

A SYSTEM F ACCOUNTING FOR SCALARS

PABLO ARRIGHI^a AND ALEJANDRO DÍAZ-CARO^b

^a ENS-Lyon, Laboratoire LIP and Université de Grenoble, Laboratoire LIG, France
e-mail address: pablo.arrighi@imag.fr

^b Université de Grenoble, Laboratoire LIG and Université Paris-Nord, Laboratoire LIPN, France
e-mail address: alejandro.diaz-caro@lipn.univ-paris13.fr

ABSTRACT. The algebraic lambda-calculus and the linear-algebraic lambda-calculus extend the lambda-calculus with the possibility of making arbitrary linear combinations of terms. In this paper we provide a fine-grained, System F -like type system for the linear-algebraic lambda-calculus. We show that this “*Scalar*” type system enjoys both the subject-reduction property and the strong-normalisation property, our main technical results. The latter yields a significant simplification of the linear-algebraic lambda-calculus itself, by removing the need for some restrictions in its reduction rules. But the more important, original feature of the *Scalar* type system is that it keeps track of ‘the amount of a type’ that is present in each term. As an example of its use, we show that it can serve as a guarantee that the normal form of a term is barycentric, i.e that its scalars are summing to one.

1. INTRODUCTION

(Linear-)algebraic λ -calculi. A number of recent works seek to endow the λ -calculus with a vector space structure. This agenda has emerged simultaneously in two different contexts.

- The field of *Linear Logic* considers a logic of resources where the propositions themselves stand for those resources – and hence cannot be discarded nor copied. When seeking to find models of this logic, one obtains a particular family of vector spaces and differentiable functions over these. It is by trying to capture back these mathematical structures into a programming language that Ehrhard and Regnier have defined the *differential λ -calculus* [ER03], which has an intriguing differential operator as a built-in primitive and an algebraic module of the λ -calculus terms over natural numbers. Lately Vaux [Vau07, Vau09] has focused his attention on a ‘differential λ -calculus without differential operator’, extending the algebraic module to positive real numbers. He obtained a confluence result

1998 ACM Subject Classification: F.4.1.

Key words and phrases: linear-algebraic λ -calculus; type theory; barycentric calculus.

An extended abstract of an earlier version of this paper has appeared in [ADC11].

^b This work was partially supported by grants from DIGITEO and Région Île-de-France.

in this case, which stands even in the untyped setting. More recent works on this *algebraic λ -calculus* tend to consider arbitrary scalars [Ehr10, Tas09].

- The field of *Quantum Computation* postulates that, as computers are physical systems, they may behave according to the quantum theory. It proves that, if this is the case, novel, more efficient algorithms are possible [Sho97, Gro96] – which have no classical counterpart. Whilst partly unexplained, it is nevertheless clear that the algorithmic speed-up arises by tapping into the parallelism granted to us ‘for free’ by the *superposition principle*; which states that if \mathbf{t} and \mathbf{u} are possible states of a system, then so is the formal linear combination of them $\alpha.\mathbf{t} + \beta.\mathbf{u}$ (with α and β some arbitrary complex numbers, up to a normalizing factor). The idea of a module of λ -terms over an arbitrary scalar field arises quite naturally in this context. This was the motivation behind the *linear-algebraic λ -calculus* by Dowek and one of the authors [AD08], who obtained a confluence result which holds for arbitrary scalars and again covers the untyped setting.

These two languages are rather similar: they both merge higher-order computation, be it terminating or not, in its simplest and most general form (namely the untyped λ -calculus) together with linear algebra in its simplest and most general form also (the axioms of vector spaces). In fact they can simulate each other [DCPTV10]. Our starting point is the second one, also referred to as *Lineal* in this work: because its confluence proof allows arbitrary scalars and because one has to make a choice. Whether the models developed for the first language, and the type systems developed for the second language, carry through to one another via their reciprocal simulations, is a current topic of investigation.

Other motivations to study (linear-)algebraic λ -calculi. The two languages are also reminiscent of other works in the literature:

- *Algebraic and symbolic computation.* The functional style of programming is based on the λ -calculus together with a number of extensions, so as to make everyday programming more accessible. Hence since the birth of functional programming there have been several theoretical studies on extensions of the λ -calculus in order to account for basic algebra (see for instance Dougherty’s algebraic extension [Dou92] for normalising terms of the λ -calculus) and other basic programming constructs such as pattern-matching [CKL01, AMR09], together with sometimes non-trivial associated type theories [Pet09]. Whilst this was not the original motivation behind (linear-)algebraic λ -calculi, they could still be viewed as just an extension of the λ -calculus in order to handle operations over vector spaces and make programming more accessible upon them. The main difference in approach is that the λ -calculus is not seen here as a control structure which sits on top of the vector space data structure, controlling which operations to apply and when. Rather, the λ -calculus terms themselves can be summed and weighted, hence they actually are vectors, upon which they can also act.
- *Parallel and probabilistic computation.* The above intertwinings of concepts are essential if seeking to represent parallel or probabilistic computation as it is the computation itself which must be endowed with a vector space structure. The ability to superpose λ -calculus terms in that sense takes us back to Boudol’s parallel λ -calculus [Bou94]. It may also be viewed as being part of a series of works on probabilistic extensions of calculi, *e.g.* [BH03, HP00] and [dP95, DHW05, DZ11] for λ -calculus more specifically.

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

Imposing the existence of a norm in (linear-)algebraic λ -calculi. We do not explore any of the above-mentioned fields of applications specifically in this paper. The theory of vector spaces has many fields of applications, but also many theoretical refinements that must be studied in their own right. We take the view that, in the same way, the theory of vector spaces plus λ -calculus has got theoretical refinements that we must study. Moreover, some of these theoretical refinements are often pre-requisite in order to address other applications, as is notoriously the case with the notion of norm. For instance if we want to be able to interpret a linear combination of terms $\sum \alpha_i.t_i$ as a probability distribution, we need to make sure that it has norm one. The same is true if we want to interpret $\sum \alpha_i.t_i$ as quantum superposition, but with a different norm¹. Yet the very definition of a norm is difficult in our context: deciding whether a term terminates is undecidable; but these terms can produce infinities (*e.g.* a non terminating term which reduces to itself plus 1), hence convergence of the norm is undecidable. Related to this precise topic, Vaux has studied simply typed algebraic λ -calculus, ensuring convergence of the norm [Vau07, Vau09]. Tasson has studied model-theoretic properties of the *barycentric* fragment of this simply-typed algebraic λ -calculus. Ehrhard has proven [Ehr10] the convergence of a Taylor series expansion of algebraic λ -calculus terms, via a System F type system [Gir71, Rey74].

Therefore, it can be said that standard type systems provide part of the solution: they ensure the convergence of (the existence of the norm of) a term. And indeed it is not so hard to define a simple extension of System F that fits *Lineal* – just by providing the needed rules to type additions, scalar products and the null vector in some trivial manner (see Definition 5.1). In this paper we do provide a full-blown proof of strong normalisation for this type system. A byproduct of this result is that we are able to lift several conditions that were limiting the reduction rules of *Lineal* and still prove its confluence. This is because the purpose of these conditions was really to prevent indefinite forms from reducing (*e.g.* $\mathbf{t} - \mathbf{t}$, with \mathbf{t} a divergent term). Hence the result together with its byproduct make *Lineal* into a simpler language.

Imposing a given norm in (linear-)algebraic λ -calculi. Standard type systems only provide a part of the solution; they are unable for instance to impose that well-typed terms have normal form $\sum \alpha_i.t_i$ with the constraint that $\sum \alpha_i = 1$, a property which is referred to as barycentricity in the previous literature [Tas09]. Our main contribution in this paper is to have managed a type system of a novel kind, *Scalar*, which has the ability to keep track of ‘the amount of a type’ by summing the amplitudes of its contributing terms and reflecting this amount through scalars within the types. As an example application of the *Scalar* type system, we have demonstrated that the type system can be used to enforce the barycentric property (see section 6), thereby specializing *Lineal* into a barycentric, higher-order λ -calculus.

¹Whereas it is clear already that *Lineal* is a quantum λ -calculus, in the sense that any quantum algorithm can be expressed in this language [AD08], the converse, alas, is not true, in the sense that some programs in *Lineal* express evolutions which are not valid quantum algorithms. This is precisely because *Lineal* does not restrict its vectors to be normalized $\sum |\alpha_i|^2 = 1$ and its applications to be isometries relative to this norm.

Endowing *Lineal* with a non-trivial, more informative, fine-grained type system was quite a challenge, as the reader will judge. We believe that fine-grained type theories for these non-deterministic, parallel, probabilistic or quantum extensions of calculi, which capture how many processes are in what class of states (with what weight), constitute truly promising routes towards relevant, less ad-hoc quantitative logics.

Plan. Section 2 presents an overview of the linear-algebraic λ -calculus (*Lineal*) [AD08]. Section 3 presents the *Scalar* type system with its grammar, equivalences and inference rules. Section 4 shows the subject reduction property giving consistency to the system. Section 5 shows the strong normalisation property for this system, allowing us to lift the above discussed restrictions in the reduction rules. In section 6 we formalize the type system \mathcal{B} for barycentric calculi. Section 7 concludes.

2. LINEAR-ALGEBRAIC λ -CALCULUS

Intuitions. As a language of terms, *Lineal* is just the λ -calculus together with the possibility to make arbitrary linear combinations of terms: if \mathbf{t} and \mathbf{r} are two terms, $\alpha.\mathbf{t} + \beta.\mathbf{r}$ is also a term. Regarding the operational semantics, *Lineal* merges higher-order computation, be it terminating or not, in its simplest and most general form (the β -reduction of the untyped λ -calculus) together with linear algebra in its simplest and most general form also (the oriented axioms of vector spaces). Care must be taken, however, when merging these two families of reduction rules.

For instance the term $(\lambda x xx)(\alpha.\mathbf{t} + \beta.\mathbf{u})$ maybe thought of as reducing to $(\alpha.\mathbf{t} + \beta.\mathbf{u})(\alpha.\mathbf{t} + \beta.\mathbf{u})$ in a call-by-name-oriented view or to $\alpha.(\mathbf{tt}) + \beta.(\mathbf{uu})$ in a call-by-value-oriented view, also compatible with the view that application should be bilinear, just as in linear algebra (*cf. Application rules*, below). Leaving both options open would break confluence, the second option was chosen, which entails restricting the β -reduction to terms not containing sums or scalars in head position (*cf. Beta reduction rule*, below).

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

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

and consider a term \mathbf{b} such that \mathbf{Yb} is β -equivalent to $\mathbf{b} + \mathbf{Yb}$ and so on. Modulo equality over vector spaces, the theory would be inconsistent, as the term $\mathbf{Yb} - \mathbf{Yb}$ would then be equal to $\mathbf{0}$, but would also reduce to $\mathbf{b} + \mathbf{Yb} - \mathbf{Yb}$ and hence also be equal to \mathbf{b} . Instead, this problem can be fixed by restricting rules such as $\alpha.\mathbf{u} + \beta.\mathbf{u} \rightarrow (\alpha + \beta).\mathbf{u}$ to terms that cannot reduce forever (*cf. Factorisation rules*, below), matching the old intuition that indefinite forms ‘ $\infty - \infty$ ’ must be left alone. Moreover, oriented axioms of vector spaces define vector spaces and no more than vector spaces, just as well as the original axioms do, as was shown in [AD08]. Plus the orientation serves a purpose: it presents the vector in its canonical form.

Definitions. The untyped *Lineal* calculus, as defined in [AD08], is presented in Figure 1. Terms contain a subclass of *base terms*, that are the only ones that can be substituted for a variable in a β -reduction step. We use the convention that terms equal up to associativity and commutativity axioms on $+$ (that is an *AC-rewrite system*). In particular, we have $\mathbf{t} + \mathbf{u} = \mathbf{u} + \mathbf{t}$ and $\mathbf{t} + (\mathbf{u} + \mathbf{r}) = (\mathbf{t} + \mathbf{u}) + \mathbf{r}$. That rewriting modulo equational theory has been extensively studied [JK86, PS81] to a high-level of rigour, and largely discussed in the context of this calculus in [AD08]. Scalars (denoted by $\alpha, \beta, \gamma, \dots$) are members of a commutative ring $(\mathcal{S}, +, \times, 0)$. Scalars can be defined as forming a rewrite system by itself [AD08], however for the sake of simplicity, it is enough to consider them as members of a ring. For example, the term $6.\mathbf{t}$ and the term $(2 \times 3).\mathbf{t}$ are considered to be the same term. Also, α -equivalence is considered implicitly, that is we rename variables when needed to avoid variable capturing. To be fully formal we would need to introduce these three equivalences, however they are standard so we do not mention them explicitly as long as it does not lead to confusion. The parentheses associations are the standard. We add the following conventions: $\alpha.\beta.\mathbf{t}$ stands for $\alpha.(\beta.\mathbf{t})$ and $\mathbf{t}_1 \dots \mathbf{t}_n + \mathbf{r}_1 \dots \mathbf{r}_n$ stands for $(\mathbf{t}_1 \dots \mathbf{t}_n) + (\mathbf{r}_1 \dots \mathbf{r}_n)$.

The confluence of this calculus has been formally proven in [AD08].

The set of free variables of a term (notation: $\text{FV}(\mathbf{t})$) is defined as expected. The operation of substitution on terms (notation: $\mathbf{t}[\mathbf{b}/x]$) is defined as usual (*i.e.* taking care of renaming bound variables when needed in order to prevent variable capture), with $(\alpha.\mathbf{t} + \beta.\mathbf{u})[\mathbf{b}/x] = \alpha.(\mathbf{t}[\mathbf{b}/x]) + \beta.(\mathbf{u}[\mathbf{b}/x])$. Finally, $\text{nf}(\mathbf{t})$ denotes the unique normal form of the term \mathbf{t} and it is used only when it is clear that \mathbf{t} has a unique normal form. As usual, \rightarrow^* is the application of zero or more reduction rules.

Example 2.1. One may be tempted to think $\lambda x 2.x \sim 2.\lambda x x$. Consider the following reduction. $(\lambda x 2.x + \lambda x x)(3.\mathbf{b}_1 + 2.\mathbf{b}_2)$

$$\begin{aligned} &\rightarrow^* 3.((\lambda x 2.x)\mathbf{b}_1) + 3.((\lambda x x)\mathbf{b}_1) + 2.((\lambda x 2.x)\mathbf{b}_2) + 2.((\lambda x x)\mathbf{b}_2) \\ &\rightarrow^* 3.2.\mathbf{b}_1 + 3.\mathbf{b}_1 + 2.2.\mathbf{b}_2 + 2.\mathbf{b}_2 \\ &\rightarrow^* 6.\mathbf{b}_1 + 3.\mathbf{b}_1 + 4.\mathbf{b}_2 + 2.\mathbf{b}_2 \\ &\rightarrow^* 9.\mathbf{b}_1 + 6.\mathbf{b}_2 \end{aligned}$$

However while these terms behave the same in this case, they are different: the first one is the function (a base term) duplicating its argument, while the second one is twice the identity function. The display below is clarifying:

$$\begin{aligned} (\lambda x y)(\lambda x 2.x) &\rightarrow y, \\ (\lambda x y)(2.\lambda x x) &\rightarrow 2.(\lambda x y)(\lambda x x) \rightarrow 2.y. \end{aligned}$$

3. THE *Scalar* TYPE SYSTEM

The grammar of types, *cf.* Figure 2, defines the set of types (denoted by T, R, S) and its syntactic subclass (denoted by U, V, W) of what we call *unit types*. Note that the grammar for unit types does not allow for scalars except to the right of an arrow. More generally, note the novelty of having scalars at the level of types. We will see that these scalars keep track of the sum of the coefficients of the terms contributing to the type.

Type variables are denoted by X, Y , etc. and can only ever be substituted by a unit type. Contexts are denoted by Γ, Δ , etc. and are defined as sets $\{x:U, \dots\}$, where x is a term

<i>Terms:</i>	$\mathbf{r}, \mathbf{t}, \mathbf{u} ::= \mathbf{b} \mid \mathbf{tr} \mid \mathbf{0} \mid \alpha.\mathbf{t} \mid \mathbf{t} + \mathbf{r}$	
<i>Base terms:</i>	$\mathbf{b} ::= x \mid \lambda x \mathbf{t}$	
<i>Elementary rules:</i>	<i>Factorisation rules:</i>	<i>Application rules:</i>
$\mathbf{t} + \mathbf{0} \rightarrow \mathbf{t},$	$\alpha.\mathbf{t} + \beta.\mathbf{t} \rightarrow (\alpha + \beta).\mathbf{t}$ (*),	$(\mathbf{t} + \mathbf{r})\mathbf{u} \rightarrow \mathbf{tu} + \mathbf{ru}$ (**),
$0.\mathbf{t} \rightarrow \mathbf{0},$	$\alpha.\mathbf{t} + \mathbf{t} \rightarrow (\alpha + 1).\mathbf{t}$ (*),	$\mathbf{u}(\mathbf{t} + \mathbf{r}) \rightarrow \mathbf{ut} + \mathbf{ur}$ (**),
$1.\mathbf{t} \rightarrow \mathbf{t},$	$\mathbf{t} + \mathbf{t} \rightarrow (1 + 1).\mathbf{t}$ (*).	$(\alpha.\mathbf{t})\mathbf{r} \rightarrow \alpha.\mathbf{tr}$ (*),
$\alpha.\mathbf{0} \rightarrow \mathbf{0},$		$\mathbf{r}(\alpha.\mathbf{t}) \rightarrow \alpha.\mathbf{rt}$ (*),
$\alpha.(\beta.\mathbf{t}) \rightarrow (\alpha \times \beta).\mathbf{t},$	<i>Beta reduction:</i>	$\mathbf{0t} \rightarrow \mathbf{0},$
$\alpha.(\mathbf{t} + \mathbf{r}) \rightarrow \alpha.\mathbf{t} + \alpha.\mathbf{r}.$	$(\lambda x \mathbf{t})\mathbf{b} \rightarrow \mathbf{t}[\mathbf{b}/x]$ (***)	$\mathbf{t0} \rightarrow \mathbf{0}.$
<i>Contextual rules:</i> If $\mathbf{t} \rightarrow \mathbf{r}$, then for any term \mathbf{u} and variable x ,		
$\mathbf{tu} \rightarrow \mathbf{ru},$	$\mathbf{t} + \mathbf{u} \rightarrow \mathbf{r} + \mathbf{u},$	$\alpha.\mathbf{t} \rightarrow \alpha.\mathbf{r}$ and
$\mathbf{ut} \rightarrow \mathbf{ur},$	$\mathbf{u} + \mathbf{t} \rightarrow \mathbf{u} + \mathbf{r},$	$\lambda x \mathbf{t} \rightarrow \lambda x \mathbf{r}.$

where $+$ is an associative-commutative (*AC*) symbol and

(*) these rules apply only if \mathbf{t} is a closed normal term.

(**) these rules apply only if $\mathbf{t} + \mathbf{r}$ is a closed normal term.

(***) the rule applies only when \mathbf{b} is a base term.

Restriction (***) is the one that limits the β -reduction, whereas restrictions (*) and (**) are those that avoid confluence problems related to infinities and indefinite forms, as discussed above.

Figure 1: Syntax and reduction rules of *Lineal*.

variable appearing only once in the set and U is a unit type. We usually omit the brackets of the set. The substitution of V for X in U is defined as usual and is written $U[V/X]$. We sometimes use the vectorial notation $U[\vec{V}/\vec{X}]$ for $U[V_1/X_1] \cdots [V_n/X_n]$ if $\vec{X} = X_1, \dots, X_n$ and $\vec{V} = V_1, \dots, V_n$. In order to avoid capture, we consider that X_i cannot appear free in V_j , with $j < i$. Also we assume that free and bound variables of a type are distinct. We also may abuse notation and say $\vec{X} \notin S$ meaning that none of the X_i from \vec{X} are in the set S . We write $\Gamma[U/X]$ to the substitution of U for X in each type of Γ . Also we write $(\alpha.T)[U/X]$ for $\alpha.T[U/X]$. $\text{FV}(T)$ designates the set of free variables of the type T , defined in the usual manner, and $\text{FV}(\Gamma)$ is the union of the sets of free variables of each type in Γ . Scalars are denoted by $\alpha, \beta, \gamma \dots$ and are members of the same commutative ring $(\mathcal{S}, +, \times)$ as those of terms.

We also define an equivalence relation upon types as follows:

Definition 3.1. For any α, β and T . We define the type equivalence \equiv to be the least congruence such that

$$\bullet \alpha.\bar{0} \equiv \bar{0} \quad \bullet 0.T \equiv \bar{0} \quad \bullet 1.T \equiv T \quad \bullet \alpha.(\beta.T) \equiv (\alpha \times \beta).T \quad \bullet \forall X.\alpha.T \equiv \alpha.\forall X.T$$

The complete set of typing rules is shown in Figure 2. Splitting the grammar into general types and unit types is a necessary consequence of the fact that we want scalars in the types to reflect scalars in the terms (*e.g.* $\alpha.\lambda x \mathbf{t}$ should have a type $\alpha.U$). Indeed if we did not have the restriction on the left hand side of an arrow being a unit type, *i.e.* $U \rightarrow T$, then we would have types like $(\alpha.X) \rightarrow X$, which *a priori* do not make sense, because

<i>Types:</i>	$R, S, T ::= U \mid \forall X.T \mid \alpha.T \mid \bar{0}$	
<i>Unit types:</i>	$U, V, W ::= X \mid U \rightarrow T \mid \forall X.U$	
$\frac{}{\Gamma, x:U \vdash x:U} ax$	$\frac{\Gamma \vdash \mathbf{t}:T \quad T \equiv S}{\Gamma \vdash \mathbf{t}:S} \equiv$	$\frac{\Gamma \vdash \mathbf{t}:\alpha.(U \rightarrow T) \quad \Gamma \vdash \mathbf{r}:\beta.U}{\Gamma \vdash \mathbf{tr}:(\alpha \times \beta).T} \rightarrow_E$
$\frac{\Gamma, x:U \vdash \mathbf{t}:T}{\Gamma \vdash \lambda x \mathbf{t}:U \rightarrow T} \rightarrow_I$	$\frac{\Gamma \vdash \mathbf{t}:\forall X.T}{\Gamma \vdash \mathbf{t}:T[U/X]} \forall_E$	$\frac{\Gamma \vdash \mathbf{t}:T \quad X \notin \text{FV}(\Gamma)}{\Gamma \vdash \mathbf{t}:\forall X.T} \forall_I$
$\frac{}{\Gamma \vdash \mathbf{0}:\bar{0}} ax_{\bar{0}}$	$\frac{\Gamma \vdash \mathbf{t}:\alpha.T \quad \Gamma \vdash \mathbf{r}:\beta.T}{\Gamma \vdash \mathbf{t} + \mathbf{r}:(\alpha + \beta).T} +_I$	$\frac{\Gamma \vdash \mathbf{t}:T}{\Gamma \vdash \alpha.\mathbf{t}:\alpha.T} s_I$

Figure 2: Types and typing rules of *Scalar*

abstractions receive only base terms as arguments. This could be fixed by adding the equivalence $(\alpha.A) \rightarrow B \equiv \alpha.(A \rightarrow B)$, making sure that α is non-zero. But still we would need to keep the \rightarrow_E rule restricted to having a unit type on the left of the arrow, otherwise we would break the required correspondence between scalars-in-types and scalars-in-terms, *e.g.*:

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

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

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

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

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

By the rule (3), we must have:

$$\frac{\Gamma \vdash \mathbf{t}:\alpha.(U \rightarrow T) \quad \Gamma \vdash \mathbf{r}:U}{\Gamma \vdash \mathbf{tr}:\alpha.T}$$

By the rule (4), we must have:

$$\frac{\Gamma \vdash \mathbf{t}:U \rightarrow T \quad \Gamma \vdash \mathbf{r}:\beta.U}{\Gamma \vdash \mathbf{tr}:\beta.T}$$

By combining these two we obtain the rule \rightarrow_E presented in Figure 2.

Remark 3.2. A good insight into the type system is that, due to the equivalence relation, scalars in types occur only at the top-level and in the target subtypes of arrow types. This fits very well with the idea that in *Lineal* all term constructs are linear, except for abstraction. With this in mind, the syntax of arrow types could have been restricted to $U \rightarrow \alpha.V$, or even written as $U \rightarrow_{\alpha} V$ instead. In other words, we could get rid of the type equivalences (*cf.* Definition 3.1) and represent each type equivalence class by just its canonical member. Such a design choice would spare us some lemmas (*cf.* Section 4), but comes at a price:

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

4. SUBJECT REDUCTION

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

Theorem 4.1 (Subject Reduction). *For any terms t, t' , any context Γ and any type T , if $t \rightarrow t'$ and $\Gamma \vdash t:T$, then $\Gamma \vdash t':T$.*

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

4.1. Preliminary lemmas. In order to prove this theorem, we need several auxiliary lemmas giving general properties of our system. We have tried to provide an intuition of every lemma so as to make it easier to follow. Also, we divided them in four groups, reflecting the nature of their statements. Standard and straightforward lemmas are left without proof to improve readability.

4.1.1. Lemmas about types. The lemmas in this section are statements about the properties of the types themselves, *i.e.* the type equivalence relation.

It is not so hard to see that every type is equivalent to unit type multiplied by a scalar. A unit type itself can of course always be multiplied by 1.

Lemma 4.2 (α unit). *For each type T there exists a unit type U and a scalar α such that $T \equiv \alpha.U$. \square*

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

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

Lemma 4.3 (Unit does not add scalars). *For any unit types U, U' and scalars α, β , if $\alpha.U \equiv \beta.U'$ then, $\alpha = \beta$ and if $\alpha \neq 0$, then $U \equiv U'$.*

Informal proof. Following the grammar of unit types, neither U nor U' could contain scalars in this head form but only in the right hand side of an arrow. However, no equivalence rule lets it come out from the right of the arrow and get to the head-form, so if $\alpha.U \equiv \beta.U'$ that means $\alpha = \beta = 0$ or $U \equiv U'$ and $\alpha = \beta$. \square

4.1.2. *Relation between types.* We define a family of relations between types inspired by [Bar92, def. 4.2.1]:

Definition 4.4. For any types T, R any context Γ and any term \mathbf{t} such that

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

- (1) if $X \notin \text{FV}(\Gamma)$, write $T \succ_{X, \Gamma}^{\mathbf{t}} R$ if
 - either $R \equiv \forall X.T$
 - or $T \equiv \forall X.S$ and $R \equiv S[U/X]$ for some U and S .
- (2) if \mathcal{V} is a set of type variables such that $\mathcal{V} \cap \text{FV}(\Gamma) = \emptyset$, we define $\succeq_{\mathcal{V}, \Gamma}^{\mathbf{t}}$ inductively by
 - If $X \in \mathcal{V}$ and $T \succ_{X, \Gamma}^{\mathbf{t}} R$, then $T \succeq_{\{X\}, \Gamma}^{\mathbf{t}} R$.
 - If $\mathcal{V}_1, \mathcal{V}_2 \subseteq \mathcal{V}$, $T \succeq_{\mathcal{V}_1, \Gamma}^{\mathbf{t}} R$ and $R \succeq_{\mathcal{V}_2, \Gamma}^{\mathbf{t}} S$, then $T \succeq_{\mathcal{V}_1 \cup \mathcal{V}_2, \Gamma}^{\mathbf{t}} S$.
 - If $T \equiv R$, then $T \succeq_{\mathcal{V}, \Gamma}^{\mathbf{t}} R$.

Example 4.5. Let the following be a valid derivation.

$$\frac{\frac{\frac{\Gamma \vdash \mathbf{t} : T \quad X \notin \text{FV}(\Gamma)}{\Gamma \vdash \mathbf{t} : \forall X.T} \forall_I}{\Gamma \vdash \mathbf{t} : T[U/X]} \forall_E \quad \frac{Y \notin \text{FV}(\Gamma)}{\Gamma \vdash \mathbf{t} : \forall Y.T[U/X]} \forall_I}{\Gamma \vdash \mathbf{t} : R} \equiv$$

Then $T \succeq_{\{X, Y\}, \Gamma}^{\mathbf{t}} R$.

Note that this relation is stable under reduction in the following way:

Lemma 4.6 (\succeq -stability). *For any types T, R , set of type variables \mathcal{V} , terms \mathbf{t} and \mathbf{r} and context Γ , if $T \succeq_{\mathcal{V}, \Gamma}^{\mathbf{t}} R$, $\mathbf{t} \rightarrow \mathbf{r}$ and $\Gamma \vdash \mathbf{r} : T$, then $T \succeq_{\mathcal{V}, \Gamma}^{\mathbf{r}} R$.*

Proof. It suffices to show this for $\succ_{X, \Gamma}^{\mathbf{t}}$, with $X \in \mathcal{V}$. Observe that since $T \succ_{X, \Gamma}^{\mathbf{t}} R$, then $X \notin \text{FV}(\Gamma)$. We only have to prove that $\Gamma \vdash \mathbf{r} : R$ is derivable from $\Gamma \vdash \mathbf{r} : T$. We proceed now by cases:

- $R \equiv \forall X.T$, then using rules \forall_I and \equiv , we can deduce $\Gamma \vdash \mathbf{r} : R$.
- $T \equiv \forall X.S$ and $R \equiv S[U/X]$, then using rules \forall_E and \equiv , we can deduce $\Gamma \vdash \mathbf{r} : R$. \square

The following lemma states that scalars do not interfere with the relation.

Lemma 4.7 (Scalars keep order). *For any types T, R , any scalar α , context Γ , set of variable types \mathcal{V} and term t , if $T \succeq_{\mathcal{V}, \Gamma}^t R$ then $\alpha.T \succeq_{\mathcal{V}, \Gamma}^{\alpha, t} \alpha.R$. \square*

The following lemma states that if two arrow types are ordered, then they are equivalent up to some substitutions.

Lemma 4.8 (Arrows comparison). *For any types U, V, T, R , context Γ , set of type variables \mathcal{V} and term t , if $V \rightarrow R \succeq_{\mathcal{V}, \Gamma}^t U \rightarrow T$, then there exist unit types \vec{W} and variables $\vec{X} \in \mathcal{V}$ such that*

$$U \rightarrow T \equiv (V \rightarrow R)[\vec{W}/\vec{X}]$$

Proof. cf. Appendix A.1. \square

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

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

We need five generation lemmas: the classical ones, one for applications (Lemma 4.9) and one for abstractions (Lemma 4.11); and three new ones for the algebraic rules, one for products by scalars different than 0 (Lemma 4.12) other for product by 0 (Lemma 4.13) and one for sums (Lemma 4.14). All of them follow by induction on the typing derivation.

Lemma 4.9 (Generation lemma (app)). *For any terms t, r , any type T , any scalar γ and any context Γ , if $\Gamma \vdash \mathbf{tr} : \gamma.T$, then there exist $\alpha, \beta, U, \mathcal{V}$ with $\gamma.R \succeq_{\mathcal{V}, \Gamma}^{\mathbf{tr}} \gamma.T$ such that $\Gamma \vdash \mathbf{r} : \alpha.U$ and $\Gamma \vdash \mathbf{t} : \beta.(U \rightarrow R)$ with $\alpha \times \beta = \gamma$. \square*

Remark 4.10. Consider the following example: Let $\Gamma \vdash \mathbf{t} : U \rightarrow R$ and $\Gamma \vdash \mathbf{r} : U$. Then $\Gamma \vdash \mathbf{tr} : R$ and if $R \succeq_{\mathcal{V}, \Gamma}^{\mathbf{tr}} T$, then $\Gamma \vdash \mathbf{tr} : T$. Note that in general, \mathbf{t} does not have type $U \rightarrow T$. As a counterexample note that $x : X \rightarrow X \vdash x : X \rightarrow X$ but $x : X \rightarrow X \not\vdash x : X \rightarrow \forall X.X$.

Lemma 4.11 (Generation lemma (abs)). *For any term t , any type T and any context Γ , if $\Gamma \vdash \lambda x \mathbf{t} : T$ then there exist U, R and \mathcal{V} such that $\Gamma, x : U \vdash \mathbf{t} : R$ and $U \rightarrow R \succeq_{\mathcal{V}, \Gamma}^{\lambda x} t.T$. \square*

Proof. cf. Appendix A.2. \square

Lemma 4.12 (Generation lemma (sc)). *For any scalar $\alpha \neq 0$, any context Γ , any term t and any type T , if $\Gamma \vdash \alpha.t : \alpha.T$, then $\Gamma \vdash \mathbf{t} : T$. \square*

Lemma 4.13 (Generation lemma (sc-0)). *For any context Γ , any term t and any type T , if $\Gamma \vdash 0.t : T$, then there exists R such that $\Gamma \vdash \mathbf{t} : R$. \square*

Lemma 4.14 (Generation lemma (sum)). *For any terms t, r , any scalar α , any unit type U and any context Γ , if $\Gamma \vdash \mathbf{t} + \mathbf{r} : \alpha.U$, then there exist $\delta, \gamma \in \mathcal{S}$ such that $\Gamma \vdash \mathbf{t} : \delta.U$ and $\Gamma \vdash \mathbf{r} : \gamma.U$ with $\delta + \gamma = \alpha$. \square*

Remark 4.15. Note that the assumption of the previous lemma could be weakened as $\Gamma \vdash \mathbf{t} + \mathbf{r} : T$, since by Lemma 4.2, any type T can be written as $\alpha.U$ for some scalar α and unit type U .

The following lemma is quite standard in proofs of subject reduction for System F -like systems and can be found for example in [Bar92, prop. 4.1.19] and [Kri90, prop. 8.2 and 8.5]. It ensures that when substituting types for type variables or terms for term variables, in an adequate manner, then the type derived remains valid.

Lemma 4.16 (Substitution lemma). *For any term \mathbf{t} , any base term \mathbf{b} , any types T, \vec{U} and any context Γ ,*

(1) *If $\Gamma \vdash \mathbf{t} : T$ then $\Gamma[\vec{U}/\vec{X}] \vdash \mathbf{t} : T[\vec{U}/\vec{X}]$.*

(2) *If $\Gamma, x : U \vdash \mathbf{t} : T$ and $\Gamma \vdash \mathbf{b} : U$ then $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T$.* □

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

Lemma 4.17 (Scaling unit). *For any term \mathbf{t} , scalar α , type T and context Γ , if $\Gamma \vdash \alpha.\mathbf{t} : T$ then there exists U and $\gamma \in \mathcal{S}$ such that $T \equiv \alpha.\gamma.U$.* □

Base terms are typed by unit types or its equivalent (proof by induction on the typing derivation).

Lemma 4.18 (Base terms in unit). *For any base term \mathbf{b} , context Γ and type T , if $\Gamma \vdash \mathbf{b} : T$ then there exists U such that $T \equiv U$.* □

By $ax_{\bar{0}}$, it is easy to see that $\mathbf{0}$ has type $\bar{0}$, but also by using equivalences between types we have that $\forall X.\bar{0}$ is equivalent to $\bar{0}$. Moreover, for any T, Γ and \mathcal{V} with $\mathcal{V} \cap \text{FV}(\Gamma) = \emptyset$, such that $\bar{0} \succeq_{\mathcal{V}, \Gamma}^{\mathbf{0}} T$, we have $T \equiv \bar{0}$. Then we can state the following lemma (proof by induction on the typing derivation).

Lemma 4.19 (Type for $\mathbf{0}$). *For any context Γ and type T , if $\Gamma \vdash \mathbf{0} : T$ then $T \equiv \bar{0}$.* □

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

Theorem 4.20 (Uniqueness of scalars). *For any term \mathbf{t} , any context Γ , any scalars α, β and any unit types U, V , if $\Gamma \vdash \mathbf{t} : \alpha.U$ and $\Gamma \vdash \mathbf{t} : \beta.V$, then $\alpha = \beta$.*

Proof. Structural induction over \mathbf{t} .

(1) $\mathbf{t} = \mathbf{0}$. Then by Lemmas 4.19 and 4.3, $\alpha = \beta = 0$.

(2) $\mathbf{t} = x$ or $\mathbf{t} = \lambda x \mathbf{t}'$. Then by Lemmas 4.18 and 4.3, $\alpha = \beta = 1$.

(3) $\mathbf{t} = \gamma.\mathbf{t}'$. Then by Lemma 4.17, there exist σ, δ, U', V' , such that $\alpha.U \equiv \gamma.\sigma.U'$ and $\beta.V \equiv \gamma.\delta.V'$. If $\gamma = 0$, then $\gamma \times \sigma = \gamma \times \delta = 0$ and then by Lemma 4.3, $\alpha = \beta = 0$. If $\gamma \neq 0$, then by Lemma 4.12, $\Gamma \vdash \mathbf{t}' : \sigma.U'$ and $\Gamma \vdash \mathbf{t}' : \delta.V'$, so by the induction hypothesis $\sigma = \delta$. Note that, by Lemma 4.3, $\alpha = \gamma \times \sigma$ and $\beta = \gamma \times \delta$, so $\alpha = \gamma \times \sigma = \gamma \times \delta = \beta$.

- (4) $\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2$. Then by Lemma 4.14, there exist γ_1, γ_2 such that $\Gamma \vdash \mathbf{t}_1 : \gamma_1.U$ and $\Gamma \vdash \mathbf{t}_2 : \gamma_2.U$ with $\gamma_1 + \gamma_2 = \alpha$; and also by the same lemma, there exist δ_1, δ_2 such that $\Gamma \vdash \mathbf{t}_1 : \delta_1.V$ and $\Gamma \vdash \mathbf{t}_2 : \delta_2.V$ with $\delta_1 + \delta_2 = \beta$. Then by the induction hypothesis $\gamma_1 = \delta_1$ and $\gamma_2 = \delta_2$, so $\alpha = \gamma_1 + \gamma_2 = \delta_1 + \delta_2 = \beta$.
- (5) $\mathbf{t} = \mathbf{t}_1 \mathbf{t}_2$. Then by Lemma 4.9, there exist $\gamma_1, \gamma_2, W, \mathcal{V}$ and $\alpha.T \succeq_{\mathcal{V}, \Gamma}^{\mathbf{t}_1 \mathbf{t}_2} \alpha.U$ such that $\Gamma \vdash \mathbf{t}_1 : \gamma_1.(W \rightarrow T)$ and $\Gamma \vdash \mathbf{t}_2 : \gamma_2.W$ with $\gamma_1 \times \gamma_2 = \alpha$; and also by the same lemma, there exist $\delta_1, \delta_2, W', \mathcal{V}'$ and $\beta.R \succeq_{\mathcal{V}', \Gamma}^{\mathbf{t}_1 \mathbf{t}_2} \beta.V$ such that $\Gamma \vdash \mathbf{t}_1 : \delta_1.(W' \rightarrow R)$ and $\Gamma \vdash \mathbf{t}_2 : \delta_2.W'$ with $\delta_1 \times \delta_2 = \beta$. Then by the induction hypothesis $\gamma_1 = \delta_1$ and $\gamma_2 = \delta_2$, so $\alpha = \gamma_1 \times \gamma_2 = \delta_1 \times \delta_2 = \beta$. \square

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

Corollary 4.21 (Uniqueness of $\bar{0}$). *For any term \mathbf{t} and any context Γ , if $\Gamma \vdash \mathbf{t} : \bar{0}$ then for each $T \neq \bar{0}$, $\Gamma \not\vdash \mathbf{t} : T$.*

Proof. Assume $\Gamma \vdash \mathbf{t} : T$, then by Lemma 4.2, $T \equiv \alpha.U$. Since $\Gamma \vdash \mathbf{t} : \bar{0} \equiv 0.U$ we obtain by Theorem 4.20 that $\alpha = 0$. \square

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

Lemma 4.22 (Linearity of $\mathbf{0}$). *For any term \mathbf{t} , any context Γ and any type T ,*

- (1) *If $\Gamma \vdash \mathbf{0}\mathbf{t} : T$ then $T \equiv \bar{0}$.* (2) *If $\Gamma \vdash \mathbf{t}\mathbf{0} : T$ then $T \equiv \bar{0}$.*

Proof.

- (1) Let $\Gamma \vdash \mathbf{0}\mathbf{t} : T$. By Lemma 4.2, $T \equiv \gamma.U$. Moreover, by Lemma 4.9, there exist $\alpha, \beta, U', \mathcal{V}$ and $\gamma.R \succeq_{\mathcal{V}, \Gamma}^{\mathbf{0}\mathbf{t}} \gamma.U$ such that $\Gamma \vdash \mathbf{0} : \beta.(U' \rightarrow R)$ and $\Gamma \vdash \mathbf{t} : \alpha.U'$ with $\gamma = \alpha \times \beta$. Hence, by Corollary 4.21, $\beta.(U' \rightarrow R) \equiv \bar{0} \equiv 0.U$, so by Lemma 4.3, $\beta = 0$, then $\gamma = \alpha \times 0 = 0$, so $T \equiv \gamma.U \equiv \bar{0}$.
- (2) Analogous to (1). \square

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

Lemma 4.23 (Product). *For any term \mathbf{t} , any scalars α and β and any context Γ , if $\Gamma \vdash \alpha.(\beta.\mathbf{t}) : T$ then $\Gamma \vdash (\alpha \times \beta).\mathbf{t} : T$.*

Proof. By Lemma 4.17, there exist U and $\gamma \in \mathcal{S}$ such that $T \equiv \alpha.\gamma.U$. We proceed now by cases:

- $\alpha \neq 0$ **and** $\beta \neq 0$: By Lemma 4.12, $\Gamma \vdash \beta.\mathbf{t} : \gamma.U$. Moreover, by Lemma 4.17 again, there exist U' and γ' such that $\gamma.U \equiv \beta.\gamma'.U'$. So, by Lemma 4.12, $\Gamma \vdash \mathbf{t} : \gamma'.U'$, from which, using the rule s_I one can derive $\Gamma \vdash (\alpha \times \beta).\mathbf{t} : (\alpha \times \beta).\gamma'.U'$. Note that $(\alpha \times \beta).\gamma'.U' \equiv \alpha.\beta.\gamma'.U' \equiv \alpha.\gamma.U \equiv T$.
- $\alpha \neq 0$ **and** $\beta = 0$: By Lemma 4.12, $\Gamma \vdash \mathbf{0}\mathbf{t} : \gamma.U$. Moreover, by Lemma 4.13, there exists R such that $\Gamma \vdash \mathbf{t} : R$, from which, using the rule s_I , one can derive $\Gamma \vdash (\alpha \times 0).\mathbf{t} : (\alpha \times 0).R = 0.R$. Note that by Lemma 4.2, there exists δ and V such that $R \equiv \delta.V$, so $0.R \equiv 0.V \equiv \bar{0}$. Since $(\alpha \times 0).\mathbf{t} = \mathbf{0}\mathbf{t}$, by Corollary 4.21, $\gamma = 0$, and note that $0.U \equiv \bar{0}$, so $0.R \equiv \bar{0} \equiv 0.U = (\alpha \times 0).U \equiv \alpha.0.U \equiv T$.

$\alpha = 0$: By Lemma 4.13, there exists R such that $\Gamma \vdash \beta.\mathbf{t}:R$. Then by Lemma 4.12 or 4.13, depending if $\beta = 0$ or not, there exists S such that $\Gamma \vdash \mathbf{t}:S$, from which, using the rule s_I , one can derive $\Gamma \vdash 0.\mathbf{t}:0.S$. Note that $0.S \equiv \bar{0} \equiv 0.\gamma.U \equiv T$. Also note that $0.\mathbf{t} = (0 \times \beta).\mathbf{t}$. \square

Lemma 4.24 (Distributivity). *For any terms \mathbf{t} and \mathbf{r} , any scalar α , any context Γ and any type T , if $\Gamma \vdash \alpha.(\mathbf{t} + \mathbf{r}):T$ then $\Gamma \vdash \alpha.\mathbf{t} + \alpha.\mathbf{r}:T$.*

Proof. Let $\Gamma \vdash \alpha.(\mathbf{t} + \mathbf{r}):T$. By Lemma 4.17, there exist α, R such that $T \equiv \alpha.R$. We proceed now by cases:

$\alpha \neq 0$: By Lemma 4.12, $\Gamma \vdash \mathbf{t} + \mathbf{r}:R$. By Lemma 4.2, there exist σ, U such that $R \equiv \sigma.U$. So by Lemma 4.14 there exist δ and γ such that $\Gamma \vdash \mathbf{t}:\delta.U$ and $\Gamma \vdash \mathbf{r}:\gamma.U$ with $\delta + \gamma = \sigma$. Then by rules s_I and \equiv , $\Gamma \vdash \alpha.\mathbf{t}:(\alpha \times \delta).U$ and $\Gamma \vdash \alpha.\mathbf{r}:(\alpha \times \gamma).U$, from which using the rule $+_I$ one can derive $\Gamma \vdash \alpha.\mathbf{t} + \alpha.\mathbf{r}:(\alpha \times \delta + \alpha \times \gamma).U$. Note that $(\alpha \times \delta + \alpha \times \gamma).U \equiv \alpha.\sigma.U \equiv \alpha.R \equiv T$.

$\alpha = 0$: By Lemma 4.13, there exists S such that $T \equiv 0.S \equiv \bar{0}$ with $\Gamma \vdash \mathbf{t} + \mathbf{r}:S$. In addition, by Lemma 4.2, there exist δ, V such that $S \equiv \delta.V$. Then by Lemma 4.14, there exist σ, ς such that $\Gamma \vdash \mathbf{t}:\sigma.V$ and $\Gamma \vdash \mathbf{r}:\varsigma.V$. By rules s_I and \equiv , $\Gamma \vdash 0.\mathbf{t}:0.V$ and $\Gamma \vdash 0.\mathbf{r}:0.V$, from which using the rule $+_I$ one can derive $\Gamma \vdash 0.\mathbf{t} + 0.\mathbf{r}:0.V$. Note that $0.V \equiv \bar{0} \equiv T$. \square

Lemma 4.25 (Factorisation). *For any term \mathbf{t} , scalars α and β , type T and context Γ , if $\Gamma \vdash \alpha.\mathbf{t} + \beta.\mathbf{t}:T$ then $\Gamma \vdash (\alpha + \beta).\mathbf{t}:T$.*

Proof. Let $\Gamma \vdash \alpha.\mathbf{t} + \beta.\mathbf{t}:T$. By Lemma 4.2, there exist σ, V such that $T \equiv \sigma.V$. So, by Lemma 4.14, there exist $\delta, \gamma \in \mathcal{S}$ such that $\Gamma \vdash \alpha.\mathbf{t}:\delta.V$ and $\Gamma \vdash \beta.\mathbf{t}:\gamma.V$ with $\delta + \gamma = \sigma$. Then by Lemma 4.17, there exist ϕ, φ, U and U' such that $\delta.V \equiv \alpha.\phi.U$ and $\gamma.V \equiv \beta.\varphi.U'$. So, by Lemma 4.3, $\delta = \alpha \times \phi$ and $\gamma = \beta \times \varphi$. We proceed now by cases:

$\alpha = 0$: Then $\delta = 0$ and so $\gamma = \sigma$. Thus $\Gamma \vdash \beta.\mathbf{t}:T$. Note that $\beta = \alpha + \beta$.

$\alpha \neq 0, \beta = 0$: Analogous to the previous case.

$\alpha, \beta \neq 0, \delta = 0$: Then $\gamma = \sigma$, so $\Gamma \vdash \beta.\mathbf{t}:T \equiv \beta.\varphi.U'$, then by Lemma 4.12, $\Gamma \vdash \mathbf{t}:\varphi.U'$. In addition, $\Gamma \vdash \alpha.\mathbf{t}:0.V \equiv \bar{0} \equiv \alpha.\bar{0}$, so by Lemma 4.12, $\Gamma \vdash \mathbf{t}:\bar{0}$. Then by Corollary 4.21, $\varphi.U' \equiv \bar{0}$, so by definition of \equiv and Lemma 4.3, $\varphi = 0$ and then $\gamma = 0$, so $\sigma = 0$ and $T \equiv \bar{0}$. Using the rule s_I one can derive $\Gamma \vdash (\alpha + \beta).\mathbf{t}:(\alpha + \beta).\bar{0} \equiv \bar{0} \equiv T$.

$\alpha, \beta \neq 0, \gamma = 0$: Analogous to the previous case.

$\alpha, \beta, \gamma, \delta \neq 0$: Then $\phi, \varphi \neq 0$, so by Lemma 4.3, $V \equiv U \equiv U'$. Then $\Gamma \vdash \alpha.\mathbf{t}:\alpha.\phi.V$ and $\Gamma \vdash \beta.\mathbf{t}:\beta.\varphi.V$. Hence by Lemma 4.12, $\Gamma \vdash \mathbf{t}:\phi.V$ and $\Gamma \vdash \mathbf{t}:\varphi.V$. Then by Theorem 4.20, $\phi = \varphi$ and then by the rule s_I , one can derive $\Gamma \vdash (\alpha + \beta).\mathbf{t}:(\alpha + \beta).\phi.V$. Note that $(\alpha + \beta).\phi.V \equiv ((\alpha + \beta) \times \phi).V = (\alpha \times \phi + \beta \times \varphi).V = (\delta + \gamma).V = \sigma.V \equiv T$. \square

4.2. Subject reduction proof. Now we are able to prove the subject reduction property (Theorem 4.1).

Proof. We proceed by checking that every reduction rule preserves the type. We give two cases as example, the full proof can be found in Appendix A.3.

rule $(\mathbf{t} + \mathbf{r})\mathbf{u} \rightarrow \mathbf{t}\mathbf{u} + \mathbf{r}\mathbf{u}$: Let $\Gamma \vdash (\mathbf{t} + \mathbf{r})\mathbf{u}:T \equiv 1.T$. Then, by Lemma 4.9, there exist α, β, U, V and $T' \succeq_{\nu, \Gamma}^{(\mathbf{t} + \mathbf{r})\mathbf{u}} T$ such that $\Gamma \vdash \mathbf{u}:\alpha.U$ and $\Gamma \vdash \mathbf{t} + \mathbf{r}:\beta.(U \rightarrow T') \equiv 1.\beta.(U \rightarrow T')$ with $\alpha \times \beta = 1$. Then by Lemma 4.14, there exist δ and γ such that $\Gamma \vdash \mathbf{t}:\delta.(U \rightarrow T')$ and $\Gamma \vdash \mathbf{r}:\gamma.(U \rightarrow T')$ with $\delta + \gamma = \beta$. Then by the rule \rightarrow_E , $\Gamma \vdash \mathbf{t}\mathbf{u}:(\delta \times \alpha).T'$ and

$\Gamma \vdash \mathbf{ru} : (\gamma \times \alpha).T'$. Using the rule $+_I$, one can derive $\Gamma \vdash \mathbf{tu} + \mathbf{ru} : (\delta \times \alpha + \gamma \times \alpha).T' = 1.T' \equiv T'$ and by Lemma 4.6, $T' \succ_{\mathcal{V}, \Gamma}^{\mathbf{tu} + \mathbf{ru}} T$.

rule $(\lambda x \mathbf{t})\mathbf{b} \rightarrow \mathbf{t}[\mathbf{b}/x]$: Let $\Gamma \vdash (\lambda x \mathbf{t})\mathbf{b} : T$. By the rule \equiv , $\Gamma \vdash (\lambda x \mathbf{t})\mathbf{b} : 1.T$, so by Lemma 4.9, there exist $\alpha, \beta, U, \mathcal{V}, T' \succeq_{\mathcal{V}, \Gamma}^{(\lambda x \mathbf{t})\mathbf{b}} T$ such that $\Gamma \vdash \lambda x \mathbf{t} : \beta.(U \rightarrow T')$ and $\Gamma \vdash \mathbf{b} : \alpha.U$ with $\alpha \times \beta = 1$. Since \mathbf{b} is a base term, by Lemmas 4.18 and 4.3, $\alpha = 1$ and so $\beta = 1$. Then by Lemma 4.11, there exist V, R and \mathcal{V}' with $V \rightarrow R \succeq_{\mathcal{V}', \Gamma}^{\lambda x \mathbf{t}} U \rightarrow T'$. Then by Lemma 4.8, there exist \vec{W} and $\vec{X} \in \mathcal{V}'$ such that $U \equiv V[\vec{W}/\vec{X}]$, $T' \equiv R[\vec{W}/\vec{X}]$ and since $\vec{X} \notin \text{FV}(\Gamma)$, $\Gamma = \Gamma[\vec{W}/\vec{X}]$. So, by Lemma 4.16(1), $\Gamma, x : U \vdash \mathbf{t} : T'$. Thus, by Lemma 4.16(2), $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T'$, from which, by Lemma 4.6, one obtain $\Gamma \vdash \mathbf{t}[\mathbf{b}/x] : T$. \square

5. STRONG NORMALISATION, SIMPLIFIED REDUCTION RULES AND CONFLUENCE

The *Scalar* type system is now proved to have the strong normalisation property, *i.e.* every typable term is strongly normalising, it cannot reduce forever. In order to show this we first set up another type system, which simply ‘forgets’ the scalars. Hence this simpler type system is just a straightforward extension of System F , called here $\lambda 2^{la}$ (Definition 5.1). In the literature surrounding not *Lineal* but its cousin, the algebraic λ -calculus, one finds such a System F in [Ehr10], which extends the simply typed algebraic λ -calculus of [Vau07, Vau09] – our $\lambda 2^{la}$ is very similar. Secondly we prove strong normalisation for it (Theorem 5.9). Thirdly we show that every term which has a type in *Scalar* has a type in $\lambda 2^{la}$ (Lemma 5.11), which entails strong normalisation in *Scalar* (Theorem 5.12).

This strong normalisation proof constitutes the second main technical contribution of the paper. The confluence of a simplified version of *Lineal* is derived from it.

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

Definition 5.1. The type grammar of $\lambda 2^{la}$ is the following:

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

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

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

Example 5.2. In *Scalar* it is possible to derive $\vdash 3.\lambda x(x+x) : 3.(X \rightarrow 2.X)$, while in $\lambda 2^{la}$ we have $\vdash 3.\lambda x(x+x) : X \rightarrow X$.

In order to prove strong normalisation we extend the proof for $\lambda 2$. The standard method was invented by Tait [Tai67] for simply typed λ -calculus and generalized to System F by Girard [Gir72]. Our presentation follows [Bar92, sec. 4.3] (who uses an analogous to Krivine’s saturated sets [Kri90, thm. 8.9]). The following definitions are taken from this reference – with slight modifications to handle the extra $\lambda 2^{la}$ rules.

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

Definition 5.3. $\text{SN} = \{\mathbf{t} \in \Lambda \mid \mathbf{t} \text{ is strongly normalising}\}$.

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

Definition 5.4.

- (1) A subset $X \subseteq \text{SN}$ is called *saturated* if
 - (a) $\mathbf{0} \in X$;
 - (b) for each x and $\vec{\mathbf{t}} \in \text{SN}$, $x\vec{\mathbf{t}} \in X$;
 - (c) if $\mathbf{b} \in \text{SN}$ and $\mathbf{t}[\mathbf{b}/x]\vec{\mathbf{r}} \in X$ then $(\lambda x \mathbf{t})\mathbf{b}\vec{\mathbf{r}} \in X$;
 - (d) if (for each $i \in I$, $\mathbf{t}_i\vec{\mathbf{r}} \in X$) then $(\sum_{i \in I} \mathbf{t}_i)\vec{\mathbf{r}} \in X$;
 - (e) if (for each $i \in I$, $\mathbf{u}\mathbf{t}_i\vec{\mathbf{r}} \in X$) then $\mathbf{u}(\sum_{i \in I} \mathbf{t}_i)\vec{\mathbf{r}} \in X$;
 - (f) for each $\alpha \in \mathcal{S}$, $\mathbf{t} \in X$ if and only if $\alpha.\mathbf{t} \in X$;
 - (g) $\alpha.\mathbf{t}_1\mathbf{t}_2 \dots \mathbf{t}_n \in X$ if and only if $\mathbf{t}_1\mathbf{t}_2 \dots (\alpha.\mathbf{t}_k) \dots \mathbf{t}_n \in X$ ($1 \leq k \leq n$);
 - (h) for each $\vec{\mathbf{t}} \in \text{SN}$, $\mathbf{0}\vec{\mathbf{t}} \in X$;
 - (i) for each $\mathbf{t}, \vec{\mathbf{u}} \in \text{SN}$, $\mathbf{t}\mathbf{0}\vec{\mathbf{u}} \in X$.
 where I is a finite set of indices.
- (2) $\text{SAT} = \{X \subseteq \Lambda \mid X \text{ is saturated}\}$

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

Definition 5.5.

- (1) A *valuation* in SAT is a map $\xi: \mathbb{V} \rightarrow \text{SAT}$, where \mathbb{V} is the set of type variables.
- (2) For any $A, B \subseteq \Lambda$, we define $A \Rightarrow B = \{\mathbf{t} \in \Lambda \mid \text{for each } \mathbf{r} \in A, \mathbf{t}\mathbf{r} \in B\}$.
- (3) Given a valuation ξ in SAT, we define for every $T \in \mathbb{T}(\lambda 2^{la})$ a set $\llbracket T \rrbracket_\xi \subseteq \Lambda$ as follows:

$$\begin{aligned} \llbracket X \rrbracket_\xi &= \xi(X), \text{ where } X \in \mathbb{V} \\ \llbracket A \rightarrow B \rrbracket_\xi &= \llbracket A \rrbracket_\xi \Rightarrow \llbracket B \rrbracket_\xi \\ \llbracket \forall X.A \rrbracket_\xi &= \bigcap_{Y \in \text{SAT}} \llbracket A \rrbracket_{\xi(X:=Y)} \end{aligned}$$

Lemma 5.6.

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

Proof. cf. Appendix A.4. □

Just like in Definition 5.5, we define another valuation, this time from term variables to base terms. We use it to check what happens when we change every free variable of a term for any other base term. The basic idea is the following: we define $\rho, \xi \models \mathbf{t}: A$ to be the property of changing every free term variable in \mathbf{t} for another term with the help of the valuation ρ (a base term, since term variables only run over base terms) and still having the resulting term in the set $\llbracket A \rrbracket_\xi$. So, we define $\Gamma \models \mathbf{t}: A$ to be the same property, when the property holds for every pair in Γ and for every valuations ρ and ξ .

This is formalised in the following definition (Definition 5.7) and with this definition, we prove that if a term has a type in a valid context, then the property above holds

(Theorem 5.8), which yields the strong normalisation theorem (Theorem 5.9) via the concept of *saturated set* (because *saturated sets* are subsets of SN).

Definition 5.7.

- A *valuation* in Λ_b is a map $\rho: V \rightarrow \Lambda_b$, where V is the set of term variables and $\Lambda_b = \{\mathbf{b} \in \Lambda \mid \mathbf{b} \text{ is a base term}\}$.
- Let ρ be a valuation in Λ_b . Then $\llbracket \mathbf{t} \rrbracket_\rho = \mathbf{t}[x_1 := \rho(x_1), \dots, x_n := \rho(x_n)]$, where $\vec{x} = x_1, \dots, x_n$ is the set of free variables in \mathbf{t} .
- Let ρ be a valuation in Λ_b and ξ a valuation in SAT. Then
 - ρ, ξ *satisfies* $\mathbf{t}: A$, notation $\rho, \xi \models \mathbf{t}: A$, if and only if $\llbracket \mathbf{t} \rrbracket_\rho \in \llbracket A \rrbracket_\xi$.
 - $\rho, \xi \models \Gamma$ if and only if $\rho, \xi \models x: A$ for all $x: A$ in Γ
 - $\Gamma \models \mathbf{t}: A$ if and only if for every ρ, ξ , $\rho, \xi \models \Gamma$ implies $\rho, \xi \models \mathbf{t}: A$.

Theorem 5.8 (Soundness). *If $\Gamma \Vdash \mathbf{t}: A$ then $\Gamma \models \mathbf{t}: A$.*

Proof. We proceed by induction on the derivation of $\Gamma \Vdash \mathbf{t}: T$. We show one case as an example. The full proof is in Appendix A.5.

$$\frac{\Gamma \Vdash \mathbf{t}: A \rightarrow B \quad \Gamma \Vdash \mathbf{r}: A}{\Gamma \Vdash \mathbf{tr}: B} \rightarrow_E^{\triangleleft}$$

By the induction hypothesis, $\Gamma \models \mathbf{t}: A \rightarrow B$ and $\Gamma \models \mathbf{r}: A$. Assume $\rho, \xi \models \Gamma$ in order to show $\rho, \xi \models \mathbf{tr}: B$. Then $\rho, \xi \models \mathbf{t}: A \rightarrow B$, i.e. $\llbracket \mathbf{t} \rrbracket_\rho \in \llbracket A \rightarrow B \rrbracket_\xi = \llbracket A \rrbracket_\xi \Rightarrow \llbracket B \rrbracket_\xi$ and $\llbracket \mathbf{r} \rrbracket_\rho \in \llbracket A \rrbracket_\xi$. Then $\llbracket \mathbf{tr} \rrbracket_\rho = \llbracket \mathbf{t} \rrbracket_\rho \llbracket \mathbf{r} \rrbracket_\rho \in \llbracket B \rrbracket_\xi$, so $\rho, \xi \models \mathbf{tr}: B$. \square

Theorem 5.9 (Strong normalisation for $\lambda 2^{la}$). *If $\Gamma \Vdash \mathbf{t}: A$, then \mathbf{t} is strongly normalising.*

Proof. Let $\Gamma \Vdash \mathbf{t}: A$. Then by Theorem 5.8, $\Gamma \models \mathbf{t}: A$. Define $\rho_0(x) = x$ for all x and let ξ be a valuation in SAT. Then $\rho_0, \xi \models \Gamma$ (i.e. for all $(x: B) \in \Gamma$, $\rho_0, \xi \models x: B$ since $x \in \llbracket B \rrbracket_\xi$ holds because $\llbracket B \rrbracket_\xi$ is saturated). Therefore $\rho_0, \xi \models \mathbf{t}: A$, hence $\mathbf{t} = \llbracket \mathbf{t} \rrbracket_{\rho_0} \in \llbracket A \rrbracket_\xi \subseteq \text{SN}$. \square

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

Definition 5.10. Let $(\cdot)^\natural$ be the following mapping from types in *Scalar* to $\mathbb{T}(\lambda 2^{la})$:

$$\bullet X^\natural = X \quad \bullet (\forall X.T)^\natural = \forall X.T^\natural \quad \bullet (U \rightarrow T)^\natural = U^\natural \rightarrow T^\natural \quad \bullet (\alpha.T)^\natural = T^\natural \quad \bullet \bar{0}^\natural = A$$

where A is an arbitrary fixed type in $\mathbb{T}(\lambda 2^{la})$.

We also extend this definition to contexts: $\Gamma^\natural = \{(x: T^\natural) \mid (x: T) \in \Gamma\}$.

The following lemma ensures that if it is possible to give a type to a term in *Scalar* then it is possible to give to the term the mapped type in $\lambda 2^{la}$. Example 5.2 may help to see this.

Lemma 5.11 (Correspondence with $\lambda 2^{la}$). *If $\Gamma \vdash \mathbf{t}: T$ then $\Gamma^\natural \Vdash \mathbf{t}: T^\natural$.*

Proof. We proceed by induction on the derivation of $\Gamma \vdash \mathbf{t}: T$. We show one case as example. The full proof is in Appendix A.6.

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

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

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

Theorem 5.12 (Strong normalisation). *If $\Gamma \vdash \mathbf{t}: T$ then \mathbf{t} is strongly normalising.*

Proof. By Lemma 5.11, $\Gamma^\natural \Vdash \mathbf{t}: T^\natural$, then by Theorem 5.9, \mathbf{t} is strongly normalising. \square

Taking up again the example of Section 2, terms like \mathbf{Y} are simply not allowed in this typed setting, since Theorem 5.12 ensures that all the typable terms have a normal form. So we do not have infinities and hence the intuitive reasons for having restrictions (*) on the *Factorisation rules* of the linear-algebraic calculus (*cf.* the reduction rules in Section 2) have now vanished. If we drop them, the example becomes as follows.

Example 5.13. Consider some arbitrary typable and hence normalising term \mathbf{t} . Then $\alpha.\mathbf{t} - \alpha.\mathbf{t}$ can be reduced by a factorisation rule into $(\alpha - \alpha).\mathbf{t}$. This reduces in one step to $\mathbf{0}$, without the need to reduce \mathbf{t} .

It turns out that, in general, for typable terms we can indeed drop the restrictions (*) and (**) without breaking the confluence of *Lineal*. These restrictions were there only due to the impossibility of checking for the normalisation property in the untyped setting. Having this property, the confluence of the system follows naturally.

Theorem 5.14 (Confluence). *Let \mathbf{t} be a term of *Lineal*, as it appears in Figure 1, but without restrictions (*) and (**). If \mathbf{t} is typable in *Scalar*, $\mathbf{t} \rightarrow^* \mathbf{u}$ and $\mathbf{t} \rightarrow^* \mathbf{r}$, then there exists a term \mathbf{t}' such that $\mathbf{u} \rightarrow^* \mathbf{t}'$ and $\mathbf{r} \rightarrow^* \mathbf{t}'$.*

Proof. First let us introduce some notation. Let \rightarrow_β be a β -reduction and \rightarrow_a any reduction from Figure 1 but the β -reduction, without restrictions (*) and (**). Moreover, \rightarrow_β^* and \rightarrow_a^* are the application of zero or more of such rules, as usual.

The proof follows in several steps.

- (1) First we prove local confluence for the algebraic fragment, *i.e.* if $\mathbf{t} \rightarrow_a \mathbf{u}$ and $\mathbf{t} \rightarrow_a \mathbf{r}$, then there exists a term \mathbf{t}' such that $\mathbf{u} \rightarrow_a^* \mathbf{t}'$ and $\mathbf{r} \rightarrow_a^* \mathbf{t}'$.
- (2) Then we prove local confluence for the β -reduction, *i.e.* if $\mathbf{t} \rightarrow_\beta \mathbf{u}$ and $\mathbf{t} \rightarrow_\beta \mathbf{r}$, then there exists a term \mathbf{t}' such that $\mathbf{u} \rightarrow_\beta^* \mathbf{t}'$ and $\mathbf{r} \rightarrow_\beta^* \mathbf{t}'$.
- (3) Finally, we prove that algebraic rules and β -reduction commutes, *i.e.* if $\mathbf{t} \rightarrow_a \mathbf{u}$ and $\mathbf{t} \rightarrow_\beta \mathbf{r}$, then there exists a term \mathbf{t}' such that $\mathbf{u} \rightarrow^* \mathbf{t}'$ and $\mathbf{r} \rightarrow^* \mathbf{t}'$, where \rightarrow^* is a reduction sequence of zero or more steps involving any rules.

This prove the local confluence of the system. The well known Newman's lemma states that local confluence plus strong normalisation implies confluence (*cf.* for example [TeR03, thm. 1.2.1]).

This is how the proofs of the steps look like.

- (1) Valiron did a semi-automatized proof in the interactive theorem prover Coq [Coq09]. The interested reader can find the proof in [Val11]. An easier-to-read explanation of it appears in [ADCV11].
- (2) The local confluence of the β -reduction follows from [Dou92] and a trivial extension of the confluence of lambda-calculus.
- (3) This proof goes by structural induction. *cf.* Appendix A.7. □

Note that the proofs of subject reduction (Theorem 4.1) and strong normalisation (Theorem 5.12) have been done in the general case, without restrictions (*) and (**), so they are still valid for the simplified calculus.

Having dropped restrictions (*) and (**) is an important simplification of the linear-algebraic λ -calculus, which becomes really just an oriented version of the axioms of vector spaces [AD04] together with the call-by-value strategy (*i.e.* restriction (***) remains, of course, to make all functions remain linear in their arguments, in the sense of linear-algebra).

6. BARYCENTRIC λ -CALCULUS

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

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

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

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

Although this seems feasible, we do not develop a full-blown barycentric higher-order λ -calculus and associated properties in this paper. We just show that the *Scalar* type system accomplishes part of the job by checking for the barycentric property, *i.e.* checking that the normal form of a term has amplitudes summing to one. A barycentric λ -calculus fragment of the algebraic λ -calculus has already been studied for its own sake [Tas09], however in this work the calculus was endowed with a simple type system, not one that would recognize barycentric terms amongst other terms.

To this end let us define a type system with the rules and grammar of *Scalar*, but where the valid types are the classic ones (*i.e.* types exempt of any scalar, which we have referred to as $\mathbb{T}(\lambda 2^{la})$ in Definition 5.1), whilst all the other types are just intermediate types:

Definition 6.1. We define the type system \mathcal{B} for the barycentric calculus to be the *Scalar* type system with the following restrictions:

- $\mathcal{S} = \mathbb{R}$,
- Contexts are sets of tuples $(x:A)$, with $A \in \mathbb{T}(\lambda 2^{la})$,
- Type variables run over $\mathbb{T}(\lambda 2^{la})$ instead of unit types, *i.e.* the rule \forall_E accepts only $A \in \mathbb{T}(\lambda 2^{la})$,
- The final sequent has type in $\mathbb{T}(\lambda 2^{la})$.

Note that the type derivation may have, as intermediate sequents, sequents that do not have their type in $\mathbb{T}(\lambda 2^{la})$ but in *Scalar*.

Example 6.2. The sequent $y: X \vdash 2.(\lambda x \frac{1}{2}.x)y: X$ is valid in \mathcal{B} , even if intermediate sequents do not have their types restricted to $\mathbb{T}(\lambda 2^{la})$:

$$\frac{\frac{\frac{\frac{}{y: X, x: X \vdash x: X} ax}{y: X, x: X \vdash \frac{1}{2}.x: \frac{1}{2}.X} s_I} y: X \vdash \lambda x \frac{1}{2}.x: X \rightarrow \frac{1}{2}.X} \rightarrow_I \quad \frac{}{y: X \vdash y: X} ax}{\frac{y: X \vdash (\lambda x \frac{1}{2}.x)y: \frac{1}{2}.X}{y: X \vdash 2.(\lambda x \frac{1}{2}.x)y: 2.\frac{1}{2}.X} s_I} \rightarrow_E} 2.\frac{1}{2}.X \equiv X \equiv \frac{}{y: X \vdash 2.(\lambda x \frac{1}{2}.x)y: X} \equiv$$

In order to show that this type system does the job, let us define the weight function:

Definition 6.3. Let $\omega : \Lambda \rightarrow \mathbb{R}$ be a function defined inductively by:

$$\begin{aligned} \omega(\mathbf{0}) &= 0 & \omega(\mathbf{b}) &= 1 & \omega(\mathbf{t}_1 + \mathbf{t}_2) &= \omega(\mathbf{t}_1) + \omega(\mathbf{t}_2) \\ \omega(\mathbf{t}_1 \mathbf{t}_2) &= \omega(\mathbf{t}_1) \times \omega(\mathbf{t}_2) & \omega(\alpha \cdot \mathbf{t}) &= \alpha \times \omega(\mathbf{t}) \end{aligned}$$

where \mathbf{b} is a base term.

Example 6.4. Continuing Example 6.2, note that $2.(\lambda x \frac{1}{2}.x)y \rightarrow^* y$. The weight function cannot check the barycentric property on the first term (indeed $\omega(2.(\lambda x \frac{1}{2}.x)y) = 2$), even when this term reduces to a barycentric term. For this matters, we use the type system \mathcal{B} to check this property, as stated by the following theorem.

Theorem 6.5 (Normal-form of terms in \mathcal{B} have weight 1). *Let $\Gamma \vdash \mathbf{t} : A$ with $A \in \mathbb{T}(\lambda 2^{la})$, then $\omega(\text{nf}(\mathbf{t})) = 1$.*

We need the following three lemmas in order to prove this theorem. The first lemma says that an application in normal form can only happen when a variable is in head position.

Lemma 6.6. *If $\mathbf{t}_1 \mathbf{t}_2$ is in normal form, then $\mathbf{t}_1 = x\vec{r}$.*

Proof. Structural induction on \mathbf{t}_1 . *cf.* Appendix ??.

□

The following lemma says that if there is an application in normal form (which by the previous lemma has to have the form $x\vec{r}$), then it cannot be typed in the type system \mathcal{B} with a type having a scalar at the right of an arrow, the only possible scalar have to be in the head position of the type.

Lemma 6.7. $\Gamma \vdash x\vec{r} : T$ and $x\vec{r}$ is in normal form, then there exist $A \in \mathbb{T}(\lambda 2^{la})$ and $\alpha \in \mathcal{S}$ such that $T \equiv \alpha.A$.

Proof. Induction on the derivation of $\Gamma \vdash x\vec{r} : T$. *cf.* Appendix A.8.

□

The last lemma shows that scalars cannot be removed by subsumption.

Lemma 6.8. *Let $A \in \mathbb{T}(\lambda 2^{la})$. For any $T, \mathcal{V}, \Gamma, \mathbf{t}$, if $\alpha.T \succeq_{\mathcal{V}, \Gamma}^{\mathbf{t}} \alpha.A$, then $T \equiv B$ with $B \in \mathbb{T}(\lambda 2^{la})$.*

Proof. Case analysis. *cf.* Appendix A.9

□

Using these lemmas, the proof of Theorem 6.5 goes as follows.

Proof. Instead, we prove the more general case: If $\Gamma \vdash \mathbf{t} : \alpha.A$ then $\omega(\text{nf}(\mathbf{t})) = \alpha$, by structural induction on $\text{nf}(\mathbf{t})$. We take $\Gamma \vdash \text{nf}(\mathbf{t}) : \alpha.A$, which is true by Theorem 4.1. We show two cases as an example. The full proof is in Appendix A.10.

- Case $\text{nf}(\mathbf{t}) = \gamma \cdot \mathbf{t}'$. Then $\omega(\text{nf}(\mathbf{t})) = \gamma \times \omega(\mathbf{t}')$. By Lemma 4.17, there exist U and δ such that $\alpha.A \equiv \gamma \cdot \delta.U$. Note that $\mathbb{T}(\lambda 2^{la})$ is included in the set of unit types, so by Lemma 4.3, $\alpha = \gamma \times \delta$. We consider two cases:
 - $\alpha = 0$: Then either $\gamma = 0$, and so $\omega(\gamma \cdot \mathbf{t}') = 0 \times \omega(\mathbf{t}') = 0$, or $\gamma \neq 0$ but $\delta = 0$, and so by Lemma 4.12, $\Gamma \vdash \mathbf{t}' : 0.U \equiv 0.A$, so by the induction hypothesis $\omega(\mathbf{t}') = 0$, and then $\omega(\gamma \cdot \mathbf{t}') = \gamma \times 0 = 0$.
 - $\alpha \neq 0$: Then $A \equiv U$, so by Lemma 4.12, $\Gamma \vdash \mathbf{t}' : \delta.A$. Then by the induction hypothesis $\omega(\mathbf{t}') = \delta$. Note that $\omega(\text{nf}(\mathbf{t})) = \gamma \times \omega(\mathbf{t}') = \gamma \times \delta = \alpha$.

- Case $\text{nf}(\mathbf{t}) = \mathbf{t}_1\mathbf{t}_2$. Then $\omega(\text{nf}(\mathbf{t})) = \omega(\mathbf{t}_1) \times \omega(\mathbf{t}_2)$. By Lemma 4.9, there exist U, T, β, δ and \mathcal{V} such that $\Gamma \vdash \mathbf{t}_1 : \beta.(U \rightarrow T)$ and $\Gamma \vdash \mathbf{t}_2 : \delta.U$ with $\alpha.T \succeq_{\mathcal{V}, \Gamma}^{\mathbf{t}_1\mathbf{t}_2} \alpha.A$ and $\beta \times \delta = \alpha$. Since $A \in \mathbb{T}(\lambda 2^{la})$ by assumption, using Lemma 6.8 we have $T \equiv B$ with $B \in \mathbb{T}(\lambda 2^{la})$ and since $\mathbf{t}_1\mathbf{t}_2$ is in normal form, by Lemma 6.6, \mathbf{t}_1 is a variable applied to something else, so by Lemma 6.7, $U \rightarrow B \in \mathbb{T}(\lambda 2^{la})$, which implies that $U \in \mathbb{T}(\lambda 2^{la})$. Then by the induction hypothesis, $\omega(\mathbf{t}_1) = \beta$ and $\omega(\mathbf{t}_2) = \delta$, so $\omega(\text{nf}(\mathbf{t})) = \omega(\mathbf{t}_1) \times \omega(\mathbf{t}_2) = \beta \times \delta = \alpha$. \square

Remarks 6.9.

- As first proved in [AD08, prop. 2], closed normal terms have form

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

Thus the Theorem 6.5 entails that $\sum_{i=1}^n \alpha_i + m = 1$. Hence the type system \mathcal{B} , an easy variation of the *Scalar* type system, checks for the barycentric property, *i.e.* it checks that a given term reduces to a barycentric distribution of terms.

- A bit more elaborated example than the one given in Example 6.4. It is easy to prove that

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

But note that $\omega((2.\lambda x \lambda y (\frac{1}{4}.x + \frac{1}{4}.y))zw) = 2$, even when $(2.\lambda x \lambda y (\frac{1}{4}.x + \frac{1}{4}.y))zw \rightarrow^* \frac{1}{2}.z + \frac{1}{2}.w$, whose weight is equal to one. So, *a priori* this ω function cannot tell us that this term yields a barycentric term. However the fact that has type A in $\mathbb{T}(\lambda 2^{la})$, according to the Theorem 6.5, anticipates this result.

- One might think that unit types are just as good as barycentric types \mathcal{B} for the sake of obtaining Theorem 6.5, via a combination of Subject-reduction and the uniqueness of scalars property (*cf.* Theorem 4.20). This is not quite the case, here is a counter-example:

$$x : U \rightarrow 2.U, y : U \vdash x \frac{1}{2}.y : U \quad \text{but } \omega(x \frac{1}{2}.y) = \frac{1}{2}$$

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

7. SUMMARY OF CONTRIBUTIONS

In summary, we have defined a System F -like type system for an extension of *Lineal*, a λ -calculus which allows making arbitrary linear combinations of λ -calculus terms $\alpha.\mathbf{t} + \beta.\mathbf{u}$. The *Scalar* type system is fine-grained in that it keeps track of the ‘amount of a type’, *i.e.* the scalar of the type of a term is the sum of the scalars of the types of the contributing subterms.

Our main technical contributions were:

- A proof of the subject reduction property of the *Scalar* type system (Theorem 4.1). This came out after having proven a set of lemmas related to the equivalence relation intrinsic to the types and another set of lemmas explaining how the scalars within the types are related to the scalars within the terms. Once all of the important properties were known,

we were able to use them to decompose and recompose any term before and after applying a reduction rule, so as to show that every reduction rule preserves the types.

- A proof of the strong normalisation property of the *Scalar* type system (Theorem 5.12). The technique used to prove the strong normalisation property was by proving that such property would hold for a simpler system and then to show the correspondence between the two systems.
- A proof that under strong normalisation, most of the conditions upon the *Lineal* reduction rules can be lifted (*e.g.* allowing the factorisation not only of closed normal terms but of any term) without jeopardizing confluence, thereby simplifying the *Lineal* language.
- A proof that the *Scalar* type system can be used to check that a term has the barycentric property, *i.e.* that the amplitudes of its normal form are summing to one.

Arguably a denotational semantics approach might have led to less syntactic proofs of the properties of the type system, sustained by the guiding intuition about an underlying mathematical space. On the other hand, the complexity of the proofs in this paper is largely due to the large number of rules (16 rules plus associativity and commutativity of $+$). Moreover the issue of models of (linear-)algebraic λ -calculus is a challenging, active topic of the current research. We know of the categorical model of simply typed *Lineal* [Val10] and the finiteness space model of simply typed algebraic λ -calculus [Ehr05, Tas09]. Moreover, even if both calculi simulate each other [DCPTV10], it is not clear whether the translation applies at the level of models. Hence known models are intricate and tend not to cover the set of terms under consideration in this paper. Note also that since the models of untyped λ -calculus are uncountable, the models of (linear-)algebraic λ -calculus are likely to be vector spaces of uncountable dimensions. These are fascinating, open questions.

ACKNOWLEDGEMENT

We would like to thank to Gilles Dowek, Jonathan Grattage, Philippe Jorrand, Simon Perdrix, Barbara Petit, Frédéric Prost and Benoît Valiron for enlightening discussions. We thank also the anonymous referee for his careful reading, which enabled us to improve the level of rigour of our proofs.

REFERENCES

- [AD04] Pablo Arrighi and Gilles Dowek. A computational definition of the notion of vectorial space. In Narciso Martí-Oliet, editor, *Proceedings of WRLA-2004*, volume 117 of *Electronic Notes in Theoretical Computer Science*, pages 249–261. Elsevier, 2004.
- [AD08] Pablo Arrighi and Gilles Dowek. Linear-algebraic lambda-calculus: higher-order, encodings, and confluence. In Andrei Voronkov, editor, *Proceedings of RTA-2008*, volume 5117 of *Lecture Notes in Computer Science*, pages 17–31. Springer, 2008.
- [ADC11] Pablo Arrighi and Alejandro Díaz-Caro. Scalar System F for linear-algebraic λ -calculus: Towards a quantum physical logic. In Bob Coecke, Prakash Panangaden, and Peter Selinger, editors, *Proceedings of QPL-2009*, volume 270/2 of *Electronic Notes in Theoretical Computer Science*, pages 219–229. Elsevier, 2011.
- [ADCV11] Pablo Arrighi, Alejandro Díaz-Caro, and Benoît Valiron. A type system for the vectorial aspects of the linear-algebraic lambda-calculus. In *Proceedings of the 7th International Workshop on Developments of Computational Methods (DCM 2011)*, Zurich, Switzerland, 2011. To appear in EPTCS. Draft at <http://www.diaz-carro.info/vectorial.pdf>.
- [AMR09] Ariel Arbiser, Alexandre Miquel, and Alejandro Ríos. The λ -calculus with constructors: Syntax, confluence and separation. *Journal of Functional Programming*, 19(5):581–631, 2009.

- [Bar92] Henk P. Barendregt. *Lambda calculi with types*. Handbook of Logic in Computer Science: Volume 2. Oxford University Press, 1992.
- [BH03] Olivier Bournez and Mathieu Hoyrup. Rewriting logic and probabilities. In Robert Nieuwenhuis, editor, *Proceedings of RTA-2003*, volume 2706 of *Lecture Notes in Computer Science*, pages 61–75. Springer, 2003.
- [Bou94] Gérard Boudol. Lambda-calculi for (strict) parallel functions. *Information and Computation*, 108(1):51–127, 1994.
- [CKL01] Horatiu Cirstea, Claude Kirchner, and Luigi Liquori. The Rho Cube. In Furio Honsell and Marino Miculan, editors, *Proceedings of FOSSACS-2001*, volume 2030 of *Lecture Notes in Computer Science*, pages 168–183. Springer, 2001.
- [Coq09] Coq Dev. Team. *The Coq proof assistant reference manual*. INRIA, 8.2 edition, February 2009.
- [DCPTV10] Alejandro Díaz-Caro, Simon Perdrix, Christine Tasson, and Benoît Valiron. Equivalence of algebraic λ -calculi. In *Informal Proceedings of the 5th International Workshop on Higher-Order Rewriting, HOR-2010*, pages 6–11, Edinburgh, UK, July 14, 2010.
- [DHW05] Alessandra Di Pierro, Chris Hankin, and Herbert Wiklicky. Probabilistic λ -calculus and quantitative program analysis. *Journal of Logic and Computation*, 15(2):159–179, 2005.
- [Dou92] Daniel J. Dougherty. Adding algebraic rewriting to the untyped lambda calculus. *Information and Computation*, 101(2):251–267, 1992.
- [dP95] Ugo de’Liguoro and Adolfo Piperno. Non deterministic extensions of untyped λ -calculus. *Information and Computation*, 122(2):149–177, 1995.
- [DZ11] Ugo Dal Lago and Margherita Zorzi. Probabilistic operational semantics for the lambda calculus. Preprint at [arXiv:1104.0195](https://arxiv.org/abs/1104.0195), April 2011.
- [Ehr05] Thomas Ehrhard. Finiteness spaces. *Mathematical Structures in Computer Science*, 15(4):615–646, 2005.
- [Ehr10] Thomas Ehrhard. A finiteness structure on resource terms. In *Proceedings of LICS-2010*, pages 402–410. IEEE Computer Society, 2010.
- [ER03] Thomas Ehrhard and Laurent Regnier. The differential lambda-calculus. *Theoretical Computer Science*, 309(1):1–41, 2003.
- [Gir71] Jean-Yves Girard. Une extension de l’interprétation de Gödel à l’analyse, et son application à l’élimination des coupures dans l’analyse et la théorie des types. In Jens Erik Fenstad, editor, *Proceedings of the 2nd Scandinavian Logic Symposium*, volume 63 of *Studies in Logic and the Foundations of Mathematics*, pages 63–92. North-Holland, 1971.
- [Gir72] Jean-Yves Girard. *Interprétation fonctionnelle et élimination des coupures dans l’arithmétique d’ordre supérieure*. PhD thesis, Université Paris Diderot, Paris, France, 1972.
- [Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on Theory of computing, STOC-96*, pages 212–219. ACM, 1996.
- [HP00] Oltea Mihaela Herescu and Catuscia Palamidessi. Probabilistic asynchronous π -calculus. In Jerzy Tiuryn, editor, *Proceedings of FOSSACS-2000*, volume 1784 of *Lecture Notes in Computer Science*, pages 146–160. Springer, 2000.
- [JK86] Jean-Pierre Jouannaud and Hélène Kirchner. Completion of a set of rules modulo a set of equations. *SIAM Journal on Computing*, 15(4):1155–1194, 1986.
- [Kri90] Jean-Louis Krivine. *Lambda-calcul: types et modèles*. Études et recherches en informatique. Masson, 1990.
- [LBR03] Didier Le Botlan and Didier Rémy. ML^F : raising ML to the power of System F. In *Proceedings of the eighth ACM SIGPLAN international conference on Functional programming, ICFP’03*, pages 27–38, New York, NY, USA, 2003. ACM.
- [Pet09] Barbara Petit. A polymorphic type system for the lambda-calculus with constructors. In Pierre-Louis Curien, editor, *Proceedings of TLCA-2009*, volume 5608 of *Lecture Notes in Computer Science*, pages 234–248. Springer, 2009.
- [PS81] Gerald E. Peterson and Mark E. Stickel. Complete set of reductions for some equational theories. *Journal of the ACM*, 28(2):233–264, 1981.
- [Rey74] John C. Reynolds. Towards a theory of type structure. In Bernard Robinet, editor, *Programming Symposium: Proceedings of the Colloque sur la Programmation*, volume 19 of *Lecture Notes in Computer Science*, pages 408–425. Springer, 1974.

- [Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.
- [Tai67] William W. Tait. Intensional interpretations of functionals of finite type I. *The Journal of Symbolic Logic*, 32(2):198–212, 1967.
- [Tas09] Christine Tasson. Algebraic totality, towards completeness. In Pierre-Louis Curien, editor, *Proceedings of TLCA-2009*, volume 5608 of *Lecture Notes in Computer Science*, pages 325–340. Springer, 2009.
- [TeR03] TeReSe. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.
- [Val10] Benoît Valiron. Semantics of a typed algebraic lambda-calculus. In S. Barry Cooper, Prakash Panangaden, and Elham Kashefi, editors, *Proceedings DCM-2010*, volume 26 of *Electronic Proceedings in Theoretical Computer Science*, pages 147–158. Open Publishing Association, 2010.
- [Val11] Benoît Valiron. Coq proof. www.monoidal.net/vectorial-lvec-coqproof.tar.bz2, 2011.
- [Vau07] Lionel Vaux. On linear combinations of lambda-terms. In Franz Baader, editor, *Proceedings of RTA-2007*, volume 4533 of *Lecture Notes in Computer Science*, pages 374–388. Springer, 2007.
- [Vau09] Lionel Vaux. The algebraic lambda calculus. *Mathematical Structures in Computer Science*, 19(5):1029–1059, 2009.

APPENDIX A. OMITTED PROOFS

A.1. Proof of Lemma 4.8. A map $(\cdot)^\circ$ from types of *Scalar* to $\mathbb{T}(\lambda 2^{la})$ is defined by

$$X^\circ = X \quad (U \rightarrow T)^\circ = U \rightarrow T \quad (\forall X.T)^\circ = T^\circ \quad (\alpha.T)^\circ = \alpha.T^\circ \quad (\bar{0})^\circ = \bar{0}$$

We need three intermediate results.

- (1) If $T \equiv R$, then $T^\circ \equiv R^\circ$.
- (2) For any types T and U , exists V such that $(T[U/X])^\circ \equiv T^\circ[V/X]$
- (3) For any types T, R , set of type variables \mathcal{V} , context Γ and term \mathbf{t} , if $T \succeq_{\mathcal{V}, \Gamma}^{\mathbf{t}} R$ then there exists \vec{U} and $\vec{X} \in \mathcal{V}$ such that $R^\circ \equiv T^\circ[\vec{U}/\vec{X}]$

Proofs.

- (1) By induction on the equivalence rules. We only show the basic cases since the inductive step, given by the context where the equivalence is applied, is trivial.

$$\alpha.\bar{0} \equiv \bar{0}: (\alpha.\bar{0})^\circ = \alpha.\bar{0}^\circ = \alpha.\bar{0} \equiv \bar{0} = \bar{0}^\circ.$$

$$0.T \equiv \bar{0}: (0.T)^\circ = 0.T^\circ \equiv \bar{0} = \bar{0}^\circ.$$

$$1.T \equiv T: (1.T)^\circ = 1.T^\circ \equiv T^\circ.$$

$$\alpha.(\beta.T) \equiv (\alpha \times \beta).T: (\alpha.(\beta.T))^\circ = \alpha.(\beta.T)^\circ \equiv (\alpha \times \beta).T^\circ = ((\alpha \times \beta).T)^\circ.$$

$$\forall X.\alpha.T \equiv \alpha.\forall X.T: (\forall X.\alpha.T)^\circ = (\alpha.T)^\circ = \alpha.T^\circ = \alpha.(\forall X.T)^\circ = (\alpha.\forall X.T)^\circ.$$

- (2) Structural induction on T

$$T = X: \text{ then } (X[U/X])^\circ = U^\circ = X[U^\circ/X] = X^\circ[U^\circ/X].$$

$$T = Y: \text{ then } (Y[U/X])^\circ = Y = Y^\circ[U/X].$$

$$T = V \rightarrow R: \text{ then } ((V \rightarrow R)[U/X])^\circ = (V[U/X] \rightarrow R[U/X])^\circ = V[U/X] \rightarrow R[U/X] = (V \rightarrow R)[U/X] = (V \rightarrow R)^\circ[U/X].$$

$$T = \forall Y.R: \text{ then } ((\forall Y.R)[U/X])^\circ = (\forall Y.R[U/X])^\circ = (R[U/X])^\circ, \text{ which, by the induction hypothesis, is } \equiv \text{ to } R^\circ[V/X] = (\forall Y.R)^\circ[V/X].$$

$$T = \bar{0}: \text{ analogous to } T \equiv Y.$$

$$T = \alpha.R: \text{ then } (\alpha.R[U/X])^\circ = \alpha.(R[U/X])^\circ, \text{ which, by the induction hypothesis, is } \equiv \text{ to } \alpha.(R^\circ)[V/X] = (\alpha.R)^\circ[V/X].$$

- (3) It suffices to show this for $T \succ_{X, \Gamma}^{\mathbf{t}} R$.

Case 1: $R \equiv \forall X.T$. Then $R^\circ \equiv T^\circ$.

Case 2: $T \equiv \forall X.S$ and $R \equiv S[U/X]$ then by the intermediate results 1 and 2, one has $R^\circ \equiv (S[U/X])^\circ \equiv S^\circ[V/X] \equiv T^\circ[V/X]$ for some V .

Proof of the lemma. $U \rightarrow T \equiv (U \rightarrow T)^\circ$, by the intermediate result 3, this is \equiv to $(V \rightarrow R)^\circ[\vec{U}/\vec{X}] = (V \rightarrow R)[\vec{U}/\vec{X}]$. \square

A.2. Proof of Lemma 4.11. Induction on the depth of the derivation.

(1) $\frac{\Gamma, x:U \vdash \mathbf{t}:T}{\Gamma \vdash \lambda x \mathbf{t}:U \rightarrow T} \rightarrow_I$ This is the trivial case. Note that for each $\mathcal{V} \not\subseteq \text{FV}(\Gamma)$, one has $U \rightarrow T \succ_{\mathcal{V}, \Gamma}^{\lambda x \mathbf{t}} U \rightarrow T$.

(2) $\frac{\Gamma \vdash \lambda x \mathbf{t}:\forall X.T}{\Gamma \vdash \lambda x \mathbf{t}:T[V/X]} \forall_E$ By the induction hypothesis there exist U, R and \mathcal{V} such that $\Gamma, x:U \vdash \mathbf{t}:R$ and $U \rightarrow R \succeq_{\mathcal{V}, \Gamma}^{\lambda x \mathbf{t}} \forall X.T \succ_{X, \Gamma}^{\lambda x \mathbf{t}} T[V/X]$. (Observe that since X is bounded in $\forall X.T$, any variable in $\text{FV}(\Gamma)$ is distinct from it).

(3) $\frac{\Gamma \vdash \lambda x \mathbf{t}:T \quad X \notin \text{FV}(\Gamma)}{\Gamma \vdash \lambda x \mathbf{t}:\forall X.T} \forall_I$ By the induction hypothesis there exist U, R and \mathcal{V} such that $\Gamma, x:U \vdash \mathbf{t}:R$ and $U \rightarrow R \succeq_{\mathcal{V}, \Gamma}^{\lambda x \mathbf{t}} T \succ_{X, \Gamma}^{\lambda x \mathbf{t}} \forall X.T$.

(4) $\frac{\Gamma \vdash \lambda x \mathbf{t}:T \quad T \equiv R}{\Gamma \vdash \lambda x \mathbf{t}:R} \equiv$ By the induction hypothesis there exist U, S and \mathcal{V} such that $\Gamma, x:U \vdash \mathbf{t}:S$ and $U \rightarrow S \succeq_{\mathcal{V}, \Gamma}^{\lambda x \mathbf{t}} T \equiv R$. \square

A.3. Proof of Theorem 4.1. We proceed by checking that every reduction rule preserves the type. Let $\mathbf{t} \rightarrow \mathbf{r}$ and $\Gamma \vdash \mathbf{t}:T$. To show that $\Gamma \vdash \mathbf{r}:T$, we proceed by induction on the derivation of the typing judgement.

Elementary rules:

rule $\mathbf{t} + \mathbf{0} \rightarrow \mathbf{t}$: Let $\Gamma \vdash \mathbf{t} + \mathbf{0}:T$. By Lemma 4.2, there exist U and α such that $T \equiv \alpha.U$, then by Lemma 4.14, there exist $\delta, \gamma \in \mathcal{S}$ such that $\Gamma \vdash \mathbf{t}:\delta.U$ and $\Gamma \vdash \mathbf{0}:\gamma.U$ with $\delta + \gamma = \alpha$. By Lemma 4.19, $\gamma.U \equiv \bar{0}$, so $\gamma = 0$ and then $\delta = \alpha$.

rule $\mathbf{0.t} \rightarrow \mathbf{0}$: Let $\Gamma \vdash \mathbf{0.t}:T$. By Lemma 4.17, there exists R such that $T \equiv \mathbf{0.R} \equiv \bar{0}$ and by the rule $ax_{\bar{0}}$, $\Gamma \vdash \mathbf{0}:\bar{0}$.

rule $\mathbf{1.t} \rightarrow \mathbf{t}$: Let $\Gamma \vdash \mathbf{1.t}:T \equiv \mathbf{1.T}$. By Lemma 4.12, $\Gamma \vdash \mathbf{t}:T$.

rule $\alpha.\mathbf{0} \rightarrow \mathbf{0}$: Let $\Gamma \vdash \alpha.\mathbf{0}:T$. By Lemma 4.17, there exists R such that $T \equiv \alpha.R$. We proceed now by cases:

$\alpha \neq 0$: By Lemma 4.12, $\Gamma \vdash \mathbf{0}:R$. Thus, by Lemma 4.19, $R \equiv \bar{0}$ and so $T \equiv \alpha.R \equiv \bar{0}$.

$\alpha = 0$: By Lemma 4.13 there exists S such that $\Gamma \vdash \mathbf{0}:S$ and by Lemma 4.19 $S \equiv \bar{0}$.

rule $\alpha.(\beta.\mathbf{t}) \rightarrow (\alpha \times \beta).\mathbf{t}$: True by Lemma 4.23

rule $\alpha.(\mathbf{t} + \mathbf{r}) \rightarrow \alpha.\mathbf{t} + \alpha.\mathbf{r}$: True by Lemma 4.24.

Factorisation rules:

rule $\alpha.\mathbf{t} + \beta.\mathbf{t} \rightarrow (\alpha + \beta).\mathbf{t}$: True by Lemma 4.25.

rule $\alpha.\mathbf{t} + \mathbf{t} \rightarrow (\alpha + 1).\mathbf{t}$: Let $\Gamma \vdash \alpha.\mathbf{t} + \mathbf{t}:T$. Using the rule s_I one can derive $\Gamma \vdash \mathbf{1.}(\alpha.\mathbf{t} + \mathbf{t}):T$. Then by Lemma 4.24, $\Gamma \vdash \mathbf{1.}\alpha.\mathbf{t} + \mathbf{1.t}:T$. Moreover, by Lemma 4.2, there exist β, U such that $T \equiv \beta.U$ and then by Lemma 4.14, $\Gamma \vdash \mathbf{1.}\alpha.\mathbf{t}:\gamma.U$ and $\Gamma \vdash \mathbf{1.t}:\delta.U$ with $\gamma + \delta = \beta$. So, by Lemma 4.23, $\Gamma \vdash \alpha.\mathbf{t}:\gamma.U$. Then using the rule $+_I$ one can derive $\Gamma \vdash \alpha.\mathbf{t} + \mathbf{1.t}:\alpha.U$. We conclude, by Lemma 4.25, with $\Gamma \vdash (\alpha + 1).\mathbf{t}:\alpha.U \equiv T$.

rule $\mathbf{t} + \mathbf{t} \rightarrow (1 + 1).\mathbf{t}$: Let $\Gamma \vdash \mathbf{t} + \mathbf{t} : T$. Then by the rule s_I , $\Gamma \vdash 1.(\mathbf{t} + \mathbf{t}) : 1.T$. By Lemma 4.24, $\Gamma \vdash 1.\mathbf{t} + 1.\mathbf{t} : 1.T$ and by Lemma 4.25, $\Gamma \vdash (1 + 1).\mathbf{t} : 1.T \equiv T$.

Application rules:

rule $(\mathbf{t} + \mathbf{r})\mathbf{u} \rightarrow \mathbf{tu} + \mathbf{ru}$: Given in the main text of the paper.

rule $\mathbf{u}(\mathbf{t} + \mathbf{r}) \rightarrow \mathbf{ut} + \mathbf{ur}$: Analogous to the previous case.

rule $(\alpha.\mathbf{t})\mathbf{r} \rightarrow \alpha.\mathbf{tr}$: Let $\Gamma \vdash (\alpha.\mathbf{t})\mathbf{r} : T \equiv 1.T$. Then by Lemma 4.9, there exist $\gamma, \beta, U, \mathcal{V}$ and $T' \succeq_{\mathcal{V}, \Gamma}^{(\alpha.\mathbf{t})\mathbf{r}} T$ such that $\Gamma \vdash \mathbf{r} : \gamma.U$ and $\Gamma \vdash \alpha.\mathbf{t} : \beta.(U \rightarrow T')$ with $\gamma \times \beta = 1$. Moreover, by Lemma 4.17, $\beta.(U \rightarrow T') \equiv \alpha.\delta.U'$ then by Lemma 4.3, $U \rightarrow T' \equiv U'$ and $\beta = \alpha \times \delta$ (note that $\beta \neq 0$ because $\gamma \times \beta = 1$). So by Lemma 4.12, $\Gamma \vdash \mathbf{t} : \delta.(U \rightarrow T')$. By using the rule \rightarrow_E one can derive $\Gamma \vdash \mathbf{tr} : \delta.\gamma.T'$ from which, using the rule s_I one can deduce $\Gamma \vdash \alpha.\mathbf{tr} : \alpha.\delta.\gamma.T'$. Note that $\alpha.\delta.\gamma.T' \equiv (\alpha \times \delta \times \gamma).T' = (\beta \times \gamma).T' = 1.T' \equiv T'$ and by Lemma 4.6, $T' \succeq_{\mathcal{V}, \Gamma}^{\alpha.\mathbf{tr}} T$.

rule $\mathbf{r}(\alpha.\mathbf{t}) \rightarrow \alpha.\mathbf{rt}$: Analogous to the previous case.

rule $0\mathbf{t} \rightarrow 0$: True by Lemma 4.22 and the rule $ax_{\bar{0}}$.

rule $\mathbf{t}0 \rightarrow 0$: True by Lemma 4.22 and the rule $ax_{\bar{0}}$.

Beta reduction: Given in the main text of the paper.

AC equivalences:

Commutativity: Let $\Gamma \vdash \mathbf{t} + \mathbf{r} : T$. By Lemma 4.2, there exist α, U such that $T \equiv \alpha.U$. Then, by Lemma 4.14, there exist δ and γ such that $\Gamma \vdash \mathbf{t} : \delta.U$ and $\Gamma \vdash \mathbf{r} : \gamma.U$ with $\delta + \gamma = \alpha$. Then using rules $+_I$ and \equiv , one can derive $\Gamma \vdash \mathbf{r} + \mathbf{t} : T$.

Associativity: Let $\Gamma \vdash (\mathbf{t} + \mathbf{r}) + \mathbf{u} : T$. By Lemma 4.2, there exist α, U such that $T \equiv \alpha.U$. Then, by Lemma 4.14, there exist δ and γ such that $\Gamma \vdash \mathbf{t} + \mathbf{r} : \delta.U$ and $\Gamma \vdash \mathbf{u} : \gamma.U$ with $\delta + \gamma = \alpha$. Then, by Lemma 4.14 again, there exist δ' and γ' such that $\Gamma \vdash \mathbf{t} : \delta'.U$ and $\Gamma \vdash \mathbf{r} : \gamma'.U$ with $\delta' + \gamma' = \delta$. Then with the rule $+_I$ one can deduce $\Gamma \vdash \mathbf{r} + \mathbf{u} : (\gamma' + \gamma).U$ and with the same rule, $\Gamma \vdash \mathbf{t} + (\mathbf{r} + \mathbf{u}) : (\delta' + \gamma' + \gamma).U \equiv T$. The inverse is analogous: if $\Gamma \vdash \mathbf{t} + (\mathbf{r} + \mathbf{u}) : T$ then $\Gamma \vdash (\mathbf{t} + \mathbf{r}) + \mathbf{u} : T$.

Contextual rules: Let $\mathbf{t} \rightarrow \mathbf{r}$ and assume as the induction hypothesis that for any context Γ and type T , if $\Gamma \vdash \mathbf{t} : T$ then $\Gamma \vdash \mathbf{r} : T$.

$\mathbf{tu} \rightarrow \mathbf{ru}$: Let $\Gamma \vdash \mathbf{tu} : T$. By the rule \equiv , $\Gamma \vdash \mathbf{tu} : 1.T$. Then by Lemma 4.9, there exist α, β, U, R and \mathcal{V} such that $\Gamma \vdash \mathbf{t} : \alpha.(U \rightarrow R)$ and $\Gamma \vdash \mathbf{u} : \beta.U$ with $R \succeq_{\mathcal{V}, \Gamma}^{\mathbf{tu}} T$ and $\alpha \times \beta = 1$. By the induction hypothesis $\Gamma \vdash \mathbf{r} : \alpha.(U \rightarrow R)$, from which, using the rule \rightarrow_E , one can deduce $\Gamma \vdash \mathbf{ru} : \alpha \times \beta.R$. Note that $\alpha \times \beta.R = 1.R \equiv R$ and by Lemma 4.6, $R \succeq_{\mathcal{V}, \Gamma}^{\mathbf{ru}} T$.

$\mathbf{ut} \rightarrow \mathbf{ur}$: Analogous to the previous case.

$\mathbf{t} + \mathbf{u} \rightarrow \mathbf{r} + \mathbf{u}$: Let $\Gamma \vdash \mathbf{t} + \mathbf{u} : T$. By Lemma 4.2, $T \equiv \alpha.U$, so by the rule \equiv , $\Gamma \vdash \mathbf{t} + \mathbf{u} : \alpha.U$. Then by Lemma 4.14, $\Gamma \vdash \mathbf{t} : \delta.U$ and $\Gamma \vdash \mathbf{u} : \gamma.U$ with $\delta + \gamma = \alpha$. By the induction hypothesis $\Gamma \vdash \mathbf{r} : \delta.U$, so using the rule $+_I$ one can deduce $\Gamma \vdash \mathbf{r} + \mathbf{u} : (\delta + \gamma).U$. Note that $(\delta + \gamma).U = \alpha.U \equiv T$.

$\mathbf{u} + \mathbf{t} \rightarrow \mathbf{u} + \mathbf{r}$: Analogous to the previous case.

$\alpha.\mathbf{t} \rightarrow \alpha.\mathbf{r}$: Let $\Gamma \vdash \alpha.\mathbf{t} : T$. By Lemma 4.17, there exist γ and U such that $T \equiv \alpha.\gamma.U$. We proceed now by cases:

$\alpha \neq 0$: By Lemma 4.12, $\Gamma \vdash \mathbf{t} : \gamma.U$. Moreover, by the induction hypothesis $\Gamma \vdash \mathbf{r} : \gamma.U$ and using the rule s_I one can derive $\Gamma \vdash \alpha.\mathbf{r} : \alpha.\gamma.U \equiv T$.

$\alpha = 0$: By Lemma 4.13, there exists R such that $\Gamma \vdash \mathbf{t} : R$. Moreover, by the induction hypothesis $\Gamma \vdash \mathbf{r} : R$ and using the rule s_I one can derive $\Gamma \vdash \alpha.\mathbf{r} : \alpha.R \equiv \bar{0} \equiv T$.

$\lambda x \mathbf{t} \rightarrow \lambda x \mathbf{r}$: Let $\Gamma \vdash \lambda x \mathbf{t} : T$. By Lemma 4.11, there exist U, R and \mathcal{V} such that $\Gamma, x : U \vdash \mathbf{t} : R$ with $U \rightarrow R \succeq_{\mathcal{V}, \Gamma}^{\lambda x \mathbf{t}} T$. Then by the induction hypothesis $\Gamma, x : U \vdash \mathbf{r} : R$, and using the rule \rightarrow_I one can derive $\Gamma \vdash \lambda x \mathbf{r} : U \rightarrow R$. By Lemma 4.6, $U \rightarrow R \succeq_{\mathcal{V}, \Gamma}^{\lambda x \mathbf{r}} T$. \square

A.4. Proof of Lemma 5.6.

(1) Obviously $\text{SN} \subseteq \text{SN}$. We need to prove it satisfies each point of the definition of saturation.

- (a) $\mathbf{0} \in \text{SN}$.
- (b) for each x and $\vec{\mathbf{t}}, x\vec{\mathbf{t}} \in \text{SN}$.
- (c) Assume $\mathbf{t}[\mathbf{b}/x]\vec{\mathbf{r}} \in \text{SN}$, then the term

$$(\lambda x \mathbf{t})\mathbf{b}\vec{\mathbf{r}} \tag{A.1}$$

must terminate because \mathbf{t}, \mathbf{b} and $\vec{\mathbf{r}}$ terminate since they are SN by assumption ($\mathbf{t}[\mathbf{b}/x]$ is a sub-term of a term in SN, hence itself is SN; but then \mathbf{t} is also SN), After finitely many steps reducing terms in A.1 we obtain $(\lambda x \mathbf{t}')\mathbf{b}'\vec{\mathbf{r}}'$ with $\mathbf{t} \rightarrow^* \mathbf{t}'$, $\mathbf{b} \rightarrow \mathbf{b}'$ and for each i , $\mathbf{r}_i \rightarrow \mathbf{r}'_i$. Then the contraction of $(\lambda x \mathbf{t}')\mathbf{b}'\vec{\mathbf{r}}'$ gives

$$\mathbf{t}'[\mathbf{b}'/x]\vec{\mathbf{r}}' \tag{A.2}$$

This is a reduct of $\mathbf{t}[\mathbf{b}/x]\vec{\mathbf{r}}$ and since this is SN, also A.2 and $(\lambda x \mathbf{t})\mathbf{b}\vec{\mathbf{r}}$ are SN.

(d) First note that if $\mathbf{t}, \mathbf{u} \in \text{SN}$, then $\mathbf{t} + \mathbf{u} \in \text{SN}$. Now, assume that for each $i \in I$, $\mathbf{t}_i\vec{\mathbf{r}} \in \text{SN}$, which implies that \mathbf{t}_i and $\vec{\mathbf{r}}$ are SN. Also, note that $(\sum_{i \in I} \mathbf{t}_i)\vec{\mathbf{r}} \rightarrow^* \sum_{i \in I} \mathbf{t}_i\vec{\mathbf{r}}$ which is the sum of SN terms, so is SN. We need to prove that any other reduction starting from $\sum_{i \in I} \mathbf{t}_i\vec{\mathbf{r}}$ is also SN. We proceed by induction on I . To simplify the notation, we take $I = \{1, \dots, n\}$ with $n \geq 1$.

- If $I = \{1\}$, then we are done, since for each $i \in I$, $\mathbf{t}_i\vec{\mathbf{r}} \in \text{SN}$.
- Assume it is true for $I = \{1, \dots, n\}$, that is $(\sum_{i=1}^n \mathbf{t}_i)\vec{\mathbf{r}} \in \text{SN}$.
- Let $I = \{1, \dots, n+1\}$, then we must prove that $(\sum_{i=1}^n \mathbf{t}_i + \mathbf{t}_{n+1})\vec{\mathbf{r}} \in \text{SN}$. We proceed by a case analysis of every possible reductions of this term. Note that any reduction in $\vec{\mathbf{t}}$ or $\vec{\mathbf{r}}$ is finite since these terms are in SN, so the amount of addends is the same.

Elementary rules:

- One $\mathbf{t}_k = \mathbf{0}$ and the rule $\mathbf{t} + \mathbf{0} \rightarrow \mathbf{t}$ applies. Then the induction hypothesis closes the case.
- One $\mathbf{t}_k = \alpha.(\mathbf{t}_{k_1} + \mathbf{t}_{k_2})$ and reduces to $\alpha.\mathbf{t}_{k_1} + \alpha.\mathbf{t}_{k_2}$, it still can be considered as one addend since \mathbf{t}_k is in SN.
- In other case, it is just a reduction in one \mathbf{t}_i or one of $\vec{\mathbf{r}}$.

Factorisation rules: This case follows by the induction hypothesis.

Application rules:

- Again, reductions on any \mathbf{t}_k are not considered since these cases are trivial by the strong normalisation of \mathbf{t}_k .
- $(\sum_{i=1}^n \mathbf{t}_i + \mathbf{t}_{n+1})\vec{\mathbf{r}} \rightarrow (\sum_{i=1}^n \mathbf{t}_i)\vec{\mathbf{r}} + (\mathbf{t}_{n+1})\vec{\mathbf{r}}$, then since by the induction hypothesis $(\sum_{i=1}^n \mathbf{t}_i)\vec{\mathbf{r}} \in \text{SN}$ and by assumption also $(\mathbf{t}_{n+1})\vec{\mathbf{r}} \in \text{SN}$, its sum is in SN.
- $(\sum_{i=1}^n \mathbf{t}_i + \mathbf{t}_{n+1})\vec{\mathbf{r}} \rightarrow (\sum_{i=1}^k \mathbf{t}_i)\vec{\mathbf{r}} + (\sum_{i=k+1}^{n+1} \mathbf{t}_i)\vec{\mathbf{r}}$ the induction hypothesis applies to both addends.

- Any other case does not involve a sum, so they are either included in the case of reduction of one \mathbf{t}_k or one from $\vec{\mathbf{r}}$, or in the basic case of the induction. Recall that the terms are considered modulo AC axioms for $+$ so it is not relevant to consider the permutations or different parentheses locations in the big sums.

Beta reduction: This is either the basic case or a reduction in one \mathbf{t}_k or one of $\vec{\mathbf{r}}$.

- (e) for each $i \in I$, $\mathbf{u}\mathbf{t}_i\vec{\mathbf{r}} \in \text{SN}$. Note that $\mathbf{u}(\sum_{i \in I} \mathbf{t}_i)\vec{\mathbf{r}} \rightarrow^* \sum_{i \in I} \mathbf{u}\mathbf{t}_i\vec{\mathbf{r}}$ which is the sum of SN terms. This case is analogous to 1d.
 - (f) if $\mathbf{t} \in \text{SN}$, then for each $\alpha \in \mathcal{S}$, $\alpha.\mathbf{t} \in \text{SN}$ and vice-versa.
 - (g) $\alpha.\mathbf{t}_1\mathbf{t}_2 \dots \mathbf{t}_n \in \text{SN}$ then for each k , the term $\mathbf{t}_1 \dots (\alpha.\mathbf{t}_k) \dots \mathbf{t}_n$ must terminate because $\vec{\mathbf{t}}$ terminate since these terms are SN by assumption, so after finitely many reduction steps reducing $\mathbf{t}_1 \dots (\alpha.\mathbf{t}_k) \dots \mathbf{t}_n$ we obtain $\alpha.\mathbf{u}$, with $\mathbf{t}_1 \dots \mathbf{t}_n \rightarrow^* \mathbf{u}$. So $\alpha.\mathbf{u}$ is a reduct of $\alpha.\mathbf{t}_1 \dots \mathbf{t}_n$ and since this term is SN, for each k , $\mathbf{t}_1 \dots (\alpha.\mathbf{t}_k) \dots \mathbf{t}_n$ are SN.
 - (h) $\mathbf{0}\vec{\mathbf{t}} \rightarrow^* \mathbf{0}\vec{\mathbf{t}}'$ and since $\vec{\mathbf{t}}$ is SN, assume $\vec{\mathbf{t}}'$ is in normal form, so $\mathbf{0}\vec{\mathbf{t}}'$ can only reduce to $\mathbf{0}$, then $\mathbf{0}\vec{\mathbf{t}} \in \text{SN}$.
 - (i) Consider the term $\mathbf{t}\mathbf{0}\vec{\mathbf{u}}$. The term reduces to $\mathbf{0}\vec{\mathbf{u}}$, which then reduces to $\mathbf{0}$. But we could also choose to first reduce \mathbf{t} or $\vec{\mathbf{u}}$. In any case, the terms \mathbf{t} and \mathbf{u} do not interact with any other term than with $\mathbf{0}$, and when they do they reach $\mathbf{0}$. This will necessarily happen since \mathbf{t} and \mathbf{u} are in SN by assumption. Hence the entire term is in SN.
- (2) Let $A, B \in \text{SAT}$, then $x \in A$ by definition of saturated sets. For each $\mathbf{t} \in A \Rightarrow B$, $\mathbf{t}x \in B$. Since $B \in \text{SAT}$, then $B \subseteq \text{SN}$, so $\mathbf{t}x \in \text{SN}$ and so \mathbf{t} is strongly normalising. Therefore $A \Rightarrow B \subseteq \text{SN}$. Now we need to show $A \Rightarrow B$ is saturated by showing each point in the definition of saturated sets.
- (a) By saturation of B , for each $\mathbf{u} \in A$, $\mathbf{0}\mathbf{u} \in B$, then $\mathbf{0} \in A \Rightarrow B$.
 - (b) Let $\vec{\mathbf{t}} \in \text{SN}$, we need to show that $x\vec{\mathbf{t}} \in A \Rightarrow B$, *i.e.* for each $\mathbf{u} \in A$, $x\vec{\mathbf{t}}\mathbf{u} \in B$, which is true since $A \in \text{SAT}$ implies that $\mathbf{u} \in \text{SN}$, so $B \in \text{SAT}$ implies that $x\vec{\mathbf{t}}\mathbf{u} \in B$.
 - (c) Let $\mathbf{t}[\mathbf{b}/x]\vec{\mathbf{r}} \in A \Rightarrow B$, then for each $\mathbf{u} \in A$, $\mathbf{t}[\mathbf{b}/x]\vec{\mathbf{r}}\mathbf{u} \in B$ and since B is saturated, $(\lambda x \mathbf{t})\mathbf{b}\vec{\mathbf{r}}\mathbf{u} \in B$, so $(\lambda x \mathbf{t})\mathbf{b}\vec{\mathbf{r}} \in A \Rightarrow B$.
 - (d) Let for each $i \in I$, $\mathbf{t}_i\vec{\mathbf{r}} \in A \Rightarrow B$, then for each $\mathbf{u} \in A$ and for each $i \in I$, $\mathbf{t}_i\vec{\mathbf{r}}\mathbf{u} \in B$, then by the saturation of B , $(\sum_{i \in I} \mathbf{t}_i)\vec{\mathbf{r}}\mathbf{u} \in B$, so $(\sum_{i \in I} \mathbf{t}_i)\vec{\mathbf{r}} \in A \Rightarrow B$.
 - (e) Let for each $i \in I$, $\mathbf{u}\mathbf{t}_i\vec{\mathbf{r}} \in A \Rightarrow B$, then for each $\vec{\mathbf{t}}' \in A$, $\mathbf{u}\mathbf{t}_i\vec{\mathbf{r}}\vec{\mathbf{t}}' \in B$, then by saturation of B , $\mathbf{u}(\sum_{i \in I} \mathbf{t}_i)\vec{\mathbf{r}}\vec{\mathbf{t}}' \in B$, so $\mathbf{u}(\sum_{i \in I} \mathbf{t}_i)\vec{\mathbf{r}} \in A \Rightarrow B$.
 - (f) Let $\mathbf{t} \in A \Rightarrow B$ then for each $\mathbf{u} \in A$, $\mathbf{t}\mathbf{u} \in B$, then by the saturation of B , for each $\alpha \in \mathcal{S}$, $\alpha.\mathbf{t}\mathbf{u} \in B$, then also by the saturation of B , $(\alpha.\mathbf{t})\mathbf{u} \in B$, so $\alpha.\mathbf{t} \in A \Rightarrow B$.
Let $\alpha.\mathbf{t} \in A \Rightarrow B$, then for each $\mathbf{u} \in A$, $\alpha.\mathbf{t}\mathbf{u} \in B$, so by the saturation of B , $\alpha.\mathbf{t}\mathbf{u} \in B$, and again, by the saturation of B , $\mathbf{t}\mathbf{u} \in B$, so $\mathbf{t} \in A \Rightarrow B$.
 - (g) Let $\alpha.\mathbf{t}_1 \dots \mathbf{t}_n \in A \Rightarrow B$, then for each $\mathbf{u} \in A$, $(\alpha.\mathbf{t}_1 \dots \mathbf{t}_n)\mathbf{u} \in B$, then by the saturation of B , $\alpha.\mathbf{t}_1 \dots \mathbf{t}_n\mathbf{u} \in B$, and so, by the saturation of B again, for each k , $\mathbf{t}_1 \dots (\alpha.\mathbf{t}_k) \dots \mathbf{t}_n\mathbf{u} \in B$, then for each k , $\mathbf{t}_1 \dots (\alpha.\mathbf{t}_k) \dots \mathbf{t}_n \in A \Rightarrow B$.
The inverse follows analogously: let $\mathbf{t}_1 \dots (\alpha.\mathbf{t}_k) \dots \mathbf{t}_n \in A \Rightarrow B$, then for each $\mathbf{u} \in A$, $\mathbf{t}_1 \dots (\alpha.\mathbf{t}_k) \dots \mathbf{t}_n\mathbf{u} \in B$, so by the saturation of B , one has $\alpha.\mathbf{t}_1 \dots \mathbf{t}_n\mathbf{u} \in B$ and then, also by the saturation of B , one has $(\alpha.\mathbf{t}_1 \dots \mathbf{t}_n)\mathbf{u} \in B$, then $\alpha.\mathbf{t}_1 \dots \mathbf{t}_n \in A \Rightarrow B$.

- (h) For each $\mathbf{u} \in A$, $\mathbf{u} \in \text{SN}$ and then, by the saturation of B , for each $\vec{\mathbf{t}} \in \text{SN}$, $\mathbf{0}\vec{\mathbf{t}}\mathbf{u} \in B$. Then $\mathbf{0}\vec{\mathbf{t}} \in A \Rightarrow B$.
- (i) For each $\mathbf{r} \in A$, $\mathbf{r} \in \text{SN}$ and then, by the saturation of B , for each $\mathbf{t}, \vec{\mathbf{u}} \in \text{SN}$, $\mathbf{t}\mathbf{0}\vec{\mathbf{u}}\mathbf{r} \in B$. Then $\mathbf{t}\mathbf{0}\vec{\mathbf{u}} \in A \Rightarrow B$.
- (3) Let $\{A_i\}_{i \in I}$ be a collection of members of SAT, then for each $i \in I$, $A_i \subseteq \text{SN}$, so $\bigcap_{i \in I} A_i \subseteq \text{SN}$. We have to show that $\bigcap_{i \in I} A_i$ is saturated.
- Conditions (a), (b), (h) and (i) follow trivially: all these conditions have the form “ $\mathbf{t} \in X$ ”. Since by the saturation of A_i , for each $i \in I$, $\mathbf{t} \in A_i$, then $\mathbf{t} \in \bigcap_{i \in I} A_i$.
 - Conditions (c), (d), (e), (f) and (g) are also straightforward: all these conditions have the form “If \mathbf{t} in X , then \mathbf{r} in X ”. Let $\mathbf{t} \in \bigcap_{i \in I} A_i$, then for each $i \in I$, $\mathbf{t} \in A_i$ and so, by the saturation of A_i , $\mathbf{r} \in A_i$, from where one can deduce $\mathbf{r} \in \bigcap_{i \in I} A_i$.
- (4) By structural induction on A .
- $A := X$: Then $\llbracket A \rrbracket_\xi = \xi(X) \in \text{SAT}$.
- $A := B \rightarrow C$: Then $\llbracket A \rrbracket_\xi = \llbracket B \rrbracket_\xi \Rightarrow \llbracket C \rrbracket_\xi$. By the induction hypothesis $\llbracket B \rrbracket_\xi$ and $\llbracket C \rrbracket_\xi \in \text{SAT}$, then by Lemma 5.6(2), $\llbracket B \rrbracket_\xi \Rightarrow \llbracket C \rrbracket_\xi \in \text{SAT}$.
- $A := \forall X.A'$: Then $\llbracket A \rrbracket_\xi = \bigcap_{Y \in \text{SAT}} \llbracket A' \rrbracket_{\xi(X:=Y)}$. By the induction hypothesis $\forall Y \in \text{SAT}$, $\llbracket A' \rrbracket_{\xi(X:=Y)} \in \text{SAT}$, then by Lemma 5.6(3), $\bigcap_{Y \in \text{SAT}} \llbracket A' \rrbracket_{\xi(X:=Y)} \in \text{SAT}$. \square

A.5. Proof of Theorem 5.8. We proceed by induction on the derivation of $\Gamma \Vdash \mathbf{t}:T$.

- (1) $\frac{}{\Gamma, x:A \Vdash x:A} ax^\triangleleft$ Note that if $\rho, \xi \models \Gamma, x:A$, then by definition $\rho, \xi \models x:A$.
- (2) $\frac{}{\Gamma \Vdash \mathbf{0}:A} ax_0^\triangleleft$ Then for each ξ and ρ , by the saturation of $\llbracket A \rrbracket_\xi$ one has $\mathbf{0} \in \llbracket A \rrbracket_\xi$. Since $\llbracket \mathbf{0} \rrbracket_\rho = \mathbf{0}$, then $\rho, \xi \models \mathbf{0}:A$, and so for every Γ , $\Gamma \models \mathbf{0}:A$.
- (3) $\frac{\Gamma \Vdash \mathbf{t}:A \rightarrow B \quad \Gamma \Vdash \mathbf{r}:A}{\Gamma \Vdash \mathbf{t}\mathbf{r}:B} \rightarrow_E^\triangleleft$ Given in the main text of the paper.
- (4) $\frac{\Gamma, x:A \Vdash \mathbf{t}:B}{\Gamma \Vdash \lambda x \mathbf{t}:A \rightarrow B} \rightarrow_I^\triangleleft$ Assume $\rho, \xi \models \Gamma$ in order to show $\rho, \xi \models \lambda x \mathbf{t}:A \rightarrow B$. That is, we must show $\llbracket \lambda x \mathbf{t} \rrbracket_\rho \mathbf{u} \in \llbracket B \rrbracket_\xi$ for all $\mathbf{u} \in \llbracket A \rrbracket_\xi$. Assume $\mathbf{u} \in \llbracket A \rrbracket_\xi$, then $\rho(x := \mathbf{u}) \models \Gamma, x:A$ and hence by the induction hypothesis $\llbracket \mathbf{t} \rrbracket_{\rho(x:=\mathbf{u})} \in B$. Since $\llbracket \lambda x \mathbf{t} \rrbracket_\rho \mathbf{u} = (\lambda x \mathbf{t})[\vec{y} := \rho(\vec{y})]\mathbf{u} \rightarrow_\beta \mathbf{t}[\vec{y} := \rho(\vec{y}), x := \mathbf{u}] = \llbracket \mathbf{t} \rrbracket_{\rho(x:=\mathbf{u})}$, it follows from the saturation of $\llbracket B \rrbracket_\xi$ that $\llbracket \lambda x \mathbf{t} \rrbracket_\rho \mathbf{u} \in \llbracket B \rrbracket_\xi$.
- (5) $\frac{\Gamma \Vdash \mathbf{t}:\forall X.A}{\Gamma \Vdash \mathbf{t}:A[B/X]} \forall_E^\triangleleft$ Assume $\rho, \xi \models \Gamma$ in order to show $\rho, \xi \models \mathbf{t}:A[B/X]$. By the induction hypothesis $\llbracket \mathbf{t} \rrbracket_\rho \in \llbracket \forall X.A \rrbracket_\xi$ and this set is equal to $\bigcap_{Y \in \text{SAT}} \llbracket A \rrbracket_{\xi(X:=Y)}$, hence $\llbracket \mathbf{t} \rrbracket_\rho \in \llbracket A \rrbracket_{\xi(X:=\llbracket B \rrbracket_\xi)}$ = $\llbracket A[B/X] \rrbracket_\xi$.
- (6) $\frac{\Gamma \Vdash \mathbf{t}:A \quad X \notin \text{FV}(\Gamma)}{\Gamma \Vdash \mathbf{t}:\forall X.A} \forall_I^\triangleleft$ Assume $\rho, \xi \models \Gamma$ in order to show $\rho, \xi \models \mathbf{t}:\forall X.A$. Since $X \notin \text{FV}(\Gamma)$, one also has $\forall Y \in \text{SAT}$ that $\rho, \xi(X := Y) \models \Gamma$, therefore $\forall Y \in \text{SAT}$, $\llbracket \mathbf{t} \rrbracket_\rho \in \llbracket A \rrbracket_{\xi(X:=Y)}$, then by definition of $\llbracket \forall X.A \rrbracket_\xi$, $\llbracket \mathbf{t} \rrbracket_\rho \in \llbracket \forall X.A \rrbracket_\xi$, *i.e.* $\rho, \xi \models \mathbf{t}:\forall X.A$.
- (7) $\frac{\Gamma \Vdash \mathbf{t}:A \quad \Gamma \Vdash \mathbf{r}:A}{\Gamma \Vdash \mathbf{t}+\mathbf{r}:A} +_I^\triangleleft$ Assume $\rho, \xi \models \Gamma$ in order to show $\rho, \xi \models \mathbf{t}+\mathbf{r}:A$. By the induction hypothesis one has $\Gamma \models \mathbf{t}:A$ and $\Gamma \models \mathbf{r}:A$, so $\llbracket \mathbf{t} \rrbracket_\rho \in \llbracket A \rrbracket_\xi$ and $\llbracket \mathbf{r} \rrbracket_\rho \in \llbracket A \rrbracket_\xi$. Since $\llbracket \mathbf{t}+\mathbf{r} \rrbracket_\rho = (\mathbf{t}+\mathbf{r})[\vec{x} := \rho(\vec{x})] = \mathbf{t}[\vec{x} := \rho(\vec{x})] + \mathbf{r}[\vec{x} := \rho(\vec{x})] = \llbracket \mathbf{t} \rrbracket_\rho + \llbracket \mathbf{r} \rrbracket_\rho$, it follows from the saturation of $\llbracket A \rrbracket_\xi$ that $\llbracket \mathbf{t}+\mathbf{r} \rrbracket_\rho \in \llbracket A \rrbracket_\xi$.

$$(8) \frac{\Gamma \Vdash \mathbf{t} : A}{\Gamma \Vdash \alpha.\mathbf{t} : A} s_I^\triangleleft \quad \text{Suppose } \rho, \xi \Vdash \Gamma \text{ in order to show } \rho, \xi \Vdash \alpha.\mathbf{t} : A. \text{ By the induction hypothesis one has } \Gamma \Vdash \mathbf{t} : A, \text{ then } \llbracket \mathbf{t} \rrbracket_\rho \in \llbracket A \rrbracket_\xi. \text{ Since } \llbracket \alpha.\mathbf{t} \rrbracket_\rho = (\alpha.\mathbf{t})[\vec{x} := \rho(\vec{x})] = \alpha.\mathbf{t}[\vec{x} := \rho(\vec{x})] = \alpha.\llbracket \mathbf{t} \rrbracket_\rho, \text{ it follows from the saturation of } \llbracket A \rrbracket_\xi \text{ that } \llbracket \alpha.\mathbf{t} \rrbracket_\rho \in \llbracket A \rrbracket_\xi. \quad \square$$

A.6. Proof of Lemma 5.11. We proceed by induction on the derivation of $\Gamma \vdash \mathbf{t} : T$.

$$(1) \frac{}{\Gamma, x : U \vdash x : U} ax \quad (\Gamma, x : U)^\natural = \Gamma^\natural, x : U^\natural, \text{ so by } ax^\triangleleft, (\Gamma, x : U)^\natural \Vdash x : U^\natural.$$

$$(2) \frac{}{\Gamma \vdash \mathbf{0} : \bar{\mathbf{0}}} ax\bar{\mathbf{0}} \quad \text{By } ax_0^\triangleleft, \Gamma^\natural \Vdash \mathbf{0} : A \text{ for any } A \in \mathbb{T}(\lambda 2^{la}), \text{ so take } A = \bar{\mathbf{0}}^\natural.$$

$$(3) \frac{\Gamma \vdash \mathbf{t} : \alpha.(U \rightarrow T) \quad \Gamma \vdash \mathbf{r} : \beta.U}{\Gamma \vdash \mathbf{tr} : (\alpha \times \beta).T} \rightarrow_E \quad \text{By the induction hypothesis } \Gamma^\natural \Vdash \mathbf{t} : U^\natural \rightarrow T^\natural \text{ and } \Gamma^\natural \Vdash \mathbf{r} : U^\natural, \text{ so by the rule } \rightarrow_E^\triangleleft, \Gamma^\natural \Vdash \mathbf{tr} : T^\natural = ((\alpha \times \beta).T)^\natural.$$

$$(4) \frac{\Gamma, x : U \vdash \mathbf{t} : T}{\Gamma \vdash \lambda x \mathbf{t} : U \rightarrow T} \rightarrow_I \quad \text{By the induction hypothesis } \Gamma^\natural, x : U^\natural \Vdash \mathbf{t} : T^\natural, \text{ so by the rule } \rightarrow_I^\triangleleft, \Gamma^\natural \Vdash \lambda x \mathbf{t} : U^\natural \rightarrow T^\natural = (U \rightarrow T)^\natural.$$

$$(5) \frac{\Gamma \vdash \mathbf{t} : \forall X.T}{\Gamma \vdash \mathbf{t} : T[U/X]} \forall_E \quad \text{By the induction hypothesis } \Gamma^\natural \Vdash \mathbf{t} : (\forall X.T)^\natural = \forall X.T^\natural, \text{ so by the rule } \forall_E^\triangleleft, \Gamma^\natural \Vdash \mathbf{t} : T^\natural[U^\natural/X].$$

$$(6) \frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \mathbf{t} : \forall X.T} \forall_I \quad \text{By the induction hypothesis } \Gamma^\natural \Vdash \mathbf{t} : T^\natural, \text{ so by the rule } \forall_I^\triangleleft, \Gamma^\natural \Vdash \mathbf{t} : \forall X.T^\natural = (\forall X.T)^\natural. \text{ (Note that } FV(\Gamma) = FV(\Gamma^\natural)\text{).}$$

$$(7) \frac{\Gamma \vdash \mathbf{t} : \alpha.T \quad \Gamma \vdash \mathbf{r} : \beta.T}{\Gamma \vdash \mathbf{t} + \mathbf{r} : (\alpha + \beta).T} +_I \quad \text{Given in the main text of the paper.}$$

$$(8) \frac{\Gamma \vdash \mathbf{t} : T}{\Gamma \vdash \alpha.\mathbf{t} : \alpha.T} s_I \quad \text{By the induction hypothesis } \Gamma^\natural \Vdash \mathbf{t} : T^\natural, \text{ so by the rule } s_I^\triangleleft, \Gamma^\natural \Vdash \alpha.\mathbf{t} : T^\natural = (\alpha.T)^\natural. \quad \square$$

A.7. Proof of Theorem 5.14(3). Before proving this theorem, we need to prove the following properties:

Lemma A.1.

- (1) Let \mathbf{t} and \mathbf{r} be any terms. If $\mathbf{t} \rightarrow_a \mathbf{r}$, then for any base term \mathbf{b} , $\mathbf{t}[\mathbf{b}/x] \rightarrow_a \mathbf{r}[\mathbf{b}/x]$.
- (2) Let \mathbf{b}_1 and \mathbf{b}_2 be base terms. If $\mathbf{b}_1 \rightarrow_a \mathbf{b}_2$, then for any term \mathbf{t} , $\mathbf{t}[\mathbf{b}_1/x] \rightarrow_a^* \mathbf{t}[\mathbf{b}_2/x]$.

Proof.

- (1) We proceed by structural induction on \mathbf{t} .
 - (a) \mathbf{t} cannot be a variable or $\mathbf{0}$, since it reduces.
 - (b) $\mathbf{t} = \lambda x \mathbf{t}'$ and $\mathbf{r} = \lambda x \mathbf{r}'$, with $\mathbf{t}' \rightarrow_a \mathbf{r}'$. Then $\lambda x \mathbf{t}'[\mathbf{b}/x] = \lambda x (\mathbf{t}'[\mathbf{b}/x])$, which by the induction hypothesis \rightarrow_a -reduces to $\lambda x (\mathbf{r}'[\mathbf{b}/x]) = \lambda x \mathbf{r}'[\mathbf{b}/x]$.
 - (c) $\mathbf{t} = \alpha.\mathbf{t}'$. We proceed now by cases:
 - (i) $\alpha = 0$ and $\mathbf{r} = 0$, then $(0.\mathbf{t}')[\mathbf{b}/x] = 0.\mathbf{t}'[\mathbf{b}/x] \rightarrow_a \mathbf{0} = \mathbf{0}[\mathbf{b}/x]$.
 - (ii) $\alpha = 1$ and $\mathbf{r} = \mathbf{t}'$, then $(1.\mathbf{t}')[\mathbf{b}/x] = 1.\mathbf{t}'[\mathbf{b}/x] \rightarrow_a \mathbf{t}'[\mathbf{b}/x]$.
 - (iii) $\mathbf{t} = \mathbf{0}$ and $\mathbf{r} = \mathbf{0}$, then $(\alpha.\mathbf{0})[\mathbf{b}/x] = \alpha.\mathbf{0}[\mathbf{b}/x] = \alpha.\mathbf{0} \rightarrow_a \mathbf{0} = \mathbf{0}[\mathbf{b}/x]$.
 - (iv) $\mathbf{t}' = \beta.\mathbf{t}''$ and $\mathbf{r} = (\alpha \times \beta).\mathbf{t}''$, then $(\alpha.\beta.\mathbf{t}'')[\mathbf{b}/x] = \alpha.\beta.\mathbf{t}''[\mathbf{b}/x] \rightarrow_a (\alpha \times \beta).\mathbf{t}''[\mathbf{b}/x] = ((\alpha \times \beta).\mathbf{t}'')[\mathbf{b}/x]$.

- (v) $\mathbf{t}' = \mathbf{t}_1 + \mathbf{t}_2$ and $\mathbf{r} = \alpha.\mathbf{t}_1 + \alpha.\mathbf{t}_2$, then $(\alpha.(\mathbf{t}_1 + \mathbf{t}_2))[\mathbf{b}/x] = \alpha.(\mathbf{t}_1[\mathbf{b}/x] + \mathbf{t}_2[\mathbf{b}/x]) \rightarrow_a \alpha.\mathbf{t}_1[\mathbf{b}/x] + \alpha.\mathbf{t}_2[\mathbf{b}/x] = (\alpha.\mathbf{t}_1 + \alpha.\mathbf{t}_2)[\mathbf{b}/x]$.
 - (vi) $\mathbf{t}' \rightarrow_a \mathbf{r}'$, and $\mathbf{r} = \alpha.\mathbf{r}'$, then $(\alpha.\mathbf{t}')[\mathbf{b}/x] = \alpha.\mathbf{t}'[\mathbf{b}/x]$, which by the induction hypothesis \rightarrow_a -reduces to $\alpha.\mathbf{r}'[\mathbf{b}/x] = (\alpha.\mathbf{r}')[\mathbf{b}/x]$.
- (d) $\mathbf{t} = \mathbf{t}_1 + \mathbf{r}_2$. We proceed now by cases:
- (i) $\mathbf{t}_2 = 0$ and $\mathbf{r} = \mathbf{t}_1$, then $(\mathbf{t}_1 + \mathbf{0})[\mathbf{b}/x] = \mathbf{t}_1[\mathbf{b}/x] + 0 \rightarrow_a \mathbf{t}_1[\mathbf{b}/x]$.
 - (ii) $\mathbf{t}_1 = \alpha.\mathbf{t}'$, $\mathbf{t}_2 = \beta.\mathbf{t}'$ and $\mathbf{r} = (\alpha + \beta).\mathbf{t}'$, then $(\alpha.\mathbf{t}' + \beta.\mathbf{t}')[\mathbf{b}/x] = \alpha.\mathbf{t}'[\mathbf{b}/x] + \beta.\mathbf{t}'[\mathbf{b}/x] \rightarrow_a (\alpha + \beta).\mathbf{t}'[\mathbf{b}/x] = ((\alpha + \beta).\mathbf{t}')[\mathbf{b}/x]$.
 - (iii) $\mathbf{t}_1 = \alpha.\mathbf{t}'$, $\mathbf{t}_2 = \mathbf{t}'$ and $\mathbf{r} = (\alpha + 1).\mathbf{t}'$. Analogous to the previous case.
 - (iv) $\mathbf{t}_1 = \mathbf{t}_2 = \mathbf{t}'$, and $\mathbf{r} = (1 + 1).\mathbf{t}'$. Analogous to the previous case.
 - (v) $\mathbf{t}_1 \rightarrow_a \mathbf{t}'$ and $\mathbf{r} = \mathbf{t}' + \mathbf{t}_2$, then $(\mathbf{t}_1 + \mathbf{t}_2)[\mathbf{b}/x] = \mathbf{t}_1[\mathbf{b}/x] + \mathbf{t}_2[\mathbf{b}/x]$, which by the induction hypothesis \rightarrow_a -reduces to $\mathbf{t}'[\mathbf{b}/x] + \mathbf{t}_2[\mathbf{b}/x] = (\mathbf{t}' + \mathbf{t}_2)[\mathbf{b}/x]$.
 - (vi) $\mathbf{t}_2 \rightarrow_a \mathbf{t}'$ and $\mathbf{r} = \mathbf{t}_1 + \mathbf{t}'$. Analogous to the previous case.
- (e) $\mathbf{t} = \mathbf{t}_1\mathbf{t}_2$. We proceed now by cases:
- (i) $\mathbf{t}_1 = \mathbf{t}_{11} + \mathbf{t}_{12}$ and $\mathbf{r} = \mathbf{t}_{11}\mathbf{t}_2 + \mathbf{t}_{12}\mathbf{t}_2$, then $((\mathbf{t}_{11} + \mathbf{t}_{12})\mathbf{t}_2)[\mathbf{b}/x] = (\mathbf{t}_{11}[\mathbf{b}/x] + \mathbf{t}_{12}[\mathbf{b}/x])\mathbf{t}_2[\mathbf{b}/x] \rightarrow_a \mathbf{t}_{11}[\mathbf{b}/x]\mathbf{t}_2[\mathbf{b}/x] + \mathbf{t}_{12}[\mathbf{b}/x]\mathbf{t}_2[\mathbf{b}/x] = (\mathbf{t}_{11}\mathbf{t}_2)[\mathbf{b}/x] + (\mathbf{t}_{12}\mathbf{t}_2)[\mathbf{b}/x] = (\mathbf{t}_{11}\mathbf{t}_2 + \mathbf{t}_{12}\mathbf{t}_2)[\mathbf{b}/x]$.
 - (ii) $\mathbf{t}_2 = \mathbf{t}_{21} + \mathbf{t}_{22}$ and $\mathbf{t} = \mathbf{t}_1\mathbf{t}_{21} + \mathbf{t}_1\mathbf{t}_{22}$. Analogous to the previous case.
 - (iii) $\mathbf{t}_1 = \alpha.\mathbf{t}'$ and $\mathbf{r} = \alpha.\mathbf{t}'\mathbf{t}_2$, then $((\alpha.\mathbf{t}')\mathbf{t}_2)[\mathbf{b}/x] = \alpha.\mathbf{t}'[\mathbf{b}/x]\mathbf{t}_2[\mathbf{b}/x] \rightarrow_a \alpha.\mathbf{t}'[\mathbf{b}/x]\mathbf{t}_2[\mathbf{b}/x] = (\alpha.\mathbf{t}'\mathbf{t}_2)[\mathbf{b}/x]$.
 - (iv) $\mathbf{t}_2 = \alpha.\mathbf{t}'$ and $\mathbf{r} = \alpha.\mathbf{t}_1\mathbf{t}'$. Analogous to the previous case.
 - (v) $\mathbf{t}_1 = \mathbf{r} = \mathbf{0}$, then $(\mathbf{0}\mathbf{t}_2)[\mathbf{b}/x] = \mathbf{0}\mathbf{t}_2[\mathbf{b}/x] \rightarrow_a \mathbf{0} = \mathbf{0}[\mathbf{b}/x]$.
 - (vi) $\mathbf{t}_2 = \mathbf{r} = \mathbf{0}$. Analogous to the previous case.
 - (vii) $\mathbf{t}_1 \rightarrow_a \mathbf{t}'$ and $\mathbf{r} = \mathbf{t}'\mathbf{t}_2$, then $(\mathbf{t}_1\mathbf{t}_2)[\mathbf{b}/x] = \mathbf{t}_1[\mathbf{b}/x]\mathbf{t}_2[\mathbf{b}/x]$, which by the induction hypothesis \rightarrow_a -reduces to $\mathbf{t}'[\mathbf{b}/x]\mathbf{t}_2[\mathbf{b}/x] = (\mathbf{t}'\mathbf{t}_2)[\mathbf{b}/x]$.
 - (viii) $\mathbf{t}_2 \rightarrow_a \mathbf{t}'$ and $\mathbf{r} = \mathbf{t}_1\mathbf{t}'$. Analogous to the previous case.
- (2) We proceed by structural induction over \mathbf{t} .
- (i) $\mathbf{t} = x$, then $x[\mathbf{b}_1/x] = \mathbf{b}_1 \rightarrow_a \mathbf{b}_2 = x[\mathbf{b}_2/x]$.
 - (ii) $\mathbf{t} = y$, then $y[\mathbf{b}_1/x] = y = y[\mathbf{b}_2/x]$.
 - (iii) $\mathbf{t} = \mathbf{0}$, analogous to the previous case.
 - (iv) $\mathbf{t} = \lambda y \mathbf{t}'$, then $(\lambda y \mathbf{t}')[\mathbf{b}_1/x] = \lambda y \mathbf{t}'[\mathbf{b}_1/x]$. By the induction hypothesis $\mathbf{t}'[\mathbf{b}_1/x] \rightarrow_a^* \mathbf{t}'[\mathbf{b}_2/x]$, so $\lambda y \mathbf{t}'[\mathbf{b}_1/x] \rightarrow_a^* \lambda y \mathbf{t}'[\mathbf{b}_2/x]$.
 - (v) $\mathbf{t} = \mathbf{t}_1\mathbf{t}_2$. Then $(\mathbf{t}_1\mathbf{t}_2)[\mathbf{b}_1/x] = \mathbf{t}_1[\mathbf{b}_1/x]\mathbf{t}_2[\mathbf{b}_1/x]$. By the induction hypothesis $\mathbf{t}_1[\mathbf{b}_1/x] \rightarrow_a^* \mathbf{t}_1[\mathbf{b}_2/x]$ and $\mathbf{t}_2[\mathbf{b}_1/x] \rightarrow_a^* \mathbf{t}_2[\mathbf{b}_2/x]$, so $\mathbf{t}_1[\mathbf{b}_1/x]\mathbf{t}_2[\mathbf{b}_1/x] \rightarrow_a^* \mathbf{t}_1[\mathbf{b}_2/x]\mathbf{t}_2[\mathbf{b}_2/x] = (\mathbf{t}_1\mathbf{t}_2)[\mathbf{b}_2/x]$.
 - (vi) $\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2$. Then $(\mathbf{t}_1 + \mathbf{t}_2)[\mathbf{b}_1/x] = \mathbf{t}_1[\mathbf{b}_1/x] + \mathbf{t}_2[\mathbf{b}_1/x]$. By the induction hypothesis $\mathbf{t}_1[\mathbf{b}_1/x] \rightarrow_a^* \mathbf{t}_1[\mathbf{b}_2/x]$ and $\mathbf{t}_2[\mathbf{b}_1/x] \rightarrow_a^* \mathbf{t}_2[\mathbf{b}_2/x]$, so one has $\mathbf{t}_1[\mathbf{b}_1/x] + \mathbf{t}_2[\mathbf{b}_1/x] \rightarrow_a^* \mathbf{t}_1[\mathbf{b}_2/x] + \mathbf{t}_2[\mathbf{b}_2/x] = (\mathbf{t}_1 + \mathbf{t}_2)[\mathbf{b}_2/x]$.
 - (vii) $\mathbf{t} = \alpha.\mathbf{t}'$. Then $(\alpha.\mathbf{t}')[\mathbf{b}_1/x] = \alpha.\mathbf{t}'[\mathbf{b}_1/x]$. By the induction hypothesis one has $\mathbf{t}'[\mathbf{b}_1/x] \rightarrow_a^* \mathbf{t}'[\mathbf{b}_2/x]$, so $\alpha.\mathbf{t}'[\mathbf{b}_1/x] \rightarrow_a^* \alpha.\mathbf{t}'[\mathbf{b}_2/x] = (\alpha.\mathbf{t}')[\mathbf{b}_2/x]$. \square

Now, we can prove Theorem 5.14(3), using this property.

Proof. Let $\mathbf{t} \rightarrow_a \mathbf{u}$ and $\mathbf{t} \rightarrow_\beta \mathbf{r}$. We proceed by structural induction on \mathbf{t} .

- (1) \mathbf{t} cannot be a variable or $\mathbf{0}$, since it reduces.
- (2) $\mathbf{t} = \lambda x \mathbf{t}_1$. Then $\mathbf{u} = \lambda x \mathbf{u}_1$ and $\mathbf{r} = \lambda x \mathbf{r}_1$ with $\mathbf{t}_1 \rightarrow_a \mathbf{u}_1$ and $\mathbf{t}_1 \rightarrow_\beta \mathbf{r}_1$. Then by the induction hypothesis \mathbf{t}
- (3) $\mathbf{t}_1 = x$. Done.

- (4) $\mathbf{t}_1 = \lambda x \mathbf{r}$, then $\mathbf{t}_1 \mathbf{t}_2 \rightarrow \mathbf{r}[\mathbf{t}_2/x]$, which is a contradiction.
- (5) $\mathbf{t}_1 = \mathbf{0}$, then $\mathbf{t}_1 \mathbf{t}_2 \rightarrow \mathbf{0}$, which is a contradiction.
- (6) $\mathbf{t}_1 = \alpha.\mathbf{r}$, then $\mathbf{t}_1 \mathbf{t}_2 \rightarrow \alpha.\mathbf{r}\mathbf{t}_2$, which is a contradiction.
- (7) $\mathbf{t}_1 = \mathbf{r} + \mathbf{u}$, then $\mathbf{t}_1 \mathbf{t}_2 \rightarrow \mathbf{r}\mathbf{t}_2 + \mathbf{u}\mathbf{t}_2$, which is a contradiction.
- (8) $\mathbf{t}_1 = \mathbf{u}_1 \mathbf{u}_2$, then by the induction hypothesis $\mathbf{u}_1 = x\vec{\mathbf{r}}$, so $\mathbf{u}_1 \mathbf{u}_2 = x\vec{\mathbf{r}}'$, where $\vec{\mathbf{r}}' = \vec{\mathbf{r}}, \mathbf{u}_2$. \square

A.8. **Proof of Lemma 6.7.** First we need some remarks.

Remark A.2.

- (1) Variables can only have type $T \in \mathbb{T}(\lambda 2^{la})$, since contexts have only classic types in the type system \mathcal{B} , and forall eliminations do not introduce types with scalars.
- (2) If $A \in \mathbb{T}(\lambda 2^{la})$ and U is a unit type, with $\alpha.U \equiv \beta.A$, then it is easy to check that $U \in \mathbb{T}(\lambda 2^{la})$, $\alpha = \beta$ and $U \equiv A$, since there is no way using the equivalences to remove a scalar at the right of an arrow.

Now, we prove Lemma 6.7

Proof. I induction on the derivation of $\Gamma \vdash x\vec{\mathbf{r}}:T$. We only consider the cases ending with an application.

- (1)
$$\frac{\Gamma \vdash x:U \rightarrow T \quad \Gamma \vdash \mathbf{t}:\beta.U}{\Gamma \vdash x\mathbf{t}:\beta.T} \rightarrow_E \quad \text{Since } U \rightarrow T \in \mathbb{T}(\lambda 2^{la}), \text{ then } T \in \mathbb{T}(\lambda 2^{la}).$$
- (2)
$$\frac{\Gamma \vdash x\vec{\mathbf{r}}:\alpha.(U \rightarrow T) \quad \Gamma \vdash \mathbf{t}:\beta.U}{\Gamma \vdash (x\vec{\mathbf{r}})\mathbf{t}:(\alpha \times \beta).T} \rightarrow_E \quad \begin{array}{l} \text{Then by the induction hypothesis, } U \rightarrow \\ T \in \mathbb{T}(\lambda 2^{la}), \text{ so } T \in \mathbb{T}(\lambda 2^{la}). \end{array}$$
- (3)
$$\frac{\Gamma \vdash x\vec{\mathbf{r}}:\forall X.T}{\Gamma \vdash x\vec{\mathbf{r}}:T[A/X]} \forall_E \quad \begin{array}{l} \text{Then by the induction hypothesis there exist } B \in \mathbb{T}(\lambda 2^{la}) \\ \text{and } \alpha \in \mathcal{S} \text{ such that } \forall X.T \equiv \alpha.B, \text{ so there exists } C \in \\ \mathbb{T}(\lambda 2^{la}) \text{ such that } T \equiv \alpha.C, \text{ then } T[A/X] \equiv (\alpha.C)[A/X] \equiv \\ \alpha.C[A/X]. \text{ Note that } C[A/X] \in \mathbb{T}(\lambda 2^{la}). \end{array}$$
- (4)
$$\frac{\Gamma \vdash x\vec{\mathbf{r}}:T}{\Gamma \vdash x\vec{\mathbf{r}}:\forall X.T} \forall_I \quad \begin{array}{l} \text{Then by the induction hypothesis there exist } A \in \mathbb{T}(\lambda 2^{la}) \text{ and} \\ \alpha \in \mathcal{S} \text{ such that } T \equiv \alpha.A, \text{ so } \forall X.T \equiv \forall X.\alpha.A \equiv \alpha.\forall X.A. \quad \square \end{array}$$

A.9. **Proof of Lemma 6.8.** First note that if $A \in \mathbb{T}(\lambda 2^{la})$ and $\alpha.T \equiv \alpha.A$, then $T = A$ or $T = 1.A$: The only equivalence rule involving a type that could be in $\mathbb{T}(\lambda 2^{la})$ is $1.T \equiv T$, so if $\alpha.T \equiv \alpha.A$, T must be either $1.A$ or A .

Now we prove the Lemma for $\succ_{X,\Gamma}^t$, and in the light of Lemma 4.7, we can remove the scalar α . We proceed now by cases:

- $A \equiv \forall X.T$, then either $\forall X.T = A$, so $T \in \mathbb{T}(\lambda 2^{la})$, or $\forall X.T = 1.A$, in which case $T = 1.B$ with $B \in \mathbb{T}(\lambda 2^{la})$.
- $T \equiv \forall X.R$ and $A \equiv R[U/X]$. Since $R[U/X] \in \mathbb{T}(\lambda 2^{la})$, then $R \in \mathbb{T}(\lambda 2^{la})$ and so $\forall X.R \in \mathbb{T}(\lambda 2^{la})$, then either $T \in \mathbb{T}(\lambda 2^{la})$ or $T \equiv 1.B$ with $B \in \mathbb{T}(\lambda 2^{la})$. \square

A.10. Proof of Theorem 6.5. Instead, we prove the most general case: If $\Gamma \vdash \mathbf{t} : \alpha.A$ then $\omega(\text{nf}(\mathbf{t})) = \alpha$, by structural induction on $\text{nf}(\mathbf{t})$. We take $\Gamma \vdash \text{nf}(\mathbf{t}) : \alpha.A$, which is true by Theorem 4.1.

- (1) $\text{nf}(\mathbf{t}) = \mathbf{0}$. Then $\omega(\text{nf}(\mathbf{t})) = 0$. In addition, by Lemma 4.19, $\alpha.A \equiv \bar{0}$, so by Theorem 4.20, $\alpha = 0$.
- (2) $\text{nf}(\mathbf{t}) = x$ or $\text{nf}(\mathbf{t}) = \lambda x \mathbf{t}'$. Then $\omega(\text{nf}(\mathbf{t})) = 1$. In addition, by Lemma 4.18, $\alpha = 1$.
- (3) $\text{nf}(\mathbf{t}) = \gamma.\mathbf{t}'$. Given in the main text of the paper.
- (4) $\text{nf}(\mathbf{t}) = \mathbf{t}_1 + \mathbf{t}_2$. Then $\omega(\text{nf}(\mathbf{t})) = \omega(\mathbf{t}_1) + \omega(\mathbf{t}_2)$. By Lemma 4.14, there exist $\sigma, \phi \in \mathcal{S}$ such that $\Gamma \vdash \mathbf{t}_1 : \sigma.A$ and $\Gamma \vdash \mathbf{t}_2 : \phi.A$ with $\sigma + \phi = \alpha$. Then by the induction hypothesis $\omega(\mathbf{t}_1) = \sigma$ and $\omega(\mathbf{t}_2) = \phi$, so $\omega(\mathbf{t}_1) + \omega(\mathbf{t}_2) = \sigma + \phi = \alpha$.
- (5) $\text{nf}(\mathbf{t}) = \mathbf{t}_1\mathbf{t}_2$. Given in the main text of the paper. □