vec{W}_j / \vec{X}]) \) as a consequence of \( \Gamma \vdash t \cdot (\sum_{i=1}^{n'} \alpha_i \cdot \forall \vec{Y} \cdot (V \rightarrow T_i)) \). Then by the induction hypothesis \( \Gamma[U/X] \vdash t : (\sum_{i=1}^{n'} \alpha_i \cdot \forall \vec{Y} \cdot (V[U/X] \rightarrow T_i[U/X]))[U/X] \) and \( \Gamma[U/X] \vdash (\sum_{i=1}^{n'} \alpha_i \cdot \forall \vec{Y} \cdot (V[U/X] \rightarrow T_i[U/X]))[U/X] \) Notice that, up to variable renaming, \( T_j[U/X] = (V[U/X])(\vec{U}_j / \vec{Y}) \). Then using rule \( \rightarrow_E \), \( \Gamma[U/X] \vdash (t \cdot (\sum_{i=1}^{n'} \alpha_i \times \beta_j \cdot T_i[U/X]))[\vec{U}_j / \vec{Y}] \). Notice that, up to variable renaming, we can conclude the case by realizing that \( \sum_{i=1}^{n'} \sum_{j=1}^{m} \alpha_i \times \beta_j \cdot T_i[U/X] \) is equal to \( (\sum_{i=1}^{n'} \sum_{j=1}^{m} \alpha_i \times \beta_j \cdot T_i[U/X])(\vec{U}_j / \vec{Y}) \).
   - Let \( \Gamma \vdash \lambda x : T \cdot e : T \) as a consequence of \( \Gamma, x : V \vdash t : T \) and rule \( \rightarrow_I \). Then by the induction hypothesis \( \Gamma[U/X], x : V[U/X] \vdash t : T[U/X] \). So using rule \( \rightarrow_I \), \( \Gamma[U/X] \vdash \lambda x : T \cdot e : T[U/X] \). Notice that \( V[U/X] \rightarrow T[U/X] = (V \rightarrow T)[U/X] \).
   - Let \( \Gamma \vdash t : (\sum_{i=1}^{n} \alpha_i \cdot \forall \vec{Y} \cdot V_i[U/X]) \) as a consequence of \( \Gamma \vdash t : (\sum_{i=1}^{n} \alpha_i \cdot \forall \vec{Y} \cdot V_i[U/X])[U/X] \) and rule \( \forall_E \). Then by the induction hypothesis one has that \( \Gamma[U/X] \vdash t : (\sum_{i=1}^{n} \alpha_i \cdot \forall \vec{Y} \cdot V_i[U/X]) \). Notice that \( (\sum_{i=1}^{n} \alpha_i \cdot \forall \vec{Y} \cdot V_i[U/X]) \) is equal to \( \sum_{i=1}^{n} \alpha_i \cdot \forall \vec{Y} \cdot V_i[U/X] \). So by rule \( \forall_E \), we can derive the following sequent \( \Gamma[U/X] \vdash t : (\sum_{i=1}^{n} \alpha_i \cdot \forall \vec{Y} \cdot V_i[U/X])[W/Y] \). Notice that, up to variable renaming \( (\sum_{i=1}^{n} \alpha_i \cdot \forall \vec{Y} \cdot V_i[U/X])[W/Y] = (\sum_{i=1}^{n} \alpha_i \cdot \forall \vec{Y} \cdot V_i[U/X])[W/Y] \).
   - Let \( \Gamma \vdash t + r : T + R \) as a consequence of \( \Gamma \vdash t : T \) and \( \Gamma \vdash r : R \) with rule \( +_R \). Then by the induction hypothesis \( \Gamma[U/X] \vdash t : T[U/X] \) and \( \Gamma[U/X] \vdash r : R[U/X] \). So by rule \( +_R \), \( \Gamma[U/X] \vdash t + r : T[U/X] + R[U/X] \). Notice that \( T[U/X] + R[U/X] = (T + R)[U/X] \).
   - Let \( \Gamma \vdash \alpha \cdot t : \alpha \cdot T \) as a consequence of \( \Gamma \vdash t : T \) and rule \( \alpha_I \). The by the induction hypothesis \( \Gamma[U/X] \vdash t : T[U/X] \). So by rule \( \alpha_I \), \( \Gamma[U/X] \vdash \alpha \cdot t : \alpha \cdot T[U/X] \). Notice that \( \alpha \cdot T[U/X] = (\alpha \cdot T)[U/X] \).
   - Let \( \Gamma \vdash t : R \) as a consequence of \( \Gamma \vdash t : T \) with \( T \equiv R \), and rule \( \equiv \). Then by the induction hypothesis \( \Gamma[U/X] \vdash t : T[U/X] \). By lemma 3.2[3] \( T \equiv R \Rightarrow T[U/X] \equiv R[U/X] \). So by rule \( \equiv \), \( \Gamma[U/X] \vdash t : R[U/X] \)

2. Induction on the typing derivation of \( \Gamma, x : U \vdash t : T \).
– Let $\Gamma, x : U \vdash x : U$ as a consequence of rule $ax$. Trivial since $x(b/x) = b$.
– Let $\Gamma, x : U \vdash 0 : 0. \cdot T$ as a consequence of $\Gamma, x : U \vdash t : T$ and rule $0_t$. Then by the induction hypothesis $\Gamma \vdash t[b/x] : T[\vec{W}/\vec{X}]$. Then by rule $0_t$, $\Gamma \vdash 0 : 0. \cdot T[\vec{W}/\vec{X}]$. Notice that $0 = 0[b/x]$ and $0 \cdot T[\vec{W}/\vec{X}] = (0 \cdot T)[\vec{W}/\vec{X}]$.

– Let $\Gamma, x : U \vdash (t \ r) : \sum_{i=1}^{n'} \sum_{j=1}^{n''} \alpha_i \cdot \beta_j \cdot T_i[\vec{W}_i/\vec{Y}_i]$ as a consequence of $\Gamma, x : U \vdash r : \sum_{j=1}^{n''} \beta_j \cdot V_j'$ and $\Gamma, x : U \vdash t : \sum_{i=1}^{n'} \alpha_i \cdot \forall \vec{Y}. (V \rightarrow T_i)$ where $\forall \vec{Y}_j', \exists \vec{W}_j, \forall \vec{W}_j / \vec{V}[\vec{W}_j/\vec{Y}_j] = V_j'$ by rule $\rightarrow_E$. Then by the induction hypothesis $\Gamma \vdash t[b/x] : (\sum_{i=1}^{n'} \alpha_i \cdot \forall \vec{Y}. (V \rightarrow T_i))[\vec{W}/\vec{X}]$ which is equal to $\sum_{i=1}^{n'} \alpha_i \cdot \forall \vec{Y}. (V[\vec{W}/\vec{X}] \rightarrow T_i[\vec{W}/\vec{X}])$. Also we can derive $\Gamma \vdash r[b/x] : (\sum_{j=1}^{n''} \beta_j \cdot V_j')[\vec{W}/\vec{X}] = \sum_{j=1}^{n''} \beta_j \cdot V_j'[\vec{W}/\vec{X}]$. Notice also that, up to variable renaming, $(\forall \vec{Y}. (V[\vec{W}/\vec{X}]))[\vec{W}'/\vec{Y}]$ is equal to $(\forall \vec{Y}. (V[\vec{W}'/\vec{X}]])[\vec{W}/\vec{X}] = V_j'[\vec{W}/\vec{X}]$. Then by rule $\rightarrow_E$, $\Gamma \vdash (t[b/x]) r[b/x] : \sum_{i=1}^{n'} \sum_{j=1}^{n''} \alpha_i \cdot \beta_j \cdot (T_i[\vec{W}/\vec{X}])[\vec{W}_j'/\vec{Y}_j]'$. Notice that $(t[b/x]) r[b/x]$ is equal to $(t \ r)[b/x]$ and, up to variable renaming, $\sum_{i=1}^{n'} \sum_{j=1}^{n''} \alpha_i \cdot \beta_j \cdot (T_i[\vec{W}/\vec{X}])[\vec{W}_j'/\vec{Y}_j]'$ is equal to $\sum_{i=1}^{n'} \sum_{j=1}^{n''} \alpha_i \cdot \beta_j \cdot T_i[\vec{W}_j'/\vec{Y}_j']$.

– Let $\Gamma, x : U \vdash (t \ \lambda \ y : V \rightarrow T)$ as a consequence of $\Gamma, x : U, y : V : t : T$ and rule $\rightarrow_I$. Then by item 1, $\Gamma[\vec{W}/\vec{X}], x : U[\vec{W}/\vec{X}], y : V[\vec{W}/\vec{X}] \vdash t : T[\vec{W}/\vec{X}]$. Since $\vec{X} \notin FV(\Gamma)$, we do not need to replace anything on $\Gamma$ and then this sequent is the same to $\Gamma, x : U[\vec{W}/\vec{X}], y : V[\vec{W}/\vec{X}] \vdash t : T[\vec{W}/\vec{X}]$. Notice that $y \notin \Gamma$, so $\Gamma, y : V[\vec{W}/\vec{X}] \vdash b : U[\vec{W}/\vec{X}]$. 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 $\Gamma, y, y : V[\vec{W}/\vec{X}] \vdash (t[b/x]) \ r[b/x] : (T[\vec{W}/\vec{X}])[\vec{W}[\vec{W}/\vec{X}] = T[\vec{W}/\vec{X}]$. So by rule $\rightarrow_I$, $\Gamma \vdash \lambda \ y : V[\vec{W}/\vec{X}] \rightarrow (T[\vec{W}/\vec{X}])[(\forall Y. V_i)[\vec{W}/\vec{X}]] = (T[\vec{W}/\vec{X}])[(\forall Y. V_i)[\vec{W}/\vec{X}]]$. Notice that, up to renaming of variables, the type $(\forall Y. V_i)[\vec{W}/\vec{X}]]$ is equal to $(\forall Y. V_i)[\vec{W}/\vec{X}]]$.

– Let $\Gamma, x : U \vdash t : \sum_{i=1}^{n'} \alpha_i \cdot V_i[\vec{W}'/\vec{Y}]$ as a consequence of the rule $\forall_E$ and the sequent $\Gamma, x : U \vdash t : \sum_{i=1}^{n'} \alpha_i \cdot \forall Y. V_i$. Then by the induction hypothesis $\Gamma \vdash (t[b/x]) \ r[b/x] : (\sum_{i=1}^{n'} \alpha_i \cdot \forall Y. V_i)[\vec{W}/\vec{X}] = \sum_{i=1}^{n'} \alpha_i \cdot \forall Y. V_i[\vec{W}/\vec{X}]$. So by rule $\forall_E$, $\Gamma \vdash (t[b/x]) \ r[b/x] : (\sum_{i=1}^{n'} \alpha_i \cdot V_i[\vec{W}/\vec{X}])[\vec{W}'/\vec{Y}]$. Notice that, up to renaming of variables, the type $(\sum_{i=1}^{n'} \alpha_i \cdot V_i[\vec{W}/\vec{X}])[\vec{W}'/\vec{Y}]$ is equal to $(\sum_{i=1}^{n'} \alpha_i \cdot V_i[\vec{W}/\vec{X}])[\vec{W}'/\vec{Y}]$.

– Let $\Gamma, x : U \vdash t : \sum_{i=1}^{n'} \alpha_i \cdot \forall Y. V_i$ as a consequence of the rule $\forall_I$ and the sequent $\Gamma, x : U \vdash t : \sum_{i=1}^{n'} \alpha_i \cdot V_i$. Then by the induction hypothesis we have $\Gamma \vdash (t[b/x]) \ r[b/x] : \sum_{i=1}^{n'} \alpha_i \cdot V_i[\vec{W}/\vec{X}]$. So by rule $\forall_I$, we can conclude $\Gamma \vdash (t[b/x]) \ r[b/x] : \sum_{i=1}^{n'} \alpha_i \cdot \forall Y. V_i[\vec{W}/\vec{X}]$. Notice that $\sum_{i=1}^{n'} \alpha_i \cdot \forall Y. V_i[\vec{W}/\vec{X}] = (\sum_{i=1}^{n'} \alpha_i \cdot \forall Y. V_i)[\vec{W}/\vec{X}]$.

– Let $\Gamma, x : U \vdash t + r : T + R$ as a consequence of $\Gamma, x : U \vdash t : T$ and $\Gamma, x : U \vdash r : R$ by rule $+t$. Then by the induction hypothesis $\Gamma \vdash t[b/x] : T[\vec{W}/\vec{X}]$ and $\Gamma \vdash r[b/x] : R[\vec{W}/\vec{X}]$. So by rule $+t$, $\Gamma \vdash (t[b/x] + r[b/x] : T[\vec{W}/\vec{X}] + R[\vec{W}/\vec{X}]$. Notice that $t[b/x] + r[b/x] = (t + r)[b/x]$ and $T[\vec{W}/\vec{X}] + R[\vec{W}/\vec{X}] = (T + R)[\vec{W}/\vec{X}]$. 
Lemma 4.8 (Generation lemma (application)) For any terms t, r, any context Γ and any type T, if Γ ⊢ (t) r : T, then there exists natural numbers n, m, unit types U, V, . . . , Vm, types T1, . . . , Tn and scalars α1, . . . , αn, β1, . . . , βm, such that Γ ⊢ t : \sum_{i=1}^{n} αi . ∀X. (U → Ti), Γ ⊢ r : \sum_{j=1}^{m} βj . Vj, for all Vj, there exists \( \bar{W} \) such that \( U[\bar{W}_j/\bar{X}] = V_j \) and \( \sum_{i=1}^{n} \sum_{j=1}^{m} α_i × β_j . T_i[\bar{W}_j/\bar{X}] ≥ T \).

Proof. Induction on the typing derivation.

- Let Γ ⊢ (t) r : \( \sum_{i=1}^{n} \sum_{j=1}^{m} α_i × β_j . T_i[\bar{W}_j/\bar{X}] \) as a consequence of \( Γ ⊢ t : \sum_{i=1}^{n} α_i . ∀X. U_i \) and \( Γ ⊢ r : \sum_{j=1}^{m} β_j . V_j \), where \( ∀V_j, \exists\bar{W}_j \) such that \( U[\bar{W}_j/\bar{X}] = V_j \) and \( \sum_{i=1}^{n} \sum_{j=1}^{m} α_i × β_j . T_i[\bar{W}_j/\bar{X}] ≥ T \).
- Let Γ ⊢ (t) r : S as a consequence of \( Γ ⊢ t : \sum_{i=1}^{n} α_i . ∀X. (U → Ti), Γ ⊢ r : \sum_{j=1}^{m} β_j . V_j \), for all Vj, there exists \( \bar{W} \) such that \( U[\bar{W}/\bar{X}] = V_j \) and \( \sum_{i=1}^{n} \sum_{j=1}^{m} α_i × β_j . T_i[\bar{W}/\bar{X}] ≥ T \).
- Let Γ ⊢ (t) r : \( \sum_{i=1}^{n} α_i . U_i[V/X] \) as a consequence of rule \( \forall_E \) and the sequent \( Γ ⊢ (t) r : \sum_{i=1}^{n} α_i . ∀X. U_i \). Then by the induction hypothesis Γ ⊢ t : \( \sum_{i=1}^{n} α_i . ∀X. (U → Ti) \), Γ ⊢ r : \( \sum_{j=1}^{m} β_j . V_j, ∀V_j, \exists\bar{W}_j / U[\bar{W}_j/\bar{X}] = V_j \) and \( \sum_{i=1}^{n} \sum_{j=1}^{m} α_i × β_j . T_i[\bar{W}_j/\bar{X}] ≥ \sum_{i=1}^{n} α_i . ∀X. U_i \) and \( \sum_{i=1}^{n} \sum_{j=1}^{m} α_i × β_j . T_i[\bar{W}_j/\bar{X}] ≥ T_i[\bar{W}_j/\bar{X}] ≥ \sum_{i=1}^{n} α_i . ∀X. U_i \).

Lemma 4.9 (Generation lemma (abstraction)) For any term variable x, term t, context Γ and type T, if Γ ⊢ \( \lambda x. t : R \), then there exists types U and T such that \( U → T ≤ R \) and \( Γ, x : U ⊢ t : T \).

Proof. Induction on the typing derivation.

- Let Γ ⊢ \( \lambda x. t : U → T \) as a consequence of \( Γ, x : U ⊢ t : T \) and rule \( →_I \). This is the trivial case.
- Let Γ ⊢ \( \lambda x. t : R \) as a consequence of \( Γ ⊢ \lambda x. t : S \) \( S ≤ R \) and rule \( ≡ \).
- Then by the induction hypothesis \( U → T ≤ S ≤ R \) and \( Γ, x : U ⊢ t : T \).
- Let Γ ⊢ \( \lambda x. t : \sum_{i=1}^{n} α_i . U_i[V/X] \) as a consequence of rule \( ∀E \) with the sequent \( Γ ⊢ \lambda x. t : \sum_{i=1}^{n} α_i . ∀X. U_i \). Then by the induction hypothesis \( W → T ≤ \sum_{i=1}^{n} α_i . ∀X. U_i \) and \( Γ, x : W ⊢ t : T \).
Lemma 4.12 (Generation lemma (sum)) For any context \( \Gamma \), terms \( t \) and \( r \) and type \( S \), if \( \Gamma \vdash t + r : S \), then there exists types \( T \) and \( R \) such that \( \Gamma \vdash t : T \), \( \Gamma \vdash r : R \) and \( T + R \leq S \).

Proof. Induction on the typing derivation.

- Let \( \Gamma \vdash t + r : T + R \) as a consequence of \( \Gamma \vdash t : T \), \( \Gamma \vdash r : R \) and rule \( +_1 \).
  - Trivial case.
- Let \( \Gamma \vdash t + r : S \) as a consequence of \( \Gamma \vdash t + r : S' \) where \( S' \equiv S \), and rule \( \equiv \).
  - Then by the induction hypothesis \( \Gamma \vdash t : T \), \( \Gamma \vdash r : R \) and \( T + R \leq S' \leq S \).
- Let \( \Gamma \vdash t + r : \sum_{i=1}^{n} \alpha_i \cdot U_i[V/X] \) as a consequence of \( \Gamma \vdash t + r : \sum_{i=1}^{n} \alpha_i \cdot U_i \) and rule \( \forall_E \).
  - Then by the induction hypothesis \( \Gamma \vdash t : T \), \( \Gamma \vdash r : R \) and \( T + R \leq \sum_{i=1}^{n} \alpha_i \cdot U_i[V/X] \).
- Let \( \Gamma \vdash t + r : \sum_{i=1}^{n} \alpha_i \cdot U_i \) as a consequence of \( \Gamma \vdash t + r : \sum_{i=1}^{n} \alpha_i \cdot U_i \) and rule \( \forall_f \).
  - Then by the induction hypothesis \( \Gamma \vdash t : T \), \( \Gamma \vdash r : R \) and \( T + R \leq \sum_{i=1}^{n} \alpha_i \cdot U_i \).

Lemma 4.12 (Generation lemma (scalar)) For any context \( \Gamma \), term \( t \), scalar \( \alpha \) and type \( T \), if \( \Gamma \vdash \alpha \cdot t : T \), then there exists a type \( R \) such that \( \alpha \cdot R \leq T \) and \( \Gamma \vdash \alpha \cdot t : \alpha \cdot R \).

Proof. Induction on the typing derivation.

- Let \( \Gamma \vdash \alpha \cdot t : \alpha \cdot T \) as a consequence of \( \Gamma \vdash t : T \) and rule \( \alpha_f \).
  - Trivial case.
- Let \( \Gamma \vdash \alpha \cdot t : T \) as a consequence of \( \Gamma \vdash \alpha \cdot t : S \) where \( S \equiv T \), and rule \( \equiv \).
  - Then by the induction hypothesis \( \Gamma \vdash \alpha \cdot t : \alpha \cdot R \) with \( \alpha \cdot R \leq S \equiv T \).
- Let \( \Gamma \vdash \alpha \cdot t : \sum_{i=1}^{n} \alpha_i \cdot U_i[V/X] \) as a consequence of \( \Gamma \vdash \alpha \cdot t : \sum_{i=1}^{n} \alpha_i \cdot U_i \) and rule \( \forall_E \).
  - Then by the induction hypothesis \( \Gamma \vdash \alpha \cdot t : \alpha \cdot R \) with \( \alpha \cdot R \leq \sum_{i=1}^{n} \alpha_i \cdot U_i[V/X] \).
- Let \( \Gamma \vdash \alpha \cdot t : \sum_{i=1}^{n} \alpha_i \cdot U_i \) as a consequence of \( \Gamma \vdash \alpha \cdot t : \sum_{i=1}^{n} \alpha_i \cdot U_i \) and rule \( \forall_f \).
  - Then by the induction hypothesis \( \Gamma \vdash \alpha \cdot t : \alpha \cdot R \) with \( \alpha \cdot R \leq \sum_{i=1}^{n} \alpha_i \cdot U_i \).

Lemma 4.12 (Generation lemma (zero)) For any context \( \Gamma \) and type \( T \), if \( \Gamma \vdash 0 : T \), then there exists a type \( R \) such that \( T \equiv 0 \cdot R \).

Proof. Induction on the typing derivation.

1. Let \( \Gamma \vdash 0 : 0 \cdot T \) as a consequence of \( \Gamma \vdash t : T \) and rule \( 0_f \). Trivial case.
2. Let \( \Gamma \vdash 0 : T \) as a consequence of \( \Gamma \vdash 0 : S \) where \( S \equiv T \), and rule \( \equiv \).
  - Then by the induction hypothesis \( 0 \cdot R \equiv S \equiv T \).
3. Let \( \Gamma \vdash 0 : \sum_{i=1}^{n} \alpha_{i} \cdot U_{i}[V/X] \) as a consequence of \( \Gamma \vdash 0 : \sum_{i=1}^{n} \alpha_{i} \cdot \forall X.U_{i} \) and rule \( \forall F \). Then by the induction hypothesis \( \sum_{i=1}^{n} \alpha_{i} \cdot \forall X.U_{i} \equiv 0 \cdot R \). By lemma 3.1, \( R \equiv \sum_{j=1}^{m} \beta_{j} \cdot V_{j} \) and so \( 0 \cdot R \equiv \sum_{j=1}^{m} 0 \cdot V_{j} \). Thus by lemma 3.2[2], \( \forall V_{j}, \exists W_{j} / V_{j} \equiv \forall X.W_{j} \). Then by lemma 3.2[1], \( \sum_{i=1}^{n} \alpha_{i} \cdot U_{i} \equiv \sum_{j=1}^{m} 0 \cdot W_{j} \)
and by lemma 3.2[8], \( \sum_{i=1}^{n} \alpha_{i} \cdot U_{i}[V/X] = (\sum_{i=1}^{m} \alpha_{i} \cdot U_{i})[V/X] \) which is equivalent to \( (\sum_{j=1}^{m} 0 \cdot W_{j})[V/X] = 0 \cdot \sum_{j=1}^{m} W_{j}[V/X] \).

4. Let \( \Gamma \vdash 0 : \sum_{i=1}^{n} \alpha_{i} \cdot \forall X.U_{i} \) as a consequence of \( \Gamma \vdash 0 : \sum_{i=1}^{n} \alpha_{i} \cdot U_{i} \) and rule \( \forall F \). Then by the induction hypothesis \( \sum_{i=1}^{n} \alpha_{i} \cdot U_{i} \equiv 0 \cdot R \). By lemma 3.1 \( R \equiv \sum_{j=1}^{m} \beta_{j} \cdot V_{j} \) so \( 0 \cdot R \equiv \sum_{j=1}^{m} 0 \cdot V_{j} \). Then by lemma 3.2[1], \( \sum_{i=1}^{n} \alpha_{i} \cdot \forall X.U_{i} \equiv 0 \cdot \sum_{j=1}^{m} \forall X.W_{j} \).

\[\Box\]

**Lemma 4.13 (Order characterization)** For any type \( R \), unit types \( V_{1}, \ldots, V_{m} \) and scalars \( \beta_{1}, \ldots, \beta_{m} \), if \( R \subseteq \sum_{j=1}^{m} \beta_{j} \cdot V_{j} \), then there exists a scalar \( \delta \), a natural number \( k \), a set \( N \subseteq \{1, \ldots, m\} \) and a unit type \( W \leq V_{k} \) such that \( R \equiv \delta \cdot W + \sum_{j \in N} \beta_{j} \cdot V_{j} \) and \( \sum_{j=1}^{m} \beta_{j} = \delta + \sum_{j \in N} \beta_{j} \).

**Proof.** Structural induction on \( R \).

- \( R = U \). Then by definition of \( \subseteq \), \( \exists k / U \leq V_{k} \) and \( \sum_{j=1}^{m} \beta_{j} = 1 \).
- \( R = \alpha \cdot T \). Then \( \sum_{j=1}^{m} \beta_{j} \cdot V_{j} \equiv \alpha \cdot \sum_{j=1}^{m} \gamma_{j} \cdot V_{j} \), so by the induction hypothesis \( T \equiv \delta \cdot W + \sum_{j \in N} \gamma_{j} \cdot V_{j} \) with \( N \subseteq \{1, \ldots, m\} \) and \( W \leq V_{k} \) for some \( k \). So \( R \equiv \alpha \cdot T \equiv \alpha \cdot \delta \cdot W + \alpha \cdot \sum_{j \in N} \gamma_{j} \cdot V_{j} \equiv (\alpha \cdot \delta) \cdot W + \sum_{j \in N} \beta_{j} \cdot V_{j} \).
- \( R = T + S \). Then \( \exists m' \leq m \) such that \( \sum_{j=1}^{m} \alpha_{j} \cdot V_{j} = \sum_{j=1}^{m'} \beta_{j} \cdot V_{j} + \sum_{j=m'+1}^{m} \beta_{j} \cdot V_{j} + S \) with \( T \equiv \sum_{j=1}^{m'} \beta_{j} \cdot V_{j} \), so by the induction hypothesis \( T \equiv \delta \cdot W + \sum_{j \in N} \beta_{j} \cdot V_{j} \) with \( N \subseteq \{1, \ldots, m'\} \) and \( W \leq V_{k} \) for some \( k \). So \( R \equiv T + S \equiv \delta \cdot W + \sum_{j \in N} \beta_{j} \cdot V_{j} + S \equiv \delta \cdot W + \sum_{j \in N} \beta_{j} \cdot V_{j} + S \equiv \delta \cdot W + \sum_{j \in N} \beta_{j} \cdot V_{j} \) with \( N' = N \cup \{m'+1, \ldots, m\} \). Notice that \( \sum_{j=1}^{m} \beta_{j} = \sum_{j=1}^{m} \beta_{j} + \sum_{j=m'+1}^{m} \beta_{j} = \sum_{j=m'+1}^{m} \beta_{j} + \sum_{j=1}^{m} \beta_{j} = \delta + \sum_{j \in N} \beta_{j} \).

**B Full proof of theorem 4.2**

**Theorem 4.2 (Weak subject reduction)** For any terms \( t, t' \), any context \( \Gamma \) and any type \( T \), if \( t \rightarrow_{R} t' \) and \( \Gamma \vdash t : T \), then

- If \( R \notin \text{Group F} \), then \( \Gamma \vdash t' : T \).
- If \( R \in \text{Group F} \) or the rule \( t + 0 \rightarrow t \), then \( \exists S \subseteq T \) such that \( \Gamma \vdash t' : S \) and \( \Gamma \vdash t : S \).

**Proof.** Let \( t \rightarrow_{R} t' \) and \( \Gamma \vdash t : T \). We proceed by induction. We treat separately every rule \( R \).

**Basic cases.**

**Group E**
rule $0 \cdot t \to 0$. Let $\Gamma \vdash 0 \cdot t : T$. Then by lemma 4.12, $\exists R / 0 \cdot R \leq T$ and $\Gamma \vdash 0 : t : 0 \cdot R$, then by rule $0_1$, $\Gamma \vdash 0 \cdot (0 \cdot R)$. Since $0 \cdot (0 \cdot R) \equiv 0 \cdot R \leq T$, by lemma 4.1, $\Gamma \vdash 0 : T$.

rule $1 \cdot t \to t$. Let $\Gamma \vdash 1 \cdot t : T$, then by lemma 4.12, $\exists R / 1 \cdot R \leq T$ and $\Gamma \vdash 1 \cdot t : 1 \cdot R$. Then by corollary 4.4, $\Gamma \vdash t : R$ and by $\equiv$-rule, $\Gamma \vdash t : 1 \cdot R$, so by lemma 4.1 $\Gamma \vdash t : T$.

rule $\alpha \cdot 0 \to 0$. Let $\Gamma \vdash \alpha \cdot 0 : T$, then by lemma 4.12, $\exists R / \alpha \cdot R \leq T$ and $\Gamma \vdash \alpha \cdot 0 : T$. Cases: If $\alpha \neq 0$, then by corollary 4.4, $\Gamma \vdash 0 : R$, and by lemma 4.12, $\exists S / R \equiv 0 \cdot S$. Notice that $0 \cdot 0 = \alpha \times 0 \cdot S \equiv \alpha \cdot (0 \cdot S) \equiv \alpha \cdot R \leq T$, so by lemma 4.1, $\Gamma \vdash 0 : T$. If $\alpha = 0$, then by rule $0_1$, $\Gamma \vdash 0 \cdot 0 : (0 \cdot R)$, and notice that $0 \cdot (0 \cdot R) \equiv 0 \cdot R$. Then by lemma 4.1, $\Gamma \vdash 0 : T$.

rule $\alpha \cdot (\beta \cdot t) \to (\alpha \times \beta) \cdot t$. Let $\Gamma \vdash \alpha \cdot (\beta \cdot t) : T$. Then by lemma 4.12, $\exists R$ such that $\alpha \cdot R \leq T$ and $\Gamma \vdash \alpha \cdot (\beta \cdot t) : \alpha \cdot R$. Cases: If $\alpha \neq 0$, then by corollary 4.4, $\Gamma \vdash \beta \cdot t : R$. Then by lemma 4.12 again, $\exists S / S \leq R$ and $\Gamma \vdash \beta \cdot t : \beta \cdot S$. If $\beta = 0$, then $(\alpha \times \beta) \cdot t = \beta \cdot t$ and $0 \cdot S \equiv (\alpha \cdot 0) \cdot S \equiv \alpha \cdot (0 \cdot S) \leq \alpha \cdot R \leq T$ so by lemma 4.1, $\Gamma \vdash \beta \cdot t : T$. If $\beta \neq 0$, then by corollary 4.4, $\Gamma \vdash t : S$, then by rule $\alpha_1$, $\Gamma \vdash (\alpha \times \beta) \cdot t : (\alpha \times \beta) \cdot S$. Note that $(\alpha \times \beta) \cdot S \equiv (\alpha \cdot S) \leq \alpha \cdot R \leq T$, so by lemma 4.1, $\Gamma \vdash (\alpha \times \beta) \cdot t : T$. If $\alpha = 0$, first we prove that $\Gamma \vdash \beta \cdot t : T \Rightarrow \Gamma \vdash 0 \cdot t : T$. We proceed by induction on the typing derivation.

Let $\Gamma \vdash 0 \cdot \beta \cdot t : 0 \cdot T$ as a consequence of $\Gamma \vdash \beta \cdot t : T$ and rule $\alpha_1$.

Then by lemma 4.12, there exists a type $R$ such that $\beta \cdot R \leq T$ and $\Gamma \vdash \beta \cdot t : \beta \cdot R$. Cases: If $\beta \neq 0$, then by corollary 4.4 $\Gamma \vdash t : R$ so by rule $\alpha_1$ we get $\Gamma \vdash 0 \cdot t : 0 \cdot R$. Notice that $0 \cdot R = (0 \times \beta) \cdot R \equiv 0 \cdot \beta \cdot R \leq 0 \cdot T$, so by lemma 4.1, $\Gamma \vdash 0 \cdot t : 0 \cdot R$. If $\beta = 0$, then $\Gamma \vdash 0 : t : 0 \cdot R$. Notice that $0 \cdot R = (0 \times 0) \cdot R \equiv 0 \cdot (0 \cdot R) \leq 0 \cdot T$, so by lemma 4.1, $\Gamma \vdash 0 \cdot t : 0 \cdot T$.

Let $\Gamma \vdash 0 \cdot \beta \cdot t : T$ as a consequence of $\Gamma \vdash 0 \cdot \beta \cdot t : R$ where $R \equiv T$ and rule $\equiv$. Then by the induction hypothesis $\Gamma \vdash 0 \cdot t : R$, so by rule $\equiv$, $\Gamma \vdash 0 \cdot t : T$.

Let $\Gamma \vdash 0 \cdot \beta \cdot t : \sum_{i=1}^{n} \alpha_i \cdot U_i [V / X]$ as a consequence of rule $\forall E$ and the sequent $\Gamma \vdash 0 \cdot \beta \cdot t : \sum_{i=1}^{n} \alpha_i \cdot \forall X . U_i$. Then by the induction hypothesis $\Gamma \vdash 0 \cdot t : \sum_{i=1}^{n} \alpha_i \cdot \forall X . U_i$. So by rule $\forall E$, $\Gamma \vdash 0 \cdot t : \sum_{i=1}^{n} \alpha_i \cdot U_i [V / X]$.

Let $\Gamma \vdash 0 \cdot \beta \cdot t : \sum_{i=1}^{n} \alpha_i \cdot \forall X . U_i$ as a consequence of rule $\forall I$ and the sequent $\Gamma \vdash 0 \cdot \beta \cdot t : \sum_{i=1}^{n} \alpha_i \cdot U_i$. Then by the induction hypothesis $\Gamma \vdash 0 \cdot t : U_i$. So by rule $\forall I$, $\Gamma \vdash 0 \cdot t : \sum_{i=1}^{n} \alpha_i \cdot \forall X . U_i$. With this result we can deduce that $\Gamma \vdash 0 \cdot t : 0 \cdot R$. Notice that $0 \cdot t = (0 \times \beta) \cdot t$, so by lemma 4.1, $\Gamma \vdash (0 \times \beta) \cdot t : T$.

rule $\alpha \cdot (t + r) \to \alpha \cdot t + \alpha \cdot r$. Let $\Gamma \vdash \alpha \cdot (t + r) : T$. Then by lemma 4.12, $\exists R$ such that $\alpha \cdot R \leq T$ and $\Gamma \vdash \alpha \cdot (t + r) : \alpha \cdot R$. Cases: If $\alpha \neq 0$, then by corollary 4.4, $\Gamma \vdash t + r : R$. So by lemma 4.12, $\exists S_1 , S_2$ such that $\Gamma \vdash t : S_1$, $\Gamma \vdash r : S_2$ and $S_1 + S_2 \leq R$. Then by rule $\alpha_1$, $\Gamma \vdash \alpha \cdot t : \alpha \cdot S_1$ and $\Gamma \vdash \alpha \cdot r : \alpha \cdot S_2$, and so by rule $+_I$, $\Gamma \vdash \alpha \cdot t + \alpha \cdot r : \alpha \cdot S_1 + \alpha \cdot S_2$. Notice that $\alpha \cdot S_1 + \alpha \cdot S_2 \equiv \alpha \cdot (S_1 + S_2) \leq \alpha \cdot R \leq T$, so by lemma 4.1, $\Gamma \vdash \alpha \cdot t + \alpha \cdot r : T$. 


If $\alpha = 0$, then $\Gamma \vdash 0 \cdot (t + r) : 0 \cdot R$. We show by induction the most general case: $\Gamma \vdash 0 \cdot (t + r)$ implies $S \Rightarrow \Gamma \vdash 0 \cdot t + 0 \cdot r : S$. Then since $0 \cdot R \subseteq T$, by lemma 4.1, $\Gamma \vdash 0 \cdot t + 0 \cdot r : T$.

- Let $\Gamma \vdash 0 \cdot (t + r) : 0 \cdot R$ as a consequence of $\Gamma \vdash t + r : R$ and rule $\alpha_I$. Then by lemma 4.12, $\exists T, S$ such that $\Gamma \vdash t : T$, $\Gamma \vdash r : S$ and $T + S \leq R$. So by rule $\alpha_I$, $\Gamma \vdash \text{0} \cdot t : 0 \cdot T$ and $\Gamma \vdash \text{0} \cdot r : 0 \cdot S$ and by rule $+T$, $\Gamma \vdash \text{0} \cdot t + \text{0} \cdot r : 0 \cdot T + 0 \cdot S$. Notice that $0 \cdot T + 0 \cdot S \equiv 0 \cdot (T + S) \leq 0 \cdot R$, so by lemma 4.1, $\Gamma \vdash \text{0} \cdot t + \text{0} \cdot r : 0 \cdot R$.

- Let $\Gamma \vdash 0 \cdot (t + r) : S$ as a consequence of $\Gamma \vdash 0 \cdot (t + r) : R \equiv S$ and rule $\equiv$. Then by the induction hypothesis $\Gamma \vdash 0 \cdot t + 0 \cdot r : R$. So by rule $\equiv$, $\Gamma \vdash 0 \cdot t + 0 \cdot r : S$.

- Let $\Gamma \vdash 0 \cdot (t + r) : \sum_{i=1}^{n} \alpha_i \cdot U_i[V/X]$ as a consequence of rule $\forall_E$ and the sequent $\Gamma \vdash 0 \cdot (t + r) : \sum_{i=1}^{n} \alpha_i \cdot \forall X. U_i$. Then by the induction hypothesis $\Gamma \vdash 0 \cdot t + 0 \cdot r : \sum_{i=1}^{n} \alpha_i \cdot \forall X. U_i$. So by rule $\forall_E$, $\Gamma \vdash 0 \cdot 0 \cdot t + 0 \cdot r = 0 \cdot 0 \cdot t + 0 \cdot r : \sum_{i=1}^{n} \alpha_i \cdot \forall X. U_i$. Then by the induction hypothesis $\Gamma \vdash 0 \cdot t + 0 \cdot r \subseteq T$, $\sum_{i=1}^{n} \alpha_i \cdot \forall X. U_i$. So by rule $\forall_I$, $\Gamma \vdash 0 \cdot t + 0 \cdot r : \sum_{i=1}^{n} \alpha_i \cdot \forall X. U_i$.

**rule $t + 0 \Rightarrow t$.** Let $\Gamma \vdash t + 0 : T$. Then by lemma 4.12, $\exists R, S$ such that $\Gamma \vdash t : R$, $\Gamma \vdash 0 : S$ and $R + S \leq T$. So, by lemma 4.12, $\exists S' / S \equiv 0 \cdot S'$. Notice that $R \subseteq R + 0 \cdot S' \equiv R + S \subseteq T$.

**Group F**

**rule $\alpha \cdot t + \beta \cdot t \Rightarrow (\alpha + \beta) \cdot t$.** Let $\Gamma \vdash \alpha \cdot t + \beta \cdot t : T$. Then by lemma 4.12, $\exists R, S$ such that $\Gamma \vdash \alpha \cdot t : R$, $\Gamma \vdash \beta \cdot t : S$ and $R + S \leq T$. Then by lemma 4.12, $\exists R' / \alpha \cdot R' \leq R$ and $\Gamma \vdash \alpha \cdot t : \alpha \cdot R'$, also $\exists S' / S' \leq S$ and $\Gamma \vdash \beta \cdot t : \beta \cdot S'$. Cases: If $\alpha \neq 0$ (or analogously $\beta \neq 0$), then by corollary 4.4, $\Gamma \vdash t : R'$ and so by $\alpha_I$ we conclude $\Gamma \vdash (\alpha + \beta) \cdot t : (\alpha + \beta) \cdot R'$. Notice that $(\alpha + \beta) \cdot R' \subseteq \alpha \cdot R' + \beta \cdot S' \subseteq R + S \subseteq T$. Also using rules $+I$ and $\equiv$ we conclude $\Gamma \vdash \alpha \cdot t + \beta \cdot t : (\alpha + \beta) \cdot R'$. If $\alpha = \beta = 0$, then notice that $\Gamma \vdash 0 \cdot t : 0 \cdot R'$ and $0 \cdot R' \subseteq 0 \cdot R' + 0 \cdot S' \subseteq R + S \subseteq T$. Also, using rules $+I$ and $\equiv$, we conclude $\Gamma \vdash t : 0 \cdot R'$.

**rules** $\alpha \cdot t + t \Rightarrow (\alpha + 1) \cdot t$ and $t + t \Rightarrow (1 + 1) \cdot t$. This cases are analogous to the previous case.

**Group A**

**rule $t \cdot t \Rightarrow (t) \cdot t + (r) \cdot t$.** Let $\Gamma \vdash (t + r) : u : T$. Then by lemma 4.8, $\exists n, m, U, T_1, \ldots, T_n, \alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_m, V_1, \ldots, V_m$ such that $\Gamma \vdash t + r : \sum_{i=1}^{n} \alpha_i \cdot \forall X. (U \rightarrow T_i)$, $\Gamma \vdash u : \sum_{j=1}^{m} \beta_j \cdot V_j, \forall V_j$, $\exists W_j / U[W_j / \forall X.X] = V_j$ and $\sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \cdot \beta_j \cdot T_i[W_j / \forall X.X] \leq T$. Then by lemma 4.12, $\exists R, S$ such that $\Gamma \vdash t : R$, $\Gamma \vdash r : S$ and $R + S$ which is $\leq \sum_{i=1}^{n} \alpha_i \cdot \forall X. (U \rightarrow T_i)$. Then $\exists N_1, N_2 \subseteq \{1, \ldots, n\}$ such that $R \leq \sum_{i \in N_1 \setminus N_2} \alpha_i \cdot \forall X. (U \rightarrow T_i) + \sum_{i \in N_1 \cap N_2} \delta_i \cdot \forall X. (U \rightarrow T_i)$ and $S \leq \sum_{i \in N_2 \setminus N_1} \alpha_i \cdot \forall X. (U \rightarrow T_i) + \sum_{i \in N_1 \cap N_2} \delta_i \cdot \forall X. (U \rightarrow T_i)$ and $\Gamma \vdash r : \sum_{i \in N_1 \setminus N_2} \alpha_i \cdot \forall X. (U \rightarrow T_i) + \sum_{i \in N_1 \cap N_2} \delta_i \cdot \forall X. (U \rightarrow T_i)$. Then by lemma 4.1, $\Gamma \vdash t : \sum_{i \in N_1 \setminus N_2} \alpha_i \cdot \forall X. (U \rightarrow T_i) + \sum_{i \in N_1 \cap N_2} \delta_i \cdot \forall X. (U \rightarrow T_i)$ and $\Gamma \vdash t : \sum_{i \in N_1 \setminus N_2} \alpha_i \cdot \forall X. (U \rightarrow T_i) + \sum_{i \in N_1 \cap N_2} \delta_i \cdot \forall X. (U \rightarrow T_i)$. Then by lemma 4.1, $\Gamma \vdash t : \sum_{i \in N_1 \setminus N_2} \alpha_i \cdot \forall X. (U \rightarrow T_i) + \sum_{i \in N_1 \cap N_2} \delta_i \cdot \forall X. (U \rightarrow T_i)$ and $\Gamma \vdash t : \sum_{i \in N_1 \setminus N_2} \alpha_i \cdot \forall X. (U \rightarrow T_i) + \sum_{i \in N_1 \cap N_2} \delta_i \cdot \forall X. (U \rightarrow T_i)$ and $\Gamma \vdash t : \sum_{i \in N_1 \setminus N_2} \alpha_i \cdot \forall X. (U \rightarrow T_i) + \sum_{i \in N_1 \cap N_2} \delta_i \cdot \forall X. (U \rightarrow T_i)$.
Then by rule \( \rightarrow_E \), \( \Gamma \vdash (t \cdot u) \): \( \sum_{i \in N_1 \setminus N_2} \sum_{j=1}^{m_i} \alpha_i \times \beta_j \cdot T_i[\vec{W}_j / \vec{X}] + \sum_{i \in N_1 \cap N_2} \sum_{j=1}^{m_i} \delta_i \times \beta_j \cdot T_i[\vec{W}_j / \vec{X}] \), and analogously we can also derive that \( \Gamma \vdash \alpha \cdot \beta \cdot T_i[\vec{W}_j / \vec{X}] \). So using \( +_r \), \( \Gamma \vdash (t + (u \cdot r)) \cdot u \cdot u + (r \cdot u) \) : \( \sum_{i \in N_1 \cup N_2 \setminus N_1 \cap N_2} \sum_{j=1}^{m_i} \alpha_i \times \beta_j \cdot T_i[\vec{W}_j / \vec{X}] + \sum_{i \in N_1 \cap N_2} \sum_{j=1}^{m_i} \gamma_i \times \beta_j \cdot T_i[\vec{W}_j / \vec{X}] \). So using \( +_r \), \( \Gamma \vdash (t + (u \cdot r)) \cdot u + (r \cdot u) \cdot u = T \). Then by \( \alpha \cdot \beta \cdot T_i[\vec{W}_j / \vec{X}] \), we conclude \( T \).

**rule** \((u \cdot (t + r)) \rightarrow (u \cdot t + (u \cdot r)) \). Let \( \Gamma \vdash (u \cdot (t + r)) \cdot T \). Then by \( \alpha \cdot \beta \cdot T_i[\vec{W}_j / \vec{X}] \), we conclude \( T \).

**rule** \((\alpha \cdot t) \cdot r \rightarrow \alpha \cdot (t \cdot r) \). Let \( \Gamma \vdash (\alpha \cdot t) \cdot r \cdot T \). Then by \( \alpha \cdot \beta \cdot T_i[\vec{W}_j / \vec{X}] \), we conclude \( T \).

**rule** \((\alpha \cdot t) \cdot r \rightarrow \alpha \cdot (t \cdot r) \). Let \( \Gamma \vdash (\alpha \cdot t) \cdot r \cdot T \). Then by \( \alpha \cdot \beta \cdot T_i[\vec{W}_j / \vec{X}] \), we conclude \( T \).

**rule** \((\alpha \cdot t) \cdot r \rightarrow \alpha \cdot (t \cdot r) \). Let \( \Gamma \vdash (\alpha \cdot t) \cdot r \cdot T \). Then by \( \alpha \cdot \beta \cdot T_i[\vec{W}_j / \vec{X}] \), we conclude \( T \).
\[ t: \sum_{i=1}^{n} \alpha_i \forall \vec{X}. (U \rightarrow T_i) \text{ and } \Gamma \vdash \alpha \cdot r: \sum_{j=1}^{m} \beta_j \cdot V_j \text{ with } \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j \cdot T_i[\vec{W}_j/\vec{X}] \leq T \] 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 4.12, \( \exists R \vdash \alpha \cdot R \leq \sum_{j=1}^{m} \beta_j \cdot V_j \) and \( \Gamma \vdash \alpha \cdot r: \alpha \cdot R \), then by corollary 4.4, \( \Gamma \vdash r: R \). Notice also that \( R \leq \sum_{j=1}^{m} \delta_j \cdot V_j \) where \( \forall \delta_j, \delta_j = \beta_j \). Then by lemma 4.1, \( \Gamma \vdash r: \sum_{j=1}^{m} \delta_j \cdot V_j \), so using rule \( \rightarrow_E \), followed by \( \alpha_j \) we get \( \Gamma \vdash \alpha \cdot (t) \vdash \alpha \cdot \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \delta_j \cdot T_i[\vec{W}_j/\vec{X}] \). Notice that \( \alpha \cdot \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \delta_j \cdot T_i[\vec{W}_j/\vec{X}] \equiv \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \delta_j \cdot T_i[\vec{W}_j/\vec{X}] \leq T \). Then by rule \( \rightarrow_E \), \( \Gamma \vdash \alpha \cdot (t) \vdash T \). If \( \alpha = 0 \), then by lemma 4.5 we have \( \Gamma \vdash r: \sum_{j=1}^{m} \delta_j \cdot V_j \) and \( \forall \delta_j, \delta_j = \beta_j = 0 \), so using rule \( \rightarrow_E \) followed by \( \alpha_j \), we conclude \( \Gamma \vdash 0 \cdot (t) \vdash 0 \cdot \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \delta_j \cdot T_i[\vec{W}_j/\vec{X}] \). Notice that \( 0 \cdot \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \delta_j \cdot T_i[\vec{W}_j/\vec{X}] \equiv \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j \cdot T_i[\vec{W}_j/\vec{X}] \leq T \). Then by lemma 4.1, \( \Gamma \vdash \alpha \cdot (t) \vdash T \).

**rule (0) t \rightarrow 0.** Let \( \Gamma \vdash (0) t: T \). Then by lemma 4.8, \( \exists n, m, U, T_i, \alpha_i \) and \( \beta_j \) with \( i = 1 \ldots n \) and \( j = 1 \ldots m \) such that \( \Gamma \vdash 0: \sum_{i=1}^{n} \alpha_i \cdot \forall \vec{X}. (U \rightarrow T_i) \), \( \Gamma \vdash t: \sum_{i=1}^{n} \sum_{j=1}^{m} \beta_j \cdot V_j \forall \vec{X}, \exists \vec{W}_j / U[\vec{W}_j/\vec{X}] \equiv V_j \) and \( \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j \cdot T_i[\vec{W}_j/\vec{X}] \leq T \). By lemma 4.12, \( \exists R \vdash \sum_{i=1}^{n} \alpha_i \cdot \forall \vec{X}. (U \rightarrow T_i) \equiv 0 \cdot R \), then it is equivalent to \( \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \cdot \forall \vec{X}. (U \rightarrow T_i) \equiv 0 \cdot R \). Notice that \( 0 \cdot \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j \cdot T_i[\vec{W}_j/\vec{X}] \equiv \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j \cdot T_i[\vec{W}_j/\vec{X}] \leq T \), then by rule \( \rightarrow_E \), \( \Gamma \vdash 0: T \).

**rule (t) 0 \rightarrow 0.** Let \( \Gamma \vdash (t) 0: T \). Then by lemma 4.8, \( \exists n, m, U, T_i, \alpha_i \) and \( \beta_j \) with \( i = 1 \ldots n \) and \( j = 1 \ldots m \) such that \( \Gamma \vdash t: \sum_{i=1}^{n} \alpha_i \cdot \forall \vec{X}. (U \rightarrow T_i) \), \( \Gamma \vdash 0: \sum_{i=1}^{n} \sum_{j=1}^{m} \beta_j \cdot V_j \forall \vec{X}, \exists \vec{W}_j / U[\vec{W}_j/\vec{X}] \equiv V_j \) and \( \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j \cdot T_i[\vec{W}_j/\vec{X}] \leq T \). By lemma 4.12, \( \exists R \vdash \sum_{i=1}^{n} \sum_{j=1}^{m} \beta_j \cdot V_j \equiv 0 \cdot R \), so it is equivalent to \( \sum_{i=1}^{n} \sum_{j=1}^{m} \beta_j \cdot V_j \equiv 0 \cdot R \), i.e., \( \exists M_1, \ldots, M_m \subseteq \{1, \ldots, m\} \) disjoint sets, such that \( \forall \delta_i, \forall \delta_j \in M_k \), \( \delta_i \cdot \delta_j = 0 \). Then using rule \( \rightarrow_E \) we get \( \Gamma \vdash (t) 0: \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \equiv 0 \cdot T_i[\vec{W}_j/\vec{X}] \) and by rule \( 0t \), \( \Gamma \vdash 0: 0 \cdot \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times 0 \cdot T_i[\vec{W}_j/\vec{X}] \). Notice that \( 0 \cdot \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times 0 \cdot T_i[\vec{W}_j/\vec{X}] \equiv \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times 0 \cdot T_i[\vec{W}_j/\vec{X}] \leq T \), then by lemma 4.1, \( \Gamma \vdash 0: T \).

**Group B**

**rule (λx t) b \rightarrow t[b/x].** Let \( \Gamma \vdash (λx t) b: T \). Then by lemma 4.8, \( \exists n, m, T_i, \alpha_i, \beta_j \) and \( V_j \) with \( i = 1 \ldots n \) and \( j = 1 \ldots m \) such that the following sequents are valid: \( \Gamma \vdash λx.t: \sum_{i=1}^{n} \alpha_i \cdot \forall \vec{X}. (U \rightarrow T_i) \) and \( \Gamma \vdash b: \sum_{j=1}^{m} \beta_j \cdot V_j \). Then by lemma 4.6, \( \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \times \beta_j \cdot T_i[\vec{W}_j/\vec{X}] \leq T \), where \( \forall V_j, \vec{W}_j \) is such that \( U[\vec{W}_j/\vec{X}] \equiv V_j \) and \( \forall \vec{X}. (U \rightarrow T_i) \equiv \forall \vec{X}. (U \rightarrow T_i) \) with \( T_i = T_k \) for all \( i, k = 1, \ldots, n \) and \( \sum_{j=1}^{m} \beta_j \cdot V_j \equiv V_j \).
with $V_j = V_h$ for all $j, h = 1, \ldots, m$. So $\sum_{i=1}^n \alpha_i = 1$ and $\sum_{j=1}^m \beta_j = 1$. By
\begin{align*}
\equiv\text{-rule, } \Gamma \vdash \lambda x. t : \forall X. (U \rightarrow T_i) \land \Gamma \vdash b : V_i.
\end{align*}
Then by corollary 4.11, $\Gamma, x : U \vdash t : T_i$. Notice that $V_i \equiv U[\hat{W}]$. Then, by lemma 4.7[2],
\begin{align*}
\Gamma \vdash t[b/x] : T_i[\hat{W}].
\end{align*}
Notice that $T_i[\hat{W}] \equiv (1 \times 1) \cdot T_i[\hat{W}] = \sum_{i=1}^n \sum_{j=1}^m \alpha_i \times \beta_j \cdot T_i[\hat{W}]$
and since all the $T_i$ are equivalents between them, this type is equivalent
to $\sum_{i=1}^n \sum_{j=1}^m \alpha_i \times \beta_j \cdot T_i[\hat{W}] \leq T$. Then by lemma 4.1, $\Gamma \vdash t[b/x] : T$.

**AC equivalences**

**rule** $t + r = r + t$. Let $\Gamma \vdash t + r : T$. Then by lemma 4.12, $\exists R, S / R + S \leq T, \Gamma \vdash r : R$ and $\Gamma \vdash t : S$. So using $+I$, $\Gamma \vdash r + t : S + R$. Notice that $S + R \equiv R + S \leq T$, then by lemma 4.1, $\Gamma \vdash r + t : T$.

**rule** $(t + r) + u = u + (t + r)$. Let $\Gamma \vdash (t + r) + u : T$. Then by lemma 4.12, $\exists R, S$ such that $\Gamma \vdash t + r : R, \Gamma \vdash u : S$ and $R + S \leq T$. Then, by lemma 4.12 again, $\exists R', S'$ such that $\Gamma \vdash r' : R', \Gamma \vdash s' : S'$ and $R' + S' \leq S$. Then using $+I$ in the correct order, we get $\Gamma \vdash (r + u) : R' + (S' + S)$. Notice that $R' + (S' + S) \equiv (R' + S') + S \leq R + S \leq T$, then by lemma 4.1, $\Gamma \vdash (r + u) : T$.

**Inductive cases** (Context rules)

- Let $\alpha \cdot s \rightarrow \alpha \cdot t$ as a consequence of $s \rightarrow t$. Let $\Gamma \vdash \alpha \cdot s : T$, then by lemma 4.12, $\exists R / \alpha \cdot R \leq T$ and $\Gamma \vdash \alpha \cdot s : \alpha \cdot R$. Cases: If $\alpha \neq 0$, then by corollary 4.4, $\Gamma \vdash s : R$, so by the induction hypothesis $\Gamma \vdash t : R'$ with $R' \subseteq R$, then using $\alpha_I$, $\Gamma \vdash \alpha \cdot t : \alpha \cdot R'$. Notice that $\alpha \cdot R' \subseteq \alpha \cdot R \subseteq T$. If $\alpha = 0$, then notice that $T \equiv \sum_{i=1}^\delta \beta_i \cdot U_i$, by lemma 4.5, $\Gamma \vdash s : \sum_{i=1}^\delta \beta_i \cdot U_i$ and $\forall i, \beta_i = 0$. Then by the induction hypothesis $\Gamma \vdash t : R$ with $R \subseteq \sum_{i=1}^\delta \beta_i \cdot U_i$. So using $\alpha_I$, $\Gamma \vdash 0 : t : 0 : R$. Notice that $0 : R \subseteq 0 : \sum_{i=1}^\delta \beta_i \cdot U_i \equiv \sum_{i=1}^\delta 0 : U_i \equiv T$.

- Let $r + s \rightarrow r + t$ as a consequence of $s \rightarrow t$. Let $\Gamma \vdash r + s : T$, then by lemma 4.12, $\exists R, S$ such that $\Gamma \vdash r : R, \Gamma \vdash s : S$ and $R + S \leq T$. Then by the induction hypothesis $\Gamma \vdash t : S' \subseteq S$, so using $+I$ we can conclude $\Gamma \vdash r + t : R + S'$. Notice that $R + S' \subseteq R + S \leq T$.

- Let $(r) \rightarrow (s) \rightarrow t$ as a consequence of $s \rightarrow t$. Let $\Gamma \vdash (r) \rightarrow (s) : T$. Then by lemma 4.8, $\exists n, a, n, a > 1, U, T_1, \ldots, T_n, a_1, \ldots, a_n, b_1, \ldots, b_m, V_1, \ldots, V_m$ such that $\Gamma \vdash t : \sum_{i=1}^a a_i \cdot \forall X. (U \rightarrow T_i)$ and $\Gamma \vdash s : \sum_{i=1}^m \beta_i \cdot V_j$ where for all $j$, there exists $W_j$ such that $U[\hat{W}] \equiv V_j$ and $\sum_{i=1}^n \sum_{j=1}^m \alpha_i \times \beta_j \cdot T_i[\hat{W}] \leq T$. Cases: If $s \rightarrow R$ then by the induction hypothesis $\Gamma \vdash t : \Sigma_{j=1}^m \beta_j \cdot V_j$, so using rule $\rightarrow E$ we obtain $\Gamma \vdash (r) \rightarrow (t) : \Sigma_{i=1}^a \sum_{j=1}^m \alpha_i \times \beta_j \cdot T_i[\hat{W}]$. By lemma 4.1, $\Gamma \vdash (r) \rightarrow (t) : T$. If $s \rightarrow R$ then by the induction hypothesis $\Gamma \vdash t : R$ with $R \subseteq \sum_{j=1}^m \beta_j \cdot V_j$. By lemma 4.13, $\exists \delta, k, N \subseteq \{1, \ldots, m\}$, $W \leq V_k$ such that $R \equiv \delta \cdot \hat{W} + \sum_{j \in N} \beta_j \cdot V_j$. Notice that since we have obtained $t$ from $s$ by applying one of the rules in Group F, we can safely take $W \equiv V_k$. Then notice that $\sum_{i=1}^a \sum_{j \in N} \alpha_i \times \beta_j \cdot T_i[\hat{W}] + \sum_{i=1}^a \sum_{j \in N} \alpha_i \times \beta_j \cdot T_i[\hat{W}] \leq \sum_{i=1}^a \sum_{j \in N} \sum_{j=1}^m \alpha_i \times \beta_j \cdot T_i[\hat{W}] \leq T$.

- Let $(s) \rightarrow (r) \rightarrow (t)$ as a consequence of $s \rightarrow t$. Let $\Gamma \vdash (s) \rightarrow (r) : T$. Then by lemma 4.12, $\exists n, a, n, a > 1, U, T_1, \ldots, T_n, a_1, \ldots, a_n, b_1, \ldots, b_m, V_1, \ldots, V_m$ such
that $Γ ⊢ s$: $\sum_{i=1}^{n} α_i \cdot ∀X_i(U \rightarrow T_i)$ and $Γ ⊢ r$: $\sum_{i=1}^{m} β_j \cdot V_j$ where for all $V_j$, there exists $\tilde{W}_j$ such that $U[\tilde{W}_j/\tilde{X}] = V_j$ and $\sum_{i=1}^{n} \sum_{j=1}^{m} α_i \times β_j \cdot T_i[\tilde{W}_j/\tilde{X}] \leq T$. Cases: If $s →_R t$ with $R \notin$ Group F, then by the induction hypothesis $Γ ⊢ t$: $\sum_{i=1}^{n} α_i \cdot ∀X_i(U \rightarrow T_i)$, so using rule $→_E$, $Γ ⊢ (t) r$: $\sum_{i=1}^{n} \sum_{j=1}^{m} α_i \times β_j \cdot T_i[\tilde{W}_j/\tilde{X}]$. Then by lemma 4.1, $Γ ⊢ t$: $T$. If $s →_R t$ with $R$ in Group F, then by the induction hypothesis $Γ ⊢ t$: $R$ where $R \subseteq \sum_{i=1}^{n} α_i \cdot ∀X_i(U \rightarrow T_i)$) By lemma 4.13, $\exists δ, k, N \subseteq \{1, \ldots, n\}, W \subseteq ∀X_i(U \rightarrow T_k)$ such that $R \equiv δ \cdot W + \sum_{i \in N} α_i \cdot ∀X_i(U \rightarrow T_i)$. Notice that since we have obtained $t$ from $s$ by applying one of the Group F rules, we can safely take $W \equiv ∀X_i(U \rightarrow T_k)$. Then using rule $→_E$, $Γ ⊢ (t) r$: $\sum_{i=1}^{n} δ \times β_j \cdot T_k[\tilde{W}_j/\tilde{X}]$. Notice that $\sum_{j=1}^{m} δ \times β_j \cdot T_k[\tilde{W}_j/\tilde{X}] + \sum_{i \in N} \sum_{j=1}^{m} α_i \times β_j \cdot T_i[\tilde{W}_j/\tilde{X}] \subseteq \sum_{i=1}^{n} \sum_{j=1}^{m} α_i \times β_j \cdot T_i[\tilde{W}_j/\tilde{X}] ≤ T$.

Let $λx.s: λx.t$ as a consequence of $s \rightarrow t$. Let $Γ ⊢ λx.s: T$. By lemma 4.9, $∃U, R$ such that $U \rightarrow R \leq T$ and $Γ, x: U ⊢ s: R$. Then by the induction hypothesis $Γ, x: U ⊢ t: S$, with $S \subseteq R$. So using rule $→_I$, $Γ ⊢ λx.t: U \rightarrow S$. Notice that since $S \subseteq R$, then $U \rightarrow S \subseteq U \rightarrow R \subseteq T$.

\section{Proof of Strong Normalization}

We use reducibility candidates, a well-known method described for example in [10]. A \textit{value} is a non-reducing term of the form $λx.s$, $α \cdot λx.t$ or sums of thereof. A \textit{neutral term} is a term that is not a value. The set of closed \textit{neutral terms} is denoted with $N$. We write $A_0$ for the set of closed terms and $SN_0$ for the set of closed, strongly normalizing terms. If $t$ is any term, $Red(t)$ is the set of all terms $t'$ such that $t \rightarrow t'$, and $Red_e(t)$ is the set of all terms $t'$ such that $t \rightarrow^* t'$. Both are naturally extended to sets of terms. We say that a set $S$ of closed terms is a reducibility candidate, denoted with $S \in RC$ if the following conditions are verified:

\textbf{RC}_1: Strong normalization: $S \subseteq SN_0$.
\textbf{RC}_2: Stability under reduction: $t \in S$ implies $Red(t) \subseteq S$.
\textbf{RC}_3: Stability under neutral expansion: If $t \in N$ and $Red(t) \subseteq S$ then $t \in S$.
\textbf{RC}_4: Closure under linear combinations: If $s$ and $t$ are members of $S$ then so are $0$, $α \cdot s$, and $s + t$.

If $S$ is a set of terms, we define $\overline{S}$ to be the closure of $S$ under $\textbf{RC}_3$ and $\textbf{RC}_4$.

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

- $A → B$ is the set $\{t \in A_0 | ∀u \in A, (t) u \in B\}$.
- $A ⊕ B$ is the set $A \cup B$.

\textbf{Lemma 5.2} If $A$, $B$ and all the $A_i$’s are in $RC$, then so are $A → B$, $A ⊕ B$ and $\cap_i A_i$. 
**Proof.** The main difficulty of the proof is to show that linear combinations of strongly normalizing terms are strongly normalizing. It is done by using the trick in [3].

□

**Lemma 5.3** The operation $\oplus$ on $\mathcal{RC}$ is commutative and associative, and it commutes with the intersection.

**Reducibility model** A valuation $\rho$ is a partial function from types variables to reducibility candidates. The interpretation $\llbracket T \rrbracket_\rho$ of a type $T$ is defined inductively as follows: $\llbracket X \rrbracket_\rho = \rho(X)$, $\llbracket U \rightarrow T \rrbracket_\rho = \llbracket U \rrbracket_\rho \rightarrow \llbracket T \rrbracket_\rho$, $\llbracket \forall A \in \mathcal{RC} \exists S \llbracket S \rrbracket_\rho = \bigcap_{A \in \mathcal{RC}} \llbracket S \rrbracket_{\rho,X \rightarrow A}$.

From Lemma 5.2, the interpretation of any type is a reducibility candidate. From Lemma 5.3, it is stable under type equivalence.

We extend the definition of interpretation to typing context: If $\Gamma$ is $(x_i : T_i)_i$ and if $\rho$ is a valuation, then $\llbracket \Gamma \rrbracket_\rho$ is the set of substitutions sending any $x_i$ in $\Gamma$ to $\llbracket T_i \rrbracket_\rho$. We now write $\Gamma \models t : T$ if for every valuation $\rho$, every substitution $\sigma \in \llbracket \Gamma \rrbracket_\rho$ we have $\sigma(t) \in \llbracket T \rrbracket_\rho$.

**Lemma 5.4** For every valid sequent $\Gamma \vdash t : T$ we also have $\Gamma \models t : T$.

**Proof.** The proof is done by induction on the size of $t$. Choose a typing derivation for $\Gamma \vdash t : T$, we proceed by case distinction on the first rule used.

- $(\alpha )$. In this case, $t = x$ and then $\sigma(t) = \sigma(x)$. By definition, it belongs to $\llbracket T \rrbracket_\rho$.

- $(0)$. Here, $\sigma(t) = \sigma(0) = 0$. From $\mathcal{RC}_4$ it belongs to any reducibility candidate, therefore to $\llbracket T \rrbracket_\rho$.

- $(\alpha t )$. We have $T = \alpha \cdot T'$ and $\sigma(t) = \sigma(\alpha \cdot t') = \alpha \cdot \sigma(t')$ for some term $t'$ smaller than $t$. The sequent $\Gamma \vdash t' : T'$ is valid: $\sigma(t') \in \llbracket T' \rrbracket_\rho$. From $\mathcal{RC}_4$ we conclude that $\sigma(t)$ belongs to $\llbracket T \rrbracket_\rho = \llbracket T' \rrbracket_\rho$.

- $(\lambda t )$. Similar: the set $\llbracket R \rrbracket_\rho \oplus \llbracket S \rrbracket_\rho$ is closed under sums.

- $(\epsilon )$. As we already noted, $\llbracket \cdot \rrbracket_\rho$ is closed under type equivalence.

- $(\forall X )$. Immediate by noting that the set $\llbracket [S[U/X]] \rrbracket_\rho$ is larger than the intersection $\bigcap_{A \in \mathcal{RC}} \llbracket S \rrbracket_{\rho,X \rightarrow A}$.

- $(\forall i )$. By induction hypothesis, for all $\rho$, if $\sigma \in [\Gamma]_\rho$, we have $\sigma(t) \in \llbracket \Gamma \rrbracket_\rho$. Write $\rho$ as $\rho', X \rightarrow \mathcal{A}$. Since $X$ does not appear in $\Gamma$, $\sigma$ belongs to $\llbracket \Gamma \rrbracket_{\rho', X \rightarrow A}$ for all possible value $A$. Therefore, we can take the intersection: $\sigma(t) \in \bigcap_{\mathcal{A} \in \mathcal{RC}} \llbracket U \rrbracket_{\rho', X \rightarrow \mathcal{A}}$. We can conclude using Lemma 5.3.

- $(\rightarrow t)$. We have $t = \lambda x . s$, $T = U \rightarrow S$ and $\Gamma, x : U \vdash s : S$ valid. By induction hypothesis, $\Gamma, x : U \models s : S$. In particular, $s$ is strongly normalizing.

Choose a valuation $\rho$ and a substitution $\sigma \in [\Gamma]_\rho$; we want to show that $\sigma(t) = \lambda x . \sigma(s) \in [U]_\rho \rightarrow [S]_\rho$. This means that for all $r$ in $[U]_\rho$, we have to show that $(\lambda x . \sigma(s)) r$ is in $[S]_\rho$. We proceed by induction on the size of $r$ and on the sum of the lengths of the longest rewrite sequences starting with $r$ and with $\sigma(s)$. The set $\text{Red}((\lambda x . \sigma(s)) r)$ consists of terms of the form:

- $(\lambda x . s') r$ and $(\lambda x . \sigma(s)) r'$, when $\sigma(s) \rightarrow s'$ and $r \rightarrow r'$. They are both in $[S]_\rho$ by induction hypothesis.
• \( \sigma(s)[r/x] \). This is of the form \( \sigma'(s) \) when \( \sigma' \in T \): the induction hypothesis applies again.
• \( 0 \), when \( r = 0 \): we conclude with \( RC_4 \).
• \( (\lambda x.s') r_1 + (\lambda x.s') r_2 \), when \( r = r_1 + r_2 \). Since both \( r_1 \) and \( r_2 \) are strictly smaller than \( r \), we can conclude by induction hypothesis that \( (\lambda x.s') r_1 \) and \( (\lambda x.s') r_2 \) are in \( T_\rho \). We conclude using \( RC_4 \).
• \( \alpha \cdot (\lambda x.s') r' \), when \( r = \alpha \cdot r' \). Similar to the previous case.

So \( \text{Red}(\lambda x.\sigma(s)) r \subseteq \text{Red}(s) r \); we conclude using \( RC_3 \).

\(- (\to_E) \). The term \( t \) is of the form \( (s) r \), and by induction hypothesis we have \( \Gamma \vdash t : \sum_i \alpha_i \cdot \forall \vec{X}. (U \to V_i) \) and \( \Gamma \vdash r : \sum_j \beta_j \cdot \forall \vec{Y} \). We want to show that \( \sigma((s) r) \in \oplus_{i,j} [T_i \vec{W}_j / \vec{X}]_\rho \), that is, \( \sigma((s) r) \) belongs to \( \oplus_{i,j} [T_i \vec{W}_j / \vec{X}]_\rho \). Since both \( \sigma(s) \) and \( \sigma(r) \) are strongly normalizing, we proceed by induction on the sum of the lengths of their longest rewrite sequences. The set \( \text{Red}(\sigma((s) r)) \) contains:

• \( (\sigma(s)) r' \) and \( (s') \sigma(r) \) when \( \sigma(s) \to s' \) or \( \sigma(r) \to r' \). They both belong to \( \oplus_{i,j} [T_i \vec{W}_j / \vec{X}]_\rho \), by induction hypothesis.

• A term coming from one of the rewrite of the group A. We conclude by noting that we obtain a linear combination of terms smaller that the original one. We can conclude with the induction hypothesis, Lemma 4.12 and \( RC_4 \).

• The term \( \sigma(s')[\sigma(r)/x] \), when \( s = \lambda x.s' \) and \( r \) is a base term. Note that this term is of the form \( \sigma'(s') \), where \( \sigma' \in \Gamma \). We are in the situation where the types of \( s \) and \( r \) are respectively \( \forall \vec{X}. (U \to V) \) and \( V \). There is \( \vec{W} \) such that \( V = U[\vec{W} / \vec{X}] \), so we can conclude by an argument similar to the one of the case \( (\forall E) \).

Since the set \( \text{Red}(\sigma((s) r)) \) is contained in \( \oplus_{i,j} [T_i \vec{W}_j / \vec{X}]_\rho \), we can conclude by applying \( RC_3 \).

\( \square \)

**Theorem 5.5** Suppose that \( \Gamma \vdash t : T \) is a valid term, then \( t \) is strongly normalizing.

**Proof.** If \( \Gamma \) is the list \( (x_i : U_i)_i \), the sequent \( \vdash \lambda x_1 \ldots x_n. t : U_1 \to (\cdots \to (U_n \to T) \cdots) \) is valid. Using Lemma 5.4, we deduce that for any valuation \( \rho \) and any substitution \( \sigma \in [0]_\rho \), we have \( \sigma(t) \in [T]_\rho \). By construction, \( \sigma \) does nothing on \( t : \sigma(t) = t \). Since \( [T]_\rho \) is a reducibility candidate, \( t \) is strongly normalizing.

Now suppose that \( t \) were not strongly normalizing. There would be an infinite rewrite sequence of terms \( (t_i)_i \) starting with \( t \). But then \( (\lambda \vec{x}. t_i)_i \) would then be an infinite rewrite sequence of terms starting with a strongly normalizing term: contradiction. Therefore, \( t \) is strongly normalizing. \( \square \)